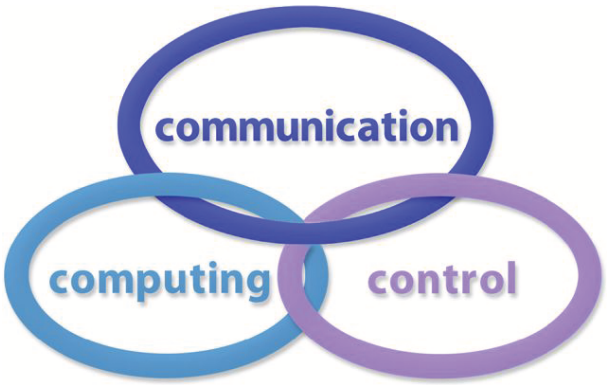
INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

With Emphasis on the Integration of Three Technologies

IJCCC



Year: 2011 Volume: 6 Number: 4 (December)

Agora University Editing House

CCC Publications

www.journal.univagora.ro

International Journal of Computers, Communications & Control

EDITORIAL ADDRESS:
R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piaţa Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Journal website: www.journal.univagora.ro

DATA FOR SUBSCRIBERS
Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Fiscal code: RO24747462
Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526
Bank: MILLENNIUM BANK, Bank address: Piata Unirii, str. Primariei, 2, Oradea, Romania
IBAN Account for EURO: RO73MILB0000000000932235
SWIFT CODE (eq.BIC): MILBROBU

International Journal of Computers, Communications & Control

EDITORIAL BOARD

Mario de J. Pérez Jiménez
Dept. of CS and Artificial Intelligence
University of Seville, Sevilla,
Avda. Reina Mercedes s/n, 41012, Spain
marper@us.es

Dana Petcu
Computer Science Department
Western University of Timisoara
V.Parvan 4, 300223 Timisoara, Romania
petcu@info.uvt.ro

Radu Popescu-Zeletin
Fraunhofer Institute for Open
Communication Systems
Technical University Berlin, Germany
rpz@cs.tu-berlin.de

Imre J. Rudas
Institute of Intelligent Engineering Systems
Budapest Tech
Budapest, Bécsi út 96/B, H-1034, Hungary
rudas@bmf.hu

Yong Shi
Research Center on Fictitious Economy
& Data Science
Chinese Academy of Sciences
Beijing 100190, China
yshi@gucas.ac.cn
and
College of Information Science & Technology
University of Nebraska at Omaha
Omaha, NE 68182, USA
yshi@unomaha.edu

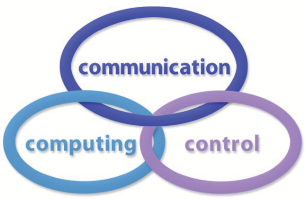
Athanasios D. Styliadis
Alexander Institute of Technology
Agiou Panteleimona 24, 551 33
Thessaloniki, Greece
styl@it.teithe.gr

Gheorghe Tecuci
Learning Agents Center
George Mason University, USA
University Drive 4440, Fairfax VA 22030-4444
tecuci@gmu.edu

Horia-Nicolai Teodorescu
Faculty of Electronics and Telecommunications
Technical University "Gh. Asachi" Iasi
Iasi, Bd. Carol I 11, 700506, Romania
hteodor@etc.tuiasi.ro

Dan Tufiş
Research Institute for Artificial Intelligence
of the Romanian Academy
Bucharest, "13 Septembrie" 13, 050711, Romania
tufis@racai.ro

Lotfi A. Zadeh
Professor,
Graduate School,
Director,
Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
Department of Electrical Engineering
& Computer Sciences
University of California Berkeley,
Berkeley, CA 94720-1776, USA
zadeh@eecs.berkeley.edu



International Journal of Computers, Communications & Control

Short Description of IJCCC

Title of journal: International Journal of Computers, Communications & Control
Acronym: IJCCC
International Standard Serial Number: ISSN 1841-9836, E-ISSN 1841-9844
Publisher: CCC Publications - Agora University
Starting year of IJCCC: 2006
Founders of IJCCC: Ioan Dzitac, Florin Gheorghe Filip and Mişu-Jan Manolescu
Logo:

Number of issues/year: IJCCC has 4 issues/odd year (March, June, September, December) and 5 issues/even year (March, September, June, November, December). Every even year IJCCC will publish a supplementary issue with selected papers from the International Conference on Computers, Communications and Control.
Coverage:

- Beginning with Vol. 1 (2006), Supplementary issue: S, IJCCC is covered by Thomson Reuters - SCI Expanded and is indexed in ISI Web of Science.
- Journal Citation Reports/Science Edition 2010:
  - Impact factor = 0.650
- Beginning with Vol. 2 (2007), No.1, IJCCC is covered in EBSCO.
- Beginning with Vol. 3 (2008), No.1, IJCCC, is covered in Scopus.

Scope: IJCCC is directed to the international communities of scientific researchers in universities, research units and industry. IJCCC publishes original and recent scientific contributions in the following fields: Computing & Computational Mathematics; Information Technology & Communications; Computer-based Control.
Unique features distinguishing IJCCC: To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (Computing, Communication, Control).
Policy: The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by at least two reviewers and will be published only after completion of the editorial workflow.

Contents

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 592-602

A Non-cooperative Game Algorithm for Task Scheduling in Wireless Sensor Networks

L. Dai, Y.L. Chang, Z. Shen

Liang Dai
School of Electronic and Control Engineering
Chang'an University
Xi'an 710064, China
E-mail: ldai1981@gmail.com

Yilin Chang, Zhong Shen
State Key Laboratory of Integrated Service Networks
Xidian University
Xi'an 710071, China
E-mail: ylchang@xidian.edu.cn, zhshen@mail.xidian.edu.cn

Abstract: Scheduling tasks in wireless sensor networks is one of the most challenging problems. Sensing tasks should be allocated and processed among sensors in minimum times, so that users can draw prompt and effective conclusions through analyzing sensed data. Furthermore, finishing sensing task faster will benefit energy saving, which is critical in system design of wireless sensor networks. But sensors may refuse to take pains to carry out the tasks due to the limited energy. To solve the potentially selfish problem of the sensors, a non-cooperative game algorithm (NGTSA) for task scheduling in wireless sensor networks is proposed. In the proposed algorithm, according to the divisible load theory, the tasks are distributed reasonably to every node from SINK based on the processing capability and communication capability. By removing the performance degradation caused by communications interference and idle, the reduced task completion time and the improved network resource utilization are achieved. Strategyproof mechanism which provide incentives to the sensors to obey the prescribed algorithms, and to truthfully report their parameters, leading to an effient task scheduling and execution. A utility function related with the total task completion time and tasks allocating scheme is designed. The Nash equilibrium of the game algorithm is proved. The simulation results show that with the mechanism in the algorithm, selfish nodes can be forced to report their true processing capability and endeavor to participate in the measurement, thereby the total time for accomplishing the task is minimized and the energy-consuming of the nodes is balanced.
Keywords: wireless sensor networks; task scheduling; divisible load theory; game algorithm; mechanism design.

1 Introduction & Motivation

Wireless sensor networks are constituted by a large number of nodes, and without fixed infrastructure and centralized control mechanisms, so the task scheduling needs of mutual cooperation in all nodes. The energy of sensors is limited, so not all the sensors are willing to participate in collaborative efforts. Lack of coordination between the nodes in wireless sensor networks is a new problem[1].

Owing to the sensors with limited energy, the task should be completed within the shortest possible amount of time, so that users can draw useful conclusions through analyzing sensed

data. Furthermore, finishing sensing task faster will benefit energy saving, which is critical in system design of wireless sensor networks. Divisible load theory [2] provides an effective solution to wireless sensor networks for task scheduling [3-6]. Different from other heuristic solutions of task scheduling problem in wireless sensor networks [7, 8], this scheme can get not only the optimal solution, but also the analytic solution, thus ensuring the consistency of the results of scheduling.

Divisible load scheduling algorithms were applied to wireless sensor networks in [3-6]. Although the authors derived closed-form solutions to obtain the optimal finish time, the network topology discussed in those papers is single-level tree structure. While in wireless sensor networks, as compared with the single-level tree structure, clustered structure (multi-level tree structure) has a great of advantages [9]. Those algorithms mentioned above designed on the premise that sensors manipulate the algorithms and perform a task to the best of their abilities. But the sensors may be owned by autonomous rational organizations or people that have no a priori motivation for cooperation. Consequently, they will manipulate the algorithms if it benefits them to do so.

Therefore, we present a non-cooperative game algorithm for task scheduling in wireless sensor networks. The goal of this algorithm is to minimize the overall execution time (hereafter called *makespan*) and fully utilize network resources, by finding an optimal strategy of splitting the original tasks received by SINK into a number of sub-tasks as well as distributing these sub-tasks to the clusters in the right order, and through the proposed mechanism to encourage collaboration.

2 Mechanism design

The energy of sensors is limited, if we do not integrate contributions with benefits, not all nodes willing to participate in collaborative efforts to perform tasks. This selfish, non-cooperative behavior will result in poor performance. If we follow the principle of more labor and more benefits, some selfish sensors, driven by the interests, will intentionally exaggerate its data-measuring capacity to increase their income. Such dishonest behavior will result in that it's difficult to determine which sensor reporting its actual capacity, and which said a lie, so we can not globally optimize tasks allocation according to each sensor's capacity. These dishonest or selfish behavior, has seriously affected the efficiency of wireless sensor network. Therefore, we should provide incentives to the nodes to obey the prescribed algorithms and to truthfully report their parameters, leading to an efficient allocation and execution. This problem can be solved by applying mechanism design of game theory.

Mechanisms are generally composed of allocation $o = o(a_1, a_2, \cdots, a_n)$ and payment $p = (p_1(a), p_2(a), \cdots, p_n(a))$ rules. Allocation rules also are called output function. In mechanism design, there are usually n agents, and each agent has some private information, which is known as the type of agent or true value. The private information is only known by agent, and is confidential for the external, for example, the type of a agent t_i can be the cost executing an assigned task. $v_i(t_i, o)$ is the value function of an agent, said the cost of performing a task. $p_i(\cdot)$ is the payment function for agent performing a task. $u_i(\cdot)$ is the utility function of an agent, that $u_i(\cdot) = p_i(\cdot) - v_i(t_i, o)$. To maximize the utility function value is the goal of agents. Mechanism should be designed to maintain unity with the goal the agent seeking for. Mechanism can induce the agent's choice or behavior by means of payment.

If in a mechanism, the only optional behavior of all agents is directly stating their preferences, then we say that the mechanism is a direct revelation mechanism. In a direct revelation mechanism, if all agents will tell their true preferences in order to maximize their utility, then we say that the mechanism is incentive compatibility. The nature of incentive compatibility allows

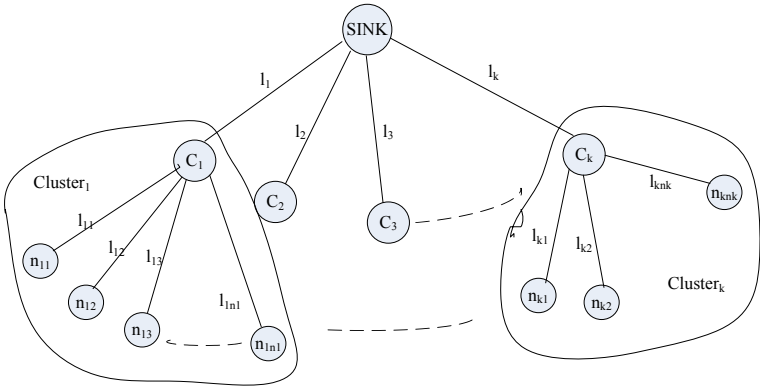


Figure 1: Network topology

us to design a mechanism to overcome the selfishness of agents, because in an incentive compatible mechanism, the agent will always choose to say their true private information in order to maximize their utility. In a direct revelation mechanism, if it's a dominant strategy that they gave their true preferences, then we claimed that the mechanism is *strategyproof*. If each agent is honest, and each agent's income is not less than zero, then the mechanism satisfies the voluntary participation condition.

3 Scheduling Algorithm

3.1 Network model

The network topology discussed in this paper is shown in Fig.1. Wireless sensor networks construct clusters several times in its life cycle. Each cluster will have a set-up phase and a steady-state phase[11]. Assuming that there are k clusters ($Cluster_i$, $i = 1, ..., k$) in the network during a steady-state phase. Each cluster head is expressed as C_i. Within the $Cluster_i$, there are n_i nodes expressed as $n_{i,j}$ $(i = 1, ...k; j = 1, ...n_i)$, respectively. Communication links between cluster heads and SINK are expressed as l_i, $i = 1, ..., k$, respectively. Communication links between intra-cluster nodes and cluster head C_i are expressed as $l_{i,j}$ $(i = 1, ...k; j = 1, ...n_i)$, respectively.

The following notations will be used throughout this paper:

α_i: The total fraction of load that is assigned by SINK to cluster head C_i;

$\alpha_{i,j}$: The fraction of load that is assigned to intra-cluster node $n_{i,j}$ in $Cluster_i$ by C_i;

By definition we can see:

$$\sum_{i=1}^{k} \alpha_i = 1 \tag{1}$$

$$\sum_{j=1}^{n_i} \alpha_{i,j} = \alpha_i \tag{2}$$

ω_i: A constant that is inversely proportional to the processing (data fusion) speed of cluster head C_i;

$y_{i,j}$: A constant that is inversely proportional to the measuring speed of intra-cluster node $n_{i,j}$;

z_i: A constant that is inversely proportional to the speed of link between SINK and cluster head C_i;

$z_{i,j}$: A constant that is inversely proportional to the speed of link between the cluster head C_i and the intra-cluster node $n_{i,j}$;

T_{cp}: Data fusion intensity constant. T_{cp} is the time that takes to fuse all the load on a cluster head when $\omega_i = 1$. The entire load can be fused on cluster head C_i in time $\omega_i T_{cp}$;

T_{ms}: Measurement intensity constant. T_{ms} is the time that takes the intra-cluster node $n_{i,j}$ to measure the entire load when $y_{i,j} = 1$. The entire assigned measurement load can be measured on the intra-cluster node $n_{i,j}$ in time $y_{i,j}T_{ms}$;

T_{cm}: Communication intensity constant. This is the time it takes to transmit all the processing loads over a link when $z_i = 1$. The entire load can be transmitted over the ith link in time $z_i T_{cm}$;

φ_i: The information utility constant [8,10] of cluster head C_i. Information utilization constant is based on a technique of information accuracy estimation. Through estimating accuracy of information, cluster head can know the approximate percentage of data fusion.

T_f: The finish time of a given task.

In this paper, we formulate the problem of task scheduling in a large scale sensor network as an optimization problem to minimize T_f.

3.2 Scheduling strategy

The proposed task scheduling algorithm is based on LEACH [11] protocol. LEACH was firstly proposed clustering protocol in wireless sensor network, and its certain governing ideas breathe through many other clustering protocols, such as TEEN and DAEA. LEACH protocol structures the single-hop networks. Each cluster's intra-cluster nodes share the same channel, while each cluster head has its own channel to SINK.

The original tasks received by SINK are divided into two stages: inter-cluster task scheduling and intra-cluster task scheduling. First, inter-cluster task scheduling partitions the entire tasks into each cluster, and then the sub-tasks in a cluster are assigned to each intra-cluster node by intra-cluster task scheduling.

1) Intra-cluster task scheduling

Fig.2 illustrates the timing diagram for a set of intra-cluster nodes, indexed from n_1 to n_k, in one cluster. From Fig.2, it can be observed that there is no time gap between every two successive nodes because the divisible workload can be transferred in the cluster. All nodes start to measure data at the same time. Once the previous node finishes transmitting data, the other one completes its measuring task and starts to report its data. As a result, the proposed timing diagram minimizes the makespan by scheduling the measuring time and reporting time of each node. Moreover, since the intra-cluster scheduling tries to avoid the transmission conflicts at the cluster head, energy spent on retransmission is conserved.

For cluster i, based on the timing diagram shown in Fig.2, one can write the following set of equations about tasks assigned to the intra-cluster nodes in $Cluster_i$:

$$\alpha_{i,j-1} y_{i,j-1} T_{ms} = \alpha_{i,j} y_{i,j} T_{ms} + \alpha_{i,j} z_{i,j} T_{cm}, i = 2, 3, ...k \tag{3}$$

A general expression for the above set of recursive equations can be written as:

$$\alpha_{i,j} = s_{i,j} \alpha_{i,j-1} \tag{4}$$

where $s_{i,j} = y_{i,j-1} T_{ms} / (y_{i,j} T_{ms} + z_{i,j} T_{cm})$.

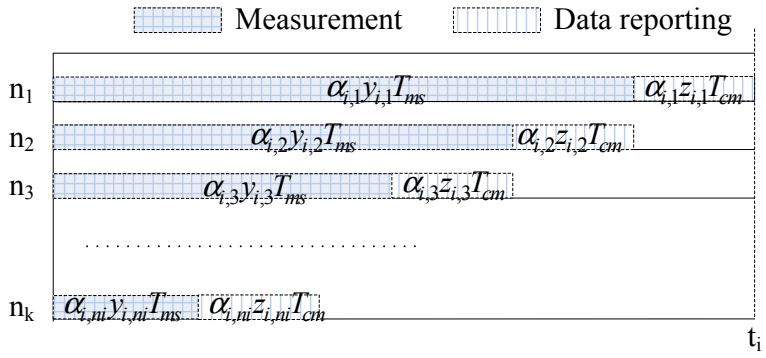


Figure 2: Timing diagram for intra-cluster task-processing

Using eq.2 and eq.4, The above recursive equation for $\alpha_{i,1}$ can be rewritten in terms of α_i only as

$$\alpha_{i,1} = \alpha_i/(1+\sum_{j=2}^{n_i}\prod_{k=2}^{j} s_{i,k}) \tag{5}$$

The cluster head C_i will use the above value of $\alpha_{i,1}$ to obtain the amount of data that has to be measured by the rest of the $n_i - 1$ sensors by using

$$\alpha_{i,j} = \alpha_i \prod_{k=2}^{j}(s_{i,k})/(1+\sum_{j=2}^{n_i}\prod_{k=2}^{j} s_{i,k}) \tag{6}$$

Then from Fig.2, the minimum time for measuring, reporting SINK's sub-task α_i can be given as

$$t_i{=}\alpha_i(y_{i,1}T_{ms} + z_{i,1}T_{cm})/(1+\sum_{j=2}^{n_i}\prod_{k=2}^{j} s_{i,k}) \tag{7}$$

2) Inter-cluster task scheduling

After cluster heads fusing the intra-cluster nodes' measured data, cluster heads can send the fused data to SINK concurrently because each cluster head has a separate channel to SINK.

In order to remove the performance degradation caused by idle, and to improve efficiency, as shown in Fig.3, we can get

$$t_i + \alpha_i w_i T_{cp} + \varphi_i \alpha_i z_i T_{cm} = t_j + \alpha_j w_j T_{cp} + \varphi_j \alpha_j z_j T_{cm}, i \neq j \tag{8}$$

In eq.7, we make $(y_{i,1}T_{ms} + z_{i,1}T_{cm})/(1+\sum\limits_{j=2}^{n_i}\prod\limits_{k=2}^{j} s_{i,k}) = g_i$, then substitute g_i into eq.8, we can get

$$\alpha_i r_i = \alpha_j r_j \tag{9}$$

where $r_i = g_i + w_i T_{cp} + \varphi_i z_i T_{cm}$.

Now using the eq.1 and eq.9, one can solve for α_i as

$$\alpha_i{=}(1/r_i)/\sum_{i=1}^{k}(1/r_i) \tag{10}$$

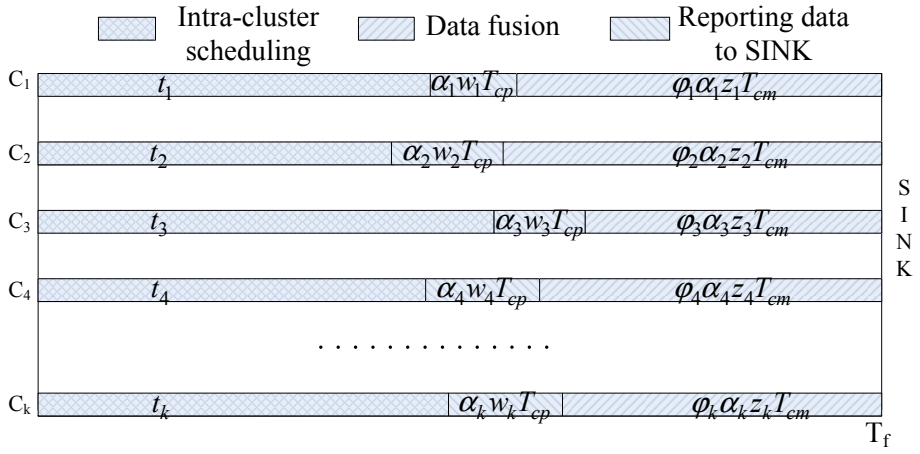


Figure 3: Timing diagram for inter-cluster task scheduling

Hereto, we can get the fraction of load cluster head C_i and the intra-cluster nodes within it received from SINK (α_i and $\alpha_{i,j}$).

And the total task execution time is as follow:

$$T_f{=}t_i + (w_i T_{cp} + \varphi_i z_i T_{cm})(1/r_i)/\sum_{i=1}^{k}(1/r_i) \tag{11}$$

For the homogeneous networks, the parameters(measurement speed, communication speed, data-fusing speed, information utility constant) of the network are the same and there are n sensors in each cluster. SINK can evenly distributed loads to each cluster. The sub-loads each cluster receiving is as follows:

$$\alpha_i{=}1/k \tag{12}$$

In the homogeneous networks, the minimum time-consuming for the total task is:

$$T_f{=}(D + wT_{cp} + \varphi z T_{cm})/k \tag{13}$$

where $D{=}\ (yT_{ms} + zT_{cm})/n$.

With the increasing density of wireless sensor networks, the number of the clusters is higher with the current clustering algorithms. From eq.13, it can be known that T_f approaches 0 with k approaching infinite.

4 Mechanism Design for NGTSA

In this section, we propose strategyproof mechanism for task scheduling of wireless sensor networks. The ith cluster is composed of n_i strategic sensors. Each task-performing sensor $n_{i,j}$ is characterized its true value $y_{i,j}$, which is equal to the time to measure the unit task. We denote by $\mathbf{y}_i = (y_{i,1}, y_{i,2}, \cdots, y_{i,n_i})$ the vector of true measuring speed. Measuring speed $y_{i,j}$ is private to $n_{i,j}$. As the sensor $n_{i,j}$ with rationality may give a bid $b_{i,j}$ which is not equal the true measuring speed $y_{i,j}$, if it benefits it to do so.

Cluster heads calculated the fraction of load that is assigned to intra-cluster nodes $\alpha_i(\mathbf{b}_i) = (\alpha_{i,1}(\mathbf{b}_i), \alpha_{i,2}(\mathbf{b}_i), \cdots, \alpha_{i,n_i}(\mathbf{b}_i))$, according to the bids of intra-cluster nodes, where $\sum_{j=1}^{n_i} \alpha_{i,j}(\mathbf{b}_i) = \alpha_i$, and $\mathbf{b}_i = (b_{i,1}, b_{i,2}, \cdots, b_{i,n_i})$ is the bid vector of intra-cluster nodes.

As the selfishness of sensors, the sensor may complete the measurement task at a relatively low speed $\tilde{y}_{i,j}$, where $\tilde{y}_{i,j} \leq y_{i,j}$. After intra-cluster nodes completed their task, cluster head node will get the actual measuring speed.

The value function of $n_{i,j}$ under task allocation $\alpha_i(\mathbf{b}_i)$ is defined as follow:

$$V_{i,j}(\alpha_i(\mathbf{b}_i), \tilde{y}_{i,j}) = -\alpha_{i,j}\tilde{y}_{i,j} \tag{14}$$

The value of $V_{i,j}(\alpha_i(\mathbf{b}_i), \tilde{y}_{i,j})$ is equivalent to the negation of the actual time required for $n_{i,j}$ to complete $\alpha_{i,j}$ of total load. It can be regarded as the cost of $n_{i,j}$ to complete its task.

The mechanism will provide payment for involving in measurement, so each node will select the appropriate strategy to maximize its benefits. The utility function of $n_{i,j}$ is defined as follow:

$$U_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) = Q_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) + V_{i,j}(\alpha_i(\mathbf{b}_i), \tilde{y}_{i,j}) \tag{15}$$

where $Q_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i)$ is the payment cluster head C_i paid to $n_{i,j}$, $\tilde{\mathbf{y}}_i = (\tilde{y}_{i,1}, \tilde{y}_{i,2}, \cdots, \tilde{y}_{i,n_i})$ is the vector of each node's actual measurement speed. Utility function equals the payment the node got from mechanism minus the cost for completing its tasks.

Designing such mechanisms involves finding an allocation and a payment scheme that minimize the makespan according to the sensors' bid $\mathbf{b}_i$ and motivate all the intra-cluster sensors to bid their true value $y_{i,j}$ and measure the task at their full measuring capacity $\tilde{y}_{i,j} = y_{i,j}$.

Define the payment function for mechanism is as follow:

$$Q_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) = C_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) + B_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) \tag{16}$$

where $C_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) = -V_{i,j}(\alpha_i(\mathbf{b}_i), \tilde{y}_{i,j})$ is the compensation to the node $n_{i,j}$.

The bonus of $n_{i,j}$ is defined as follow:

$$B_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) = T_{-i,j}(\alpha_i(\mathbf{b}_{-i,j}), \mathbf{b}_{-i,j}) - T(\alpha_i(\mathbf{b}_i), (\mathbf{b}_{-i,j}, \tilde{y}_{i,j})) \tag{17}$$

where $T_{-i,j}(\alpha(\mathbf{b}_{-i,j}), \mathbf{b}_{-i,j})$ is the optimal task completion time when $n_{i,j}$ is not involved in measurement, that is, the bonus of intra-cluster node is equal to its contribution in reducing the task completion time because of its participation for measurement.

Theorem (strategyproofness). The mechanism proposed in this paper is *strategyproof*.

Proof. According to the bids of intra-cluster nodes in cluster i, $\mathbf{b}_i = (b_{i,1}, b_{i,2}, \cdots, b_{i,n_i})$, we can get the utility of $n_{i,j}$ is as follow:

$$\begin{aligned} U_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) &= Q_{i,j}(\mathbf{b}_i, \tilde{\mathbf{y}}_i) + V_{i,j}(\alpha_i(\mathbf{b}_i), \tilde{y}_{i,j}) \\ &= T_{-i,j}(\alpha(\mathbf{b}_{-i,j}), \mathbf{b}_{-i,j}) - T(\alpha_i(\mathbf{b}_i), (\mathbf{b}_{-i,j}, \tilde{y}_{i,j})) + \alpha_{i,j}(\mathbf{b})\tilde{y}_{i,j} - \alpha_{i,j}(\mathbf{b})\tilde{y}_{i,j} \\ &= T_{-i,j}(\alpha(\mathbf{b}_{-i,j}), \mathbf{b}_{-i,j}) - T(\alpha_i(\mathbf{b}_i), (\mathbf{b}_{-i,j}, \tilde{y}_{i,j})) \end{aligned} \tag{18}$$

When $n_{i,j}$ perform a task to the best of their abilities ($\tilde{y}_{i,j} = y_{i,j}$), and bid the measurement as $b_{i,j} = y_{i,j}$, the utility of $n_{i,j}$ is as follow:

$$\begin{aligned} U^t_{i,j} &= T_{-i,j}(\alpha(\mathbf{b}_{-i,j}), \mathbf{b}_{-i,j}) - T(\alpha_i(\mathbf{b}_i), (\mathbf{b}_{-i,j}, \tilde{y}_{i,j})) \\ &= T_{-i,j}(\alpha(\mathbf{b}_{-i,j}), \mathbf{b}_{-i,j}) - T^t_{i,j} \end{aligned} \tag{19}$$

where $T^t_{i,j}$ is the shortest possible amount of time to finish total tasks according to divisible load theory. If $n_{i,j}$ bids a lower value, that is, $b_{i,j} < y_{i,j}$,the makespan is increased by a small amount as the load allocated to the other intra-cluster nodes is increased. Therefore, only when $n_{i,j}$ voluntarily gives the true measuring speed and be assiduous in its duty, they can get the maximum benefit.

Because $T_{-i,j}$ is the optimal task completion time when $n_{i,j}$ is not involved in measurement, its value must not be less than when all the nodes involved in the tasks. So, we can get $U^t_{i,j} \geq 0$, that is, all nodes within one cluster will try to do their best to carry out a task.

5 Wireless Energy Use

In this section, the energy model of the NGTSA algorithm is presented in detail and the equations of energy consumption of individual sensor nodes are derived. The model is based on first-order radio model [11].

There are three kinds of energy consumption in the wireless sensor network: measurement, data fusion, and communication. Because nodes in the sensor network cooperate with each other via data transmission. Energy consumption of communications exist in sensor nodes, cluster heads and SINK. It is not necessary for cluster heads and SINK to perform any sensing task. Thus, there is no energy cost for cluster heads due to the measurement of these nodes, while the additional energy cost of cluster heads attributes to data fusion. The energy to sense, fuse, and transmit a unit sensory data are denoted by e_s, e_p, and e_{tx}, respectively. Sensor nodes also consume the energy of e_{rx} to receive one unit of data. The distance between the sender and the receiver is d.

The energy use for each kind of nodes is outlined as follows:

Energy use for individual sensor nodes j in cluster i:

$$E_{i,j} = \alpha_{i,j}(e_s + e_{tx}d^2), i = 1, \cdots, k\,, j = 1, \cdots, n_i \tag{20}$$

Energy use for individual cluster head:

$$E_i = \alpha_i(e_{rx} + e_p + \varphi_i e_{tx}d^2), i = 1, \cdots, k \tag{21}$$

Energy use for SINK:

$$E_{SINK} = \sum_{i=1}^{k} \alpha_i \varphi_i e_{tx} \tag{22}$$

6 Performance Evaluation

In the above sections, we have obtained the close-form solution of the task completion time and the formula for the energy consumption of each node. In this section, we investigate the effects of different cases under homogeneous network environment on the makespan, energy consumption of every intra-cluster nodes, and the payment/utility of the cheating nodes.

We consider the wireless sensor network comprising 10 clusters. Within each cluster, there are 20 intra-cluster nodes. Assume that only the first intra-cluster node cheats by reporting values different than its true measuring speed, and by measurement at a different speed than the true speed in each cluster. The speed of communication link between each node were 0.05; the measuring speed of each intra-cluster node is 0.2; the data fusion speed of each cluster head is 0.1, and the information utility constant of each cluster head is 0.5.

In the simulation, the following energy parameters are adopted: transmitting a unit of sensor reading over a unit distance takes e_{tx}=200nJ, receiving one unit of sensor reading consumes e_{rx}=150nJ, measuring one unit of sensor reading needs e_s=100nJ, fusing one unit of observation consumes e_p=20nJ and the distance between the sender and the receiver is d=100m. In the simulation, we supposed that: T_{ms}=T_{cm}=T_{cp}= 1.

We simulated the following eight cases: ① $y_{i,1} = b_{i,1} = \tilde{y}_{i,1}$; ② $y_{i,1} = b_{i,1} < \tilde{y}_{i,1}$; ③ $y_{i,1} < b_{i,1} = \tilde{y}_{i,1}$; ④ $y_{i,1} = \tilde{y}_{i,1} < b_{i,1}$; ⑤ $b_{i,1} < y_{i,1} = \tilde{y}_{i,1}$; ⑥ $y_{i,1} < \tilde{y}_{i,1} < b_{i,1}$; ⑦ $y_{i,1} < b_{i,1} < \tilde{y}_{i,1}$; ⑧ $b_{i,1} < y_{i,1} < \tilde{y}_{i,1}$. where $b_{i,1}$ states the bid of the first intra-cluster node; $y_{i,1}$ represents the node's actual value; $\tilde{y}_{i,1}$ is the true value of the measurement speed.

The bid $b_{i,1}$ and the measurement value $\tilde{y}_{i,1}$ for each case are presented in Tables 1:

Table 1: Bids and Execution Values

Case	1	2	3	4	5	6	7	8
$b_{i,1}$	0.2	0.2	0.3	0.3	0.1	0.4	0.3	0.1
$\tilde{y}_{i,1}$	0.2	0.3	0.3	0.2	0.2	0.3	0.4	0.3

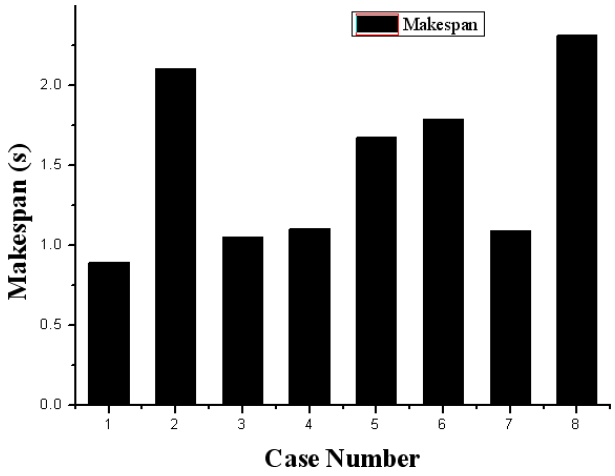


Figure 4: Makespan when the first intra-cluster node cheats

The simulation results are shown in Fig.3 and Fig.5.

Firstly, Fig.3 shows the makespan for the eight cases. Notice that case 1 ($y_{i,1} = b_{i,1} = \tilde{y}_{i,1}$) results in the minimum makespan, while all other cases result in a larger makespan. In that case, each node honestly bids, and makes effort to measure. When $\tilde{y}_{i,1} \leq b_{i,1}$, the case 3,4, and 7, the makespan is increased by a small amount as the load allocated to the other intra-cluster nodes is increased. The effect of cheating is dispersed throughout the remaining nodes. In the remaining case ($\tilde{y}_{i,1} > b_{i,1}$),the case 2,5,6, and 8, the network performance dramatically degrades as the first intra-cluster nodes is overloaded, and the other nodes are underutilized. In these cases, the first intra-cluster nodes is slowing down the entire network. The increase in makespan is large, and it is due to the impact of $\tilde{y}_{i,1}$.

Next, the second simulation is about the energy consumption of intra-cluster sensor nodes for the 8 cases. SINK and cluster heads are not taken into account, because generally, SINK has no energy constraint and the chosen cluster heads have the possibly enough energy. The network is configured with 20 clusters. Without loss of generality, the intra-cluster sensor nodes in the first cluster are chosen to study the energy consumption, as shown in Fig.5. In each case, the energy consumption of sensor nodes monotonically decreases due to the reduced workload. In the case ($y_{i,1} = b_{i,1} = \tilde{y}_{i,1}$), each node honestly bids, and make effort to measure, so the energy-consuming of each intra-cluster node is most evenly. In the case 2,5,6, and 8, because the cluster head allocates more tasks on the first intra-cluster node as it bids a rate slower than its true rate, the energy-consuming of the first intra-cluster nodes increases noticeably, which will run out power and affected the entire network performance.

Then, the third simulation is about the payment and utility of intra-cluster nodes. Fig.6 shows the payment and utility for the first intra-cluster nodes in the 8 cases. As we expected, case 1 ($y_{i,1} = b_{i,1} = \tilde{y}_{i,1}$), maximizes the first intra-cluster node's utility. In all the other cases, utility is lessened. When $\tilde{y}_{i,1} > b_{i,1}$, the utility is negative due to the impact of $\tilde{y}_{i,1}$ on the

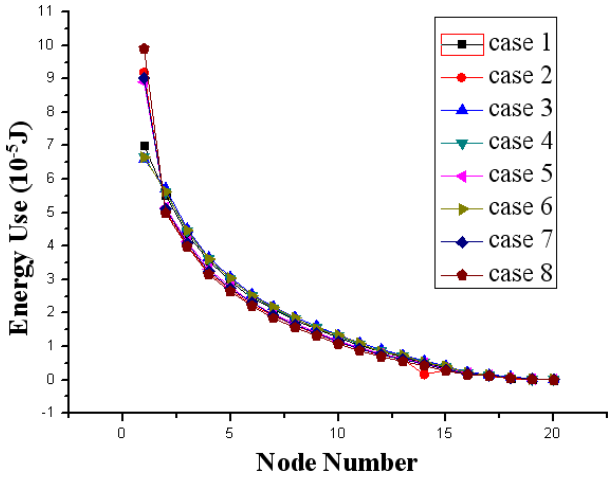


Figure 5: Energy consumption when the first intra-cluster node cheats

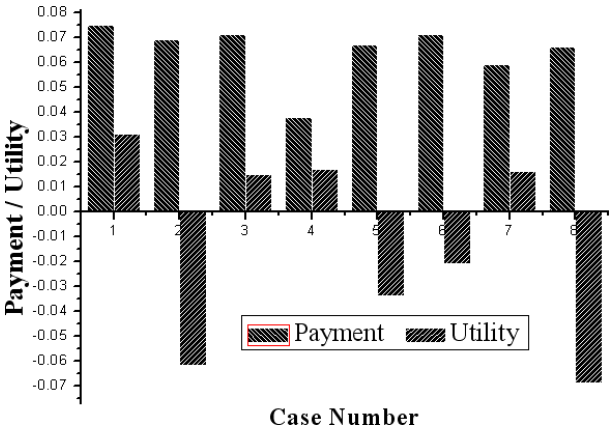


Figure 6: Payment and utility of the first intra-cluster node when it cheats

makespan. In the remaining cases, payment and utility are reduced as $\alpha_{i,1}$.

7 Conclusions

As the node in wireless sensor network has limited energy, the tasks should be completed as quickly as possible, and the network resources should be fully utilized. In this paper, we present a task scheduling algorithm (NGTSA) in clustered wireless sensor networks. The goal of this algorithm is to minimize the makespan and fully utilize network resources, by finding an optimal strategy of splitting the original load received by SINK into a number of chunks as well as distributing these chunks to the clusters in the right order.

As the gradual development of the wireless sensor networks and ubiquitous networks, this scheduling model using divisible load theory needs further study and validation. In addition, the validation of divisible load theory in wireless sensor networks still remains in the simulation phase. There are short of scanty of large-scale, practical applications to demonstrate its true potential, so applying the divisible load theory in actual wireless sensor network should be an important research direction for the next step.

Acknowledgement

The authors thank the editors and the anonymous reviewers for their valuable comments that helped to improve the paper. The work was supported by the National Natural Science Foundation of China (No.60972047), and the 111 project (No.B08038).

Bibliography

[1] C. Pandana, H. Zhu, L. Ray, Cooperation Enforcement and Learning for Optimizing Packet Forwarding in Autonomous Wireless Networks. *IEEE Transactions on Wireless Communications*, vol.7, No.8, pp.3150-3163, 2008.

[2] V. Bharadwaj, D. Ghose, T. G.ROBERTAZZI, Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems. *Cluster Computing*, vol.6, No.1, pp.7-18, 2003.

[3] M. Moges, T.G. Robertazzi, Wireless Sensor Networks: Scheduling for Measurement and Data Reporting. *IEEE Transactions on Aerospace and Electronic Systems*, vol.42, No.1, pp.327-340, 2006.

[4] H. LIU, X. YUAN, M. Moges, An Efficient Task Scheduling Method for Improved Network Delay in Distributed Sensor Networks. *In Proceedings of TridentCom 2007*, Orlando, FL, USA, 1-8, 2007.

[5] H. LIU, J. SHEN, X. YUAN, M. Moges, Performance Analysis of Data Aggregation in Wireless Sensor Mesh Networks, *In Proceedings of Earth & Space 2008*, Akron, OH, USA, 1-8, 2008.

[6] C. Kijeung , T. G. Robertazzi, Divisible Load Scheduling in Wireless Sensor Networks with Information Utility Performance. *In Proceedings of IPCCC 2008*, Austin, Texas, USA, 9-17, 2008.

[7] Z. ZENG, A. LIU, D. LI, A Highly Efficient DAG Task Scheduling Algorithm for Wireless Sensor Networks, *In Proceedings of ICYCS 2008*,Zhang Jia Jie , Hunan , China, 570-575, 2008.

[8] J. LIN, W. XIAO, F. L. Lewis, Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks. *IEEE Transactions on Instrumentation and Measurement*, vol.58, No.6, pp.1886 - 1896, 2009.

[9] P. Guo, T. Jiang, K. Zhang, H. Chen, Clustering algorithm in initialization of multi-hop wireless sensor networks. *IEEE Transactions on Wireless Communications*, vol.8, No.12, pp. 5713–5717, 2009.

[10] N. Nisan, A. Ronen. Algorithmic mechanism design, *Games and Economic Behavior*, vol.35, nos.1-2, pp.166 -196, 2001.

[11] W. Heinzelman, A. Chandrakasan, An Application-specifid Protocol Architecture for Wireless Microsensor Networks. *IEEE Transaction on Wireless Communications*, vol.1, No.4, pp. 660-670, 2002.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 603-614

Fuzzy Local Trend Transform based Fuzzy Time Series Forecasting Model

J. Dan, F. Dong, K. Hirota

Jingpei Dan, Fangyan Dong, Kaoru Hirota
Tokyo Institute of Technology
Japan, 226-8502 Yokohama, 4259 Nagatsuta, Midori-ku,
E-mail: {dan,tou,hirota}@hrt.dis.titech.ac.jp

Jingpei Dan
Chongqing University
P.R.China, 400044 Chongqing, 174 Shazhengjie, Shapingba
E-mail: danjingpei@hotmail.com

Abstract: A fuzzy local trend transform based fuzzy time series forecasting model is proposed to improve practicability and forecast accuracy by providing forecast of local trend variation based on the linguistic representation of ratios between any two consecutive points in original time series. Local trend variation satisfies a wide range of real applications for the forecast, the practicability is thereby improved. Specific values based on the forecasted local trend variations that reflect fluctuations in historical data are calculated accordingly to enhance the forecast accuracy. Compared with conventional models, the proposed model is validated by about 50% and 60% average improvement in terms of MLTE (mean local trend error) and RMSE (root mean squared error), respectively, for three typical forecasting applications. The MLTE results indicate that the proposed model outperforms conventional models significantly in reflecting fluctuations in historical data, and the improved RMSE results confirm an inherent enhancement of reflection of fluctuations in historical data and hence a better forecast accuracy. The potential applications of the proposed fuzzy local trend transform include time series clustering, classification, and indexing.
Keywords: time series forecasting, fuzzy time series, trend, transform.

1 Introduction

Based on Zadeh's works (see [1] and [2]), the concept of fuzzy time series and its models for forecasting have been proposed to solve the forecasting problems where the historical data are linguistic values (see [3]- [5]). Conventional fuzzy time series forecasting models that are based on fuzzy time series of original data are limited to forecasting specific values that do not reflect fluctuations in historical data. Local trend variations, however, are mainly concerned with real applications. For example, forecast of changing direction of stock price are more important for stock investors to make reasonable determinations than specific forecast values of stock price. In addition, the forecasted specific demand values are unreliable since historical data is distorted when it transfers along the supply chain due to the bullwhip effect, so local trend variations of demand that are not suffered from the bullwhip effect are more valuable and practical for supply chain managers. Therefore the practicability of conventional fuzzy time series forecasting methods suffers from the limitation of forecasting specific values. This study aims to improve the

practicability and forecast accuracy by forecasting local trend variations that reflect fluctuations in historical data. It should be noted that the word "trend" usually refers to the long-term trend in statistics, whereas as used in this paper word "trend" means local trend variation in short term or during one period.

Recently, some trend involved fuzzy time series models have been proposed to improve forecasting. Huarng has proposed heuristic models by integrating problem-specific heuristic knowledge with Chen's model [6] to improve forecasting by reflecting the fluctuations in fuzzy time series [7]. A trend-weighted fuzzy time series model for forecasting Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) has been proposed in [8]. Chen and Wang have proposed a method to predict TAIEX based on fuzzy-trend logical relationship groups to improve forecast accuracy [9]. The interval rearranged method has been proposed to reflect fluctuations in historical data and improve forecast accuracy of fuzzy time series in [10]. Although all these methods are involved with trends, they are intrinsically conventional fuzzy time series forecasting methods since they are all based on original data and their forecasting targets are specific values. Fuzzy local trend transform is proposed to provide a different forecasting basis by transforming original data into a linguistic representation of local trend variations called fuzzy local trend time series. In contrast to conventional fuzzy time series forecasting models, local trend variations are forecasted based on the transformed fuzzy local trend time series in the proposed model. Forecast accuracy of specific values is hence enhanced by forecasted local trend variations that reflect fluctuations in historical data.

Three typical forecasting targets, enrollment forecasting, stock index forecasting, and inventory demand forecasting are used to validate the proposed model. To make an effective evaluation, forecasts are evaluated by two measures from different aspects. Root mean squared error (RMSE) is used to evaluate forecast accuracy of specific values while mean local trend error (MLTE) measure [11] is used to evaluate how accurately forecast reflects fluctuations in historical data. For enrollment forecasting, the proposed model outperforms typical fuzzy time series models in terms of RMSE and MLTE. Especially, comparing to Chen's model [6], Huarng's model [12], and Cheng et al.' model [13], MLTE results show an improvement of 73.3%, 55.6%, and 66.7%, respectively. For TAIEX forecasting, the proposed model gets the smallest RMSE result while the second rank in terms of MLTE compared to Chen's model [6], Yu's model [14], and Cheng's model [8]. For inventory demand forecasting, compared to Huarng and Yu' model [15], Cheng et al.' model [13], and Chen and Wang' model [9], the proposed model yields about 50%, 50%, and 33.3% improvement in terms of MLTE and 48.3%, 73.3%, and 49.1% improvement in terms of RMSE. The MLTE results demonstrate that the proposed model outperforms conventional fuzzy time series models in significantly reflecting fluctuations in historical data, and the improved RMSE results confirm an inherent enhancement of reflection of fluctuations in historical data and hence a better forecast accuracy.

The rest of the paper is organized as follows: in section 2, fuzzy time series and fuzzy c-means clustering are briefly reviewed. The proposed fuzzy local trend transform based fuzzy time series forecasting model is elaborated in section 3. Empirical analyses on three forecasting targets to demonstrate the proposed model are illustrated in section 4.

2 Fuzzy Time Series and Fuzzy C-means Clustering: a Brief Review

The proposed model is based on conventional fuzzy time series forecasting model and the fuzzy c-means clustering method is applied in the proposed fuzzy local trend transform, so fuzzy time series and fuzzy c-means clustering are briefly reviewed by adjusting the notations.

2.1 Fuzzy time series

Let U be the universe of discourse, where $U = \{u_1, u_2, \cdots, u_b\}$. A fuzzy set A_i of U is defined as $A_i = \sum_{j=1}^{b} f_{A_i}(u_j)/u_j$, where f_{A_i} is the membership function of the fuzzy set A_i; $f_{A_i} : U \to [0, 1]$, u_a is a generic element of fuzzy set A_i and $f_{A_i}(u_a)$ is the degree of belongingness of u_a to A_i; $f_{A_i}(u_a) \in [0, 1]$ and $a \in [1, b]$. In [3], the general definitions of fuzzy time series are given as follows:

Definition 1. Let a subset of real numbers $Y(t)(t = \cdots, 0, 1, 2, \cdots)$ be the universe of discourse by which fuzzy sets $f_j(t)$ are defined. If $F(t)$ is a collection of $f_1(t), f_2(t), \cdots$, then $F(t)$ is called a fuzzy time series defined on $Y(t)$.

Definition 2. If fuzzy time series relationships assume that $F(t)$ is caused only by $F(t-1)$, then the relationship can be expressed as: $F(t) = F(t-1) * R(t, t-1)$, which is the fuzzy relationship between $F(t)$ and $F(t-1)$, where $*$ represents as an operator. (Note that the operator can be either max-min [4], min-max [5], or arithmetic operator [6].)

To sum up, let $F(t-1) = A_i$, and $F(t) = A_j$, the fuzzy logical relationship between $F(t)$ and $F(t-1)$ can be denoted as $A_i \to A_j$, where A_i refers to the left-hand side and A_j refers to the right-hand side of the FLR. Furthermore, these fuzzy logical relationships can be grouped to establish different fuzzy relationship. These groups are called fuzzy logical relationship groups (FLRGs). On account of its simplicity, FLR method is chosen by most researchers.

The procedure for forecasting using conventional fuzzy time series models has four main steps: (1) Define universe of discourse and intervals; (2) Define fuzzy sets and fuzzify observations in the original time series; (3) Establish fuzzy relationships; (4) Forecast and defuzzify the outcome. Assume that the current state of $F(t)$ is A_i, $F_{def}(t+1)$ can be forecasted and defuzzified by the following rules:

Rule 1: If there is $A_i \to A_j$ in fuzzy logical relationship groups, then $F(t+1) = A_j$ and defuzzified as $F_{def}(t+1) = center_j$, where $center_j$ is the center of cluster j to which A_j belongs.

Rule 2: If there is $A_i \to \#$ in fuzzy logical relationship groups, then $F(t+1) = A_i$ and defuzzified as $F_{def}(t+1) = center_i$, where $\#$ represents null value and $center_i$ is the center of cluster i to which A_i belongs.

Rule 3: If there is $A_i A_1, A_2, \cdots, A_j$ in fuzzy logical relationship groups, then the forecast at $t+1$ is calculated as $F_{def}(t+1) = (center_1 + center_2 + \cdots + center_j)/j$, where c_j is the center of cluster to which A_j belongs.

2.2 Fuzzy c-means clustering

Fuzzy c-means (FCM) clustering is a method of clustering which allows one piece of data to belong to two or more clusters, and is frequently used in pattern recognition [16].

The FCM is based on minimization of the objective function

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \|x_i - c_j\|^2, 1 \le m \le \infty, \tag{1}$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the i-th of d-dimensional measured data, c_j is the d-dimension center of the cluster, and $\| * \|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function (1), with the update of membership $u_{ij} = 1/\sum_{k=1}^{C} \left(\|x_i - c_j\|/\|x_i - c_k\|\right)^{2/m-1}$ and the cluster centers $c_j = \sum_{i=1}^{N} u_{ij}^m (x_i / \sum_{i=1}^{N} u_{ij}^m)$.

This iteration will stop when $Max_{ij}|u_{ij}^{k+1} - u_{ij}^{k}| < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas k is the iteration step. This procedure converges to a local minimum or a saddle point of J_m.

3 Proposed Fuzzy Local Trend Transform based Fuzzy Time Series Forecasting Model

In literature, trends are widely represented by using absolute variations, slopes, or relative variations between two consecutive points in literature. In section 3.1, trends or local trend variations in study are defined as relative variations, or ratios. The algorithm of fuzzy local trend transform based on local trend variations is then elaborated in section 3.2. The fuzzy time series forecasting model based on fuzzy local trend transform is presented in section 3.3.

3.1 Local trend variations

To address the limitation of absolute variations [17] and slopes([18]- [20])for representing local trend variations, relative variations, which are the ratios between two consecutive data points in a given historical time series, are adopted to indicate local trend variations in this study.

Assuming that for any given time series $P(t)$, $t = 1, 2, \cdots, n$, $n \in N$, local trend variation between time t and $t-1$ is defined as

$$r_t = (P(t) - P(t-1))/P(t-1), t = 2, 3, \cdots, n, \tag{2}$$

then the time series of local trend variations for $P(t)$ is defined as $T(t) = r_t$, $t = 2, \cdots, n$.

The reasons for forecasting based on local trend variations instead of original time series data are explained as follows. First, the forecasts based on original time series data may not reflect the fluctuations in historical data properly. In most previous studies, the forecasts are equal at some consecutive points which indicated that forecasting based on original time series are not appropriate for reflecting fluctuations [15].

Second, original time series data varies dramatically in different contexts while the time series of local trend variations varies slightly.

Finally, forecasting based on local trend variations are more suitable for reflecting fluctuations in historical data since directions and variation degrees of local trend variations can be indicated by signs and magnitudes of ratios easily. Ratios are preferable in terms of demonstrating the differences in various contexts, ratios-based lengths of intervals are hence adopted to improve fuzzy time series in [15].

Forecasting based on local trend variations are therefore considered more suitable for reflecting fluctuations in historical data and forecast accuracy should be further improved inherently.

3.2 Fuzzy local trend transform

To forecast local trend variations, original time series is represented by linguistic local trend variations of original data firstly, the algorithm of fuzzy local trend transform is hence proposed as follows:

Step 1 : Obtain the local trend variation time series $T(t)$ by calculating ratios between each two consecutive data points in the original time series $P(t)$ in terms of equation (2).

Step 2: Divide $T(t)$ into three basic clusters in terms of local trend changing direction, i.e., decreasing cluster T_d, unchanged cluster T_u, and increasing cluster T_i. It is easy to determine T_d and T_i in terms of sign of ratios after T_u is determined. Assume that the interval for unchanged

Table 1: Parameter ŚÁ for determining the interval of unchanged cluster

$Max(\|T(t)\|)(\times 10^{-2})$	$Max(\|T(t)\|) \leq 1$	$Max(\|T(t)\|) \leq 10$	$Max(\|T(t)\|) \leq 20$	$\cdots$
$\alpha(\times 10^{-2})$	0.01	0.1	0.2	$\cdots$

cluster is $[-\alpha, \alpha]$, α is determined by $Max(|T(t)|)$ according to Table 1 since it is possible that the definition of unchanged cluster varies from problem to problem. The observations in $T(t)$, of which the values are greater than α, are then assigned to T_i, while the observations of which the values less than $-\alpha$ are assigned to T_d.

Step 3: Divide T_i and T_d into c_i and c_d clusters by applying FCM, respectively. Assume that the number of observations in T_i and T_d are n_i and n_d, respectively, then the number of clusters for T_i, T_d and $T(t)$ are predefined by users as c_i $(2 \leq c_i \leq n_i)$, c_d $(2 \leq c_d \leq n_d)$, and c $(c = c_d + c_i + 1)$, respectively. T_i and T_d are then divided into c_i and c_d clusters by FCM, respectively, as described in section 2.2. Consequently, the cluster centers and memberships with respect to the clusters are obtained.

Step 4: Fuzzify the local trend time series $T(t)$ as fuzzy local trend time series $F_T(t)$. First, the clusters of $T(t)$ are achieved by combining the clustering results obtained in Step 2 and Step 3. Then the linguistic terms A_i $(i = 1, 2, \cdots, c)$ are defined corresponding to the clusters. $T(t)$ is finally fuzzified into $F_T(t)$ by assigning A_i to $T(t)$ when the maximum membership of $T(t)$ occurs at the cluster to which A_i belongs.

3.3 Fuzzy local trend transform based fuzzy time series forecasting model

Theoretically, the proposed fuzzy local trend transform can be integrated with any conventional fuzzy time series forecasting model. Because of simplicity, the proposed model is integrated with Chen's model [6] as stated in section 2.1. The proposed model differs from Chen' model in Step 1 and Step 4 as described in the following:

Step 1: Transform original time series into fuzzy local trend time series by the proposed fuzzy local trend transform as stated in section 3.2.

Step 2: Establish fuzzy logical relationships and fuzzy logical relationship groups based on fuzzy local trend time series obtained in Step 1 as described in section 2.1.

Step 3: Forecast and defuzzify the possible outcomes of local trend variations, which is denoted as $T_{def}(t)$, $t = 1, 2, \cdots, n-1$, based on fuzzy logical relationship groups as described in section 2.1.

Step 4: Calculate specific values based on forecasted local trend variations obtained in Step 3 in terms of equation(3) that is defined as

$$P_{pre}(t) = P(t) \times T_{def}(t), t = 1, 2, \cdots, n-1, \tag{3}$$

where $P_{pre}(t)$ indicates predicted specific values.

4 Empirical Analyses on Forecasting based on Fuzzy Local Trend Time Series

To validate the proposed model, three applications are used in the empirical analyses, including student enrollment forecasting (the enrollments of the University of Alabama [4]), stock index forecasting (Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)), and inventory demand forecasting [15]. The first two data sets are algebraic growth data and widely used to validate fuzzy time series models in many relevant studies while the inventory demand data set is exponential growth data and typical in supply chain management application. We

compare the proposed model with typical fuzzy time series models in terms of two forecast accuracy measures. One is conventional measure RMSE (root mean squared error) that is commonly used to measure forecast accuracy of specific values based on quantitative error in fuzzy time series forecasting, while the other is MLTE (mean local trend error), which is proposed to measure how accurately forecasts reflect fluctuations in actual data based on local trend error [11]. It is more effective and proper for comparing the models by evaluating them from two different aspects than using one or more conventional measures that based on quantitative error. For a given time series $y_t, t = 1, \cdots, n$, the prediction of y_t is f_t, $t = 1, \cdots, n$, RMSE is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(f_t - y_t)^2}, t = 1, 2, \cdots, n, n \in N, \tag{4}$$

ĄĄ while MLTE is defined as

$$MLTE = \frac{1}{n-1}\sum_{i=1}^{n-1} E_i \times 100\%, \quad E_i = \begin{cases} 1 & sign(y_{t+1} - y_t) \neq sign(f_{t+1} - f_t) \\ 0 & sign(y_{t+1} - y_t) = sign(f_{t+1} - f_t) \end{cases} \quad t = 1, 2, \cdots, n, n \in N, \tag{5}$$

where E_i indicates the number of local trend change errors, sign indicates the operator for outputting the sign of the operand. When the predicted local trend variation is inconsistent with the original one in the same interval, E_i is equal to 1, otherwise E_i equals to 0 [11].

The experiments are implemented using Matlab R2010a. In section 4.1, the forecasting algorithm is illustrated step by step with the example of enrollment forecasting. Performance of the proposed model for stock index and inventory demand forecasting are analyzed in section 4.2 and section 4.3, respectively.

4.1 Forecasting enrollments

The yearly data of student enrollments of the University of Alabama from 1971 to 1992 are commonly used in previous studies on fuzzy time series to validate fuzzy time series models. To make proper comparison with other models, the same data set is used in this study to validate the proposed model. The procedure of forecasting enrollments by the proposed model is as follows:

Step 1: Transform original time series into fuzzy local trend time series by the proposed fuzzy local trend transform.

(1) Obtain the local trend time series $T(t)$ as shown in the third column of Table 2 by calculating the local trend variations of the enrollments between each two consecutive years in the original time series $P(t)$ in terms of equation(2) .

(2) Divide $T(t)$ into decreasing cluster T_d, unchanged cluster T_u, and increasing cluster T_i. Since none of the observations in $T(t)$ is equal to 0 and maximum absolute value of the observations in $T(t)$ is 7.6675%, α is determined to be 0.1% according to Table 1. Consequently, $T_d = \{-5.8274, -3.1385, -2.3840, -2.2714, -0.9638\}$, $T_u = \{0.0466\}$, and $T_i = \{0.1189, 0.4147, 0.6664, 1.6535, 1.8872, 1.9071, 2.2414, 3.8912, 4.5179, 5.1987, 5.4145, 5.4742, 5.9643, 5.9782, 7.6576\}$.

(3) Divide T_i and T_d into c_i into c_d clusters by applying FCM, respectively. For FCM, the predefined number of clusters $c = c_d + c_i + 1$. To make fair comparison, the total number of clusters is predefined to be 7 in this study as the same as used in the previous studies. Assume that $c_d = c_i = 3$, the cluster centers of T_d and T_i and the membership grades of T_d and T_i are shown in Table 3 and Table 4, respectively. Naturally, the center of unchanged cluster T_i is 0%.

(4) Fuzzify the local trend time series $T(t)$ into fuzzy local trend time series $F_T(t)$ as shown in the fourth column of Table 2. Combine the clustering results obtained in Step (2) and Step

Table 2: Forecasting local trend variations and specific values of the enrollments

Year	Actual enrollment	Local trend variation	Fuzzified local trend variation	Forecasted local trend variation ($\times 10^{-2}$)	Forecasted enrollment
1971	13055	-	-	-	-
1972	13563	3.8912	A6	-	-
1973	13867	2.2414	A5	0.1233	13580
1974	14696	5.9782	A7	1.1623	14028
1975	15460	5.1987	A6	3.8752	15266
1976	15311	-0.96378	A3	0.1233	15479
1977	15603	1.9071	A5	1.2224	15498
1978	15861	1.6535	A5	1.1623	15784
1979	16807	5.9643	A7	1.1623	16045
1980	16919	0.66639	A5	3.8752	17458
1981	16388	-3.1385	A2	1.1623	17116
1982	15433	-5.8274	A1	-2.3003	16011
1983	15497	0.4147	A5	1.2224	15622
1984	15145	-2.2714	A2	1.1623	15677
1985	15163	0.11885	A5	-2.3003	14797
1986	15984	5.4145	A7	1.1623	15339
1987	16859	5.4742	A7	3.8752	16603
1988	18150	7.6576	A7	3.8752	17512
1989	18970	4.5179	A6	3.8752	18853
1990	19328	1.8572	A5	0.1233	18993
1991	19337	0.046565	A4	1.1623	19553
1992	18876	-2.384	A2	-2.5770	18839
RMSE					438.18
MLTE					21.0526%

(3) to achieve the clusters of $T(t)$, the linguistic variables are then defined as shown in Table 5 according to the clusters. Each local trend variation in $T(t)$ is fuzzified by the linguistic variable to which the maximum membership belongs in terms of the results in Table 3 and Table 4.

Table 3: Membership grades of decreasing cluster for each linguistic variable

clusters	Cluster Centers	Linguistic variables	$T_d(1)$	$T_d(2)$	$T_d(3)$	$T_d(4)$	$T_d(5)$
1	-2.5770	A2	0.0000018	0.9	0.9785	0.9407	0.000055
2	-5.8231	A1	0.9999974	0.0393	0.0031	0.0070	0.000006
3	-0.9758	A3	0.0000008	0.0607	0.0184	0.0523	0.999939
		$F_T(t)$	A1	A2	A2	A2	A3

Step 2: The FLRs of $F_T(t)$ are established as shown in Table 6 according to Definition 3 in 2.1. Then, the FLRs are rearranged into FLRGs as shown in Table 7.

Step 3: The possible outcomes of local trend variations from 1973 to 1992 are forecasted and defuzzified as shown in the fifth column of Table 2.

Table 4: Membership grades of increasing cluster for each linguistic variable

clusters	Cluster Centers	Linguistic variables	$T_i(1)$	$T_i(2)$	$T_i(3)$	$T_i(4)$	$T_i(5)$	$T_i(6)$	$T_i(7)$	$T_i(8)$
1	4.3997	A6	0.0603	0.0387	0.0215	0.0238	0.0639	0.0684	0.1719	0.9139
2	1.2224	A5	0.9077	0.9416	0.9680	0.9667	0.9123	0.9063	0.7715	0.0332
3	6.0036	A7	0.0319	0.0197	0.0105	0.0095	0.0238	0.0253	0.0566	0.0530
		$F_T(t)$	A5	A5	A5	A5	A5	A5	A5	A6
			$T_i(9)$	$T_i(10)$	$T_i(11)$	$T_i(12)$	$T_i(13)$	$T_i(14)$	$T_i(15)$	
1	4.3997	A6	0.9924	0.4936	0.2486	0.1929	0.00063	0.00027	0.1947	
2	1.2224	A5	0.0013	0.0199	0.0146	0.0123	0.00007	0.00003	0.0499	
3	6.0036	A7	0.0063	0.4864	0.7371	0.7948	0.9993	0.9997	0.7554	
		$F_T(t)$	A6	A6	A7	A7	A7	A7	A7	

Step 4: The enrollment values from the year 1973 to 1992 are calculated based on the forecasted local trend variations obtained in Step 3 as shown in the last column of Table 2. The number of intervals of the universe of discourse affects forecasting results [21]. When analyzing the sensitivity of c, which varies in $\{7, 9, 11, 13\}$, the RMSE and MLTE results $\{502.401, 440.3566, 380.8953, 313.2307$

Table 5: Cluster centers of local trend variations of enrollments

Cluster Centers ($\times 10^{-2}$)	Linguistic variables
-5.8231	A1(big decrease)
-2.5770	A2(decrease)
-0.9758	A3(small decrease)
0	A4(almost unchanged)
1.2224	A5(small increase)
4.3997	A6(increase)
6.0036	A7(big increase)

and $\{23.1579\%, 24.2105\%, 20\%, 13.6842\%\}$ respectively. The average RMSE and MLTE results indicate that forecast error decrease as the number of clusters increases, in other words, the bigger the number of clusters the higher the forecast accuracy.

Table 6: Fuzzy logical relationships of local trend variations of enrollments

A6→A5	A5→A7	A7→A6	A6→A3	A3→A5	
A5→A5	A5→A7	A7→A5	A5→A2	A2→A1	
A1→A5	A5→A2	A2→A5	A5→A7	A7→A7	
A7→A7	A7→A6	A6→A5	A5→A4	A4→A2	A2→#

Table 7: Fuzzy logical relationship groups of local trend variations of enrollments

Group 1: A1→A5	Group 2: A2→A1,A5,#	Group 3: A3→A5	Group 4: A4→A2
Group 5: A5→A2, A4, A5, A7	Group 6: A6→A3, A5	Group 7: A7→A5, A6, A7	

Table 8: Comparisons for enrollment forecasting under the same conditions

Year	Actual enrollment	Chen [6]	Huarng [12]	Cheng et al. [13]	The proposed model
1971	13055	-	-	-	-ĄĄ
1972	13563	14000	14000	14242	-ĄĄ
1973	13867	14000	14000	14242	13580
1974	14696	14000	14000	14242	14028
1975	15460	15500	15500	15474.3	15266
1976	15311	16000	15500	15474.3	15479
1977	15603	16000	16000	15474.3	15498
1978	15861	16000	16000	15474.3	15784
1979	16807	16000	16000	16146.5	16045
1980	16919	16833	17500	16988.3	17458
1981	16388	16833	16000	16988.3	17116
1982	15433	16833	16000	16146.5	16011
1983	15497	16000	16000	15474.3	15622
1984	15145	16000	15500	15474.3	15677
1985	15163	16000	16000	15474.3	14797
1986	15984	16000	16000	15474.3	15339
1987	16859	16000	16000	16146.5	16603
1988	18150	16833	17500	16988.3	17512
1989	18970	19000	19000	19144	18853
1990	19328	19000	19000	19144	18993
1991	19337	19000	19500	19144	19553
1992	18876	19000	19000	19144	18839
RMSE ĄĄ		646.79	477.91	466.17	438.18
MLTE		78.9474%	47.3684%	63.1579%	21.0526%

The proposed method is compared with typical models for enrollment forecasting, including Chen's model [6], Huarng's model [12], and Cheng et al.' model [13]. To make fair comparison, seven linguistic variables are defined in all the compared models and the RMSEs and MLTEs are computed with reference to the year 1973. As shown in Table 8, the comparative results show that the proposed method outperforms the other models in terms of RMSE and MLTE.

Especially, comparing to Chen's model, Huarng's model, and Cheng et al.' model, the proposed model makes about 73.3%, 55.6%, and 66.7% improvements in terms of MLTE, respectively.

4.2 Stock index forecasting

Table 9: Comparisons for TAIEX forecasting

Date	Actual Index	Actual local trend ($\times 10^{-2}$)	Chen [6]	Yu [14]	Cheng [8]	The proposed model	Forecasted local trend ($\times 10^{-2}$)
00/11/02	5,626.08	1.477	5300	5340	5463.85	5524.81	-0.35
00/11/03	5,796.08	2.933	5750	5721.67	5644.8	5611.08	-0.27
00/11/04	5,677.30	-2.0922	5450	5435	5797.8	5815.39	0.33
00/11/06	5,657.48	-0.35033	5750	5721.67	5690.9	5657.47	-0.35
00/11/07	5,877.77	3.7478	5750	5721.67	5673.06	5648.99	-0.15
00/11/08	6,067.94	3.134	5750	5760	5871.32	5897.36	0.33
00/11/09	6,089.55	0.35487	6075	6062.5	6042.47	6088.09	0.33
00/11/10	6,088.74	-0.013303	6075	6062.5	6061.92	6093.84	0.07
00/11/13	5,793.52	-5.0957	6075	6062.5	6061.19	6079.54	-0.15
00/11/14	5,772.51	-0.36397	5450	5435	5795.5	5685.36	-1.87
00/11/15	5,737.02	-0.61861	5450	5435	5776.59	5763.81	-0.15
00/11/16	5,454.13	-5.1867	5450	5435	5744.65	5728.37	-0.15
00/11/17	5,351.36	-1.9204	5300	5340	5409.92	5352.29	-1.87
00/11/18	5,167.35	-3.561	5350	5350	5317.42	5304.12	-0.88
00/11/20	4,845.21	-6.6486	5150	5150	5151.81	5070.94	-1.87
00/11/21	5,103.00	5.0517	4850	4850	4861.89	4953.18	2.23
00/11/22	5,130.61	0.53814	5150	5150	5093.9	5120.92	0.35
00/11/23	5,146.92	0.31689	5150	5150	5118.75	5134.17	0.07
00/11/24	5,419.99	5.0382	5150	5150	5213.56	5150.48	0.07
00/11/27	5,433.78	0.25378	5300	5340	5459.32	5439.04	0.35
00/11/28	5,362.26	-1.3338	5300	5340	5391.6	5425.62	-0.15
00/11/29	5,319.46	-0.80459	5350	5350	5327.23	5314.92	-0.88
00/11/30	5,256.93	-1.1895	5350	5350	5288.71	5311.49	-0.15
00/12/01	5,342.06	1.5936	5250	5250	5232.44	5210.46	-0.88
00/12/02	5,277.35	-1.2262	5350	5350	5309.05	5327.84	-0.27
00/12/04	5,174.02	-1.9971	5250	5250	5250.81	5230.78	-0.88
00/12/05	5,199.20	0.48431	5150	5150	5157.82	5128.29	-0.88
00/12/06	5,170.62	-0.55274	5150	5150	5180.48	5202.82	0.07
00/12/07	5,212.73	0.80783	5150	5150	5154.76	5162.82	-0.15
00/12/08	5,252.83	0.7634	5250	5250	5192.66	5216.33	0.07
00/12/11	5,284.41	0.59761	5250	5250	5228.75	5256.46	0.07
00/12/12	5,380.09	1.7784	5250	5250	5257.17	5288.08	0.07
00/12/13	5,384.36	0.079304	5350	5350	5343.28	5398.00	0.33
00/12/14	5,320.16	-1.2067	5350	5350	5347.12	5376.30	-0.15
00/12/15	5,224.74	-1.8263	5350	5350	5289.34	5273.20	-0.88
00/12/16	5,134.10	-1.7655	5250	5250	5203.46	5178.54	-0.88
00/12/18	5,055.20	-1.5608	5150	5150	5121.89	5088.74	-0.88
00/12/19	5,040.25	-0.29661	5450	5405	5050.88	5010.54	-0.88
00/12/20	4,947.89	-1.8667	5450	5405	5037.42	5032.72	-0.15
00/12/21	4,817.22	-2.7126	4950	4950	4954.3	4904.19	-0.88
00/12/22	4,811.22	-0.12471	4850	4850	4836.7	4800.37	-0.35
00/12/26	4,721.36	-1.9033	4850	4850	4831.3	4803.96	-0.15
00/12/27	4,614.63	-2.3129	4750	4750	4750.42	4679.69	-0.88
00/12/28	4,797.14	3.8046	4650	4650	4654.37	4598.48	-0.35
00/12/29	4,743.94	-1.1214	4750	4750	4818.62	4813.06	0.33
00/12/30	4,739.09	-0.10234	4750	4750	4770.74	4736.76	-0.15
RMSE			176.32	170.27	121.47	114.63	
MLTE			64.44%	62.22%	26.67%	31.11%	

The daily stock index, TAIEX (Taiwan Stock Exchange Capitalization Weighted Stock Index), which is the other widely used data set in fuzzy time series studies, is used to further validate the proposed model for out-sample forecasting. The TAIEX data during 2000/01/01 - 2000/10/31 are used as training data set, and the data during 2000/11/01 - 2000/12/31 are used as testing data set.

The proposed model is compared with Chen's model [6], Yu's model [14], and Cheng's model [8]. The comparison of the forecasting results is shown in Table 9. The proposed model gets the smallest RMSE result while the second rank in terms of MLTE among the compared models. Cheng's model gets the smallest MLTE since it incorporates trend-weighting into Chen's model for TAIEX forecasting. The proposed model, however, improves forecast accuracy of specific value about 6% compared to Cheng's model by reflecting fluctuations in historical data.

4.3 Inventory demand forecasting

Demand forecasting plays a very important role in supply chain management. To further validate the applicability of the proposed method for demand forecasting, an inventory demand data set [15] is used in this study. This data set has been used in several previous studies, so it is proper for fair comparison. Inventory demand data from 1 to 19 are used as training set while data from 20 to 24 are used as testing set.

Table 10: Comparisons for inventory demand forecasting

Time	Actual inventory demand	Huarng and Yu [15]	Cheng et al. [13]	Chen and Wang [9]	The proposed model
20	227	206	205.5290	209.945	215.5516
21	223	228	216.4187	224.055	231.8985
22	242	228	216.4187	224.055	243.5307
23	239	244	216.4187	234.11	246.5969
24	266	244	216.4187	244.33	261.0037
RMSE		15.1	28.7295	14.88	7.6845
MLTE		100%	100%	75%	50%

The proposed model is compared with Huarng and Yu' model [15], Cheng et al.' model [13], and Chen and Wang' model [9]. As shown in Table 10 comparing with Huarng and Yu' model, Cheng et al.' model, and Chen and Wang' model, the MLTE results are improved by the proposed model about 50%, 50%, and 33.3%, respectively. This indicates that the proposed model outperforms the comparative models significantly in reflecting fluctuations in historical data. Consequently, the forecast accuracy of specific value is improved by the proposed model about 48.3%, 73.3%, and 49.1%, respectively, in terms of RMSE, which indicates that forecast accuracy is inherently improved by the proposed model by reflecting fluctuations in historical data.

5 Conclusion

In contrast to conventional fuzzy time series forecasting models that are based on original fuzzy time series, a different forecasting basis, fuzzy local trend time series, which is the linguistic representation of local trend variations of original data, is provided by the proposed fuzzy local trend transform. Local trend variations, which are defined as ratios between any two consecutive data points in original time series, are thereby forecasted based on fuzzy local trend time series and the specific values are are calculated accordingly. Therefore the practicability and forecast accuracy are improved by reflecting fluctuations in historical data.

The proposed model is validated by using three typical forecasting targets. The results are evaluated by two measures from different aspects. Compared to conventional fuzzy time series models, the proposed model yields about 50% and 60% average improvement in terms of MLTE and RMSE, respectively for the three application areas. The MLTE results indicate that the proposed model outperforms conventional fuzzy time series models significantly in reflecting fluctuations in historical data, and the improved RMSE results confirm an inherent enhancement of reflection of fluctuations in historical data and hence a better forecast accuracy.

Theoretically, the proposed fuzzy local trend transform can be integrated with any fuzzy time series model. Optimization of the parameters predefined in the proposed fuzzy local trend transform is being considered. A linguistic representation of time series data based on the proposed fuzzy local trend transform for the application to time series data mining is ongoing.

Bibliography

[1] L. A. Zadeh, Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Trans. Systems, Man,and Cybernet*, Vol.3, No.1, pp.28-44, 1973.

[2] L.A. Zadeh, The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Part 1, *Information Sciences*, Vol.8, No.3, pp.199-249, 1975.

[3] Q. Song, B. S. Chissom, Fuzzy Time Series and its Model, *Fuzzy Sets and Systems*, Vol.54, No.3, pp.269-277, 1993.

[4] Q. Song, B. S. Chissom, Forecasting Enrollments with Fuzzy Time Series -Part I, *Fuzzy Sets and Systems*, Vol.54, No.1, pp.1-9, 1993.

[5] Q. Song, B. S. Chissom, Forecasting Enrollments with Fuzzy Time Series -Part II, *Fuzzy Sets and Systems*, Vol.62, No.1, pp.1-8, 1994.

[6] S. M. Chen, Forecasting Enrollments based on Fuzzy Time Series, *Fuzzy Sets and Systems*, Vol.81, No.3, pp.311-319, 1996.

[7] K. H. Huarng, Heuristic Models of Fuzzy Time Series for Forecasting, *Fuzzy Sets and Systems*, Vol. 123, No.3, pp.369-386, 2001.

[8] C. H. Cheng et al, Trend-weighted Fuzzy Time Series Model for TAIEX Forecasing, *Lecture Notes in Compter Science*, Part III, Vol.4234, pp.469-477, 2006.

[9] S. M. Chen, N. Y. Wang, Fuzzy Forecasting based on Fuzzy-trend Logical Relationship Groups, *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol.40, No.5, pp.1343 - 1358, 2010.

[10] H. Y. Jung et al, Fuzzy Time Series Reflecting the Fluctuation of Historical Data, *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, pp.473-477, 2010.

[11] J. Dan et al, Mean local trend error and fuzzy-inference-based multicriteria evaluation for supply chain demand forecasting, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 15, No. 2, pp.134-144, 2011.

[12] J. R. Hwang et al, Handling Forecasting Problems Using Fuzzy Time Series, *Fuzzy Sets and Systems*, Vol.100, No.1-3, pp.217-228, 1998.

[13] C. H. Cheng et al, Multi-attribute Fuzzy Time Series Method based on Fuzzy Clustering, *Expert Systems with Applications*, Vol.34, No.2, pp.1235-1242, 2008.

[14] H. K. Yu, A Refined Fuzzy Time Series Model for Forecasting, *Physica A*, Vol.346, No.3/4, pp.657-681, 2005.

[15] K. H. Huarng, H. K. Yu, Ratio-based Lengths of Intervals to Improve Fuzzy Time Series Forecasting, *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol.36, No.2, pp.328-340, 2006.

[16] J. C. Bezdek et al, FCM: The Fuzzy C-means Clustering Algorithm, *Computers & Geosciences*, Vol.10, No.2-3, pp.191-203, 1984.

[17] J. R. Hwang et al, Handling Forecasting Problems Using Fuzzy Time Series, *Fuzzy Sets and Systems*, Vol.100, No.1-3, pp.217-228, 1998.

[18] A. Udechukwu et al, Discovering All Frequent Trends in Time Series, *The 2004 winter international symposium on information and communication technologies*, pp.1-6, 2004.

[19] C. H. Chen et al, Mining Fuzzy Frequent Trends from Time Series, *Expert Systems with Applications*, Vol.36, No.2, Part 2, pp.4147-4153, 2009.

[20] I. Z. Batyrshin, L. B. Sheremetov, Perception-based Approach to Time Series Data Mining, *Applied Soft Computing*, Vol.8, No.3, pp.1211-1221, 2008.

[21] K. H. Huarng, Effective Lengths of Intervals to Impove Forecasting in Fuzzy Time Series, *Fuzzy Sets and Systems*, Vol.124, No.3, pp.387-397, 2001.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 615-621

Floating License Management - Automation Using Web Technologies

A. Doloca, O. Ţănculescu

Adrian Doloca
Gr.T. Popa University of Medicine and Pharmacy,
Department of Mathematics and Informatics
Romania, Universităţii 16, 700115 Iaşi
E-mail: ad@umfiasi.ro

Oana Ţănculescu
Gr.T. Popa University of Medicine and Pharmacy,
Department of Fixed Prosthodontics
Romania, Universităţii 16, 700115 Iaşi
E-mail: otanculescu@eprpru.umfiasi.ro

Abstract: This paper examines the use of distributed computing based on web services with application to floating license management. The main goal is to automate the processes pertaining to the management activities and ensure at the same time that the security and flexibility requirements are met. We present the challenges posed by these requirements, propose a design and some implementation aspects using the latest .NET development platform.
Keywords: floating license management, distributed system, web service, multithreading, .NET development.

1 Introduction

Software is a vital component of today's businesses. The fast development of the software industry has produced applications for automating virtually every task, thus saving time and finances. At the same time protecting software from unauthorized usage became a strong necessity. Except the public domain that contains material not owned or controlled by anyone, all other software products are subject to the copyright law which gives the author of an original work exclusive right regarding its distribution, publication and modification. The software license is for that matter a legal instrument governing the usage and redistribution of software applications protected by copyright, but at the same time a technical means which will prevent the application from running when no authorization is available. This allows a better control of the product's distribution and ensures that illegal use will be kept to a minimum allowing the producer to economically benefit from the investment and to prevent revenue loss [1].

Concerning the method of protection, two main categories of software licensing systems are most popular: node-locked and floating licenses. Other license types and a method of optimization are presented in [2].

Node-locked licenses allow an application to be used on only one machine. This is achieved by making the license machine dependant. Therefore the license will include some unique IDs of hardware components like hard drives, CPU ID, network card ID, which will be checked as part of the license verification process.

Floating licenses (roaming licenses) allow multiple users to share access to a software product. Usually the number of users simultaneously running the application is limited, this limit being embedded in the license itself. This provides a higher degree of flexibility since users are not restricted to using the application from a designated workstation. The software can be

installed on any number of client computers but if the limit is reached a new user whishing to start it is put on hold and will be able to start its session as soon as an existing session is finished.

FlexNet and the older version *FlexLM* [3] are the de facto standards in electronic licensing being a partner of major software producing companies. These products support a large variety of license management scenarios focusing at maintaining a high degree of flexibility in choosing feature and pricing configurations. Using FlexNet software producers can generate, transfer and activate licenses and can also protect their products against piracy. [4] and [5] deliver an insight into the challenges that floating license management poses and shows how FlexLM finds solutions to these challenges.

This paper presents an original software system that provides flexibility in the management of floating licenses in enterprise environments, attaining a high degree of automation by using the latest Internet development technologies. It also discusses details of implementation and benefits of this system. The presentation is structured as follows. Chapter 2 looks into the features that an advanced floating license management system is required to have. Chapter 3 shows the structural aspects and communication between system components. Security issues generated by exchanging information over public networks are addressed in chapter 4. Some implementation aspects and recommendations can be found in chapter 5. Final conclusions are presented in chapter 6.

2 System Features and Functions

The floating license management system should have at least some features to ensure that the terms of the license contract are enforced and that this is done in a secure manner, with minimum effort by automating the necessary tasks. Below we present these features in detail.

A. Limit the number of simultaneously running application instances according to the license contract

At startup the application will connect to the license server in order to check if it has the authorization to run. If the maximum number of running instances has already been reached, the requesting application is denied the right to start and will close. When an ongoing working session is finished, the corresponding seat on the license server is freed and another application can start. Of course, some requirements exist for this feature:

- the license information must be centralized on a license server;
- the license server must be accessible in a network by all client computers which have the application installed.

B. Ensure security of the license content and prevent fraud attempts

The license content resides on the license server usually in the form of a text or a binary file which contains information pertaining to the company that acquired the license and the maximum number of simultaneously running application instances. In the proposed system the following set of parameters are used:

a) *License-ID*: uniquely identifies the license in the license database residing at the producer of the software. It contains a random number and possibly the creation date and time. This will block, for example, any attempt to use older licenses that have been issued to the same company, for the same software.

b) *Company-ID*: short identifier of the company that purchased this license. It prevents license usage across companies by simply copying the license content.

c) *Application-ID*: identifies the application the license refers to. Different software can use the same license management system by using different Application-IDs. It prevents illegal use of licenses issued for other software.

d) *Configuration-ID*: this piece of information specifies the configuration/version of the software which will be running (demo version, full version, light version, professional version, enterprise version, etc.). It ensures that only the authorized version of the software will be used.

e) *Hardware-ID*: uses one or a combination of multiple serial numbers of certain hardware pieces to ensure that the license is used only on a certain machine that was authorized for license management purposes. The most popular hardware related serials are: the hard drive ID (volume serial number), network card physical address (MAC), CPU ID. This prevents installing the license on several machines and using them simultaneously for increasing the number of running software instances over the authorized limit.

f) *MNRI (Maximum number of simultaneously running application instances)*: specifies how many instances of the software can run at the same time on client computers that use the license server. If this limit is reached, additional instances are prevented from running until an existing session is terminated.

The above listed information is combined into a unique string – the license content, which is encrypted using a strong encryption algorithm and stored in the license file. The encryption protects the license from any attempts to adjust the contained parameters for using it in other environments than the one it was issued for.

C. Provide flexibility for license configuration and installation

For maximum configuration and installation flexibility the license file will be regularly synchronized with a remote server controlled by the software producer. The first time the license server is started it will contact the remote server for the first synchronization. As a result, the license file is created and subsequently used until the next synchronization takes place. During each license synchronization, any modifications in the license parameters on the remote server are automatically transmitted to the license server. In this way no manual updating process of the license is required. This process is depicted in figure 1, in chapter 3. The synchronization ensures that the license used by the client is up-to-date and it hasn't been tampered with. At the same time all contract amendments that are made at one time are quickly and automatically implemented on the client's server.

D. Provide flexibility of working with multiple application types and multiple configurations of a software product

Due to the *Application-ID* and *Configuration-ID* parameters the license is linked to a specific software product and a version/configuration. The same license structure can be used with other software products just by adapting these two parameters accordingly. On the client side, one license server will be able to handle multiple licenses for multiple applications and configurations.

E. Offer flexible administrative functions for the client company as well as for the producer of the software

Regarding the used licenses, the client company is able to perform the following tasks:

- install license during the first synchronization;

- update license information during subsequent synchronizations;

- monitor and manage active users and licenses;

- configure intranet and internet network settings, etc.

The software producer has also an administration tool that enables them to:

- create new licenses according to specifications given in section B of this chapter;

- modify existing licenses (upgrade or downgrade);

- monitor the synchronization state of each client;

- block or activate a license, etc.

3 System Architecture

The architecture of the floating license management system is based on a distributed structure (figure 1) comprising three major components:

1) **Client workstations** running the licensed software;

2) **In-house license server** which authorizes software utilization on client workstations on one hand, and performs license synchronization with the remote server on the other hand;

3) **Remote License Server** controlled by the software producer, hosts the license database and performs automatically synchronizations over the Internet with the clients, acting as a web service.

The distributed structure using an in-house license server and a remote license server ensures centralized administration of the licenses both at the client and at the software manufacturer. Also the license installation and update are done automatically with a minimum of intervention from human operators. This accomplishes two main goals: ease of administration and security of the license content. Other advantages include use of public networks (Internet) and flexibility in case of software upgrade [6].

The client computer sends a message to the in-house license server during startup requesting authorization for software utilization. At the server, a service (e.g. Windows service) receives the message and based on the content of the license residing in the license file and on the current number of running application instances sends back a response message authorizing or denying software utilization. Using a web service client, the license server checks periodically with the remote license server if there are any modifications and, if necessary it updates the license file. If this communication fails, then a warning is issued on the in-house license server but the functionality is not affected. Checking with the remote license server is attempted again and if it fails repeatedly, after a specific amount of time, the license on the in-house license server expires, which results in blocking all software utilization on client workstations. Normal functionality is resumed after a successful communication between the in-house and the remote license servers. This mechanism ensures that the license is up-to-date at all times and that fraud attempts are avoided.

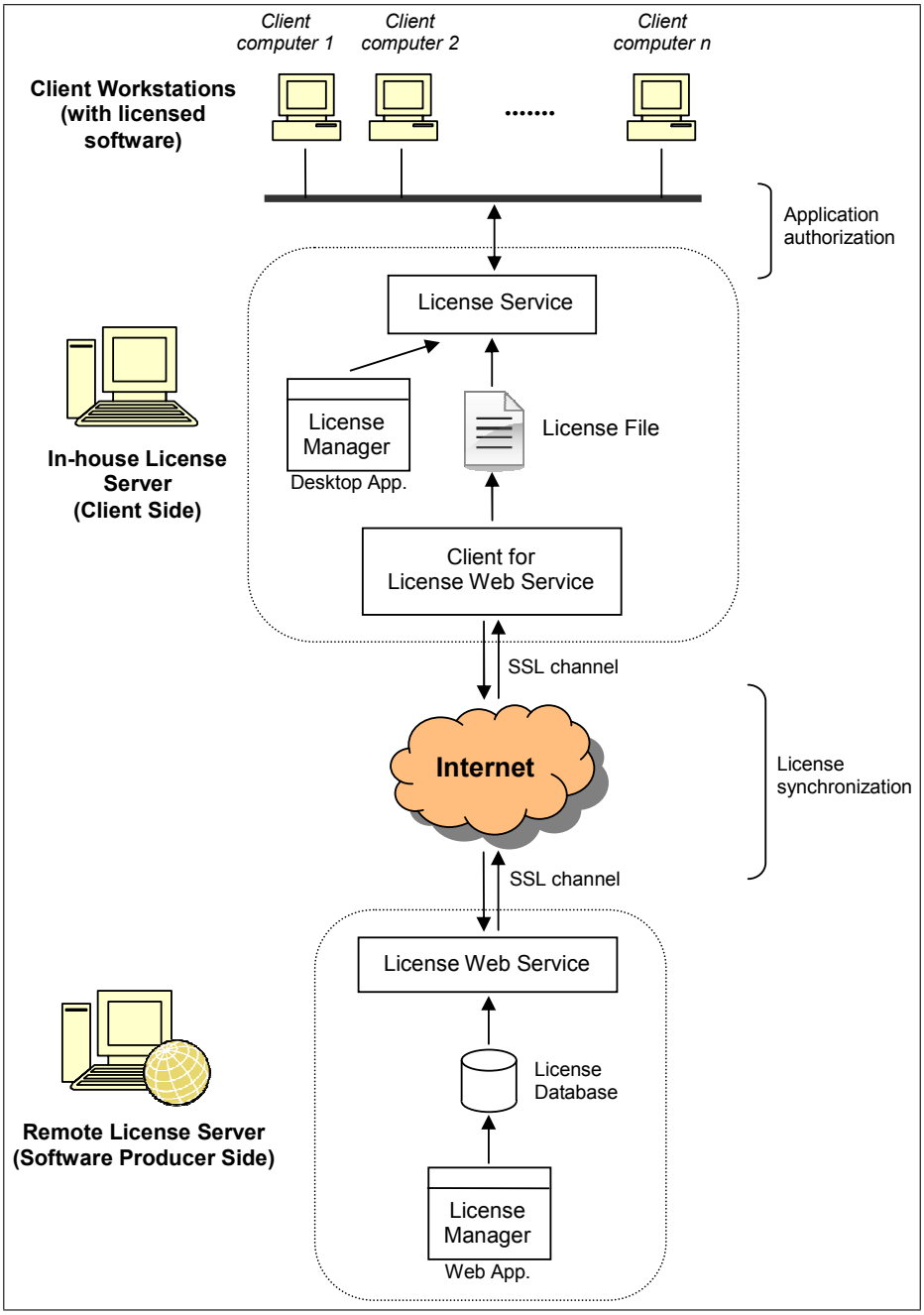


Figure 1: Floating License Management System Architecture

4 Security issues

Being a system that relies on exchanging information over public computer networks and also working with private information (i.e. the license content) in an environment that is not controlled by the software manufacturer, security has an important role. There are two aspects of this problem: securing the data transfer during the synchronization process and protecting the license content.

An effective method for dealing with the first aspect is to create an encrypted communication channel using Secure Socket Layer Connections (SSL). Thus, not only the data sent to the remote license server but also the license content received back, are protected from possible outside attacks. The subject of using SSL for Windows web services is covered in [7].

The information contained by the license file (see chapter 2, section B) must also be protected as it contains sensitive information established by the license contract. Because the information is combined into a unique string, this can be encrypted using an advanced encryption algorithm like Rijndael [8] and stored as a string representation in the license file. The license service will read and decrypt internally this information at startup.

5 Implementation aspects

The overview architecture of the license server is presented in figure 2. As the application has to be able to communicate at all times, it was implemented as a Windows service which works in the background and spawns two additional execution threads. One is the TCP Port Listener thread which receives all the requests from the local network for application start-up authorization and keeps track of the machines and active users that use the licenses. The other thread is responsible for synchronizing the license file content with the remote license server. This structure allows timely responses through both communication channels by keeping the two activities as independent as possible.

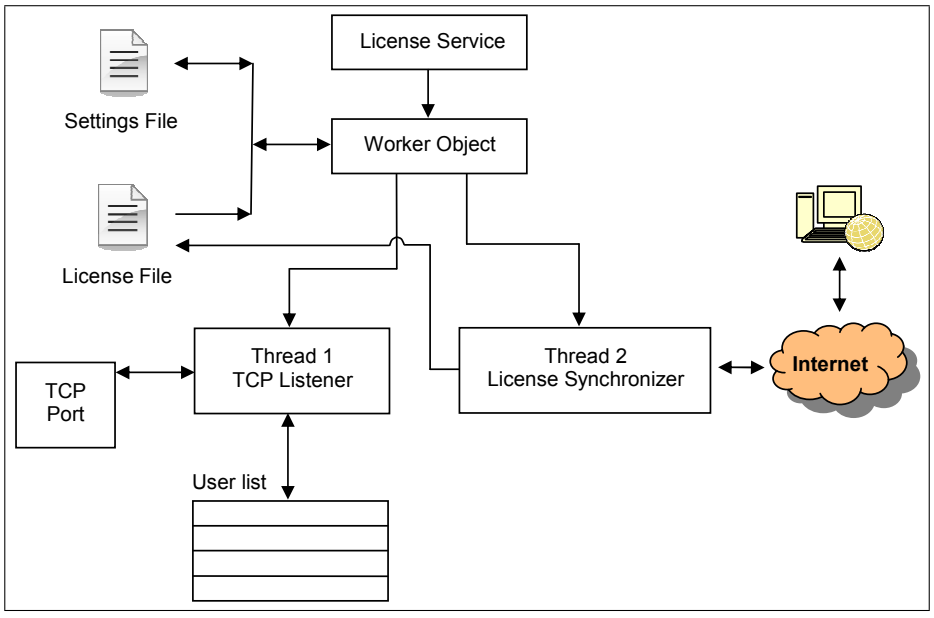


Figure 2: License Server Implementation Structure

We chose for implementation the Microsoft .NET Framework platform using the Visual Studio .NET 2008 development environment. Among the reasons:

- development support for a wide range of technologies like Web service, Windows service, Web applications and desktop applications;
- support for programming of TCP Ports;
- very good documentation.

6 Conclusions

This paper discusses the aspects of floating license management and proposes a specific system architecture that meets the flexibility requirements and offers at the same time, a high degree of automation for most of the license management activities. Using the Web service technology, license installation, activation and synchronization can be performed automatically by the system. Details of implementation using the latest .NET development framework are presented.

Bibliography

[1] R. Addy, Software License Management, *Effective IT Service Management*, pp.263-273, Springer Berlin Heidelberg, 2007.

[2] D. Gull, A. Wehrmann, Optimized Software Licensing – Combining License Types in a License Portfolio, *Gabler Verlag*, Vol.1, N0.4, pp. 277-288, August, 2009.

[3] FlexNet Producer Suite for Software Vendors, http://www.flexerasoftware.com/products/flexnet-producer-suite-software-vendors.htm.

[4] S. Sultan, Floating License Management – A Review of FlexLM, http://wob.iai.uni-bonn.de/Wob/images/36311141.pdf.

[5] FLEXlm End Users Guide, http://web.njit.edu/topics/Prog_Lang_Docs/html/flexlm/all.htm.

[6] H. Wang et al ,Web services: problems and future directions, *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol.1, Issue 3, pp. 309-320, Elsevier, April 2004.

[7] HOW TO: Secure XML Web Services with Secure Socket Layer in Windows 2000, http://support.microsoft.com/kb/307267.

[8] Advanced Encryption Standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 622-634

Network Analysis Functionality in Environmental Policy: Combining Abstract Software Engineering with Field Empiricism

N.D. Hasanagas

Nikolaos D. Hasanagas
University of Kavala
Institute of Technology Greece
E-mail:nikolaos.hasanagas@gmail.com

Abstract: An empirical application of abstract network analysis software is presented in this paper. Environmental policy networks are used as a case study. The visualization of the real network hierarchy and activity (formal and informal) is feasible only by using special software. A system of "actors" (e.g. public institutions, interest groups, enterprises) interacting with each other and dealing with a particular environmental issue constitutes a policy network, which influences the environmental policy functionality. The impacts of policy content ambiguity on network characteristics have been analyzed by using network analysis software as an example of combining algorithms with empiricism. Recommendations are made to software engineers about possible combination of algorithms with statistics and enrichment of the network analysis software with more visual analytic functions. Stronger familiarization of software engineers with policy analysis discourse and of policy analysts with positivism becomes more imperative for this purpose. On the basis of the quantitative results, environmental policy-makers are advised to invest more in trust development than in pressure and to instrumentalize more scientific information under conditions of ambiguity.
Keywords: Environmental Policy, Rural Development, Ambiguity, Network Analysis Software, Network Density.

1 Introduction

1.1 Aims and orientation

This research is presenting the results of an application of the software VISONE [3, 4] in the analysis and visualization of hierarchies and policy impacts.

The software alone is not able to be useful for analyzing practical policy issues, if the links of the network are not well defined and functionalized. This paper will show in details how software of social network analysis can be integrated into theoretical frameworks of policy analysis and in general will point out the importance of implementing software in the classical policy research in order to visualize and analyze the formal and informal hierarchies developed in policy making. The software engineers will improve the functionality of the software they develop, if they achieve to integrate it into the empirical implementation. For this purpose, they should cooperate with positivist social scientists and policy researchers.

The cases presented in this paper concern the impacts of legal acts which are characterized by ambiguous formulation. This paper is expected to interest policy makers and policy addressees (like associations of forest and land owners, environmental groups, enterprises, public agencies), as it provides information about obstacles they are going to confront with during their involvement in ambiguous environmental policy issues. Suggestions will be made for the improvement of their strategy.

Basic features of the networks will also be presented. Methodological and theoretical points will be discussed.

1.2 Literature review

The need of abstracting social phenomena in a form of algorithm-based ontology has already been pointed out on numerous research occasions [1, 2, 5, 6, 16]. Such an abstract approach is a software tool for social network analysis. It is based on Matrix Algebra and can serve as an interdisciplinary tool for in-depth comparative exploration of different policy fields and issue networks in quantitative terms. In other words, it is a functionalization of the System Theory in the analysis of policy fields [7–11, 32]. Through such a tool, the knowledge can be accessible for various learning communities [15, 17, 23, 25–30]. However, an appropriate functionalization of the links of a network on an empirical basis is necessary. Otherwise, the abstract tool will remain interesting only for mathematicians or software engineers and it will not be applicable on particular policy fields.

The necessity for cooperation between software engineers and experts of particular fields, in order to integrate applicable algorithms in the software, has been acknowledged [19–21, 33]. However, such joint projects could be many more than they are until now.

A policy network is a system of relations (exchange of information, trust, pressure) between actors (interest groups, enterprises and agencies), which deal with a particular policy issue (e.g. the certification of forestry sustainability in Denmark). What keep them together are the power relations between the actors [8,13] and of course the hope of everyone -even of the weak ones- to gain something from their networking. The power is nothing but the ability of someone to get someone else doing something, even if this may be unpleasant for the latter. The political power can be distinguished in two main dimensions: a) trust and b) pressure. Pressure may be official or unofficial and is based on the institutional dependence which may be created in transparent or intransparent framework.

The dependence on material support is a weak dimension in the networks discussed in this paper [34], it presented no substantial correlation and will not be extensively examined on this occasion.

Trust usually causes the longest-term effects because it assures legitimization to the trustee-actor and keeps the behaviour of the trustor under control even when the trustee-actor is absent and cannot supervise the trustor (e.g. the client follows the advice of the attorney even outside his office). It is understandable that the power - more precisely its unequal distribution- builds formal and informal hierarchies in a network.

The sharper the oligarchy is within the network, the fewer the conflicts are, because normally none starts a conflict, when it is evident who is going to win. Under these conditions, the network is supposed to become more sustainable [24]. However, our results allow us discussing this hypothesis more critically; It is going to be shown that though high legal ambiguity decreases oligarchy and thus lets conflicts proliferate, the network is not dissolved. In contrast, the contacts between the actors proliferate and the more policy sectors are integrated within the network (e.g. tourism, forestry, agriculture, industry, urbanization and new relationship between urban and rural areas etc).

Another usual hypothesis which is here critically examined is that the trust between the actors is restricted, when the cross-sectorality of the network increases and the contacts between the actors proliferate [13, 16, 31]. Our finding supports that though sectors and contacts proliferate under conditions of high legal ambiguity, the trust relations also proliferate and are not deconstructed. In contrast, the relations based on pressure seem to be deconstructed.

It has been found that the use of general information (political arguments) is avoided as a

mean of over-bridging sectors in networks which are characterized by high legal ambiguity ("low formalization degree") [12–14]. The hypothesis supported by our findings is that in case of high legal ambiguity the scientific information is more intensively used than the general one.

The findings will also support the hypothesis that the more non-regulative instruments (namely informative or financial means) are included in environmental policy content, the more ambiguous this content becomes [14].

1.3 Paper's Innovation

Even a software like VISONE which has been tested and proven to be appropriate for social network analysis, cannot be useful, if it is not integrated in an appropriate field-related theoretical and empirical framework. In this paper, the software VISONE is applied to forest environment policy networks by using Theory of Organized Interests. The "new relationship" between urban and rural areas can also be depicted on the interactions between interest groups and public institutions. The examination of the real (or not) "partnership" character of this relationship as provided by the European Spatial Development Perspective is also enabled.

It is an example of supporting abstract software with concrete empirical concepts. The results can be used as a feedback to software engineers as well as to policy analysts and power theorists (how network analysis software can be used for understanding policy arena hierarchies and their interaction with characteristics of policy content such as ambiguity). The innovation lies in the combination of empiricism and abstract software (Fig. 1).

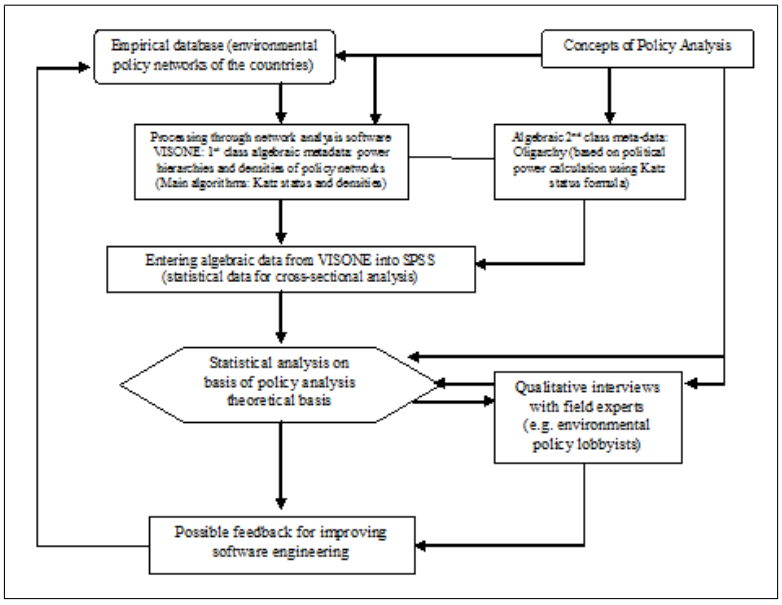


Figure 1: Supporting abstract software engineering with field research empiricism

2 The Empirical Basis

2.1 Design of the empirical research

Based on concepts of Policy Analysis such as power (trust and pressure), information, ambiguity, real policy networks are opened up in various countries and an empirical database is created. Then, the data are processed through VISONE software and statistics software SPSS

considering concepts of Policy Analysis. Qualitative interviews serve as a corrective tool during the theoretical concept building and the interpretation or the quantitative findings (Fig. 1).

The data were selected from 12 networks of environmental policy throughout 8 countries of the European Union (they will be described below). Standardized questionnaires were used. The interviewees were chairpersons, general secretaries, lobbyists or managing directors of the actors. The first actor was randomly selected from a specific list and asked to mention an environmental issue (e.g. certification of sustainability in forestry), in which he was involved in the last two years and thought that he managed to be "successful". Then, he was asked to mention the actors with whom he came in contact, within the frame of the particular issue. Next, it was asked from these actors to report those actors that they came in contact with, within the frame of the same issue, as well, and so on. Through this snowball sampling, the whole network was structured step by step till it stopped getting expanded. This is the method of the complete network analysis. Then, it can be turned into a matrix and be processed algebraically and statistically. By using Matrix Algebra, it is possible to retrace the flow of information, of trust and of the pressure, and, thus, to find the individuals that can control most of the actors, revealing the true hierarchies (formal or informal).

The variables of VISONE were processed through cross-sectional analysis (Pearson test) at significance level of 5% after normality test. Pearson test is more flexible and functional, as it offers a clearer overview of correlations among variables, in contrast to other tests such as multivariate regression which may become more rigid and restrictive regarding variable relations. The network sampling which is applied is by definition a non-random sample. However, this is no methodological weakness as long as only analytic statistics (correlations between variables) is conducted and no generalization of descriptive statistical results (e.g. mean, variance) on a larger population is aimed. Qualitative interviews with field experts (lobbyists, directors etc) have confirmed the reliability of the quantitative results and help a deeper understanding of their perception and their causal relations, as will be discussed below (Section 3).

2.2 Functionality & Variables

At first step, the details of the actor's identity are recorded so as to know whether he was a public or private one and the policy sector he is active in (agriculture, forestry, tourism, research, urbanization, etc.).

The hierarchy analysis of the network was based on the concept of oligarchy. The fewer the actors, which concentrate the power (pressure, trust), the higher the oligarchy becomes. The hierarchies are practically depicted in the form of chain control relationships: if the actor A gained actor's B trust, B gained C's trust and C gained D's trust and so on, then A could influence all: B, C and D, with just one call. Similar chains are formed in the case of pressure. An actor being on the top of numerous and longest chains, he has the highest power status in the network and he acquires the highest position in the hierarchy of the network. The previous power chains are calculated by a specific algorithm, the Katz status formula (Formula 1):

$$T = aC + a^2C^2 + ... + a^kC^k \tag{1}$$

where T is a matrix including the status values of all actors as elements, C is the matrix presenting the real network (e.g. of trust links).

For that purpose a specific software (VISONE) is used, which visualizes this hierarchy, giving it the shape of a pyramid (Fig. 2). In this way, the actors are examined as objects comparable with each other and the networks can be clearly perceived as systems [23]. Thereby, the networks become also comparable with each other, independent of the different qualitative nature of the policy issues they concern. The position (status) of each actor in the vertical Y-axis of the

pyramid is measured as power share (%) the actor possesses. Actors that are found at the same level in the pyramid hold the same percentage. The more actors at the lowest level are, the larger is the bottom X of the pyramid, and, thus, a pyramid with a large base is shaped, which shows the existence of a high oligarchy (the horizontal layout of the actors at the various levels of the pyramid has nothing to do with their power).

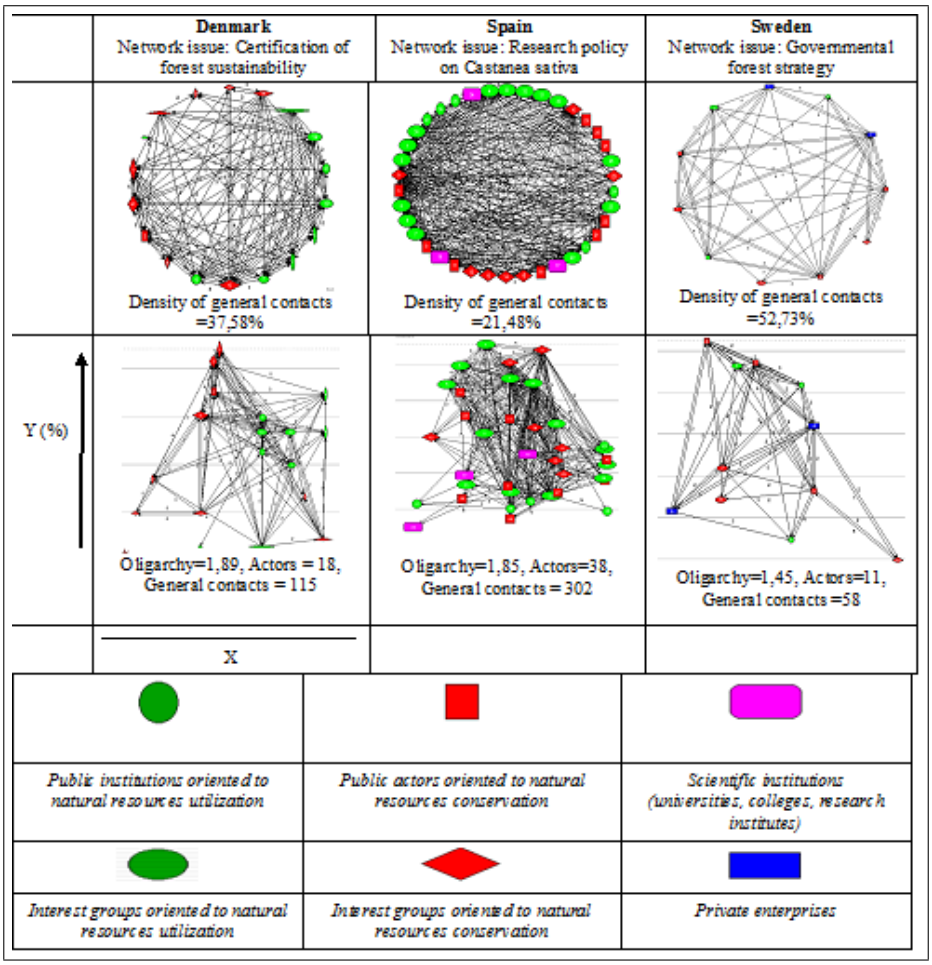


Figure 2: Examples of environmental policy networks by VISONE

The oligarchy of every network was set as the difference between the maximum (max) and the minimum (min) power value (status) that appeared in each network divided by the average power of all actors of the network and it may vary from 0 till infinity (Formula 2) [21]. Oligarchy is a meta-data derived from power and it is not measured any more as % but as a pure number. Thus, it is a mono-dimensional (single-number) indicator of the politico-administrative structure. More precisely, the links of trust and the links of pressure are primary data. The power calculated with Katz status formula is 1^{st} class meta-data and the oligarchy which calculated using power status is a 2^{nd} class meta-data [27–30].

$$Oligarchy = \frac{Status\ max - Status\ min}{Status\ Average} \tag{2}$$

Another variable is the density of the links (general contacts, conflicts, trust, pressure, exchange of general or scientific information). This variable becomes visually clear in the polygonal form of the networks in Fig. 2. If every relation is regarded as a diagonal, then the density ex-

presses the percentage of the existing diagonals divided by the total number of all possible diagonals of the polygon (Formula 3). Thus, density is a 1^{st} class meta-data based on the links number (primary data) of a particular relation type (trust, conflicts etc). In fact, high density practically means that every chance of cooperation (or conflict) within the network has been explored and used at highest possible level.

$$Density\ of\ relation\ A = Number\ of\ links\ of\ relation\ A/(N^2 - N) \tag{3}$$

where N is the number of actors participating in the network.

In general, Katz status and oligarchy provides insights about the real structure (formal and informal) of the network, about the power concentration and thereby about the coordination potential, while the density is suitable for the diagnosis of the intensity of the activity and the importance of each type of relation (general contacts, conflicts, trust, pressure, general or scientific information) for the network cohesion as well as about the saturation of policy activity (e.g. lobbying) within a network (Fig.3).

Type of data	*Structure*	*Activity*
Primary data (links)	- trust - pressure	- general contacts - conflicts - trust - pressure - information (general or scientific)
1st class meta-data	Katz status	Density
2nd class meta-data	Oligarchy	-

Figure 3: Data classification

Apart from the general contacts that are detected, more specialized relations were measured by means of the following questions:

- Trust:

To what extent would you let each of the following actors, with whom you came in contact with, to make a decision on behalf of you or you would follow their suggestions without examining them critically? 1: there is no such possibility, 2: partly, 3: to the hilt

- Institutional pressure:

Tell us for each actor, that you came in contact with, whether he was irreplaceable for you, because of a specific function he had, in the particular issue. 0: no, 1: yes

- General information:

Tell us from what actor you received information (of any kind). 0: if no information was received, 1: if there is information flow.

- Scientific information:

Name three actors from whom you received the most important scientific information. This relation was scaled with 0 and 1, as in the case of the general information.

- Conflict:

Name the actors you were in conflict with. This was scaled with 0 and 1.

- Ambiguity of policy content:

Tell us how often you needed interpretation of the laws and other rules concerning the particular issue, you were involved in, from a solicitor or other experts. 1: not at all, 2: sometimes, 3: very often. The average of the grades that the actors noted in a network showed the general ambiguity of the policy content of network issue (Fig. 3).

- Interventionism:

It has been calculated as the inverse of the ambiguity. Since the ambiguity varies from 1 to 3, the legal interventionism varies respectively from 1 to 0,33. It just constitutes a subsidiary indicator for technical-statistical reasons, which presents a significant correlation with cross-sectorality and is presented as an alternative indicator.

3 The Case study: Meta-Information of Functionality

3.1 Networks analysis

The 12 environmental policy networks are the following ones:

I) In Denmark a network of 18 "actors", concerning the certification of sustainable management of natural recourses was explored. This issue treated the enactment of specifications and control processes about the management of natural resources, so that their production potential will not be degraded. Network of the same issue (sustainability of natural resources) was opened up in II) Finland (24 actors) and in (III) Spain (38 actors). IV) In Bavaria, a network of 14 actors was opened up, which was concerning "the eco-account", namely the evaluation and the counterbalance of environmental consequences. V) Moreover, in Bavaria another network of 16 actors was opened up, which dealt with the mapping and description of forest biotopes. VI) In Sweden, a network related to the protection of biotopes (11 actors) and one more VII) related to the national forest policy (14 actors) were explored. VIII) In Greece, a network of 13 actors concerning the Article 24 of the Constitution was opened up. IX) In Scotland, the network was related to the forest policy (23 actors) and X) to the issue of the national park Loch Lomond and Trossachs (27 actors). XI) In Ireland, a network related to the provisional marketing services in forestry was examined (25 actors) and XII) in Spain, a network concerning research on Castanea genetics (21 actors) was analyzed.

All 12 networks involved 234 actors in total: 30% public agencies and 70% private actors. According to their function they can be categorized as follows: a) 43% interest groups which deal with the economic utilization of natural resources (unions of enterprises, forest industries and landowners), b) 42% actors aiming at nature conservation (e.g. institutions of environmental protection, environmental organizations), c) 6% individual enterprises (not unions of enterprises), d) 9% scientific actors (universities, research centers). In all these issue networks forest policy and rural development matters are strongly interconnected and are included in the wider field of the environmental policy discourse.

3.2 Consequences of ambiguity

Hypothesis 1

As the ambiguity increases, and the hierarchy (oligarchy) dissolves, and consequently the conflicts increase, the networks not only do not dissolve but in contrast, the contacts between the actors and the cross-sectorality increase.

In Table 1, there is a significant positive correlation between ambiguity, density of conflicts and density of general contacts. The cross-sectorality has a reciprocal relation with the legal interventionism, so practically it could be assumed that it increases with the ambiguity [12,18,31].

The more ambiguous the policy content is, the more disagreements appear about the obligations and rights of the actors and about which measures or specifications should be further institutionalized: Who will pay compensations for any restrictions imposed on natural resource utilization? What authorizations should the interest groups have? Which "risk" should be considered to be "acceptable"? etc. Such questions are steadily posed in case of ambiguous law-making. Under these conditions, the probability of conflict becomes higher. The only antidote would be a clear law. If the law is explicit enough from the beginning, it becomes clear who can participate in the network, and what he is expected or allowed to do.

As mentioned above, the fact that the networks dissolve, when the conflicts increase, is frequently supported. However, in Table 1, it is observed that apart from the conflicts that become denser to ambiguity (0,168), the general contacts among the actors become much denser (0,557). It is thus understandable that in truculent and uncertain political-administrative conditions, the actors try to broaden their coalitions in order to outbalance the chances which are lost or new threats created because of conflicts. Therefore, more and more opportunities of collaboration are used and the network is sustained, despite the increasing conflicts. Thus, the software proves that the policy networks have retroaction mechanisms.

In Table 1, it is also evident that the cross-sectorality is further developed, as the state intervensionism (inverse of ambiguity) decreases (-0,132). Actors from many different sectors can interpret the abstract aims and restrictions of a policy content, in such a way that they seem to be in accordance with their interests and capacities. An ambiguous law about the "protection of mountain water resources" makes possible the involvement of a wide spectrum of actors: from environmental organizations up to unions of construction firms, but also fresh water enterprises or angling tourism business. Of course, this may cause a proliferation of conflicts but without putting the network at risk.

In Table 1, the oligarchy decreases as the ambiguity increases (-0,441) (namely, the form of the pyramid becomes unclear, see Fig. 1). The decrease of oligarchy gives a free scope to conflicts because it is less evident who is going to win. Thus, anyone -even the weak ones- find the courage to risk a controversy. The dissolution of hierarchy is the result of the fact that it is not clear who controls whom.

Table 1. Consequences of ambiguity in conflicts, in the density of general contacts, in cross-sectorality and in oligarchy

		Ambiguity
Density of conflict-relations	Pearson Coefficient	.168
	Level of significance	.010
Density of general contacts	Pearson Coefficient	.557
	Level of significance	.000
Oligarchy (official and unofficial hierarchies)	Pearson Coefficient	-.441
	Level of significance	.000
		Interventionism
Cross-sectorality (number of involved political sectors)	Pearson Coefficient	-.132
	Level of significance	.043

Hypothesis 2

Under high ambiguity conditions (even if the general contacts get denser and there is high cross-sectorality and uncertainty in the network, as supported in Hypothesis 1), the trust relations proliferate, while the relations of pressure decrease.

The common belief is that under high ambiguity, the contacts proliferate in a network, and consequently the trust is reduced, as the potential sources of difference and threats increase and

it becomes more difficult for a larger number of participants to become familiar enough with each other and to build clear and well coordinated coalitions. However, as seen in Table 2, when the ambiguity increases, the trust seems also to increase (0,242). Simultaneously, there seems to be a minimization of the usage of pressure (-0,407). This happens because it is not clear when, how and who can activate a regulative coercive tool. Therefore, trust becomes a more preferable politico-administrative power form than pressure.

When some actors provide convincing interpretations, then they gain the trust of the rest actors. Inversely, actors, who have already gained the trust of others, thanks to their multidisciplinary staff and their stable and lawful behavior have better chances to present their interpretations as more convincing [11, 13, 18, 22, 24, 31].

Apart from that, in a network of high ambiguity a kind of "natural selection" also takes place; from the very beginning the only actors who survive are rather the reliable actors. Thus, in such a network it is more crucial for an actor to gain the trust of the others and satisfy his interest than to try to exert pressure "in the name of" an ambiguous rule with unknown results. It could be assumed that under conditions of high ambiguity, the anxiety derived from the uncertainty of the interpretation surpasses the anxiety of a new coalition. For this reason, trust can be developed.

Table 2. Consequences of ambiguity in the composition of power (trust and institutional pressure)

		Ambiguity
Density of trust relation	Pearson Coefficient	.242
	Level of significance	.000
Density of relations of pressure	Pearson Coefficient	-.407
	Level of significance	.000

Hypothesis 3

As the ambiguity increases, the use of scientific information increases, while the use of general information is restricted.

As seen in Table 3, the ambiguity creates a "friendly" or even attractive environment for the experts (lawyers, planers etc), since the need of using scientific information increases (0,250) at the expense of general information (-0,329) which can be used by any lobbyist, even by those who are not considered to be "specialized" in the particular policy issue. Because of the need of actors to be adjustable to the steadily developing "new technologies" and to the socio-political and economic changes, both expert ("scientific") knowledge on legal matters and non-legal rules (e.g. norms of nature, technology, and marketing) is demanded. Therefore, the contribution of a wide range of scientists (solicitors, archeologists, biologists etc) is indispensable, and information which is regarded as "scientific" proliferates.

As the ambiguity increases, the use of general information (matters of organizing, politics and administration) is reduced. In ambiguous issues, one expects from the scientists to play more important and intensive role in comparison with the director of public relations and the administration staff. In contrast, administrative staff can solve problems only by using formalized procedures expressed through general information and not by developing "scientific" creativeness.

Table 3. Policy impacts of ambiguity in general and scientific information

		Ambiguity
Density of scientific information	Pearson Coefficient	.250
	Level of significance	.000
Density of general information	Pearson Coefficient	-.329
	Level of significance	.000

Hypothesis 4
When a policy measure is not a direct (regulative) instrument (threat of penalty) but an indirect (economic or informational) instrument (motivating or convincing), then it is more ambiguous.

In Table 4, it is evident that the pressure from the public actors towards the private ones, which is the basis for implementing regulative tools that necessitate clear rules, tends to decrease under conditions of high ambiguity (-0,140). Ambiguity tends to appear in policy contents which include indirect instruments. Example of indirect instrument is a law which encourages the use of anti-pollution technology via the award of an ecological badge. The ambiguity of the indirect instrument is derived from the wide range of possible interpretations concerning the forms of "environment-friendly" attitude, the standards, the form, the procedure and the source of the reward (e.g. environmental tax, compensation for restrictions on the exploitation of natural resources, award of an ecological badge) and the procedure of performance assessment.

In Table 4, an increase of the number of contacts from the part of private actors towards the public ones (chances of lobbying) is also observed (0,146) under conditions of high ambiguity. This can be explained by the need of private actors to receive clear and reliable information about the law implementation. Thus, ambiguity can become a basis for new obstacles but also for new lobbying and collaborations but also between public and private actors.

Table 4. Consequences of ambiguity in the relations of public-private sector and in the institutional pressure of public actors

		Ambiguity
Number of relations of pressure exercised by public actors to private actors	Pearson Coefficient	-.140
	Level of significance	.032
Average of general contacts from the private actors towards the public ones	Pearson Coefficient	.146
	Level of significance	.026

4 Conclusion & Discussion

A software toolbox which is engineered for social network analysis can be useful for analyzing policy issues and politico-administrative systems as multi-objective entities (different types of links, hierarchies and actors). It may be applicable in improving power status, policy implementation and lobbying as well as in policy e-learning systems for lobbyists and administrators (e.g. using networks as role play scenarios). Such a software product is purposeful for detecting the real hierarchies and activities (not only the formal but also the informal ones).

Basic points for software engineers and policy analysts are the following ones:

The visualization is especially useful for this purpose. The algorithm of Katz status seems to be a useful tool, as it has been proven not only to be useful for visualizing hierarchy but also to visualize political oligarchy, at least according to the definition employed in this paper. The oligarchy indicates the potential of politico-administrative coordination. Apart from that, the oligarchy concept which is based on the power calculation and the density of trust, pressure, information and conflicts presented field-related properties such as statistical correlation with policy content ambiguity.

Another useful algorithm which presented properties interesting for policy-makers and policy analysts is this of density of various types of links, which practically indicates intensity of activity. As these properties are detected through statistical analysis, software designed for social network analysis can become more efficient, if it includes statistical functions, particularly tests, or if it provides the possibility of exporting output data to statistical software or to convert them in other

types of files which should be importable to statistical sheets. In any case, statistical analysis can hardly be isolated from social network analysis, if useful properties of social network are expected to be found. Therefore, social network analysis software should offer combined functions of Matrix Algebra and Statistics.

An additional technical advantage could be to offer the possibility of appearance of each type of links in different color within the same network file. In this way, it would be possible to visually distinguish trust from pressure links and to comparatively examine the hierarchy of trust and this of pressure within the same pyramid. Similarly, a visual comparison of different densities (trust, scientific information, general information etc) would be possible.

Before starting primary data collection and meta-data definition, the software should be integrated in theoretical frameworks of Policy Analysis and empirical research design. Without a precise definition and measurement of each type of links (e.g. trust, pressure), the quantitative metadata of such software toolbox will have no meaning for the policy field they are applied to and will not be convincing researchers and experts of empirical disciplines. Thus, a systematic collaboration between software engineers, policy analysts and policy field experts is necessary. For this purpose, software engineers should maintain narrow contact to policy analysis discourse and policy analysts should adopt more positivism.

Basic conclusions for policy-makers could be the following ones:

Although ambiguity increases the conflicts among the actors, it does not dissolve the network but increase the cross-sectorality and it lets the contacts proliferate. Moreover, it prevents the development of strict hierarchies based on pressure and favors the development of trust. In this way, it may contribute to the development of more sustainable collaborations, as trust is effective even without surveillance. In contrast, the use of pressure does not seem to be effective under conditions of ambiguity. Using information which is regarded as "scientific" is also an effective strategy. Interest groups of forest and landowners etc should pay extra attention to this point because they are normally characterized by restricted multidisciplinarity in comparison with the environmental organizations and thus, they have a more restricted persuasion potential in comparison with the environmentalists.

Actors involved in ambiguous issues must be especially well prepared for a lot of disagreements and conflicts. The role of "peacemaker" can be adopted by anyone who can gain the trust of the others and will not try to implement a forced "peace" that "suits" his interests through pressure. The successful lobbying of a private actor to the public actors is of major importance in order to assure access to explicit instructions. Ambiguous policy contents do not only cause difficulties but can also provide opportunities of satisfying interests and building new coalitions. The need for an apt and long-term lobbying becomes more imperative in the case of indirect (economic and informational) instruments, which tend to be characterized by higher ambiguity than the direct (regulative) instruments.

Acknowledgements

The research initiative proposed by this paper has been supported by the Department of Landscape Architecture, Kavala Institute of Technology, Drama, Greece), and by the Institute of Forest Policy and Nature Conservation of Goettingen University (Germany).

Bibliography

[1] D. Arotaritei, *Genetic Algorithm for Fuzzy Neural Networks using Locally Crossover.* Int. J. of Computers Communications & Control, ISSN 1841-9836, 6(1): 8-20, 2011.

[2] I. Dzitac, B.E. Bărbat, *Artificial Intelligence + Distributed Systems = Agents.* Int. J. of Computers Communications & Control, ISSN 1841-9836, 4(1): 17-26, 2009.

[3] M. Baur, M. Benkert, U. Brandes, S. Cornelsen, M. Gaertler, B. Koepf, J. Lerner, D. Wagner, *VISONE - Software for Visual Social Network Analysis.* Porc. 9th Intl. Symp. Graph Drawing, Lecture Notes in Computer Science, Springer, 2265: 463-464, 2002.

[4] U. Brandes, P. Kenis, D. Wagner *Communicating centrality in policy network drawings.* IEEE transactions on visualization and computer graphics 9(2), 241-253, 2003.

[5] P. Dolog and W. Nejdl, *Challenges and Benefits of the Semantic Web for User Modelling. Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems*, In Proc. of the 12^{th} International World Wide Web Conference, Budapest, Hungary, 2003.

[6] P. Dolog, N. Henze, W. Nejdl and M. Sintek, *Personalization in Distributed eLearning Environments.* In Proc. of the 13^{th} International World Wide Web Conference, New York, USA, 2004.

[7] N.D. Hasanagas, *Implications of the Europeanization in Trans-sectoral Environmental Policy Areas. Case study: Forestry and Nature Conservation.* Interdisciplinary Environmental Review, 3(2): 118-133, 2001.

[8] N.D. Hasanagas and R. Shoesmith, *The role of the European Parliament in forest environment issues.* European Environment, 12: 213-223, 2002.

[9] N.D. Hasanagas, A. Sachpazi, S. Kamkos, G. Tzimpridou, P. Birtsas and K. Radoglou, *Quantitative analysis of socio-political power and structures in hunting- environmental policy networks. An application of Mathematical Sociology to the Environmental Policy Analysis.* National Agriculture Research Foundation, Dept. of Publications. Athens-Thessaloniki, 2009.

[10] N.D. Hasanagas, A.D. Styliadis, E. Papadopoulou and L. Sexidis, *E- Learning & Environmental Policy: The case of a politico-administrative GIS.* Int. J. of Computers, Communications & Control, 5(5): 517-524, 2010.

[11] N.D. Hasanagas, A.D. Styliadis and E. Papadopoulou, *Environmental Policy & Science Management: Using a scientometric-specific GIS for e-learning purposes.* Int. J. of Computers, Communications & Control, 5(2): 171- 178, 2010.

[12] N. Intzesisoglou, Sociology of Law - Epistemological discussion: *The subject the Sociology of Law and its role in the Law Science* (orig. Greek). Vol. I. Publisher Sakkoula. Thessaloniki, 1990.

[13] M. Krott and N.D. Hasanagas, *Measuring bridges between sectors: Causative evaluation of cross-sectorality.* Forest Policy and Economics, 8, 555-563, 2006.

[14] E. Koutoupa-Regkakou, *Environmental Law* (orig. Greek). Publisher Sakkoula. Thessaloniki, 2005.

[15] G. McCalla, *The Fragmentation of Culture, Learning, Teaching and Technology: Implications for the Artificial Intelligence in Education Research Agenda in 2010.* International Journal of Artificial Intelligence in Education, 11: 177-196, 2000.

[16] M. Medjoudj and P. Yim, *Extraction of Critical Scenarios in a Railway Level Crossing Control System.* Int. J. of Computers, Communications & Control, 2(3): 252-268, 2007.

[17] P. Mohan, J. Greer and G. McGalla, *Instructional Planning with Learning Objects. Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems.* In Proc. Of the 18^{th} International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.

[18] Th.K. Papachristou, *Sociology of Law* (orig. Greek). Publisher Sakkoula. Thessaloniki, 1999.

[19] P.R. Polsani, *Use and Abuse of Reusable Learning Objects*". Journal of Digital information. Retrieved in September 2005 from the World Wide Web: http://jodi.ecs.soton.ac.uk/Articles/v03/ i04/Polsani, 2003.

[20] M.A. Rajan, M. Girish Chandra, L.C. Reddy and P. Hiremath, *Concepts of Graph Theory Relevant to Ad-hoc Networks*, Int. J. of Computers, Communications & Control, ISSN 1841-9836, 3(S): 465-469, 2008.

[21] A. Real and N.D. Hasanagas, *Complete Network Analysis in Research of Organized Interests and Policy Analysis: Indicators, Methodical Aspects and Challenges.* Connections 26(2), 89-106, 2005.

[22] H. Simon, *Decision-making in organizations.* Berlin, 1978.

[23] M. Stanojevi´c, M. Vujosevi´c and B. Stanojevi´c, *Number of Efficient Points in some Multiobjective Combinatorial Optimization Problems.* Int,. J. of Computers, Communications & Control, ISSN 1841-9836, 3(S): 497-502, 2008.

[24] C.N. Stone, *Systemic power in community decision-making: A restatement of stratification theory.* In: The American Political Science Review, V.74, 978-90, 1980.

[25] A.D. Styliadis, *Digital documentation of historical buildings with 3-d modeling functionality.* Automation in Construction 16, 498-510, 2007.

[26] A.D. Styliadis, *E-Learning Documentation of Historical Living Systems with 3-D Modeling Functionality.* INFORMATICA, 18(3), 419-446, 2007.

[27] A.D. Styliadis, *Historical photography-based computer-aided architectural design: Demolished buildings information modeling with reverse engineering functionality.* Automation in Construction 18, 51-69, 2008.

[28] A.D. Styliadis, D.G. Konstantinidou and K.A. Tyxola, *eCAD System Design - Applications in Architecture.* Int. J. of Computers, Communications & Control, 3(2): 204-214, 2008.

[29] A.D. Styliadis, P.G. Patias, N.C. Zestas, *3-D Computer Modeling with Intra- Component, Geometric, Quality and Topological Constraints.* INFORMATICA, 14(3), 375-392, 2003.

[30] A.D. Styliadis, I.I. Akbaylar, D.A. Papadopoulou, N.D. Hasanagas, S.A. Roussa, L.A. Sexidis, *Metadata-based heritage sites modeling with e-learning functionality.* Journal of Cultural Heritage 10, 296-312, 2009.

[31] I. Tsivakou, *The flexible boundaries of the social systems.* Publisher Nefeli. Athens, 2003.

[32] M.S. Urban and E.G. Barriocanal, *On the Integration of IEEE-LOM Metadata Instances and Ontologies.* Learning Technology Newsletter, Vol. 5 (1), 1-4, 2003.

[33] H. Wu and P. De Bra, *Suficient Conditions for Well-behaved Adaptive Hypermedia Systems.* In Proc. of the First Asia-Pacific Conference on Web Intelligence: Research and Development, Maebashi City, Japan, 2001.

[34] N.D. Hasanagas, *Power Factor Typology through Organizational and Network Analysis. Using Environmental Policy Networks as an Illustration.* Publisher Ibidem. Stuttgart, 2004.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 635-646

The Research of QoS Approach in Web Servers

Y. Hu, D. Mu, A. Gao, G. Dai

Yansu Hu, Dejun Mu, Ang Gao, Guanzhong Dai
School of Automation
Northwest Polytechnical University
Xi'an 710072, China
E-mail: huyansu@gmail.com, mudejun@nwpu.edu.cn, snailgao@gmail.com, daiguanzhong@nwpu.edu.cn

Abstract: Proportional Delay Guarantee has been widely used in the Web QoS service, and the most basic methods are the feedback of control theory and the predictive control of queuing theory. While the former belonging to passive control has a long setting time and imperfect real-time, the latter can not simulate the Web server queuing system well because of the model limitations. After the experimental verification and shortages analysis of the two methods, an improved approach is proposed in this paper. Based on the queuing feature of Web server and the HTTP 1.1 persistent connection, the improved approach predicts the delay by calculating the queue length and service rate and achieves the relative delay guarantee of different classes by adjusting their quota of worker threads. The experimental results demonstrate that the approach could maintain the relative delay guarantees well even in poor network environment and performs a much better superior compared with the traditional methods.
Keywords: Web Server, Proportional Delay Guarantee, Feedback Control, Predictive Control

1 Introduction

The increasing diversity of Web applications the last decade has witnessed an increasing demand for provisioning of different levels of quality of service (QoS) to meet changing system configuration and satisfy different client requirements. Proportional delay differentiation (PDD) service aims to ensure the QoS parameters between data flow of different classes to meet the specified proportions, so the requests with higher priority will receive the quality of service which is "at least not lower than" the low priority. There are many previous studies worked on the Web QoS and two different methods are used to achieve the differentiated service: the feedback of control theory (see [1], [2], [3], [4] and [5]) and predictive control of queuing theory (see [2], [6] and [7]). Although the algorithms are different, the QoS architecture of Web server they used is non-distinctive.

The QoS of Web application is usually related to the network layer transmission and the operating system kernel. The latter that needs to replace the operating system on all the terminals is difficult to deploy, so more and more researchers tend to the differentiation service on the application layer. It ensures the PDD service by changing the original Web FIFO mechanism and the modified Apache MPM (Multi-Processing Modules) architecture is illustrated in Figure 1.

1. The single connection queue is improved to a multi-queue structure in accordance with the classified strategy. The listener monitors the network port, accepts the client TCP connections, classifies the requests based on some classified strategy, and then puts them into the appropriate waiting queue.

2. Any class can not consume the server resource unlimitedly, so the requests of different classes must be isolated. The thread per connection structure of MPM allows all the worker threads in pool to be the resource to be allocated. So we divide the pool into several sub-pools which are isolated with each other. The number of threads in each sub-pool is known as the thread quota, and the requests are serviced in the corresponding sub-pool.

3. For N kinds of classes, let $c_i(i = 1, ..., N)$ be the thread quota allocated to the class i. When load varies, the size of sub-pool is dynamically adjusted to ensure the proportion between classes constant. And is calculated by feedback control used the delay observer to get the real delay (Figure 1 ①method), or by predictive control used the queue length observer to get the request arrival rate(Figure 1 ②method).

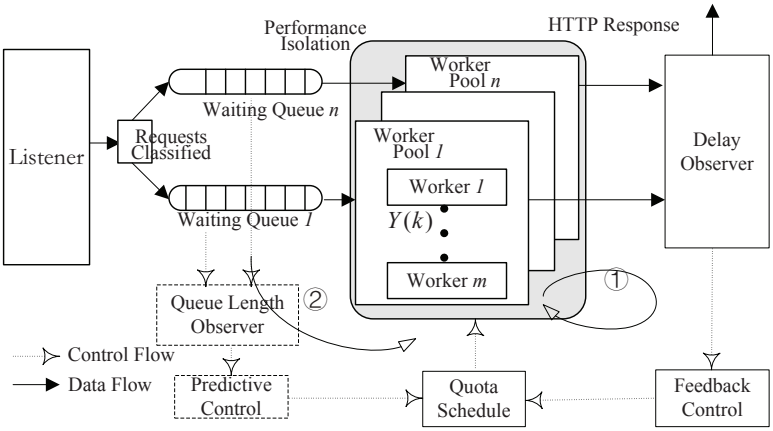


Figure 1: The Web server architecture for PDD service

But there are some imperfections and limitations in both the methods. For example, the feedback control is essentially a passive approach which works only after the deviation appeared. The delay in response is significant since the server response time is particularly slow. Although the predictive control based on queuing theory adjusts the controller output in advance, it can not get the precise plant model because of limitations itself. So an improved predictive approach is proposed to overcome their shortages and the contributions in this paper can be summarized as follows:

1) A general architecture for proportional delay service in Web server is summed up.
2) The two methods referred above are verified and compared by experiments.
3) Implement and evaluate an improved approach to get a better performance.

2 The Comparison of Two PDD Methods

2.1 The Feedback of Control Theory

Assuming C worker threads are concurrent on the Web server. For N kinds of classes, let δ_i be the constant weighting factor of class i where the higher priority has a smaller parameter value, and d_i be the expectation of class i's average measured delay, then the proportional delay guarantees can be described as follows:

$$d_i/d_j = \delta_i/\delta_j, 1 \le i \le N, 1 \le j \le N, \quad (1)$$

$$\sum_{i=1}^{N} c_i = C \tag{2}$$

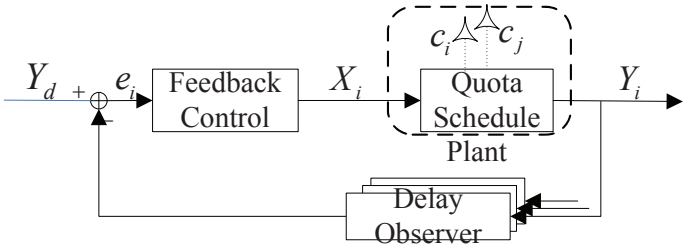


Figure 2: The feedback control model

Where c_i is the thread quota of class i. So the improved MPM is equivalent to a control model as Figure 2. The system desired output is the inherent priority parameter ratio between class i and $i+1$, which is

$$\mathbf{Y}_{\text{desire}} = [y_{1desire}, y_{2desire}, ..., y_{N-1disire}]^{\mathrm{T}}. \tag{3}$$
$$y_{idesire} = \delta_i/\delta_{i+1}, i = 1, 2, ..., N-1. \tag{4}$$

Similarly, the controller output is the thread quota ratio and $\mathbf{Y}(\mathbf{k})$ is the measured output ratio between the class i and $i+1$, which is

$$\mathbf{X}(k) = [x_1(k), x_2(k), ..., x_{N-1}(k)]^{\mathrm{T}}. \tag{5}$$
$$x_i(k) = c_i(k)/c_{i+1}(k), i = 1, 2, ..., N-1. \tag{6}$$
$$\mathbf{Y}(k) = [y_1(k), y_2(k), ..., y_{N-1}(k)]^{\mathrm{T}}. \tag{7}$$
$$y_i(k) = d_i(k)/d_{i+1}(k), i = 1, 2, ..., N-1. \tag{8}$$

Obviously, the deviation is $\mathbf{E}(k) = \mathbf{Y}(k) - \mathbf{Y}_{desire}$. On the Web server, all classes share a single host resource. So the feedback controllerďż˝ďż˝class-per-loopďż˝ďż˝adjusts $\mathbf{X}(k)$ to satisfy Equation (1) by the measured $\mathbf{E}(k)$.

The Web server is modeled as a linear system, i.e. a r-order difference equation is

$$\mathbf{Y}(k) = \sum_{l=1}^{r} [a_l \mathbf{Y}(k-l) + b_l \mathbf{X}(k-l)]. \tag{9}$$

Where a_l and b_l are the unknown parameters. The transfer function in z-domain is

$$\mathbf{G}(z) = \frac{\mathbf{Y}(z)}{\mathbf{X}(z)} = \frac{\sum_{l=1}^{r} b_l z^{r-l}}{z^r - \sum_{l=1}^{r} a_l z^{r-1}}. \tag{10}$$

The plant's mathematical model (order and parameters) is obtained by system identification which is presented by the authors in [8] in detail. Then we can design the controller based on the classical control theory. Take the PI control for example, the controller transfer function in z-domain is

$$D(z) = K_p + \frac{K_I T(z+1)}{2(z-1)}, \tag{11}$$

And the Closed-loop system transfer function is

$$G_C(z) = \frac{D(z)G(z)}{1 + D(z)G(z)}. \tag{12}$$

2.2 The Predictive Control of Queuing Theory

Although the feedback maintains the system at the balance, it takes an imperfect real-time for the lagging output. So [2, 6, 7] try to correct the deviation before the system output being affected with the help of queuing theory. In the predictive control, the queue length observer (see Figure 1②) measures the request arrival rate λ_i and service rate μ_i. Then the predictive controller reallocates the thread quota c_i for respective class.

M/M/1/∞ Queuing System

The Web server performance mainly subjects to the system bottlenecks(see [7]), so each class in Apache can be considered as a M/M/1/∞ queuing model. Define $\rho = \lambda/\mu$ be the system traffic intensity. And the residence time is the sum of queuing time (connecting time) and service time (transfer time), which is

$$l = \omega + \chi \tag{13}$$

Where ω and χ are independent from each other. According to paper [2], the mean residence time is calculated by Equation (14).

$$\bar{l} = \overline{\omega} + E[\chi] = \frac{\rho}{\mu(1-\rho)} + \frac{1}{\mu} = \frac{1}{\mu - \lambda}, (\rho < 1) \tag{14}$$

To meet the proportion relationship specified by Equation (1), we can get

$$\frac{\mu_j(k) - \lambda_j(k)}{\mu_i(k) - \lambda_i(k)} = \frac{\delta_i}{\delta_j} \tag{15}$$

While normalized the thread quota

$$s_i = c_i/C_i, 1 \le i \le N, \sum\nolimits_{i=1}^{N} s_i = 1, \tag{16}$$

Hence, for class i, the service rate can be rewritten as

$$\mu_i(k) = \mu s_i(k) \tag{17}$$

Where μ is the total service capacity. Then Equation (14) becomes $\overline{l_i} = \frac{1}{\mu s_i - \lambda_i}$. Combined with Equation (15),

$$\begin{cases} \dfrac{\mu s_2(k) - \lambda_2(k)}{\mu s_1(k) - \lambda_1(k)} = \dfrac{\delta_2}{\delta_1} \\ \dfrac{\mu s_3(k) - \lambda_3(k)}{\mu s_2(k) - \lambda_2(k)} = \dfrac{\delta_3}{\delta_2} \\ \ldots \\ \dfrac{\mu s_N(k) - \lambda_N(k)}{\mu s_{N-1}(k) - \lambda_{N-1}(k)} = \dfrac{\delta_N}{\delta_{N-1}} \\ \\ \sum_{i=1}^{N} s_i = 1 \end{cases} \tag{18}$$

Solving equations at each sampling time, we can get the thread quota for different classes $(s_1, s_2, ..., s_N)$.

M/G/1/∞ Queuing System

In M/M/1/∞ model, the service time obeys the memoryless exponential distribution. But when it is general distribution, the correlation between the current and several previous service time must be considered (see [9]). The paper [6, 10] described the "heavy tail" features of Web service time series $\{\chi_n, n \geq 1\}$, and gave a more compatible queuing model M/G/1/∞ for the Web server. According to the Pollaczek-Kinchin(PK) formula and lemma, if s_i is the normalized thread quota for class i, service time χ_i obeys the bounded pareto distribution, the mean queuing time is

$$m_{1,i} = E[\chi_i] = m_i/s_i, m_{2,i} = E[\chi_i] = m_2/{s_i}^2 \tag{19}$$

Hence the mean residence time is

$$\begin{aligned} \overline{l_i} &= \overline{\omega} + E[\chi_i] = \frac{m_{2,i}\lambda_i}{2(1-\lambda_i E[\chi])} + \frac{m_1}{s_i} = \frac{m_{2,i}\lambda_i}{2(1-\lambda_i E[\chi])} + \frac{m_1}{s_i} \\ &= \frac{m_2\lambda_i}{2s_i(s_i - m_1\lambda_i)} + \frac{m_1}{s_i} \end{aligned} \tag{20}$$

Where m_1 ,m_2 are the constants related to shape parameter of bounded pareto distribution, λ_i is the requests arrival rate which is obtained by the queue length observer. Combined Equation(20) with Equation (1),

$$\frac{s_i(s_i - m_1\lambda_i)}{s_{i+1}(s_{i+1} - m_1\lambda_{i+1})} \frac{2m_1 s_{i+1} + \lambda_{i+1}(m_2 - 2m_1^2)}{2m_1 s_i + \lambda_i(m_2 - 2m_1^2)} = \frac{\delta_{i+1}}{\delta_i} \tag{21}$$

Solve the equations similar to Equation (18) and get the results $(s_1, s_2, ..., s_N)$.

2.3 Experimental Results

The test-bed consists of a Web sever and two client machines, each with a 3.0GHz Pentium 4 professor and 521MB RAM connected with 100 Mbps Ethernet. The Web server is Apache 2(Httpd-ver2.0.53) running on Windows NT and the total number of server processes is configured to 100. The two client machines run Liunx-2.6.27 with SURGE (see [11]) (ver 1.00a) as the workload generator and each operates 120 concurrent UE. Requests are classified according to their source IP address. All the experiments are under HTTP1.1 pipeline and the number of maximum concurrent clients in SURGE is 1. Set $\delta_1/\delta_1 = 1/2$,which means the delay of high class is half of the low one. The sampling period is set to 10 sec, and all the experiments last 3000 sec. 300 valid data is measured and the controller works after 750 sec.

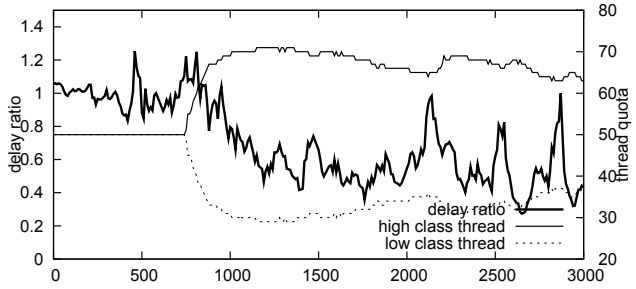
The results under PI controller and predictive controller M/G/1/∞ are respectively shown in Figure 3. Obviously, before the controllers works there is no differentiated service in different classes. But after 750 sec, the thread quota for high class increases until the expectation is satisfied. And all the methods achieve the proportional delay guarantees well.

In order to compare the features of the two methods, the mean value of the measured delay ratio l_1/l_2 and the variance relative to $\delta_1/\delta_2 = 0.5$ are defined as follows:

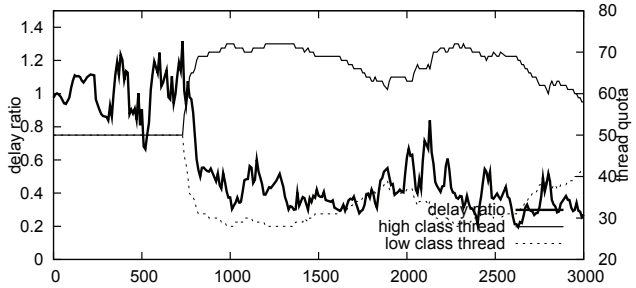
$$E_e = \frac{\sum_{k=75}^{300} l_1(k)/l_2(k)}{300 - 75} \tag{22}$$

$$D_e = \sum_{k=75}^{300} [l_1(k)/l_2(k) - \delta_1(k)/\delta_2(k)]^2 \tag{23}$$

Compute Equation (22) and (23), we get $E_e = 0.5198$,$D_e = 4.56$ under the PI controller,while $E_e = 0.4136$,$D_e = 5.5376$ under the M/G/1/∞ predictive controller.Now we can see



(a) PI control



(b) M/G/1/∞ queuing predictive control

Figure 3: The change of delay ratio and thread quota under different controllers

1. As a passive control, the PI controller works until the deviation appeared. Although the output is stabilized in the vicinity of the expectation value, there is a long setting time $t_s \approx 300s$.

2. In contrast, the latter with a much better setting time $t_s \approx 40s$ only maintains the delay proportion at 0.4. This is because the queuing theory based on PK formula asks for the statistics balance which means the requests arrival rate is equal to service rate. So the model limitation leads to the ineffaceable deviation.

3 An Improved Predictive Control

From Section 2, the queuing theory is invalid when the Web loads bursts and fluctuates dramatically. Meanwhile, for the Web queuing process, the number of requests in the front of request j can be measured. On this issue, an improved predictive control is proposed in this paper to achieve the relative delay guarantees.

3.1 Design

The relationship of the requests arrival rate, service rate and the queue length of class i are demonstrated in Figure 4(see [12]).The arrival curve $Arrvial_i(k)$ presents the rate of TCP connections, and the service curve $Service_i(k)$ presents the capacity of Web server for class i.

$$\frac{\Delta Arrival_i(k)}{\Delta k} = \lambda_i(k) \tag{24}$$

$$\frac{\Delta Service_i(k)}{\Delta k} = \mu_i(k) \tag{25}$$

When class$i$ is overload, the arrival rate is always greater than the Web service rate,i.e. $\lambda_i(k) > \mu_i(k)$.So the vertical distance between the two curves is the actual length $n_i(k)$ of waiting queue i at the k^{th} sampling time, which is $n_i(k) = Arrival_i(k) - Service_i(k)$. From the view of the tail, the waiting time $w_{i,n_i-1}$is the sum of the processing time of the top $(n_i - 1)$ requests. Considered the processing time χ_{i,n_i}itself, the mean residence time is

$$l_{i,n_i} = w_{i,n_i-1} + \chi_{i,n_i} = \sum_{j=1}^{n_i} \chi_{i,j} \tag{26}$$

If the mean residence time $l_i(k+1)$ can be predicted as $\widetilde{l_i}(k)$ based on the queue length $n_i(k)$, we can adjust the thread quota to satisfy Equation (1).

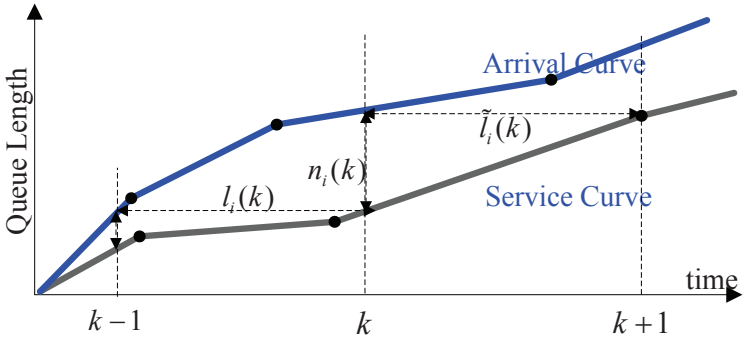


Figure 4: The arrival curve and service curve

3.2 Dealy Prediction

The Web load is actually the superposition of lots of ON/OFF process(see [11]). Just like Figure 5, Web pages are transferred during Object, and each page is composed of several embedded files which are transferred in a single TCP persistence connection under HTTP 1.1. The intervals between every embedded URL are Active OFF time corresponding to the processing time spent by browse parsing Web page. The Inactive OFF time is a longer pause corresponding to clients' "thinking procedure". Therefore, the request of a Web page consists a sequence of HTTP requests with self-similarity, which is

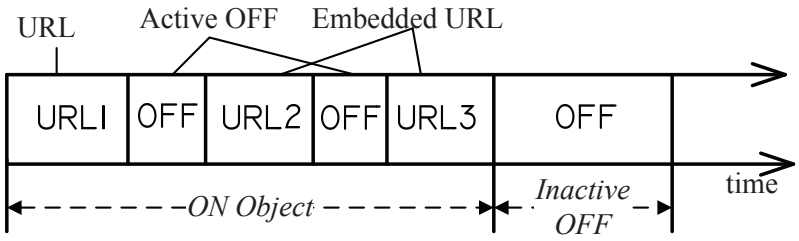


Figure 5: HTTP1.1 ON/OFF model

$$s_{i,n} = \sum_{j=1}^{y_{i,n}} x_{i,j} n = 1, 2, ...n_i(k) \tag{27}$$

Where $s_{i,n}$ means the size of the n^{th} Web page in waiting queue i, $x_{i,j}$ is the size of embedded files, and $y_{i,n}$ is the number of its embedded files. The previous research indicates that the heavy-tailed distribution is more accuracy for Web page (see [13] and [14]),and the probability mass function is:

$$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} x^{-\alpha-1}, k < x < p \tag{28}$$

Where k and p are the size of minimum and maximal file,$\alpha(0 < \alpha < 2)$ is the sharp parameter which determine the variability of distribution. Generally set $1 < \alpha < 2$to correspond to the actual communications. The number of embedded files $y_{i,n}$ follows Pareto distribution (see [11]).$y_{i,n}$ and $x_{i,j}$,$x_{i,j}$ itself are all independent from each other. The mathematical expectation of page size$s_i$is

$$E[s_i] = E[y_i]E[x_i], i = 1, 2, ...N \tag{29}$$

Let $\{s_{i,1}, s_{i,2}, ..., s_{i,n_i(k)}\}$be the size of HTTP requests on a TCP connection at the k^{th}sample. Then the total size of the page in waiting queue i to be transferred is

$$\Delta_i(k) = \sum\nolimits_{n=1}^{n_i(k)} s_{i,n} = \sum\nolimits_{n=1}^{n_i(k)} \sum\nolimits_{j=1}^{y_{i,n}} x_{i,j} \tag{30}$$

This means that the forecasting delay is the time spent by $c_i(k)$ threads concurrently processing the files of size $\Delta_i(k)$.

The paper [15] illustrates that the processing time of a static Web page is approximately linear to the size of request files. Combined the condition that the unit of the operation system is 64KB, the total size of the page to be transferred is

$$\chi(s, c_i) = b \times (s/64) + d \times c_i \tag{31}$$

Where b is the transmission coefficient, d is the overhead of threads switching and synchronization. They can be obtained by linear regression. So the total delay predicted is:

$$l_i(n_i(k), c_i(k)) = \sum\nolimits_{n=1}^{n_i(k)} \chi_i(s_{i,n}, c_i(k))/c_i(k) \tag{32}$$

$$\widetilde{l_i}(k) = E[l_i(n_i(k), c_i(k))] = \frac{n_i(k)}{64}(\frac{bE[s_i]}{n_i(k)} + 64d) \tag{33}$$

3.3 Threads Dispatch

Combined Equation (33) with Equation (1):

$$\frac{\widetilde{l_i}(k)}{\widetilde{l_{i+1}}(k)} = \frac{\delta_i}{\delta_{i+1}} = \frac{n_i(k)c_{i+1}(k)}{n_{i+1}(k)c_i(k)} \frac{bE[s_i] + 64dc_i(k)}{bE[s_{i+1}] + 64dc_{i+1}(k)} \tag{34}$$

Regardless the priority, Web pages stored in server is identical. So we have $E[x_i] = E[x_{i+1}]$. If $Z_i(k) = n_i(k)/n_{i+1}(k)$,$e = bE[s_i]$,$f = 64d$,Equation (34) can be rewritten as

$$\frac{\delta_i}{\delta_{i+1}} = Z_i(k) \frac{fc_i(k)c_{i+1}(k) + ec_{i+1}(k)}{fc_i(k)c_{i+1}(k) + ec_i(k)} \tag{35}$$

Where $Z_i(k)$ is obtained by the queue length observer, and Equation (35) can be written into the form of $c_{i+1} = F_i(c_i(k), Z_i(k))$. Take class 1 be the reference ($\delta_1 = 1$).Considered the Equation (2), the number of threads for each class is solved.

3.4 Parameter Regression

As described in Section 3.2, the parameters b and $d$in Equation (31) are calculated by the Least Square algorithm. The sampling period is set to 10 sec and the parameter regression experiments totally obtain 61 group valid data. Figure 6 presents the pages size, the predict delay and the measured delay when the thread quota varies. The identification results are $b = 9.7$,$d = 55.7$ under 0.05 significant level.

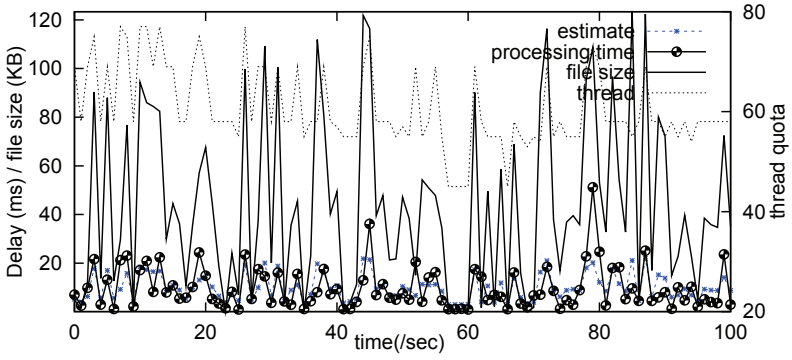


Figure 6: Parameter regression

3.5 Experimental Results

The test-bed has been shown in the section 3.3. Request rate of class1 varies from 6/s to 15/s every 300 seconds and class 2 keeps 12/s. The sampling time is $T = 10s$ and all the experiments last 2000s. The reference value is 0.5, i.e. $\delta_1/\delta_1 = 0.5$. The thread quota is equivalent for each class in the initial state. The results under different controllers are shown in Figure 7. Each sub-picture has two insets. The upper is the request arrival rate of two classes and their delay proportion. The lower presents the change of their thread quota. Compared the controllers, the results are in accordance with what we analyzed before:

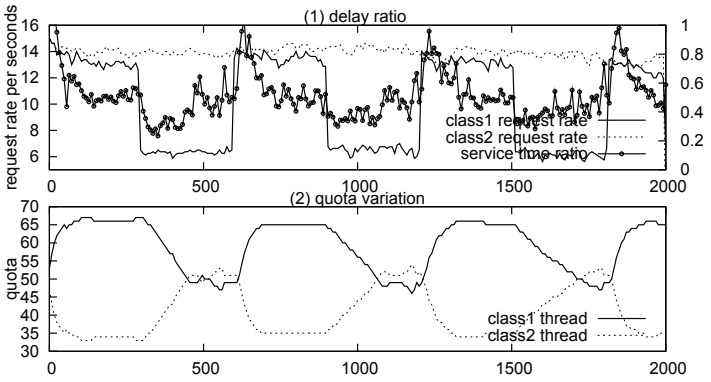
1. Whatever using feedback, $M/G/1/\infty$ predictive control or improved predictive control, they all achieve PDD in Web server. The experiments also did at sampling time $T = 8s$ and $T = 15s$ which are similar. So In order to save space, we only give the best effect figure at $T = 10s$.

2. Define the variance Ξ be the stability evaluation index ,which is

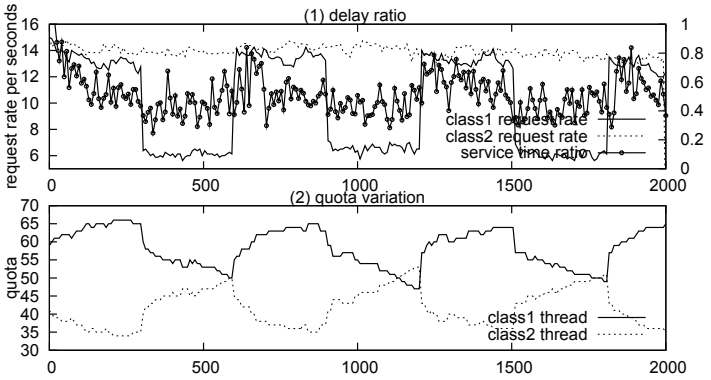
$$\Xi = \sqrt{\sum_{k=1}^{n} (y_i(k) - y_{1d})^2/n} \tag{36}$$

   Compute it under the different methods respectively, we find that the variance of proportional delay to 0.5 in improved predictive controller and $M/G/1/\infty$ predictive controller is 40.18% and 57.4% of which in PI controller. This means the predictive controller is more stable when the load changes.

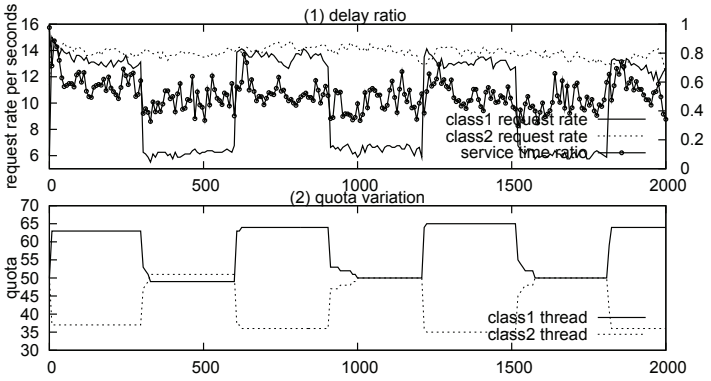
3. Compared the setting time in Figure 7a,7b and 7c. For the feedback, the setting time is nearly 100s at rising edge and 150s at the falling edge. The $M/G/1/\infty$ predictive controller is almost the same situation but a little shorter. But those of the improved predictive controller are less than 1s at both edges, which presents a much better dynamic



(a) PI control



(b) M/G/1/∞ queuing predictive control



(c) Improved predictive control

Figure 7: The change of delay ratio and thread quota under different controllers

performance. There are two reasons for this. One is the load changes influence waiting queue length first, and then this leads to the delay changes observed which exists delay. The other is the refreshed controller output works in the next sample time, while both the predictive controllers adjust the quota early.

4. Finally, compared how they adjust the thread quota when load changes in Figure 7b and 7c. Because of the limitation of PK formula, it takes a long time to adjust the quota to the steady state in Figure 7b. But the improved predictive controller can complete the

procedure in just "one step".

4 Conclusion and Future Work

The paper first presents a general Web QoS architecture, and then analyzed the defects of two traditional differentiated methods by experimental results. Finally, based on the queue length observer and parameter regression, improved predictive controller is proposed and produces a much better performance in real-time and stability.

But there are still some problems. For example, if the parameters b and d obtained online, it will be more accurate to the Web model. Meanwhile, the paper only talked about the static requests and is not compatible to the dynamic loads. In the future, we should also break the local area network, and take the network congestion into account.

Bibliography

[1] J. Wei, C. Xu, eQoS:Provisioning of client-perceived end-to-end qos guarantees in web servers. *IEEE Transactions on Computers*, Vol.55, No.12, pp.1543-1556,2006.

[2] Y. Lu, T. Abdelzaher, C. Lu, L. Sha, X. Liu, Feedback control with queuing theoretic prediction for relative delay guarantees in web servers, *The 9th IEEE Real-Time and Embedded Technology and Applications Symposium*, Washington, DC., USA,pp.208-217,2003.

[3] K.H. Chan, X. Chu, Design of a Cluster-Based Web Server with Proportional Connection Delay Guarantee, *The IEEE International conference on Communications*, Beijing,China, pp.5692-5696,2008.

[4] W. Pan, D. Mu, H. Wu, Q. Sun, Proportional Delay Differentiation Service in Web Application Servers: A Feedback Control Approach, *International Journal of Intelligent Information Technology Application*, Vol.1, No.1,pp.37-42,2008.

[5] Y. Lu, T.F. Abdelzaher, A.Saxena, Design, implementation, and evaluation of differentiated caching services, *IEEE Transactions on Parallel and Distributed Systems*, Vol.15, No.5, pp.440-452,2004.

[6] J. Wei, X. Zhou, C. Xu, Robust processing rate allocation for proportional slowdown differentiation on Internet servers, *IEEE Transactions on Computers*, Vol.54, No.8,pp.964-977,2005.

[7] L.Sha, X. Liu, U.Y. Lu, T. Abdelzaher, Queueing model based network server performance control, *The Proceedings Real-Time Systems Symposium*, Austin, USA,pp.81-90,2002.

[8] C. Lu, T. Abdelzaher, J. Stankovic, S. Son, A Feedback Control Approach for Guaranteeing Relative Delays in Web Servers, *The 7th Real-Time Technology and Applications Symposium*, Taipei, Taiwan,pp.51-62,2001.

[9] Y. Tang, X. Tang, *The queuing theory foundation and analysis*, Science Press. Beijing. 2005.

[10] X. Zhou, J. Wei, C.Z. Xu, Processing rate allocation for proportional slowdown differentiation on Internet servers. *The 18th International Parallel and Distributed Processing Symposium*, Santa Fe, USA, pp.1247-1256,2004.

[11] P. Barford, M. Crovella, Generating Representative Web Workloads for Network and Server Performance Evaluation, *Proceedings of the 1998 Joint International Conference on Measurement and Modeling of Computer Systems*, Madison, USA,pp.151-160,1998.

[12] J. Liebeherr, N. Christin, JoBS: Joint buffer management and scheduling for differentiated servers, *Proceedings of the 9th International Workshop on Quality of Service*, Karlsruhe, German,pp.404-418,2001.

[13] M. Harchol-Balter, Task assignment with unknown duration, *Journal of the ACM*, Vol.49, No.2, pp.260-288,2002.

[14] M. Arlitt, T. Jin, A workload characterization study of the 1998 world cup web site, *IEEE network*, Vol.14,No.3, pp.30-37,2000.

[15] TF Abdelzaher, N. Bhatti, Web server QoS management by adaptive content delivery, *Proceedings of the 7th International Workshop on Quality of Service*, London,UK, pp.216-225,1999.

[16] A. Gao, D. Mu, Y. Hu, W. Pan, Proportional Delay Guarantee in Web QoS Based on Predictive Control, *Proceedings of iCISE 2009*, Nanjing,China,pp.173-176,2009.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 647-655

Spiking Neural P Systems with Several Types of Spikes

M. Ionescu, G. Păun, M. J. Pérez-Jiménez, A. Rodríguez-Patón

Mihai Ionescu
University of Piteşti
Str. Târgu din Vale, nr. 1, 110040 Piteşti, Romania
E-mail: armandmihai.ionescu@gmail.com

Gheorghe Păun
Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucharest, Romania, and
Research Group on Natural Computing
Department of Computer Science and AI
University of Sevilla
Avda Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: george.paun@imar.ro, gpaun@us.es

Mario J. Pérez-Jiménez
Research Group on Natural Computing
Department of Computer Science and AI
University of Sevilla
Avda Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: marper@us.es

Alfonso Rodríguez-Patón
Department of Artificial Intelligence, Faculty of Computer Science
Polytechnical University of Madrid, Campus de Montegancedo
Boadilla del Monte 28660, Madrid, Spain
E-mail: arpaton@fi.upm.es

Abstract: With a motivation related to gene expression, where enzymes act in series, somewhat similar to the train spikes traveling along the axons of neurons, we consider an extension of spiking neural P systems, where several types of "spikes" are allowed. The power of the obtained spiking neural P systems is investigated. Some further extensions are mentioned, such as considering a process of decay in time of the spikes.
Keywords: Natural computing, Membrane computing, P system, Turing computability

1 Introduction

The present note lies at the intersection of two active research branches of bioinformatics/natural computing, namely, gene expression and membrane computing. Specifically, an extension of so-called spiking neural P systems (in short, SN P systems) is considered, with motivations related to gene expression processes.

For the reader's convenience, we shortly recall that an SN P system consists of a set of neurons placed in the nodes of a graph and sending signals (spikes) along synapses (edges of the graph), under the control of firing rules. Such a rule has the general form $E/a^c \to a^p; d$, where E is a regular expression (equivalently, we can consider it a regular language) and a denotes the spike; if the contents of the neuron is described by an element of the regular language (identified by) E, then the rule is enabled, c spikes are consumed and p are produced, and sent, after a time

delay of d steps, along the synapses leaving the neuron. There also are forgetting rules of the form $a^c \to \lambda$, with the meaning that, if the neuron contains exactly c spikes, then they can be removed (forgotten). One neuron is designated as the *output* neuron of the system and its spikes can exit into the environment, thus producing a *spike train*. Two main kinds of outputs can be associated with a computation in an SN P system: a set of numbers, obtained by considering the number of steps elapsed between consecutive spikes which exit the output neuron, and the string corresponding to the sequence of spikes which exit the output neuron.

These computing devices were introduced in [7] and then investigated in a large number of papers; we refer to the corresponding chapter from [13] and to the membrane computing website [16] for details.

In turn, gene expression is an important research area where various transcription factors appears and, important for their activity, their frequency matters – see, for instance, [1], [10], [12]. This means that a spiking like process is encountered, but with several "spikes", the regulator proteins which bind to a promotor depending on their concentration. In some sense we have here a communication process in which a signal encoded in a concentration (the transcription factor) is transduced to a frequency signal (the bursts of mRNA associated to the bindings of the transcription factor with the promotor) and again transduced back to a concentration (the level of protein produced). Thus, conceptually, we can approach this process in terms of theoretical machineries developed for spiking neurons – with the necessity of considering a variety of spikes, not only one as in the neural case. This is also suggested in [1]: "...we anticipate that frequency-modulated regulation may represent a general principle by which cells coordinate their response to signals."

Starting from these observations, we relate here the two research areas, introducing SN P systems with several types of spikes. Such a possibility was somehow forecasted already from the way the definition in [7] is given, with an alphabet, O, for the set of spikes, but with only one symbol in O; up to now, only a second type of spikes was considered, in [11], namely *anti-spikes*, which, when introduced, are immediately annihilated, in pairs with usual spikes. This extension to several types of spikes is natural also in view of the fact that all classes of P systems investigated in membrane computing work with arbitrary alphabets of objects.

As expected, having several types of spikes helps in proofs; in particular, we obtain the universality of the SN P systems with several types of spikes for systems with a very reduced number of neurons – remember that for systems with only one type of spikes the proofs do not bound the number of neurons (but such a bound can be found due to the existence of universal SN P systems, hence with a fixed number of neurons, but used in the computing mode, having both an input and an output). Three ways to define the result of a computation are considered: as the number of objects inside a specified neuron, as the number of objects sent out by the output neuron, and as the distance in time between the first two spikes sent out during the computation.

What is not investigated is the case of generating strings, in the sense of [3], [4], or even in the distributed case of [8]. Other open problems are mentioned in the rest of the paper and in the final section of it.

2 Formal Language Theory Prerequisites

We assume the reader to be familiar with basic language and automata theory, e.g., from [14] and [15], so that we introduce here only some notations and notions used later in the paper.

For an alphabet V, V^* denotes the set of all finite strings of symbols from V; the empty string is denoted by λ, and the set of all nonempty strings over V is denoted by V^+. When

$V = \{a\}$ is a singleton, then we write simply a^* and a^+ instead of $\{a\}^*, \{a\}^+$. For a language L, we denote by $sub(L)$ the set of all substrings of strings in L.

As usual in membrane computing, the multisets over a finite universe set U are represented by strings in U^* (two strings equal modulo a permutation represent the same multiset). If $u, v \in U^*$, we write the fact that u is a submultiset of v in the form $u \subseteq v$, with the understanding that there is a permutation of v having u as a substring (this can be formally formulated also in terms of Parikh mapping, but we do not enter into details). Similarly, we write $u \in sub(L)$ for a multiset u and a set L of multisets, meaning that u is a submultiset of a multiset in L.

A register machine (in the non-deterministic version) is a construct $M = (m, H, l_0, l_h, I)$, where m is the number of registers, H is the set of instruction labels, l_0 is the start label (labeling an ADD instruction), l_h is the halt label (assigned to instruction `HALT`), and I is the set of instructions; each label from H labels only one instruction from I, thus precisely identifying it. The instructions are of the following forms:

- $l_i : (\texttt{ADD}(r), l_j, l_k)$ (add 1 to register r and then go to one of the instructions with labels l_j, l_k non-deterministically chosen),
- $l_i : (\texttt{SUB}(r), l_j, l_k)$ (if register r is non-empty, then subtract 1 from it and go to the instruction with label l_j, otherwise go to the instruction with label l_k),
- $l_h : \texttt{HALT}$ (the halt instruction).

A register machine M generates a set $N(M)$ of numbers in the following way: we start with all registers empty (i.e., storing the number zero), we apply the instruction with label l_0 and we continue to apply instructions as indicated by the labels (and made possible by the contents of registers); if we reach the halt instruction, then the number n present in register 1 at that time is said to be generated by M. Without loss of generality we may assume that in the halting configuration all other registers are empty. It is known that register machines generate all sets of numbers which are Turing computable – we denote this family with NRE (RE stands for "recursively enumerable"). By $NFIN$ we denote the family of finite sets of natural numbers.

In the following sections, when comparing the power of two computing devices, number 0 is ignored (this corresponds to the fact that when comparing the power of language generating or accepting devices, the empty string λ is ignored).

3 Spiking Neural P Systems with Several Types of Spikes

We directly introduce the type of SN P systems we investigate in this paper; although somewhat far from the idea of a spike from the neural area, we still call the objects processed in our devices *spikes*.

A *spiking neural P system with several types of spikes* (abbreviated as *SN$^+$ P system*, of degree $m \geq 1$, is a construct of the form

$$\Pi = (O, \sigma_1, \ldots, \sigma_m, syn, i_0), \text{ where:}$$

1. O is the alphabet of *spikes* (we also say *objects*);
2. $\sigma_1, \ldots, \sigma_m$ are *neurons*, of the form $\sigma_i = (w_i, R_i), 1 \leq i \leq m$, where:
   a) $w_i \in O^*$ is the *initial multiset* of spikes contained in σ_i;
   b) R_i is a finite set of *rules* of the forms

(i) $E/u \to a$, where E is a regular language over O, $u \in O^+$, and $a \in O$ (spiking rules);

(ii) $v \to \lambda$, with $v \in O^+$ (forgetting rules) such that there is no rule $E/u \to a$ of type (i) with $v \in E$;

3. $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ with $i \neq j$ for each $(i, j) \in syn$, $1 \leq i, j \leq m$ (*synapses* between neurons);

4. $i_0 \in \{1, 2, \ldots, m\}$ indicates the *output neuron* (σ_{i_0}) of the system.

A rule $E/u \to a$ is applied as follows. If the neuron σ_i contains a multiset w of spikes such that $w \in L(E)$ and $u \in sub(w)$, then the rule can *fire*, and its application means consuming (removing) the spikes identified by u and producing the spike a, which will exit immediately the neuron. In turn, a rule $v \to \lambda$ is used if the neuron contains exactly the spikes identified by v, which are removed ("forgotten"). A global clock is assumed, marking the time for the whole system, hence the functioning of the system is synchronized.

If a rule $E/u \to a$ has $E = \{u\}$, then we will write it in the simplified form $u \to a$.

The spike emitted by a neuron σ_i go to all neurons σ_j such that $(i, j) \in syn$.

If several rules can be used at the same time in a neuron, then the one to be applied is chosen non-deterministically.

Using the rules as described above, one can define transitions among configurations. Any sequence of transitions starting in the initial configuration is called a *computation*. A computation halts if it reaches a configuration where no rule can be used.

There are many possibilities to associate a result with a computation, in the form of a number. Three possibilities are considered here: the number of objects in the output neuron in the halting configuration, the number of spikes sent to the environment by the output neuron, and the number of steps elapsed between the first two steps when the output neuron spikes. In the first two cases only halting computations provide an output, in the last case we can define the output also for ever going computations – but in what follows we only work with halting computations also for this case.

We denote by $N_\alpha(\Pi)$ the set of numbers generated as above by an SN$^+$ P system Π with the result defined in the mode $\alpha \in \{i, o, d\}$, where i indicate the internal output, o the external one (as the number of spikes), and d the fact that we count the distance between the first two spikes which exit the system. Then, $N_\alpha SN^+P_m$ is the family of sets of numbers $N_\alpha(\Pi)$, for SN$^+$ P systems with at most $m \geq 1$ neurons. As usual, the subscript m is replaced by $*$ if the number of neurons does not matter.

Before passing to investigate the power of the previously defined systems, let us mention that we have introduced here SN P systems of the *standard* type in what concerns the rules, i.e., producing only one spike, and without *delay*; *extended* rules are natural ($E/u \to v$, with both u and v multisets), but this is a too general case from a computability point of view, corresponding to cooperating P systems. It is important also to note that the rules we use have both additional powerful features – context sensitivity induced by the existence of the control regular language E, and strong restrictions – the produced spike (only one) should leave immediately the neuron, it cannot be further used in the same place without being sent back by the neighboring neurons. These features are essentially present in the proofs from the next section.

4 The Power of SN$^+$ P Systems

We start by considering the case when the result is counted inside the system (like in general P systems, hence somewhat far from the style of SN P systems).

Lemma 1. $NRE \subseteq N_iSN^+P_3$.

Proof: Let us consider a register machine $M = (n, H, l_0, l_h, I)$. We construct the following SN^+ P system

$$\begin{aligned}
\Pi &= (O, \sigma_1, \sigma_2, \sigma_3, syn, 1), \text{ with:}\\
O &= \{a_i \mid 1 \le i \le n\} \cup \{l, l\prime, l\prime\prime \mid l \in H\},\\
\sigma_1 &= (l_0, R_1),\\
R_1 &= \{O^*/l_i \to l\prime_i \mid (l_i : \mathtt{ADD}(r), l_j, l_k) \in I\}\\
&\cup \{O^*a^rO^*/l_ia_r \to l\prime\prime_j,\ (O^* - O^*a^rO^*)/l_i \to l\prime\prime_k \mid (l_i : \mathtt{SUB}(r), l_j, l_k) \in I\}\\
&\cup \{l_h \to \lambda\},\\
\sigma_2 &= (\lambda, R_2),\\
R_2 &= \{l\prime_i \to l_j,\ l\prime_i \to l_k \mid (l_i : \mathtt{ADD}(r), l_j, l_k) \in I\} \cup \{l\prime\prime \to l \mid l \in H\},\\
\sigma_3 &= (\lambda, R_3),\\
R_3 &= \{l\prime_i \to a_r \mid (l_i : \mathtt{ADD}(r), l_j, l_k) \in I\} \cup \{l\prime\prime \to \lambda \mid l \in H\},\\
syn &= \{(1,2),\ (1,3),\ (2,1),\ (3,1)\}.
\end{aligned}$$

The functioning of this system can be easily followed. The contents of register r is represented by the number of copies of object a_r present in the system. There are also objects associated with the labels of M.

Initially, we have only the object l_0 in neuron σ_1. In general, in the presence of a label l_i of an instruction in I, the instruction is simulated by the system Π. For the ADD instructions, the change of labels is done with the help of neuron σ_2 and the addition of a further object a_r is done in neuron σ_3. For the SUB instructions, the check for zero is performed by means of the regular language associated with the rules in R_1. The computation continues as long as the work of the machine M continues. When the label l_h is introduced – by the neuron σ_2 – the computation stops after one further step, when this object is removed from the output neuron, σ_1. Thus, in the end, this neuron only contains copies of object a_1, hence their number represents the value present in the first register of M in the end of the computation. Thus, $N(M) = L_i(\Pi)$. □

Theorem 2. $NFIN = N_iSN^+P_1 = N_iSN^+P_2 \subset N_iSN^+P_3 = NRE$.

Proof: The inclusions $N_iSN^+P_1 \subseteq N_iSN^+P_2 \subseteq N_iSN^+P_3$ are obvious from the definitions. The inclusion $N_iSN^+P_3 \subseteq NRE$ is straightforward (we can also invoke for it the Turing-Church thesis).

In an SN^+ P system with two components, the number of spikes present inside the two neurons cannot be increased (each spiking rule consumes at least one spike and produces only one spike, while there is no duplication of spikes because of multiple synapses which exit a neuron), hence we have $N_iSN^+P_2 \subseteq NFIN$.

On the other hand, $NFIN \subseteq N_iSN^+P_1$. Indeed, consider a finite set of numbers, $F = \{n_1, n_2, \ldots, n_k\}$; assume that $1 \le n_1 < n_2 < \cdots < n_k$ (remember that we ignore the number 0). We construct the system

$$\Pi = (\{a\}, (a^{n_k+1}, \{a^{n_k+1}/a^{n_k+1-n_i} \to a \mid 1 \le i \le k\}), \emptyset, 1).$$

We have $L_i(\Pi) = F$: each computation has only one step, which non-deterministically uses one of the rules in R_1. Each such rule just consumes a number of spikes, passing from the initial $n_k + 1$ spikes to any number $n_i \in F$, which cannot be further processed.

Together with Lemma 1, this concludes the proof of the theorem. □

Let us note in the construction from the proof of Lemma 1 that all neurons spike a large number of times (related to the length of the computation), not directly related to the number computed in the first register of M. This makes difficult to imagine a system with only three neurons which is universal when the result is defined as the number of spikes sent out. However, one additional neuron suffices in such a case.

Theorem 3. $NFIN = N_oSN^+P_1 \subset N_oSN^+P_2 \subseteq N_oSN^+P_3 \subseteq N_oSN^+P_4 = NRE$.

Proof: Again, the inclusions $N_oSN^+P_1 \subseteq N_oSN^+P_2 \subseteq N_oSN^+P_3 \subseteq N_oSN^+P_4 \subseteq NRE$ are obvious from the definitions.

The inclusion $NRE \subseteq N_oSN^+P_4$ can be obtained by a slight extension of the construction in the proof of Lemma 1: we replace the rule $l_h \to \lambda$ from R_1 with the rule

$$a_1^+ l_h / l_h a_1 \to l_h.$$

We also add a neuron σ_4, considered as output neuron, linked by synapses (1, 4), (4, 1) to the neuron σ_1 and containing the unique rule

$$l_h \to l_h.$$

When the computation of M stops, hence l_h is introduced in σ_1, this object removes one by one the objects a_1 and moves to the output neuron. This neuron both sends l_h out and back to σ_1, hence the number of copies of l_h sent out is equal with the number stored in the first register of M.

This time, an SN^+ P system with two components can compute an arbitrarily large number, by sending out an arbitrarily large number of spikes. For instance,

$$\Pi = (\{a\}, (a, \{a \to a\}), (aa, \{aa/a \to a,\ aa \to a\}), \{(1,2),(2,1)\}, 2),$$

has $L_o(\Pi) = \{1, 2, \dots\}$ (neuron σ_2 spikes step by step, until using the rule $aa \to a$, when only one spike remains in the system and the computation halts).

If we have only one neuron, the computation can last as may steps as many spikes are initially inside, hence $N_oSN^+P_1 \subseteq NFIN$.

On the other hand, $NFIN \subseteq N_iSN^+P_1$. Indeed, consider again a finite set of numbers, $F = \{n_1, n_2, \dots, n_k\}$ such that $1 \leq n_1 < n_2 < \cdots < n_k$ and construct the system

$$\begin{aligned} \Pi &= (\{a\}, \sigma_1, \emptyset, 1), \text{ with} \\ \sigma_1 &= (a^{n_k+1}, R_1), \\ R_1 &= \{a^{n_k+1}/a^{n_k+1-n_i+1} \to a \mid 1 \leq i \leq k\} \\ &\cup \{a^r/a \to a \mid 1 \leq r \leq n_k - 1\}. \end{aligned}$$

We have $L_i(\Pi) = F$: each computation starts with a step which uses non-deterministically a rule $a^{n_k+1}/a^{n_k+1-n_i+1} \to a$, which decreases the number of spikes from the initial n_k+1 to some $n_i - 1$; at this time, one spike was sent out. From now on, we use deterministically rules of the form $a^r/a \to a$, for all $r = 1, 2, \dots, n_i - 1$, hence for $n_i - 1$ steps, always sending out one spike. Thus, in total, we send out n_i spikes, for each $n_i \in F$.

Combining all these remarks, we have the theorem. □

It is an *open problem* whether or not the inclusions $N_oSN^+P_2 \subseteq N_oSN^+P_3 \subseteq N_oSN^+P_4$ are proper.

Theorem 4. $NFIN = N_dSN^+P_1 = N_dSN^+P_2 \subset N_dSN^+P_3 \subseteq N_dSN^+P_4 = NRE$.

Proof: As above, the inclusions $N_dSN^+P_1 \subseteq N_dSN^+P_2 \subseteq N_dSN^+P_3 \subseteq N_dSN^+P_4 \subseteq NRE$ are obvious from the definitions.

The inclusion $NFIN \subseteq N_dSN^+P_1$ is already proved for SN P systems with only one type of spikes. Like in that case, we also obtain the inclusion $N_dSN^+P_2 \subseteq NFIN$: in order to generate an arbitrarily large number, the output neuron should not spike for an arbitrarily large number of steps, but this is not possible in a system with only two neurons, because if only one neuron is working, it can perform only a number of steps bounded by the number of spikes initially present in it.

The fact that $N_dSN^+P_3$ contains infinite sets of numbers is also known for standard SN P systems.

What remains to prove is the inclusion $NRE \subseteq N_dSN^+P_4$ and this can again be obtained by an extension of the construction in the proof of Lemma 1; because this extension is not immediate, we give the construction in full details.

We consider a register machine $M = (n, H, l_0, l_h, I)$ and construct the SN$^+$ P system Π as indicated in Figure 1 – this time we do not give the system formally, but we represent it graphically, in the way usual in the SN P systems area.

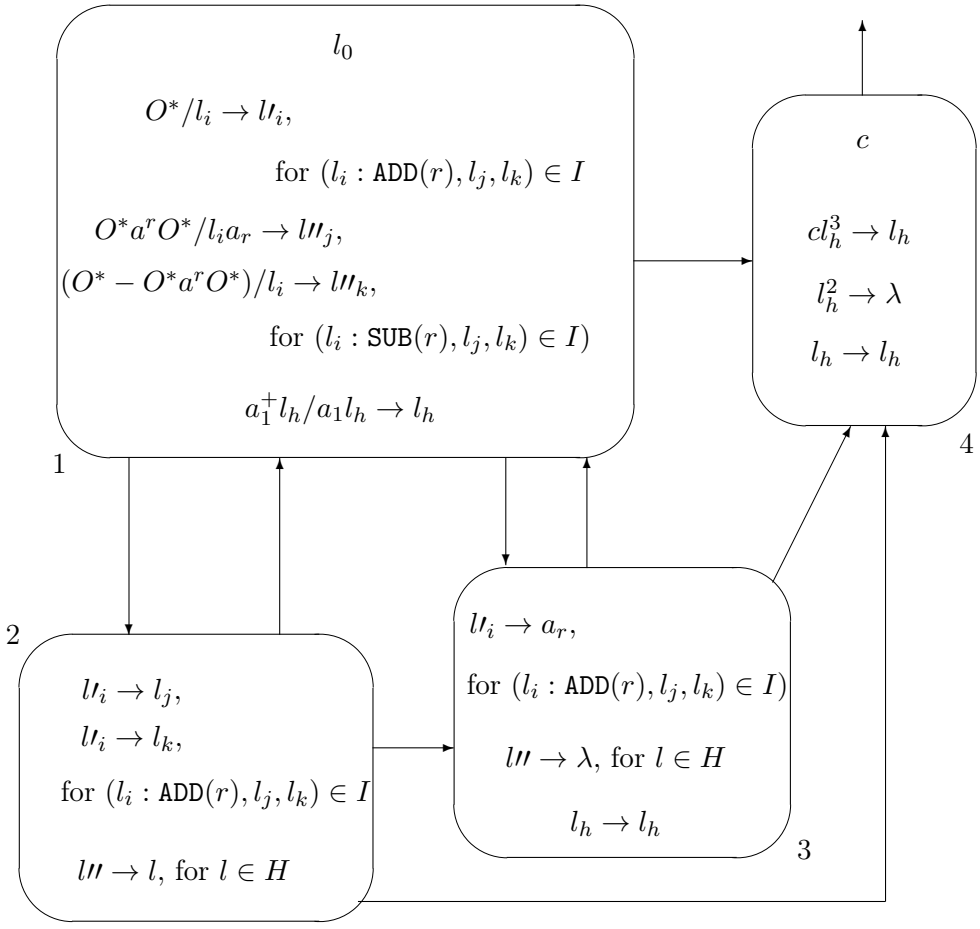


Figure 1: The SN$^+$ P system from the proof of Theorem 4

The work of this system is identical to that in the proof of Lemma 1, until producing the object l_h (the objects which arrive in the output neuron σ_4 from all other neurons remain here unused).

When σ_2 introduces the object l_h, it is sent to all other neurons. It waits unused in σ_4, but in

σ_1 and σ_3 it is reproduced in each step, hence these two neurons feed repeatedly each other with one copy of l_h. In σ_1, each use of the rule $a_1^+ l_h / a_1 l_h \to l_h$ removes one copy of a_1. In the end of step 1 (we count here only the steps after having l_h in the system, hence for the phase when the output is produced), neuron σ_4 contains three copies of l_h. Thus, in step 2, this neuron spikes.

From now on, neurons σ_1, σ_3 spike repeatedly, exchanging copies of l_h, σ_4 always forgets the two copies of l_h received from σ_1, σ_3 (while σ_2 just accumulates copies of l_h, which cannot be processed here). When the last copy of a_1 is removed from σ_1 (if m copies of a_1 were present here when l_h was introduced, then this happens in step m, after having l_h in the system), this is the last step when σ_4 receives two spikes. In the next step $(m+1)$ it receives only the spike produced by σ_3, which is used (in step $m+2$) by the rule $l_h \to l_h$ in σ_4. The computation stops. The number of steps between the two spikes sent out by the output neuron is $(m+2)-2 = m$, hence the number computed by the register machine in its first register.

The proof of the theorem is now complete. □

It is an *open problem* whether or not the inclusion $N_g SN^+ P_3 \subseteq N_g SN^+ P_4$ is proper.

5 Final Remarks

In gene expression it is also the case that the enzymes have a time dependency of their reactivity, which can be captured in terms of SN P systems by considering decaying spikes, in the sense of [6]. For instance, we can associate an *age* with each produced spike, by using rules of the form $E/u \to (a,t)$, where $t \geq 1$ is the "duration of life" of this spike. If the spike is not used in a step, then its life is decreased by one unit (this is like having rewriting rules $(a,s) \to (a,s-1)$, used in parallel for all spikes not used in spiking or forgetting rules), until reaching the state $(a,0)$, when a rule $(a,0) \to \lambda$ is assumed to be applied. This feature remains to be further investigated.

Let us close by recalling the fact that besides the synchronized (sequential in each neuron) mode of evolution, there were also introduced other modes, such as the exhaustive one, [9], and the non-synchronized one, [2], which also deserve to be considered for SN P systems with several types of spikes.

Acknowledgements

The work of M. Ionescu was possible due to CNCSIS grant RP-4 12/01.07.2009. The work of Gh. Păun was supported by Proyecto de Excelencia con Investigador de Reconocida Valía, de la Junta de Andalucía, grant P08 – TIC 04200. The work of A. Rodriguez-Paton has been supported by Spanish Ministry of Science and Innovation under project TIN2009 - 14421. The careful reading of the paper by two anonymous referees is gratefully acknowledged.

Bibliography

[1] L. Cai, C.K. Dalal, M.B. Elowitz: Frequency-modulated nuclear localization bursts coordinate gene regulation. *Nature*, 455 (25 September 2008).

[2] M. Cavaliere, E. Egecioglu, O.H. Ibarra, M. Ionescu, Gh. Păun, S. Woodworth: Asynchronous spiking neural P systems. *Theoretical Computer Science*, 410, 24-25 (2009), 2352–2364.

[3] H. Chen, R. Freund, M. Ionescu, Gh. Păun, M.J. Pérez-Jiménez: On string languages generated by spiking neural P systems. *Fundamenta Informaticae*, 75, 1-4 (2007), 141–162.

[4] H. Chen, T.-O. Ishdorj, Gh. Păun, M.J. Pérez-Jiménez: Spiking neural P systems with extended rules. In *Proc. Fourth Brainstorming Week on Membrane Computing*, Sevilla, 2006, RGNC Report 02/2006, 241–265.

[5] H. Chen, T.-O. Ishdorj, Gh. Păun, M.J. Pérez-Jiménez: Handling languages with spiking neural P systems with extended rules. *Romanian J. Information Sci. and Technology*, 9, 3 (2006), 151–162.

[6] R. Freund, M. Ionescu, M. Oswald: Extended spiking neural P systems with decaying spikes and/or total spiking. *Intern. J. Found. Computer Sci.*, 19 (2008), 1223–1234.

[7] M. Ionescu, Gh. Păun, T. Yokomori: Spiking neural P systems. *Fundamenta Informaticae*, 71, 2-3 (2006), 279–308.

[8] M. Ionescu, Gh. Păun, M.J. Pérez-Jiménez: T. Yokomori: Spiking neural dP systems. *Proc. Ninth Brainstorming Week on Membrane Computing*, Sevilla, 2011, RGNC Report 01/2011.

[9] M. Ionescu, Gh. Păun, T. Yokomori: Spiking neural P systems with exhaustive use of rules. *Intern. J. Unconventional Computing*, 3, 2 (2007), 135–154.

[10] E.M. Ozbudak, M. Thattai, I. Kurtser, A.D. Grossman, A. van Oudenaarden: Regulation of noise in the expression of a single gene. *Nature Genetics*, 31 (May 2002).

[11] L. Pan, Gh. Păun: Spiking neural P systems with anti-spikes. *Intern. J. Computers, Comm. Control*, 4, 3 (2009), 273–282.

[12] J. Paulsson: Models of stochastic gene expression. *Physics of Life Reviews*, 2 (2005), 157–175.

[13] Gh. Păun, G. Rozenberg, A. Salomaa, eds.: *Handbook of Membrane Computing.* Oxford University Press, 2010.

[14] G. Rozenberg, A. Salomaa, eds.: *Handbook of Formal Languages.* 3 volumes, Springer, Berlin, 1998.

[15] A. Salomaa: *Formal Languages.* Academic Press, New York, 1973.

[16] The P Systems Website: `http://ppage.psystems.eu`.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 656-667

An Optimal Rate Control and Routing Scheme for Multipath Networks

S. Li, W. Sun, Y. Zhang, H. Zhang

Shiyong Li, Wei Sun, Yaming Zhang
School of Economics and Management,
Yanshan University,
Qinhuangdao, 066004, P.R. China
E-mail(s): shiyongli@ysu.edu.cn, wsun@ysu.edu.cn, yaming99@ysu.edu.cn

Hongke Zhang
School of Electronic and Information Engineering,
Beijing Jiaotong University,
Beijing 100044, P.R. China
E-mail: hkzhang@bjtu.edu.cn

Abstract: This paper considers optimal rate control and routing schemes for multipath networks which can be formulated as multipath network utility maximization problems. In these schemes, maximizing the aggregated user utility over the network with multipath routes under the link capacity constraints is the objective of utility maximization problems. By adopting the Lagrangian method, sub-problems for users and paths are deduced and interpreted from an economic point of view. In order to obtain the optimal rate allocation, a novel distributed primal-dual algorithm is proposed, and the performance is evaluated through simulations under two different fairness concepts. Moreover, window-based flow control scheme is also presented since it is more convenient to realize in practical end-to-end implementation than the rate control scheme.
Keywords: multipath networks, rate control, routing, network utility maximization, optimization.

1 Introduction

Since the publication of the seminal paper [1], the single path rate control and routing schemes which can be formulated as *network utility maximization* problems have been extensively studied in the past years, mainly in the context of Internet congestion control and flow control (e.g., [2]-[5], and the reference therein).

Recently, there has been much interest in multipath rate control and routing schemes [6]-[13]. In a multipath routing scheme, each source-destination pair can have several different paths along which data packet can be transmitted. In these schemes, maximizing aggregated user utility over the network with multipath routes under the link capacity constraints is the objective of the *multipath network utility maximization* problems. These problems are indeed a combination of multipath routing and rate control and can be viewed as an example of cross-layer optimization [5] for network architectures, where additional benefits are obtained by jointly optimizing at the routing (network layer) and rate control (transport layer).

In order to solve the multipath network utility maximization problems, roughly speaking, multipath rate control and routing schemes can be classed into three categories: primal algorithms [6], [8], dual algorithms [7], [11], and primal-dual algorithms [9], [10], [12]. The primal algorithms have a dynamical law for adjusting user rate and a static law for generating link price, and conversely, the dual algorithms have a dynamical law for adjusting link price and a

static law for generating user rate. Then, the primal-dual algorithms have dynamical laws for adjusting both user rate and link price. The primal algorithms are based on a penalty function approach, i.e., they replace the capacity constraints by a penalty function in the optimization objective [1], [6], [8]. They always tend to produce biased approximates of the optimal operating points, due to the fact that penalties are only incurred when the capacity constraints are violated. (In contrast, the optimal operating point is defined to be one that satisfies the capacity constraints.) As for the dual algorithms, the advantage is that they are designed to compute the exact optimal operating points when the step sizes are driven to zero in an appropriate fashion [7], [11]. Then the primal-dual algorithms combine the advantages of both primal and dual algorithms, and possess a rapid convergence property to the optimums within reasonable convergence times [9], [10], [12].

In [1], after studying the single-path network utility maximization problem, the authors discuss the extension to the multipath case, where the flow between each source-destination pair can be globally split among multiple paths, and present a distributed rate-based congestion control algorithm. Then in [6] the utility maximization framework for multipath case is further studied, and a primal algorithm with round-trip delay is proposed and investigated. Meanwhile, a decentralized sufficient condition for local stability of the delayed algorithm is obtained. In [8], the sufficient condition for local stability of the primal algorithm with round-trip delay is improved, i.e., the gain parameter for each route is restricted by the round-trip delay of that route, but not by the round-trip delays of other routes, even those other routes serving the same users. Dual algorithms that can produce exact optimal operating points are developed in [2] for single-path networks. When extended to the multipath cases, the primal variables (i.e., the resource allocation) may not be unique although the dual variables (i.e., the price) in the dual algorithms converge. In order to obtain the unique optimum, some researchers relax the multipath network optimization problems and make them strictly concave. The authors in [9] address the problem by adding a quadratic term onto the objective problem and provide rigorous proof of convergence for the dual algorithm. Then in [11] the author proposes a novel relaxed network utility maximization problem and derives a new class of controlled splitting multipath dual algorithms. In [12], by combining the first order Lagrangian method and filtering mechanism, a new primal-dual algorithm is proposed to eliminate typical oscillation behavior in multipath networks.

In this paper we investigate the optimal rate control and routing scheme formulated as multipath network utility maximization problem, decompose the optimization problem into two sub-problems for users and paths through Lagrangian method, and give the interpretations for the sub-problems and dual problem from an economic point of view. Furthermore, for each user and its available paths, we analyze the relationship between the prices charged by paths and that paid by the user, and on basis of the difference between them we propose a novel distributed primal-dual algorithm for the optimal rate control and routing, which can converge to the optimum of the utility maximization problem, and present the practical end-to-end implementation in the Internet.

The rest of this paper is organized as follows: in Section 2, we analyze the model for jointly optimal rate control and routing scheme known as multipath network utility maximization; in Section 3, we present a novel distributed primal-dual algorithm for optimal rate control and routing; in Section 4, we give some simulation results to illustrate the convergence of the proposed algorithm; finally, we conclude the paper in Section 5.

2 Multipath Network Utility Maximization

2.1 Model Description

Consider a network consisting of a set of links L, a set of paths P and a set of users S. Each link $l \in L$ has capacity C_l. Each user identifies a unique source-destination pair. There are multiple paths between each source-destination pair. Associated with each user $s \in S$ is a set of paths $P(s)$ where each path $p \in P(s)$ is a collection of links $L(p)$. Obviously, associated with a path $p \in P(s)$ is a set of paths $P(s)$ which are all associated with the same user s, and with identical source and destination. In following analysis, if a user s transmits along a path p, then we write $p \in P(s)$; if a path p uses a link l, then we write $p \in P(l)$, where $P(l)$ is the set of paths that use link l.

In this paper we make no assumptions about whether the paths $p \in P(s)$ are link-disjointed. Obviously the ability to generate link-disjointed paths can assist in the construction of highly robust end-to-end communication for the source-destination pair which is labeled by user s, but the model also covers the case where some or all of the paths $p \in P(s)$ of user s share some common path segments.

For user s, assume the transmission rate on path p is $x_{sp}(t)$, then the total flow rate of user s is $y_s(t) = \sum_{p:p\in P(s)} x_{sp}(t)$. Meanwhile, the aggregated rate on link l is $z_l(t) = \sum_{p:p\in P(l)} x_{sp}(t)$. User s attains a utility $U_s(y_s(t))$ when it transmits at rate $y_s(t)$. The multipath network utility maximization is modeled as the following nonlinear optimization problem

$$\begin{aligned} \max \quad & \sum_{s:s\in S} U_s(y_s(t)) \\ \text{subject to} \quad & \sum_{p:p\in P(s)} x_{sp}(t) = y_s(t) \\ & \sum_{p:p\in P(l)} x_{sp}(t) \le C_l \\ \text{over} \quad & x_{sp}(t) \ge 0. \end{aligned} \tag{1}$$

Here, maximizing the aggregated utility of the user rate $y_s(t)$ over all users with the constraints of link capacities is the objective of network. We are interested in the following class of utility functions proposed in [14]

$$U_s(y_s(t)) = \begin{cases} w_s \log y_s(t), & \text{if } \alpha_s = 1, \\ w_s \dfrac{y_s(t)^{1-\alpha_s}}{1-\alpha_s}, & \text{if } \alpha_s \neq 1, \end{cases} \tag{2}$$

where w_s is considered as the willingness to pay of user s, α_s is the parameter of fairness concept. This family of utility functions are known to characterize a large class of fairness concepts and have been investigated extensively [13], [15], [12]. In particular, if we set $\alpha_s = 0$, the optimization problem reduces to throughput maximization. If $\alpha_s = 1$, proportional fairness among competing users is obtained; if $\alpha_s = 2$, then harmonic mean fairness; and if $\alpha_s = \infty$, then max-min fairness [14]. For example, these utility functions are considered in wireless networks and optimal network performance and energy efficiency is achieved by jointly optimizing congestion control and power control [13].

2.2 Model Analysis

The utility maximization problem (1) with utility functions (2) is regarded as the primal problem with primal variables $x = (x_{sp}(t), p \in P(s))$ and $y = (y_s(t), s \in S)$. Then we can obtain the following theorem

Theorem 1. *The rate allocation model (1) with utility functions (2) is a convex programming problem. The optimal total flow rate of each user, i.e. y_s^*, exists and is unique. However, the optimal flow rate of each user on every path, i.e. $x_{sp}(t)$, may be not unique.*

Proof: From the nonlinear programming theory [13], the objective of (1) is concave with respect to primal variables. The constraints of (1) are linear, and thereby all of them are convex. So the feasible set is compact. We deduce that (1) is a convex programming problem. Furthermore, the optimization problem is strictly convex with respect to primal variable $y = (y_s(t), s \in S)$, then the optimal flow rate $y = (y_s^*, s \in S)$ exists and is unique as a consequence of strict convexity. However, the objective of (1) is not strictly concave with respect to primal variable $x = (x_{sp}(t), s \in S, p \in P)$, thus the optimal flow rate of each user on every path may be not unique. □

In order to investigate the optimum of model (1), we firstly give the Lagrangian of this nonlinear optimization problem

$$L(x, y; \lambda, \mu) = \sum_{s:s\in S} \left(U_s(y_s(t)) + \lambda_s \left(\sum_{p:p\in P(s)} x_{sp}(t) - y_s(t) \right) \right) + \sum_{l:l\in L} \mu_l \left(C_l - \sum_{p:p\in P(l)} x_{sp}(t) - \varepsilon_l^2 \right), \tag{3}$$

where $\lambda = (\lambda_s, s \in S)$ is the price vector with element $\lambda_s \geq 0$, which is the Lagrange multiplier associated with the equality on user s, and can be considered as the price per unit bandwidth paid by user s; $\mu = (\mu_l, l \in L)$ is the price vector with element $\mu_l \geq 0$, which is the Lagrange multiplier associated with the inequality constraint on link l, and can be considered as the price per unit bandwidth charged by link l; ε_l^2 is the slack variable, which can be considered as the spare capacity of link l.

Then the Lagrangian (3) can be rewritten as

$$L(x, y; \lambda, \mu) = \sum_{s:s\in S} (U_s(y_s(t)) - \lambda_s y_s(t)) + \sum_{s:s\in S} \sum_{p:p\in P(s)} x_{sp}(t) \left(\lambda_s - \sum_{l:l\in L(p)} \mu_l \right) + \sum_{l:l\in L} \mu_l \left(C_l - \varepsilon_l^2 \right). \tag{4}$$

Notice that the first term in (4) is separable in $y_s(t)$, and the second term is separable in $x_{sp}(t)$. Thus the objective function of the dual problem is

$$D(\lambda, \mu) = \max_{x,y} L(x, y; \lambda, \mu) = \sum_{s:s\in S} A_s(\lambda_s) + \sum_{s:s\in S} \sum_{p:p\in P(s)} B_{sp}(\lambda_s, \gamma_{sp}) + \sum_{l:l\in L} \mu_l \left(C_l - \varepsilon_l^2 \right), \tag{5}$$

where

$$A_s(\lambda_s) = \max_{y_s(t)} U_s(y_s(t)) - \lambda_s y_s(t), \tag{6}$$

$$B_{sp}(\lambda_s, \gamma_{sp}) = \max_{x_{sp}(t)} x_{sp}(t)(\lambda_s - \gamma_{sp}), \gamma_{sp} = \sum_{l:l\in L(p)} \mu_l. \tag{7}$$

Then, the optimal flow rate can be denoted as

$$y_s^*(\lambda_s) = \text{argmax} \quad U_s(y_s(t)) - \lambda_s y_s(t),$$

$$x_{sp}^*(\gamma_{sp}) = \text{argmax} \quad x_{sp}(\lambda_s)(\lambda_s - \gamma_{sp}),$$

where $\sum_{p:p\in P(s)} x_{sp}(\lambda_s) = y_s^*(\lambda_s), \gamma_{sp} = \sum_{l:l\in L(p)} \mu_l$.

Hence, the dual problem is

$$\begin{array}{ll} \min & D(\lambda,\mu) \\ \text{over} & \lambda_s \geq 0, \mu_l \geq 0, s \in S, l \in L. \end{array} \tag{8}$$

We can interpret the sub-problems (6)(7) from an economic point of view as follows.

The sub-problem (6) is regarded as the USER problem. In this problem, every user s tries to maximize its own utility which depends on the total flow rate $y_s(t)$. Meanwhile, the user has to pay a price for its using bandwidth. Since λ_s is the price per unit bandwidth paid by user s, then $\lambda_s y_s(t)$ is the total cost that user is willing to pay. Thus, the USER problem (6) is a nonlinear optimization problem that every user is to maximize its own profit.

The sub-problem (7) is regarded as the PATH problem. In this problem, μ_l is the price per unit bandwidth charged by link l, then γ_{sp} is the total price associated with the path p of user s. The product $x_{sp}(t)\lambda_s$ is the cost paid by user s for path p, and $x_{sp}(t)\gamma_{sp}$ is the total cost charged by path p. Hence, the PATH problem (7) is a linear optimization problem that every path is to maximize its own revenue.

Meanwhile, the dual problem (8) can be considered as the NETWORK problem. The objective is to minimize the total price charged by all links under the constraints that users are guaranteed with certain levels of satisfaction. Indeed, the price paid by a user and those charged by its paths is an equilibrium from the game theory point of view.

For model (1) and its Lagrangian (4) we can obtain the following theorem.

Theorem 2. *At the positive optimum of model (1), the optimal total prices associated with paths for a user are all equal to the optimal price paid by the user.*

Proof: Indeed, at the optimum of model (1), from the Karush-Kuhn-Tucker condition for optimality of an optimization problem, we can obtain

$$U\prime_s(y_s^*) - \lambda_s^* \Rightarrow \begin{cases} = 0, & \text{if } \ y_s^* > 0, \\ \leq 0, & \text{if } \ y_s^* = 0, \end{cases} \quad \forall s \in S,$$

$$\lambda_s^* - \gamma_{sp}^* = \lambda_s^* - \sum_{l:l\in L(p)} \mu_l^* \Rightarrow \begin{cases} = 0, & \text{if } \ x_{sp}^* > 0, \\ \leq 0, & \text{if } \ x_{sp}^* = 0, \end{cases} \quad \forall s \in S, \forall p \in P(s).$$

Thus, for the multiple paths that a user uses, e.g., $p_1, p_2 \in P(s)$, the optimal total price associated with each path is

$$\sum_{l:l\in L(p_1)} \mu_l^* = \sum_{l:l\in L(p_2)} \mu_l^* = \lambda_s^* = \frac{w_s}{\left(\sum_{p:p\in P(s)} x_{sp}^*\right)^{\alpha_s}}. \tag{9}$$

□

3 Distributed Algorithm

3.1 Rate Control Algorithm

To obtain the optimal flow rate in a decentralized architecture, we present the following distributed primal-dual algorithm, which depends on locally available information.

Each user s updates its flow rate $x_{sp}(t)$ on path p with the following *primal algorithm*

$$\frac{dx_{sp}(t)}{dt} = \left(\kappa x_{sp}(t)(\lambda_s(t) - \gamma_{sp}(t))\right)^+_{x_{sp}(t)}, \tag{10}$$

$$\lambda_s(t) = \frac{w_s}{\left(\sum_{p:p\in P(s)} x_{sp}(t)\right)^{\alpha_s}}, \tag{11}$$

$$\gamma_{sp}(t) = \sum_{l:l\in L(p)} \mu_l(t), \tag{12}$$

where $\kappa > 0$ is the step size; $a = (b)_c^+$ means $a = b$ if $c > 0$ and $a = \max\{0, b\}$ if $c = 0$.

Each link l updates its price $\mu_l(t)$ with the following *dual algorithm*

$$\frac{d\mu_l(t)}{dt} = \left(\upsilon \frac{z_l(t) - C_l}{C_l}\right)_{\mu_l(t)}^+, \tag{13}$$

$$z_l(t) = \sum_{p:p\in P(l)} x_{sp}(t), \tag{14}$$

where $\upsilon > 0$ is the step size.

In the primal-dual algorithm, user s computes the price $\lambda_s(t)$ paid for path p according to (11), obtains the total price $\gamma_{sp}(t)$ charged by the path from (12), and updates its rate $x_{sp}(t)$ according to (10), which is a fluid model depending on the difference between the price paid by user s and the price charged by path p. Meanwhile, link l observes the aggregated rate $z_l(t)$ on it, and updates its price $\mu_l(t)$ according to (13), which is identical to the packet loss rate on the link.

Obviously, the primal-dual algorithm is distributed, which only depends on locally available information. Obviously, the rule for rate update (10)(11)(12) is a scaled gradient algorithm for the PATH sub-problem, and can be reduced to the single-path rate control algorithm when only a single path is available for each user. Meanwhile, the rule for price update (13)(14) is also a gradient algorithm, and is similar to the single-path price algorithms designed for solving the dual problems.

3.2 End-to-End Implementation

In practical end-to-end implementation, window-based flow control where the window size is increased or decreased upon receipt of acks (*positive* acknowledgments) or nacks (*negative* acknowledgments) is more convenient to implement than rate-based flow control mechanism since the former is inherently self-clocking. In order to obtain a window flow control scheme, we discretize the system (10)(11)(12) and obtain

$$\frac{x_{sp}(t+\delta) - x_{sp}(t)}{\delta} = \frac{\kappa x_{sp}(t)}{\left(\sum_{p:p\in P(s)} x_{sp}(t)\right)^{\alpha_s}} \left(w_r - \left(\sum_{p:p\in P(s)} x_{sp}(t)\right)^{\alpha_s} \sum_{l:l\in L(p)} \mu_l(t)\right). \tag{15}$$

Here, we assume $x_{sp}(t) > 0$.

Let $W_{sp}(t)$ be the window size of user s on its path p at time t. We follow the approximation relating data transmission rate and window size [14]

$$x_{sp}(t) \approx \frac{W_{sp}(t)}{RTT_p},$$

where RTT_p is the round trip time of path p.

Let $A_p(t, t+\delta)$ and $N_p(t, t+\delta)$ denote the numbers of acks and nacks received by user s on path p in the time interval $[t, t+\delta)$, respectively. Then

$$\frac{A_p(t, t+\delta)}{\delta} \approx x_{sp}(t) \approx \frac{W_{sp}(t)}{RTT_p},$$

and

$$\frac{N_p(t, t+\delta)}{\delta} \approx x_{sp}(t) \sum_{l:l\in L(p)} \mu_l(t).$$

Thus,

$$\frac{x_{sp}(t+\delta) - x_{sp}(t)}{\delta} = \frac{W_{sp}(t+\delta) - W_{sp}(t)}{A_p(t, t+\delta)} \frac{A_p(t, t+\delta)}{RTT_p \delta} = \frac{W_{sp}(t+\delta) - W_{sp}(t)}{A_p(t, t+\delta)} \frac{W_{sp}(t)}{RTT_p^2}.$$

Using the approximations above, the window-based flow control mechanism becomes

$$W_{sp}(t+\delta) - W_{sp}(t) = \frac{\kappa w_s RTT_p}{(\sum_{p:p\in P(s)} W_{sp}(t)/RTT_p)^{\alpha_s}} A_p(t, t+\delta) - \kappa RTT_p N_p(t, t+\delta). \quad (16)$$

In order to damp the oscillation when window size is too small, we modify (16) into

$$W_{sp}(t+\delta) - W_{sp}(t) = \kappa w_s RTT_p A_p(t, t+\delta) - \kappa RTT_p \left(\sum_{p:p\in P(s)} \frac{W_{sp}(t)}{RTT_p} \right)^{\alpha_s} N_p(t, t+\delta). \quad (17)$$

The window-based flow control mechanism (17) can be interpreted as follows: when each ack is received, the window size is increased by a fixed amount which is in proportion to RTT_p; when each nack is received, the window size is decreased by a fixed amount which is in proportion to $RTT_p(\sum_{p:p\in P(s)} W_{sp}(t)/RTT_p)^{\alpha_s}$. Hence, the window flow control scheme realizes the adaptive increase/multiplicative decrease (AIMD) principle which is used in the original TCP version and other variants.

For the total price associated with each path in practical end-to-end implementation, we present a way to communicate path prices to users *implicitly*, i.e., a user deduces the aggregated price on each path from observed round trip time.

Notice that queuing length $b_l(t)$ evolves according to

$$\frac{db_l(t)}{dt} = (z_l(t) - C_l)^+. \quad (18)$$

Comparing (18) with (13), we can observe that the link price $\mu_l(t)$ at time t is proportional to the current queuing delay $q_l(t)$, i.e.,

$$\mu_l(t) = \upsilon \frac{b_l(t)}{C_l} = \upsilon q_l(t).$$

Then the price on path p is proportional to the end-to-end queuing delay at time t

$$\gamma_{sp}(t) = \sum_{l:l\in L(p)} \mu_l(t) = \upsilon \sum_{l:l\in L(p)} q_l(t).$$

Given the end-to-end propagation delay D_p, the price associated with path p can be deduced from the round trip time RTT_p observed at the user s

$$\gamma_{sp}(t) = \upsilon \sum_{l:l\in L(p)} q_l(t) = \upsilon(RTT_p - D_p).$$

In practical implementation, D_p can be estimated by the minimum RTT_p observed so far.

4 Simulation

In this section, we investigate the performance of the proposed primal-dual algorithm and give simulation results to illustrate the convergence of our algorithm.

We make no assumption that the paths for a user are link-disjointed and consider the general multipath communication where some of the paths for a user can share some path segments in common, just as shown in Figure 1. There are two users in this simple network. Two paths are available for each user, i.e., paths A→B→C and D→E→F→G for user 1 (pair of S_1 and D_1), and paths D→E→F→G and H→I→F→G for user 2 (pair of S_2 and D_2). Obviously, the two paths for user 2 share a common link which is labeled by L_5. There are seven unidirectional links with capacities $C = (C_1, C_2, C_3, C_4, C_5, C_6, C_7) = (3, 2, 3, 4, 4, 2, 3)$Mbps. In the proposed algorithm, we choose $w = (w_1, w_2) = (2, 3)$, and $\kappa = \upsilon = 0.05$.

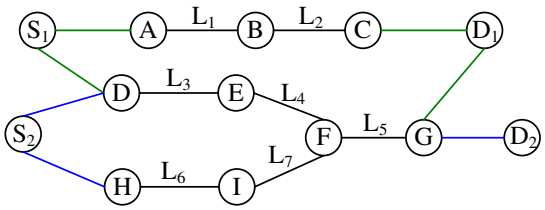


Figure 1: Multipath network topology

4.1 Proportional Fairness

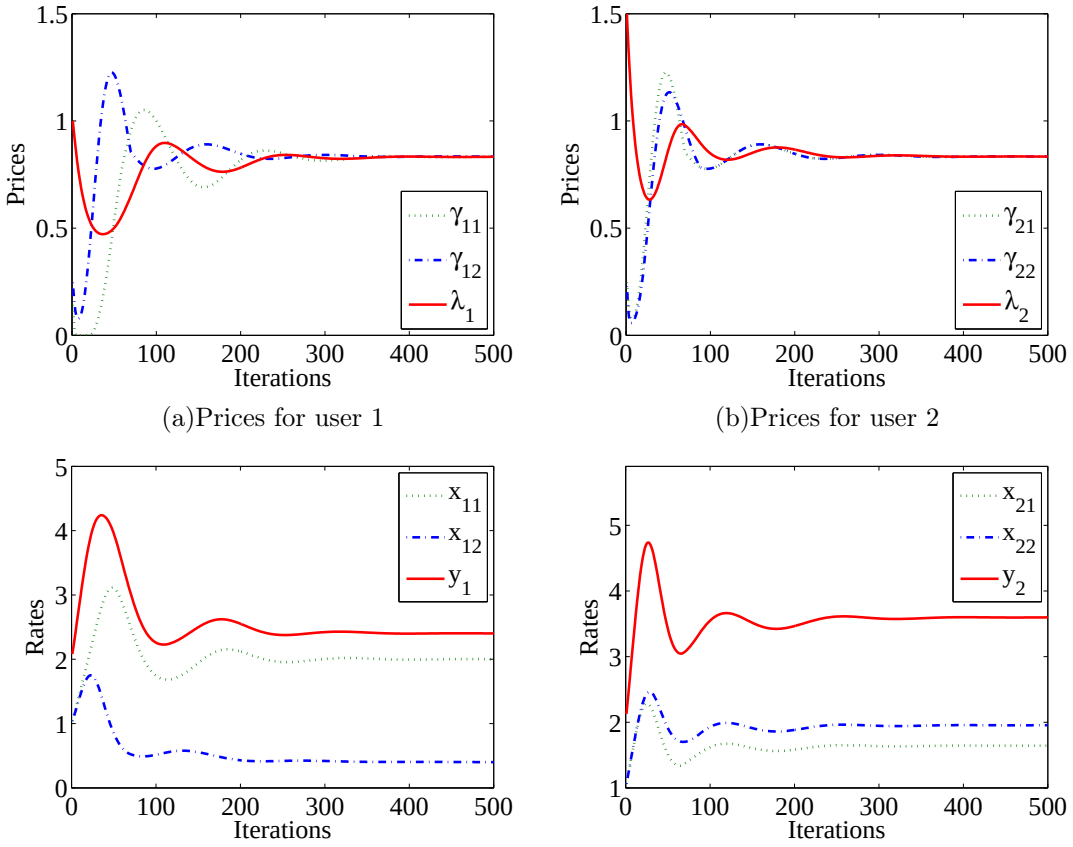
We firstly consider the proportional fairness among competing users, thus the fairness parameter is $\alpha_1 = \alpha_2 = 1$. Without loss of generality, assume the initial rate of each user on every available path is 1Mbps. The optimum obtained by the proposed algorithm is listed in Table 1. The optimal solution solved by nonlinear programming software LINGO is also presented in the table.

variable	x_{11}^*	x_{12}^*	x_{21}^*	x_{22}^*	y_1^*	y_2^*
algorithm	2.0000	0.4000	1.6440	1.9560	2.4000	3.6000
LINGO	2.0000	0.4000	1.7360	1.8640	2.4000	3.6000

Table 1. The optimum under multipath network topology: proportional fairness

Obviously, the algorithm is convergent and efficient to solve the optimum of multipath network utility maximization problem under the general multipath network topology. Also as shown in Table 1, the optimal rate of user 2 on its paths is not unique, which has been proved in Theorem 1. However, the total optimal rate for each user is unique since the optimization problem is strictly convex with respect to primal variable $y = (y_s(t), s \in S)$.

The simulation results for proportional fairness in this general multipath network topology are shown in Figure 2, where (a) and (b) are the optimal prices for users 1 and 2, respectively, and (c) and (d) are the optimal rates of users 1 and 2, respectively. Obviously, both the prices for users and rates of users converge to the optimum. It can also be observed that at the optimum the optimal total price associated with each path that a user uses is equal to the optimal price paid by the user.



(a)Prices for user 1 (b)Prices for user 2

(c)Rates of user 1 (d)Rates of user 2

Figure 2: Performance of the algorithm: proportional fairness

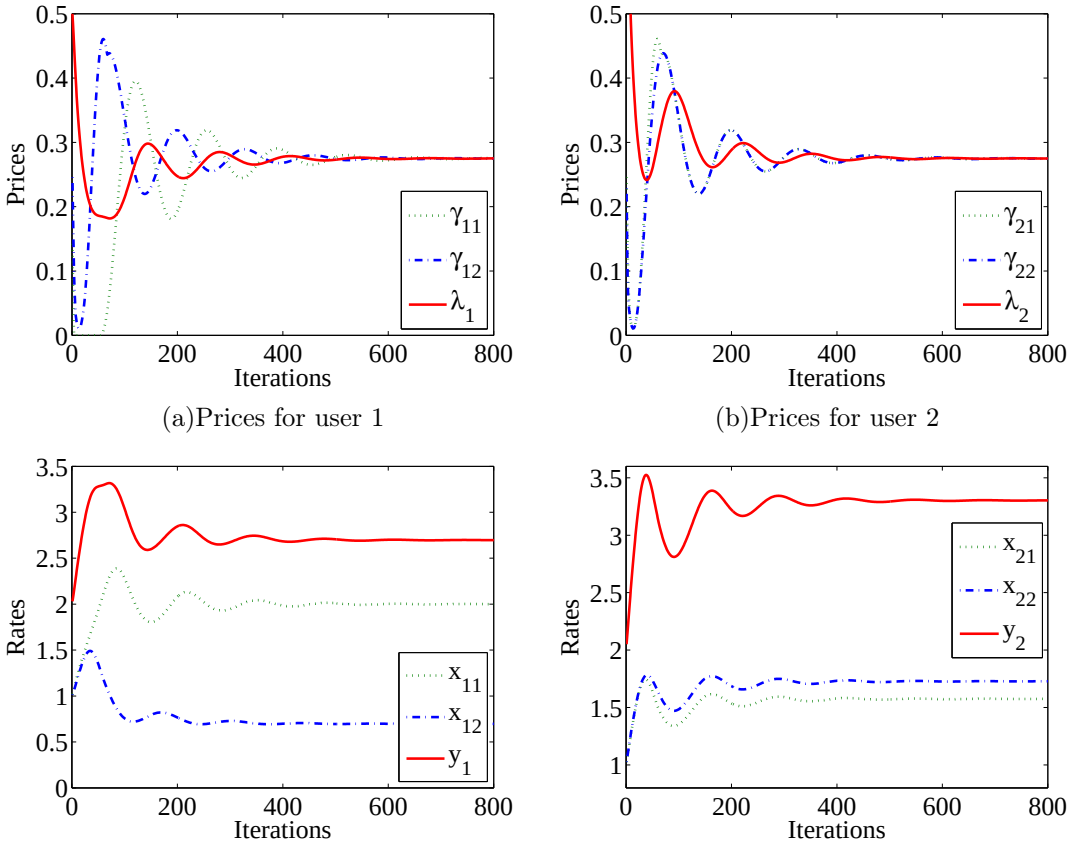
4.2 Harmonic Mean Fairness

Now we consider the harmonic mean fairness among competing users, thus the fairness parameter is $\alpha_1 = \alpha_2 = 2$. The optimums obtained by the proposed algorithm and LINGO are both listed in Table 2. Obviously, the algorithm is convergent and efficient to solve the optimum of utility maximization problem. Also, the optimal rate of user 2 on its paths is not unique, however, the total optimal rate for each user is unique since the optimization problem is strictly convex with respect to $y = (y_s(t), s \in S)$.

variable	x_{11}^*	x_{12}^*	x_{21}^*	x_{22}^*	y_1^*	y_2^*
algorithm	2.0000	0.6969	1.5748	1.7283	2.6969	3.3031
LINGO	2.0000	0.6969	1.6070	1.6961	2.6969	3.3031

Table 2. The optimum under multipath network topology: harmonic mean fairness

The simulation results for harmonic mean fairness in this general multipath network topology are shown in Figure 3, where (a) and (b) are the optimal prices for users 1 and 2, respectively, and (c) and (d) are the optimal rates of users 1 and 2, respectively. Obviously, the algorithm is convergent to the optimum of multipath network utility maximization problem. Meanwhile, at the optimum the optimal total price associated with each path of a user is equal to the optimal price paid by the user.



(a)Prices for user 1 (b)Prices for user 2

(c)Rates of user 1 (d)Rates of user 2

Figure 3: Performance of the algorithm: harmonic mean fairness

5 Conclusions

In this paper, we have studied the optimal rate control and routing schemes in communication networks with multipath routing, which are formulated as multipath network utility maximization problems. We decompose the utility optimization problem into two sub-problems, deduce the dual problem through the Lagrangian method, and give the interpretation for them from an economic point of view. In order to obtain the optimum, we propose a novel primal-dual algorithm for jointly optimizing rate control and routing, and give the window-based flow control scheme in practical end-to-end implementation. We don't assume that whether the paths of a user are node-disjoint, thus the algorithm also covers the case where the paths for a user are node-disjoint in parallel, which is regarded as concurrent multipath communication. Then, we evaluate the performance of the proposed algorithm through simulations under two different fairness concepts. Simulation results confirm that the algorithm can achieve the optimum within reasonable convergence times.

Acknowledgements

The authors would like to thank the support from the National Natural Science Foundation of China under Grants 71101124, 71171174 and 60974018, and the National Basic Research Program of China (973 Program) under Grant 2007CB307100.

Bibliography

[1] F. P. Kelly, A. Maulloo, D. Tan, Rate control in communication networks: Shadow prices, proportional fairness and stability, *Journal of the Operational Research Society*, 49(3): 237-252, 1998.

[2] S. H. Low, D. E. Lapsley, Optimization flow control, I: basic algorithm and convergence, *IEEE/ACM Transactions on Networking*, 7(6): 861-874, 1999.

[3] F. P. Kelly, Fairness and stability of end-to-end congestion control, *European Journal of Control*, 9(2-3): 159-176, 2003.

[4] S. H. Low, A duality model of TCP and queue management algorithms, *IEEE/ACM Transactions on Networking*, 11(4): 525-536, 2003.

[5] M. Chiang, S. H. Low, A. R. Calderbank, J. C. Doyle, Layering as optimization decomposition: a mathematical theory of network architectures, *Proceedings of the IEEE*, 95(1): 255-312, 2007.

[6] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, D. Towsley, Overlay TCP for multi-path routing and congestion control, presented at ENS-INRIA ARC-TCP Workshop, Paris, France, 2003.

[7] W. H. Wang, M. Palaniswami, S. H. Low, Optimal flow control and routing in multi-path networks, *Performance Evaluation*, 52(2-3): 119-132, 2003.

[8] F. P. Kelly, T. Voice, Stability of end-to-end algorithms for joint routing and rate control, *ACM Computer Communication Review*, 35(2): 5-12, 2005.

[9] X. Lin, N. B. Shroff, Utility maximization for communication networks with multipath routing, *IEEE Transactions on Automatic Control*, 51(5): 766-781, 2006.

[10] H. Han, S. Shakkottai, C. V. Hollot, D. Towsley, Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity in the Internet, *IEEE/ACM Transactions on Networking*, 14(6): 1260-1271, 2006.

[11] T. Voice, Stability of multi-path dual congestion control algorithms, *IEEE/ACM Transactions on Networking*, 15(6): 1231-1239, 2007.

[12] J. Jin, W. H. Wang, M. Palaniswami, Utility max-min fair resource allocation for communication networks with multipath routing, *Computer Communications*, 32(17): 1802-1809, 2009.

[13] J. Zhang, H. N. Lee, Energy-efficient utility maximization for wireless networks with/without multipath routing, *AEU-International Journal of Electronics and Communications*, 64(2): 99-111, 2010.

[14] J. Mo, J. Walrand, Fair end-to-end windows-based congestion control, *IEEE/ACM Transactions on Networking*, 8(5): 556-567, 2000.

[15] J. W. Lee, M. Chiang, R. A. Calderbank, Utility-optimal random access control, *IEEE Transactions on Wireless Communications*, 6(7): 2741-2751, 2007.

[16] C. Long, B. Li, Q. Zhang, B. Zhao, B. Yang, X. Guan, The end-to-end rate control in multiple-hop wireless networks: Cross-layer formulation and optimal allocation, *IEEE Journal on Selected Areas in Communications*, 26(4): 719-731, 2008.

[17] D. Bertsekas, Nonlinear Programming. 2nd ed., Athena Scientific, Nashua, USA, 1999.

[18] T. V. Lakshman, U. Madhow, The performance of TCP/IP for networks with high bandwidth-delay products and random loss, *IEEE/ACM Transactions on Networking*, 5(3): 336-350, 1997.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 668-680

Effectiveness of Program Visualization in Learning Java: a Case Study with Jeliot 3

S. Maravić Čisar, D. Radosav, R. Pinter, P. Čisar

Sanja Maravić Čisar, Robert Pinter
Subotica Tech-College of Applied Sciences, Department of Informatics
Serbia, 24000 Subotica, Marka Oreškovića 16
E-mail: sanjam@vts.su.ac.rs, probi@vts.su.ac.rs

Dragica Radosav
University of Novi Sad, Technical Faculty "Mihajlo Pupin" Zrenjanin,
Department of Informatics
Serbia, 23000 Zrenjanin, Djure Djakovica bb
E-mail: radosav@tfzr.uns.ac.rs

Petar Čisar
Telekom Srbija, Serbia, 24000 Subotica, Prvomajska 2-4
E-mail: petarc@telekom.rs

Abstract: This study was carried out to observe, measure and analyze the effects of using software visualization in teaching programming with participants from two institutions of higher educations in Serbia. Basic programming learning is notorious for complex for many novice students at university level. The visualizations of examples of program code or programming tasks could help students to grasp programming structures more easily. This paper describes an investigation about the possibilities of enhancement of learning Java using the visualization software Jeliot. An analysis of 400 students' test results indicates that a significant percentage of students had achieved better results when they were using a software visualization tool. In the experience of the authors Jeliot may yield the best results if implemented in with students who are new to the art of programming.
Keywords: software visualization, computer assisted learning, programming learning, Jeliot 3.

1 Introduction

Programming is a difficult cognitive skill to learn. Mastering the basis of a programming language is a huge problem for many students. In order to write a simple program they need to have a basic knowledge of variables, input/output of data, control structures and other areas. An even greater problem is mastering the more complex concepts such as pointers, abstraction or exception handling. And even when they have the necessary theoretical knowledge, the problem arises when they have to apply their knowledge as a whole and actually write a programming code.

Jenkins [1] identified reasons for these difficulties, such as:

- The need for good competence in problem solving;
- Students are used to courses that depend mostly on theoretical knowledge and memorization, but learning the basics of programming needs a more practical approach, based mostly on problem solving activities;

- Traditional teaching, generally based on lectures and specific programming language syntaxes, often fails to motivate students to get involved in meaningful programming activities;
- Programs have a dynamic nature, but most learning materials have a static format which makes it difficult to analyze the program's dynamic behavior;
- Students' learning needs are frequently very different within the same group. Different learning styles and previous experience make a common approach to the whole group rather difficult. Group sizes often make an individualized support to students difficult.

Educational software visualization tools are used to give students graphical representations of aspects of software systems that are inherently intangible and not obvious from them, such as the exact control flow, side effects in expression evaluations, and data dependencies. When students are given the ability to visually explore programs and algorithms, teachers expect them to be able to make better sense of program executions and programming concepts.

2 Related work

Boyle, Bradley, Chalk, Jones and Pickard [2] defined the new curriculum for London Metropolitan University's course of introductory programming. They specifically focused on visual approach. Over 600 students took part in the course. The increase in pass rates was from 12% to 23% compared to the previous year. Boyle et al. reported some significant problem concerning handling the course transition, but on average they described the graphical approach as 'very successful with the students'. Kannusmäki, Moreno, Myller and Sutinen [3] evaluated the use of the Jeliot 3 program visualization system during the second course of programming in the Virtual Studies of Computer Science distance learning program at the University of Joensuu, Finland. Gathered data showed that those who were most successful in the course used Jeliot more than the other groups involved in the research. However, most of the students (in general) by and large still used other tools to code and test their programs. The usage problems reported were mostly of a technical nature or related to the usability of the editor. The animation was criticized for being too slow and some students even found the whole system unnecessary and unsuitable for advanced courses. The positive aspects identified in the feedback included the ability to make conditional statements, loops, and objects more understandable.

Hundhausen, Douglas and Stasko [4] conducted a meta-study, analyzing 24 experimental studies on the effectiveness of algorithm visualization. They state that one of the main reasons for why visualizations are not widely used is because the teachers responsible for the courses refuse to use new methods in teaching. They also found out that the main focus in articles about visualizations is normally on their graphical means of expression instead of their learning benefits. Of the 24 studies examined, 11 showed statistically significant results of visualizations positive effects on learning, meaning that the group using a visualization system gained better learning results than the control group. Hundhausen et al. [4] also discovered that the sole use of visualization systems does not necessarily improve the learning results; it is more important to engage the learners in the subject using visualization system as an aid. Myller and Bednarik [5] presented their experiences with three empirical methodologies to study human behavior, interaction and learning with program visualization. Each of the approaches provides the research agenda with important source of data. Classroom studies inform about the practices taking place in this context and can generate testable hypotheses. Controlled experiments, when designed well, can provide answers to the previously established hypotheses and can give accurate insights into interaction and cognitive processes involved in programming. Data from surveys and questionnaire studies can be used both to collect data related to attitudes and current practices,

and generate testable hypotheses. Furthermore, all these methods can indicate issues for further development in the form of usability problems or unexpected behavior of users.

Programming is one of the essential areas taught in university studies of Computer Science and other engineering degrees and some experiences in teaching programming with e-learning application could be found in [6] and [7].

3 Software visualization

Software visualization is "the visualization of artefacts related to software and its development process" [8] and is used in the presentation, navigation and analysis of software systems. Price, Baecker and Small [9] presents the following general definition of software visualization: "Software visualization is the use of the crafts of typography, graphic design, animation and cinematography with modern human-computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software." Given that the underlying purpose of algorithm visualization is to be educationally effective, it is worth nothing that eight extant taxonomic reviews of algorithm visualization and software visualization technology have focused largely on system expressiveness, as shown in Table 1.

SV taxonomy	Descriptive Dimensions
Myers	Aspect (code, data, algorithms) × Form (static, animated)
Shu	What is visualized (data or information about data, program and/or execution, software design)
Brown	Content (direct, synthetic) × Persistence (current, history) × Transformation (incremental, discrete)
Stasko and Patterson	Aspect × Abstractness × Animation × Automation
Kraemer and Stasko	Visualization task (data collection, data analysis, storage, display) × Visualization purpose (debugging, performance evaluation or optimization, program visualization)
Roman and Cox	Scope × Abstraction × Specification method × Interface × Presentation
Price et al.	Scopex Content × Form × Method × Interaction × Effectiveness

Table 1: The descriptive dimensions of the eight extant taxonomies of SV [4]

In particular, these taxonomies have focused on three main questions [9]:

1. What kinds of programs can be visualized with a given visualization system?
2. What kinds of visualizations can a given visualization system produce?
3. What methods can one use to produce and interact with visualizations?

The primary goal of visualization is to convey information. It should convey this information in an understandable, effective, easy-to-remember way [4]. Despite its intuitive appeal as a pedagogical aid, algorithm visualization technology has failed to catch on in mainstream computer science education [9]. While those few educators who are also algorithm visualization technology developers tend to employ their own algorithm visualization technology, the majority of computer science educators tend to stick to more traditional pedagogical technologies, such as blackboards, whiteboards and overhead projectors. Why do computer science educators tend not to use algorithm visualization technology? Instructors commonly cite several reasons, including the following:

- They feel they do not have the time to learn about it.

- They feel that using it would take away time needed for other class activities.
- They feel that creating visualizations for classroom use requires too much time and effort. Note that, in the algorithm visualization technology literature, this reason is frequently used to motivate new technology that is easier to use, and that supports the more rapid creation of visualizations.
- They feel that it is simply not educationally effective.

The reason that "it is simply not educationally effective" stands out as very important because there is no reason to adopt new technology if it does not bring some improvements in the educational process. So, the aim of this paper was to research the possibilities of enhancement in learning programming language Java using the visualization software Jeliot 3. The authors concentrate on effectiveness of (algorithm) visualizations i.e. how well one performs with the visualization tool compared to others who do not use the tool. The results of successful solution of certain programming problems are compared, solutions given by students who have used this tool, and those who have not. This will lead to answers regarding the efficient application of visualization software in education.

A number of classroom studies have been conducted in order to evaluate and analyze the usage of tools in the Jeliot family by using both qualitative and quantitative methods [8].

3.1 Jeliot 3

Jeliot 3 is a program visualization application. It visualizes how a Java program is interpreted. Method calls, variables, operation are displayed on a screen as the animation goes on, allowing the student to follow the execution of a program step by step as shown in Figure 1. Programs can be created from scratch or they can be modified from previously stored code examples; all the visualization is automatically generated [10]. Jeliot 3 understands most of the Java constructs and is able to animate them.

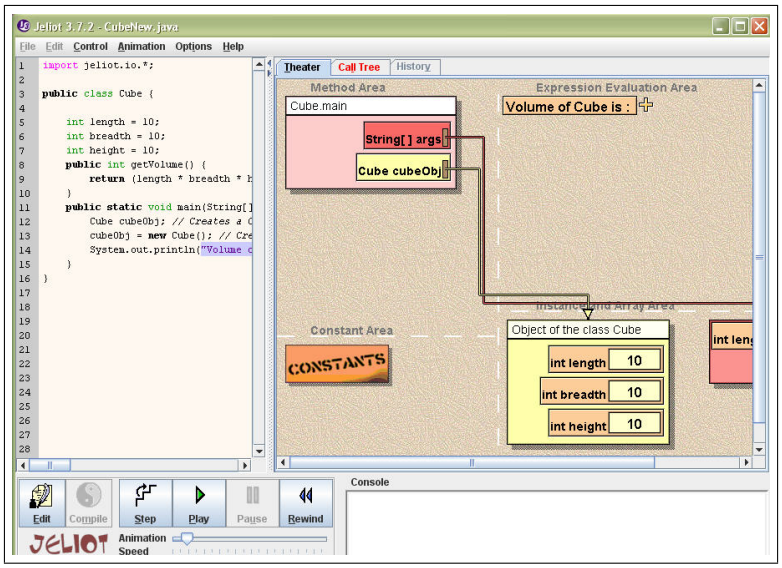


Figure 1: A screenshot of Jeliot 3

Jeliot 3 can be used in several ways for teaching and learning to program. Here are some examples from [3]:

- Lecturers can use Jeliot 3 as a part of the lecture material. They can explain the different concepts of programming through Jeliot animations. This will facilitate the construction by the students of the correct relationship between the animation and the concept, and enable them to apply it later with a reduced possibility of error [11].
- The students may use Jeliot 3 by themselves after lectures to do assignments.
- Jeliot 3 can be used in an interactive laboratory session, where students may utilize their recently acquired knowledge by writing programs and debugging them through Jeliot 3.
- Finally, Jeliot 3 provides a tool that can aid in courses where external help is not available (e.g. in distance education). Its visualization paradigm creates a reference model that can be used to explain problems by creating a common vocabulary between students and the teacher [11].

One of the possibilities that the software Jeliot 3 provides is the ability to select the *Ask Questions During Animation* option from the main menu. Whenever an expression is to be evaluated, a popup window will ask for the result. However, currently questions are generated only for assignment statements (Figure 2). The continuation of the animation is not possible until a student gives an answer to the question. In this way students have the opportunity to self evaluate their knowledge.

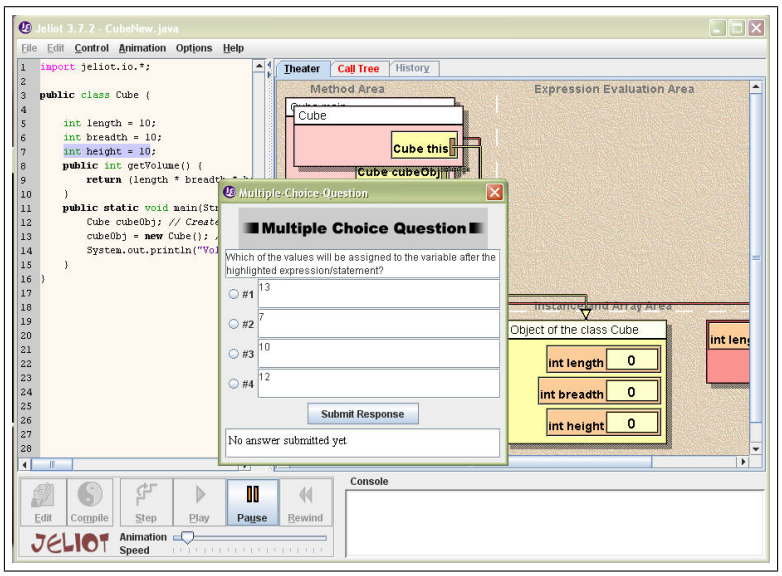


Figure 2: A screenshot with multiple choice questions

The Jeliot family's key feature has been the fully or semi-automatic visualization of the data and control flows. The development of the Jeliot family has taken more than ten years with different kinds of stages. Several versions of the concept have been developed, namely Eliot (developed at University of Helsinki, Finland), Jeliot I (developed at University of Helsinki, Finland), Jeliot 2000 (developed at Weizmann Institute, Israel). This has led to the stage when, as a product the software has become both usable and stable. The new version Jeliot 3 is a free piece of software published under General Public License (GPL). This means that the future platforms can be developed by networked teams presenting the idea of learning communities. In these communities the distinction between a teacher, a learner and a developer disappears, thus the learner can develop the tools he or she needs with the other members of the community.

Jeliot-together with its documentation, research publications, and learning materials-can be downloaded for free from http://www.cs.joensuu.fi/jeliot/.

4 Research

Wolfgram [12] states the following, "People only remember 15 percent of what they hear and 25 percent of what they see, but they remember 60 percent of what they interact with". The aim of this study was to explore the impact of the software for visualization Jeloit 3 in learning the programming language Java. Emphasis is placed on examining whether there are differences in the achievements of students who were taught Java with the help of Jeliot and those who were not. Java was selected as it has evolved to be most predominant and popular general purpose programming language of the current age. The research was the result of cooperation between two institutions of post-secondary education in Serbia: the Technical Faculty "Mihajlo Pupin" in Zrenjanin and Subotica Tech-College of Applied Science in Subotica. Preparations for the survey began in 2006, a year before the start of the experimental part of the research. Because the students from both institutions have similar education profile and many students from Subotica Tech continue their education in Zrenjanin's Technical Faculty, a team-based approach was taken for adjustment of the teaching materials. In this way one of the principals of the Bologna process of students' mobility was satisfied. The semester lasts 15 weeks and the students have a weekly lecture. The Java course consists of 30 hours of lectures and 30 hours of lab works. The lesson duration is 2x45 minutes. During the lectures the assignments for lab works were given to students. They had the task to analyze the problem at home and to draw an algorithm flow chart based on which they could write the program code at school, at lab works. At these practices students worked in pairs, two students per one PC.

Considering the fact that the participants in the survey were students from two different higher education institutions with different experience in object-oriented programming, and the fact that they were mainly beginners in programming, teaching process involved procedural programming and an introduction to object-oriented paradigm as in the material [13]. During the programming course the following topics were taught: variables, operators, decision-making statements (if-then, if-then-else, switch), the looping statements (for, while, do-while), the branching statements (break, continue, return), arrays (one-dimensional and two-dimensional), methods, constructors and inheritance. The research lasted for two school years (2007-2009) and 400 students were included. In the first year of study the sample consisted of 200 students. From the total number of respondents 47 were female (23.5%) and 153 were male students (76.5%). In the second year of study the sample consisted again of 200 students, among them 34 were female (17%) and 166 were male (83%). A certain part of the results of this research relating to only 45 students from Subotica Tech collected in the spring of 2009 could be found in [14]. The paper includes the results of a questionnaire that was used to collect reflections on programming and the application of the Jeliot program of students. The aim of this questionnaire was to find out how students had accommodated Jeliot into their learning processes.

In the research the model of pedagogical experiment with parallel groups was applied. Students were divided into three groups. The control group learned and worked in the traditional way, in the classrooms and under teachers monitoring. This group has lectures in the traditional way. The lecturers in this group used blackboard or whiteboard to present teaching material. All necessary explanations were drawn on the board, such as explaining the program code that was used as example during the class, or what took place during checking conditions in if-then-else statement, or in an algorithm flow chart. Students of the control group used JCreator (lightweight development environment for Java technologies) for writing and debugging code examples at the lab exercises, as well as for solving homework tasks. Students of this group used a textbook for

studying. For the first experimental group, in addition to the traditional ways of learning, an experimental factor was introduced - learning using visualization software Jeliot and PowerPoint presentations (blended learning) while in the second experimental group only e-teaching materials were applied. The lectures notes were in form of PowerPoint presentations. For explaining the program code that was used as example during the classes the lecturers used Jeliot. In this way the lectures become more interactive. Students of both experimental groups had a two-hour workshop at the beginning of the course to become familiar with it. The main purpose of Jeliot 3 is not to be used as a debugging tool, but it is convenient for analyzing smaller program codes as was used in this research. During the first month of the semester (a total of 8 lab exercises) they solved their tasks with the help of the lecturer. Based on the algorithm flow chart which they had to do as preparation for the class, they had to write program code in Jeliot and to use Jeliot as compiler. The students debugged the errors which were indicated by Jeliot together with the lecturer. After this first period of learning with Jeliot they had to complete all task alone (in groups of two students). Of course, during the entire class the lecturer was present and his role was to give an extra explanation if it was necessary.

Most of the time, students are trained to develop very simple programs starting from scratch. In fact, many students will be involved in evolution-related software after completion of their studies, so it is very important for them to know how to read and change existing large program codes [14]. However, evolving and understanding large existing software includes quite different activities such as recovering/understanding the actual architecture of the system, understanding some part of its source code and documentation (may be in some programming language that the student had not studied before). Including a new feature in such software then requires performing impact analysis, regression testing, etc [15]. Considering this, the next type of problems that students had was to figure out what an unknown program code did and to create appropriate test cases. Depending on which group they belonged to, students used Jeliot or JCreator to complete this task.

The comparison of the results of the control and experimental groups was performed by testing the hypothesis. Significant differences of the three groups of students were determined based on the critical values for probability levels of 0.05 and 0.01. The following hypothesis was stated: *Application of software for visualization Jeliot 3 does not affect the process of learning Java programming language.* The test was used to examine the students' knowledge. The exam was a traditional paper-and-pencil test with 20 multiple choice questions, where a standard grading procedure has been applied to grade these test items (one point for a correct answer, zero points for an incorrect one). The statistical analysis of the performed test included calculation of the arithmetic mean, standard deviation, standard error of the mean and interval of variation. The one-way ANOVA was used to test for differences among three groups of students. In order to determine which groups differ from each other the post-test for pair-wise comparisons Tukey's HSD test (honestly significant difference) was implemented. Statistical analysis of the results was done using the site VassarStats [16] that offers the possibility of online statistical calculations.

The total number of students who participated in the first year of the research was 200. The control group consisted of 60 students who had lectures in the traditional way. In the first experimental group there were 65 students who had the traditional way of lecturing accompanied with the PowerPoint presentations and software for the visualizations. In the second experimental group there were 75 students who only had e-learning material. The results of the research are given in Table 2.

The analysis of the descriptive statistical parameters has shown that the average value of the scored points for the first experimental group was the largest (11.05) with a standard deviation of 4.78 points. The lowest average of points had a control group, 7.33 with standard deviation of 4.85 points. Based on these results, it can be said that the first experimental group had the

Group	N	Mean	Std.Dev.	Std.Err.	Min	Max
Control	60	7.3333	4.8526	0.6265	0	18
Experimental I	65	11.0462	4.7777	0.5926	0	20
Experimental II	75	10.9067	5.5585	0.6418	0	20

Table 2: Descriptive statistical parameters of the first year of the research

highest average score and thus showed the best results on the test. Graphical representation of the average number of points is given in Figure 3.

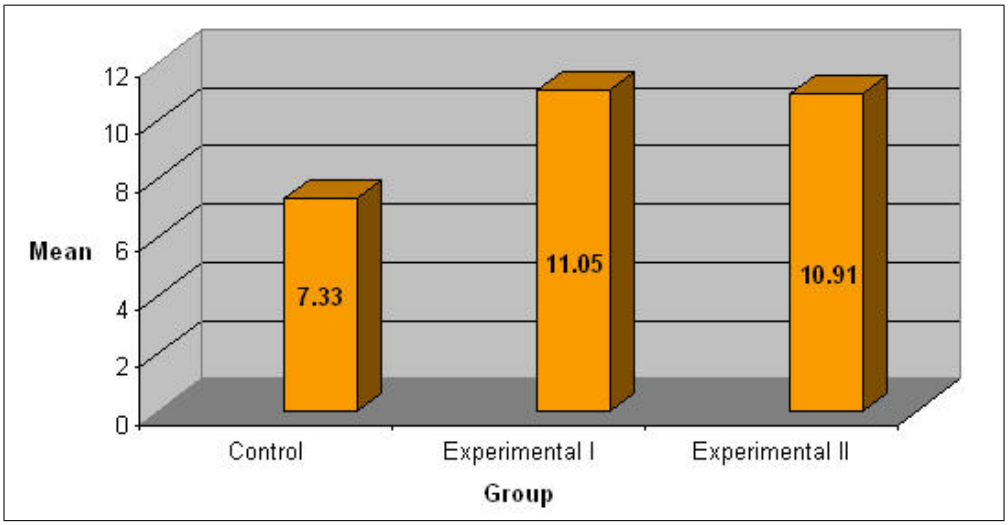


Figure 3: The average number of points in the first year of the experiment

The results of the ANOVA analysis are shown in Table 3 ($p \leq 0.05$).

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	556.5784615	2	278.2892308	10.67313056	3.97E-05	3.04175303
Within Groups	5136.541538	197	26.07381492			
Total	5693.12	199				

Table 3: The results of the ANOVA analysis in the first year of the experiment ($p \leq 0.05$)

The results of the ANOVA analysis are shown in Table 4 ($p \leq 0.01$).

The calculated value for F of 10.67 indicates significant differences between the studied groups. Significant differences between individual groups were determined using the Tukey test. The absolute (unsigned) difference between any two sample means required for significance at the designated level (M1=mean of the control group, M2=mean of the first experimental group and M3=mean of the second experimental group) is:
M1-M2=3.7162
M1-M3=3.5767
M2-M3=0.1395

Next the HSD is computed (HSD). If the difference is larger than the HSD, then the difference is said to be significant.

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	556.5784615	2	278.2892308	10.67313056	3.97E-05	4.71452043
Within Groups	5136.541538	197	26.07381492			
Total	5693.12	199				

Table 4: The results of the ANOVA analysis in the first year of the experiment ($p \leq 0.01$)

HSD[.05]=2.1; HSD[.01]=2.62
M2 vs. M3 insignificant
M1 vs. M3 $P < .01$
M1 vs. M2 $P < .01$

Based on the results of HSD, it can be concluded that there is a significant difference between the control and experimental groups I and II ($p < 0.01$). The hypothesis of equality of the control and experimental groups cannot be accepted, which means that the implementation of software for visualization Jeliot 3 does have an influence on the process of learning Java. No significant differences between the experimental groups have been recognized.

The total number of students who participated in the second year of the research was 200. The control group consisted from 68 students who had lectures in the traditional way. In the first experimental group there were 66 students who had the traditional way of lecturing accompanied with the PowerPoint presentations and software for the visualizations. In the second experimental group there were 76 students who had only e-learning material. The results of the research are given bellow in Table 5.

Group	N	Mean	Std.Dev.	Std.Err.	Min	Max
Control	68	7.2206	4.6548	0.5645	0	17
Experimental I	66	10.8333	4.6923	0.5776	1	19
Experimental II	66	11.5	5.3787	0.6621	1	20

Table 5: Descriptive statistical parameters of the second year of the research

The analysis of the descriptive statistical parameters has shown that the average value of the scored points for the second experimental group was the largest (11.5) with a standard deviation of 5.378 points. The lowest average of points had a control group, 7.22 with a standard deviation of 4.65 points. Based on these results, it can be said that the second experimental group had the highest average score and thus showed the best results on the test. The graphical representation of the average number of points is given in Figure 4.

The results of the ANOVA analysis are shown in Table 6 ($p < 0.05$).

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	713.5172	2	356.7586	14.7546	1.07E-06	3.04175303
Within Groups	4763.358	197	24.17948			
Total	5476.875	199				

Table 6: The results of the ANOVA analysis in the second year of the experiment ($p \leq 0.05$)

The calculated value for F of 14.75 indicates the existence of significant differences between

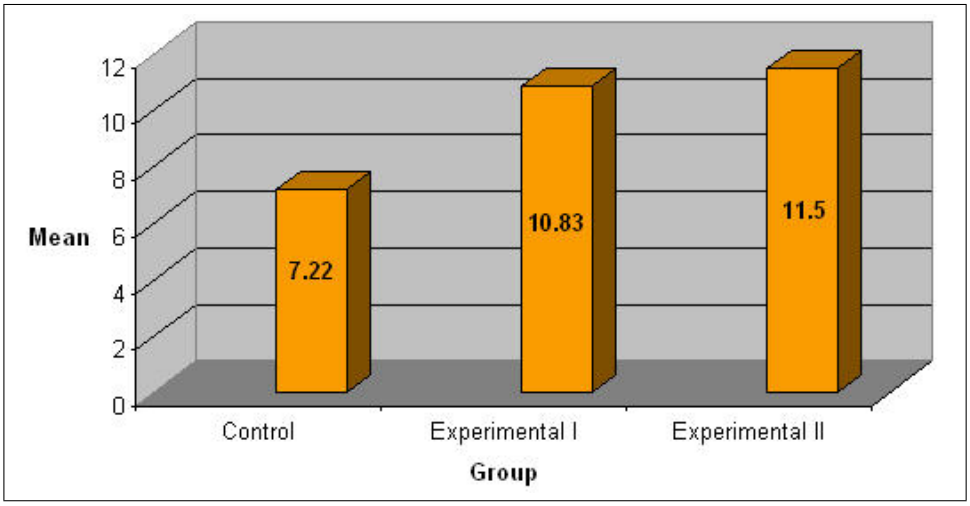


Figure 4: The average number of points in the second year of the experiment

the studied groups. Significant differences between individual groups were determined using the Tukey test. The absolute (unsigned) difference between any two sample means required for significance at the designated level is:
M1-M2=-3.6127
M1-M3=-4.2794
M2-M3=0.6667.

The following step is computing HSD:
HSD[.05]=2.01; HSD[.01]=2.51
M1 vs. M2 $P < .01$
M1 vs. M3 $P < .01$
M2 vs. M3 insignificant.

The results of the ANOVA analysis are shown in Table 7 ($p \leq 0.01$).

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	713.5172	2	356.7586	14.7546	1.07E-06	4.71452043
Within Groups	4763.358	197	24.17948			
Total	5476.875	199				

Table 7: The results of the ANOVA analysis in the second year of the experiment ($p \leq 0.01$)

Based on the results of HSD, it can be concluded that there is a significant difference between the control and experimental groups I and II ($p < 0.01$). The hypothesis of equality of the control and experimental groups cannot be accepted, which means that the implementation of the software for visualization Jeliot 3 does have an influence on the process of learning Java. There are no recognizable differences between the experimental groups. Based on the results of the two-year research in which a total of 400 students have been included it can be stated that the experimental groups I and II have been highly successful compared with the control group in learning Java. Thus, the set hypothesis that application of the software for visualization Jeliot 3 has no influence on the process of learning Java programming language was refuted. Between the experimental groups no significant statistical difference was found.

Students who were in the experimental groups after completing the course filled out the survey in order to obtain information about their opinion on learning using visualization software. First,

at the middle of the semester, students' satisfaction with their progress in learning and their motivation for further work was examined. To express views on the motivation of the students the Likert scale with 5 responses was applied (very high, high, medium, low and very low). The results are shown in Fig. 5 (all values are expressed in percentages).

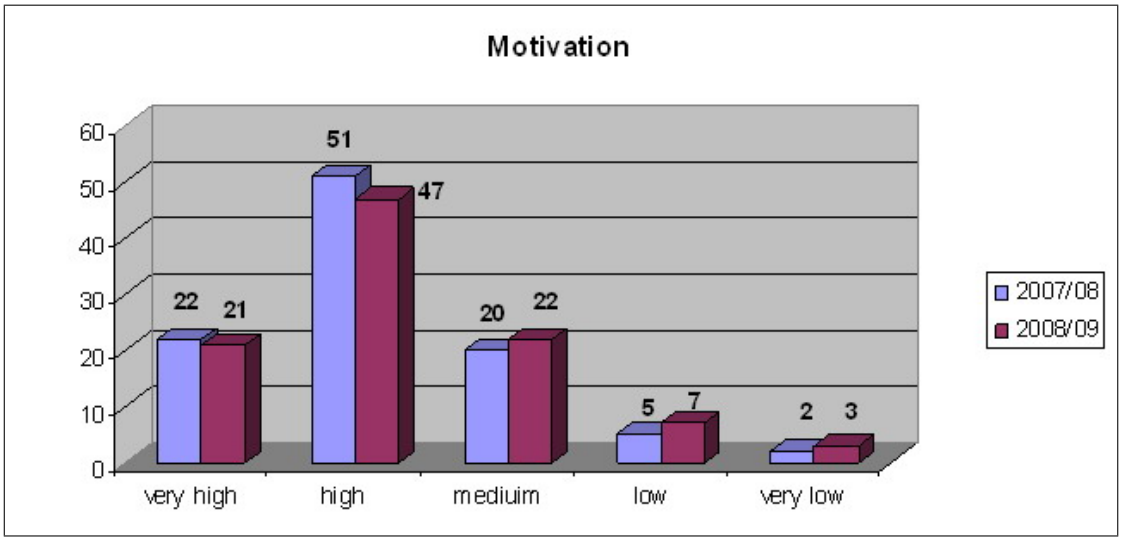


Figure 5: The motivation of the students

Figure 6 shows the results (given in percentages) of students' satisfaction in their learning progress. They could express their attitude with the following responses: very satisfied, satisfied, neutral, dissatisfied, very dissatisfied (Likert scale).

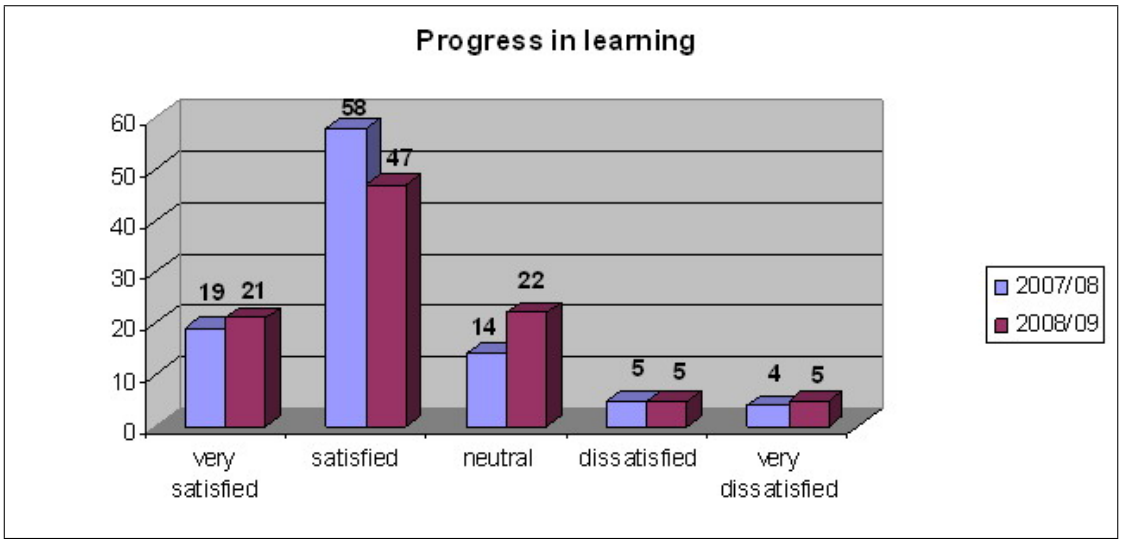


Figure 6: Student's satisfaction in learning progress

The students of the experimental group were asked what they think about the educational possibilities of Jeliot 3. Their answers depended on the level of knowledge of Java and object-oriented programming concepts in general. Responses were ranged from the answer that Jeliot3 is a great help for beginners, that it is a powerful educational tool thanks to the visualization and because of that, it made learning Java easy, to the opinions of some students who have already had experience in Java programming that it was a pure waste of time for them. Students

who expressed a negative opinion about Jeliot said that it was difficult for them to adapt their knowledge of object-oriented programming and their style of programming to the new tool, also they had objections about the elements of visualization code which were disturbing their attention. What all the negative comments had in common was that they were given by students who had already used the Java language, but almost all of them pointed out that they believed that Jeliot could be very useful at the beginning of the process of learning Java. The results of the research showed that the passing rate of the exam increased for a few percent when using the Jeliot3 (for 3% in the first year and 2.67% in the second year of research). But, in order to be able to claim that result of increased passing is the consequence of using Jeliot3, it would be necessary to perform some additional research focused solely on that piece of information.

5 Conclusions

The paper discussed the problem of the applications of software for visualization Jeliot3 in learning the programming language Java. The study involved 400 students of two higher education institutions in Serbia. Based on the research, which lasted for two years, it can be stated that there are significant differences in the achievements of students who were taught in the traditional way, and those who have used Jeliot3. Through well-defined messages from the compiler Jeliot indicates to the user where the syntax errors in code are. The goal is to help novices understand basic concepts of algorithms and programming like assignment, I/O and control flow, whose dynamic aspects are not easily grasped just by looking at the static representation of an algorithm in a programming language [11]. Software Visualization tools are intended to be used in the early stages of the learning path of a programmer, teaching the students the basics of programming, algorithms, and the software development cycle [17]. While a future experiment involving Jeliot and more advanced students (in terms of programming) may provide interesting results, it is the belief of the authors that this particular program works best with beginner students.

It is not possible to "eject" teachers from the teaching process and replace them with a computer, but it is necessary to ensure their active participation during the animation code so that students receive the necessary explanation. Using visualization tools should be intense, in the sense that teachers need to use it in the teaching process, and students in laboratory exercises as well as when working at home. Of course, further research in this field is required. Although this study did not directly deal with the influence of Jeliot to accelerate learning processes, based on previous research of the authors [18] about the visualization of some selected parts of the course Analogue and Digital Electronics that has shown that interactive animations can significantly contribute to increasing the speed of learning, it can be assumed that Jeliot could be especially helpful to beginners in learning Java in the same way. It must be noted that it is already clear there is a need for different tools for learning depending on the level of students' knowledge of the studied materials. Advanced students and even students with only superficial experience in programming are very sensitive to the change of tools that are used as a code editor, if it does not provide significant improvement over the tools they are used on. In other words, the individual characteristics of students, including the level of knowledge, must be taken into thorough consideration because the demands of students are changing rapidly.

Bibliography

[1] T. Jenkins, "On the Difficulty of Learning to Program", *in Proc. of 3rd Annual LTSN-ICS Conference*,Loughborough University, UK, 53-58, 2002.

[2] T. Boyle, C. Bradley, P. Chalk, R. Jones, P. Pickard, Using blended learning to improve student success rates in learning to program. *Journal of Educational Media, special edition on Blended Learning*, 28(2-3): 165-178, 2003.

[3] O. Kannusmäki, A. Moreno, N. Myller, E. Sutinen. What a novice wants: Students using program visualization in distance programming course, *Proc. of the Third Program Visualization Workshop (PVW'04)*, Warwick, UK, pp. 126-133, 2004.

[4] C. D. Hundhausen, S. A. Douglas, J. T. Stasko, A Meta-Study of Algorithm Visualization Effectiveness, *Journal of Visual Languages and Computing*, 259-290, 2002.

[5] N. Myller, R. Bednarik, Methodologies for studies of program visualization, *Proc. of the Methods, Materials and Tools for Programming Education Conference*, 37-42, 2006.

[6] M. D. Afonso Suarez, C. Guerra Artal, F. M. Tejera Hernandez, E-learning multimedia applications:Towards an engineering of content creation, *Int. J. of Computers, Communications & Control*, 3(2): 116-124, 2008.

[7] C. Guerra Artal, M. D. Afonso Suarez, I. Santana Perez, R. Quesada Lopez, OLC, On-Line Compiler to Teach Programming Languages, *Int. J. of Computers, Communications & Control*, 3(1): 69-79, 2008.

[8] S. Diehl, Evolution, *In Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software Springer Verlag*, pp. 149-160, 2007. [Online]. Available: `http://www.springerlink.com/content/m373254212740552/fulltext.pdf`

[9] B. A. Price, R. M. Baecker, I. S. Small, An Introduction to Software Visualization, in Software Visualization, J. Stasko, J. Dominque, M. Brown, B. Price (Eds.), London, England MIT Press, 4-26, 1998.

[10] [Online]. Available: `http://cs.joensuu.fi/jeliot/description.php`

[11] R. Ben-Bassat Levy, M. Ben-Ari, P. A. Uronen, The Jeliot 2000 program Animation System, *Computers & Education*, 40(1): 1-15, 2003.

[12] D. E. Wolfgram, Creating multimedia presentations, Que Corp, IN, USA, 1994.

[13] [Online]. Available: `http://stwww.weizmann.ac.il/g-cs/benari/lov/lov.html`

[14] S. Maravić Čisar, R. Pinter, D. Radosav, P. Čisar, Software Visualization: the Educational Tool to Enhance Student Learning, *Proc. of 33rd International Convention MIPRO 2010, Computers in Education*, May 24-28, 2010, Opatija, Croatia, ISSN 1847-3938, ISBN 978-953-233-054-0, 4: 234-238, 2010.

[15] A. Van Deursen, J. M. Favre, Experiences in Teaching Software Evolution and Program Comprehension. Available: `http://www.tzi.de/st/papers/teaching-iwpc03.pdf`

[16] Available: `http://faculty.vassar.edu/lowry/VassarStats.html`

[17] A. Moreno, M. S. Joy, Jeliot 3 in a Demanding Educational Setting, *Fourth International Program Visualization Workshop*, 29-30 June 2006, Florence, Italy

[18] R. Pinter, D. Radosav, S. Maravić Čisar, Interactive Animation in Developing e-Learning Contents, *Proceedings of 33rd International Convetion MIPRO 2010, Computers in Education*, May 24-28, 2010, Opatija, Croatia, ISSN 1847-3938, ISBN 978-953-233-054-0, 4: 251-254

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 681-700

A Novel Parallel Transmission Strategy for Data Grid

Q. Ming-Cheng, W. Xiang-Hu, Y. Xiao-Zong

QU Ming-Cheng, WU Xiang-Hu, Yang Xiao-Zong
School of Computer Science and Technology, Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
E-mail: qumingcheng@126.com, wuxianghu@hit.edu.com, yangxz@hit.edu.com

Abstract: Creation of multi-copies accelerates data transmission and reduces network traffic, but it causes overhead storage and additional network traffic. A variety of parallel transmission algorithms based on GridFTP and multi-copy can be used to accelerate data transmission further, but they can not adapt to a wide range of network, and they can not be used to solve the problems of storage space and network traffic waste. GridTorrent combined with BitTorrent and GridFTP has compatibility with grid and has flexible scalability, but the speed is very slow when there are few peers, to solve this problem multi-copy is needed also. To achieve multiple optimization objectives of storage space saving, suitable for two kinds of application modes(i.e. parallel transfer based on GridFTP and BitTorrent), adaptability for wide range of network and higher performance when there are fewer peers, based on the idea of GridTorrent, a distributed storage model, parallel transfer algorithm and virtual peer strategy are proposed. In experiments the performance is compared among the verification system VPG-Torrent and original parallel transfer algorithm (DCDA) only based on GridfTP & multi-copy and GridTorrent. When the same amount of data is deployed VPG-Torrent has better performance than DCDA, and when there are fewer peers VPG-Torrent also exceed GridTorrent, which prove the effectiveness of VPG-Torrent.
Keywords: Data grid, distributed storage model, parallel transmission.

1 Introduction

Data Grid is used during data-intensive computing applications to facilitate the efficient use of distributed data resources. It focuses on the management of data in a wide, heterogeneous, and distributed environment; acquisition of data from a variety of heterogeneous data resources, and extraction of useful information from the data source through collaboration and geographical distribution operation [1]. Data grid has been widely used in many fields currently. Such as in scientific computing, physics, biology, astronomy, oceans, atmosphere, manufacturing and so on [2–4], and its desktop operation interface has been implemented well [5–7]. Combining computing grid and data grid for data-intensive computing applications has become a trend now [8–11]. However, to meet time requirement of data-intensive computing applications, how to accelerate transmission is the key problem faced by data grid [12].

In order to improve the user's access speed and reduce the network load effectively, much work has been done in recent years on how to create a reasonable number of copies and locate them [13, 14] efficiently. Data transmission speed is improved and network traffic is reduced. However, these studies overlooked the mass properties of data in data grid, and storage and network overhead were caused by the creation of multiple copies. If there is a huge amount of data, then it is more harmful than good to create multi-copies [15, 16].

Bittorrent is a typical application example of P2P technology. In BT system every peers should upload data when they download. The download speed mainly depends on the number

of simultaneous peers. Now BT is mainly used in file sharing of public community where there are mass accesses. In scientific computing areas there are a few users in general, and computing tasks have strict time requirements in most cases, so Bittorrent reveals some shortcomings.

OverSim is a flexible overlay network simulation framework. All OverSim protocol implementations can be used without code modifications in real networks. Several applications like i3, Scribe, P2PNS, and SimMUD based on these protocols are available. All these implementations can be used for both simulation as well as real world networks. OverSim utilizes the graphical interface of OMNeT++ to display overlay and underlay topology and all network packets. The paper [17, 18] shows that with OverSim simulations of overlay networks with up to 100,000 nodes are feasible. Moreover it provides a fully configurable network topology with realistic bandwidths, packet delays and packet losses.

2 Related Works and Ideas

GridFTP provides a striped transmission mode to enable the GridFTP client to download the same data in different data blocks from multiple GridFTP servers in parallel mode, see Figure 1. Based on GridFTP protocol and multi-copy technologies, many parallel transmission algorithms have been developed [19–22], In these studies, a common feature is to divide original data into sub-blocks, then download these sub-blocks from different nodes through some decision-making in parallel mode, when download is finished the sub-blocks are merged into original data.

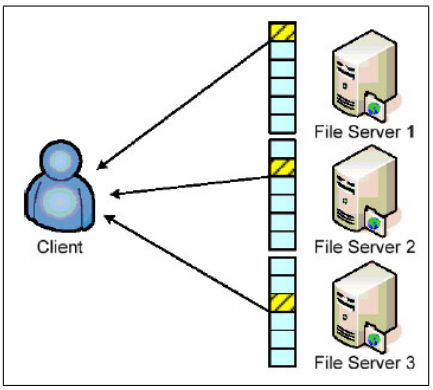


Figure 1: Striped transmission mode of GridFTP

For these parallel transfer algorithms based on client/server(C/S) mode, the performance will decrease dramatically as the amount of access increase, so new copy must be deployed to solve this problem. In spite of this defect, this C/S mode is still necessary for scientific computing community and computing tasks which have higher requirement in reliability, security and some other special data service. But the problems of storage resource waste and network traffic caused by the the deployment of multiple copies can not be neglected. So, how to use a less redundant storage and make full use of GridFTP protocol to improve data transmission speed is a challenge faced by data grid now.

Bittorrent has better adaptability in a wide range of network. But its performance also has much to be desired when there are a few peers or the peers do not allow to upload or upload speed is strictly limited. Integrated the advantages of GridFTP and Bittorrent, some scholars proposed GridTorrent as shown in [23–25].

GridTorrent (see Figure 2) is an implementation of the popular BitTorrent protocol designed to interface and integrate with well-defined and deployed Data Grid components and protocols (e.g. GridFTP, RLS). Just like BitTorrent, GridTorrent is based on peer-to-peer technique, that

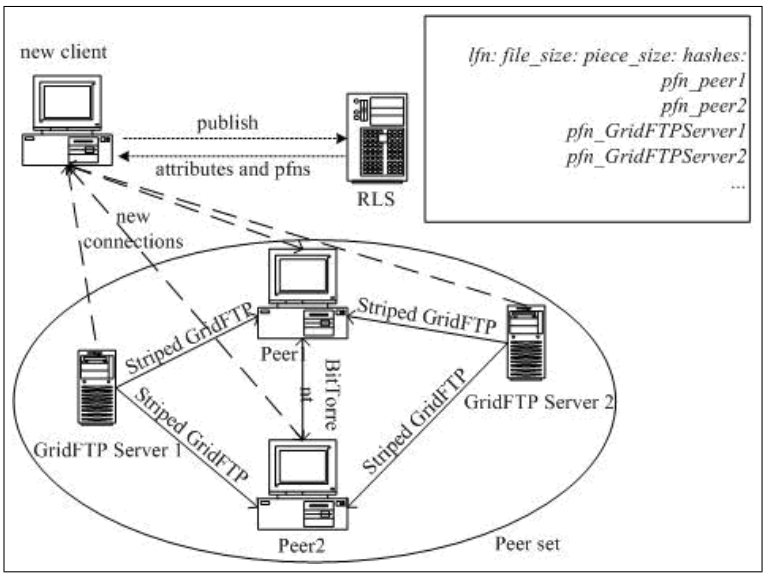


Figure 2: GridTorrent

allow clients to download files from multiple sources while uploading them to other users at the same time, rather than obtaining them from a central server. By dividing files into fragments, GridTorrent can combine the best out of the two protocols [25].

The data in Bittorrent community is generally obtained from other people's works, like music or movie file, whereas every scientific data is generated in scientific community. Therefore, scientific data are more sensitive than data used in Bittorrent community. So the number of concurrent peers will be very limited.

From Figure 2, when the sum of peers is small, the performance of GridTorrent will be degraded as GridFTP. So like other parallel transfer algorithms based on GridFTP and multi-copies, a number of copies should be deployed to enhance the transmission speed. Accordingly storage space and network traffic waste are serious.

Bittorrent algorithms do not rely on global data storage knowledge to schedule. Because the number of participating peers, network speed, peer-owned data blocks are dynamically changing. So if we directly divide overall data into sub-blocks of the same size and then deploy them to some servers, the parallel transmission capacity of the servers can not be maximized. And for the previous parallel transmission algorithms, it will inevitably lead to some of the nodes complete their download task first, i.e., all the data stored by the nodes have been downloaded. It can not guarantee each task continuously download from start to finish and all the tasks finish at or tend to the same time.

This paper is based on the idea of GridTorrent, and study how to achieve higher speed using less storage when the number of participating peers is small or P2P is not allowed, so that speed does not depend on the number of concurrent peers. To solve the problem from three levels: first, P2P is not allowed (only download from GridFTP servers); second, P2P is allowed, but the number of peers is small; third, large-scale peer transmission. Therefore, a data grid distributed storage model which uses less data redundancy and can ensure the reliability of data is proposed in this paper. A parallel scheduler and a parallel transmission algorithm based on the model and GridTorrent are thus presented.

Main contributions: a distributed storage model is put forward, the model can ensure data reliability using less storage. A scheduler is presented based on model, and further a parallel transmission algorithm (PTABM) is put forward based on model and scheduler. Comparative

experiments are carried, our algorithm PTABM achieved the same performance compared with DCDA, meanwhile the strategy proposed in this paper has absolute advantage in storage space, and also network load can be reduced greatly during copy deployment and migration.

Virtual peer(Vpeer) is presented in section 4.4 based on PTABM and the idea of GridTorrent. During the scheduling, Vpeer is regarded as a general peer, only the transmission from Vpeer is based on PTABM. This system is called VPG-Torrent, and it can adapt to the situation of large-scale networks.

Compared with GridTorrent, when the number of peers is small (dozens of peers), VPG-Torrent make the average speed of every peers be higher, as the number of peers increases gradually, the performance of them gets closer gradually, it is because that BitTorrent accounts for a dominant role when large-scale peers exist. The outcome achieved by this paper is:Using less storage space achieved objective of rapid transmission; and the performance is not affected by the sum of concurrent peers; VPG-Torrent can met two kinds of application modes, i.e., one is parallel transmission only based on GridFTP, the other one is BitTorrent service in large-scale network. This is the lack of previous studies.

The experiments include two parts: parallel transmission only based on GridFTP, simulation of VPG-Torrent.

(1)The first part is to verify the performance of VPG-Torrent when BitTorrent is not allowed. We implemented a Data Dispatcher, RLS server and PTABM, and built environment platform composed of 7 computers. Good results were achieved compared with previous parallel transmission algorithm (DCDA).

(2)The second part is to verify the performance of VPG-Torrent when BitTorrent is allowed, and compare with GridTorrent.

In order to simulate large-scale concurrent access, oversim is used in simulation experiment, and also good results were achieved.

In following chapters, firstly a distributed storage model is put forward, then a scheduler based on model is proposed, further a parallel transmission algorithm is given, and then expounded the verification system modules and work process, at last experiments and analysis are given.

3 Distributed Storage Model

In this section we will put forward a block-devision and block-storage strategy (called distributed storage model), so some necessary definitions will be given first. Definitions 1-6 are given to explain how to divide data file and how to storage the blocks into grid nodes. Definitions 7-10 are used to show the properties of distributed storage model

Definition 1. (metasum partition and metadata) Let M represent the total amount of data. Divide the overall data into sub-blocks of equal size, so that metablks=k(k-1)*metasum, and k is the number of copies, and metablks is divided shares. firstly the data is divided into k(k-1) shares, then each share is divided into metasum shares, and metasum is a variable parameter. So let the amount of data for each share be metadata, and can be expressed as:

$$metadata = \frac{M}{k(k-1) * metasum} \tag{1}$$

Definition 2. (local data) The k(k-1)*metasum shares of data defined in definition 1 are evenly distributed among k nodes. So each node contains (k-1)*metasum shares of data. These data is called local data LND_i of node N_i.

Definition 3. (local data virtual group) The local data of N_i is classified into some virtual groups.The metasum shares of data is classified as one group, and the classified groups are numbered uniquely within nodes. From definition 2, we can see that the local data of each node can be classified into (k-1) virtual groups. Let the virtual group be $G_i{}^j$ ($0 \le i \le$ k-1, $0 \le j \le$ k-2). The sum of data blocks that one group contains is metasum. Let G_i represent all the vitural groups that node N_i contains, and G represents the current vitural group.

Definition 4. (Remaining Node Set) After excluding node N_i, all remainder grid nodes is called remaining node set: $\overline{N}_i^e (0 \le e \le k-2)$ *and* $\bigcup_{y=0}^{k-2} \overline{N}_i^y = \bigcup_{j=0, j\neq i}^{k-1} N_j$, $\overline{N}_i^y = N_j$, $\begin{cases} y = j & \text{if } j < i \\ y = j-1 & \text{if } j > i \end{cases}$

Definition 5. (Cross Storage) Croup G_i of N_i is distributed into $\overline{N}_i^e$, that is $G_i \to \overline{N}_i^e$, and $\left(\bigcup_{r=e}^{w} (G_i^r \to \overline{N}_i^e) \right) \Big|_{e=0}^{k-2} \{w = (e + (p-1)) \bmod k, p \le k-1\}$, where p is the specified constant, and symbol $\to$ means 'storage into'. This is why such storage rule is called cross storage.

Definition 6. (Distributed storage Model) All the data AND_i that node N_i contains includes local data LND_i and cross storage data OND_i. From the definitions 3 to 5 above: AND_i=(LND_i)∪

(OND_i)= $\left(\bigcup_{j=0}^{k-2} G_i^j \right) \cup \left(\bigcup_{(a=0, a\neq i)}^{k-1} \left(\bigcup_{r=e}^{w} G_a^r \right) \right) \begin{cases} e = i-1 (a < i) \\ e = i \quad (a > i) \end{cases}$ &

$\begin{cases} w = (e + (p-1)) \bmod k \\ p \le k-1 \end{cases}$. p is a specified constant, so such storage mode is called Distributed storage Model. (Note: OND represents local data and OND_i represents local data of node N_i)

Theorem 1. *(p integrity) If data storage meets Distributed Storage Model, when there are arbitrary p nodes are not available, the merger of data at the remaining k-p nodes is still equivalent to M, that is, the data is complete, this property is called p integrity.*

Proof: For an arbitrary local data G_i^x, $i \in (0, k-1)$, $x \in (0, k-2)$, *where* G_i^x represents arbitrary one local virtual group of one arbitrary node N_i.

From definition 5, we can know that $e \in (x-(p-1), k-2)$, $r \in (e, w)$, and w=(e+(p-1) mod k), p≤k-1. So from e to w there are p numbers. Retrieve a subset e and $e \in (x-(p-1), x) \subset (0, k-2)$.Because of each value of e, r starts from e will get p values.

When e varies from x-(p-1) to x, r is in the range of { (x - (p - 1),x),...,(x,x + (p - 1))} .The 'x' is repeated p times because e has undergone P changes, that is,G_i^x has been stored into node $\overline{N}_i^e$ for p times. Here e∈(x-(p-1),x).

Adding the node that virtual group G_i^x lies in, there are p +1 nodes where store G_i^x Therefore, if there are arbitrary P nodes are unavailable, a node store G_i^x is available. As G_i^x is arbitrary, so all the virtual groups of all the nodes meet the above derivation. The proof is completed. □

Lemma 1. *If the data storage met the Distributed storage Model, then the overall amount of stored data can be expressed as: k(1+p)(k-1)*metasum.*

Proof: Derivated from definition 6:

$$\begin{aligned} \sum_{i=0}^{k-1} AND_i &= \sum_{i=0}^{k-1} ((k-1) * metasum|_{LND_i} + (1+p) * metasum * (k-1)|_{OND_i}) \\ &= k(k-1)(1+p) * metasum \end{aligned}$$

The proof is completed. □

Definition 7. (Storage Space Usage Rate:SSUR) The total amount of data that stored by Distributed storage Model divided by the total amount of data that stored by k-complete copies is defined as SSUR. From definition 1 and lemma 1:

$$SSUR = \frac{k(k-1)(1+p) * metasum}{k * k(k-1) * metasum} = \frac{1+p}{k} \tag{2}$$

Definition 8. (Limit Speed Ratio: Vratio) Let Vratio=V_i/V_j represents the limit ratio of the amount of data that downloaded from nodes N_i and N_j seperately. Then:

$$\frac{1}{(k-1)(1+p) * metasum} \leq V_{ratio} \leq (k-1)(1+p) * metasum \tag{3}$$

Definition 9. (Maximum Speed Ceiling: V_top) During the period of downloading data, the amount of data downloaded from any node can not excess AND_i. Then: $\left(V_{max}/\sum_{i=0}^{k-1} V_i\right) k(k-1) * metasum \leq (k-1)(1+p) * metasum \Rightarrow V_{top} \leq \frac{(1+p)}{k}(\sum_{i=0}^{k-1} V_i)$. V_top is called Maximum Speed Ceiling.

Example 1 Given: k=4,P=2,metasum=4. Questions: (A), Give the calculus course that distributes the data to grid nodes according to model. (B). Calculate SSUR.

Solution A:

(1) From definition 1: The data divided shares=4*(4-1)*4=48. These data blocks are as follows: (1), (2),..., (48).

(2) From definition 2: LND_0=(1),(2)...(12); LND_1=(13),(14)...(24); LND_2=(25),(26),...,(36); LND_3= (37),(38),...,(48)

(3) From definition 3:

(a) G_0^0=(1),(2),(3),(4), G_0^1=(5),(6),(7),(8), G_0^2=(9),(10),(11),(12);

(b) G_1^0=(13),(14),(15),(16), G_1^1=(17),(18),(19),(20), G_1^2=(21),(22),(23),(24);

(c) G_2^0=(25),(26),(27),(28), G_2^1=(29),(30),(31),(32), G_2^2=(33),(34),(35),(36);

(d) G_3^0 =(37),(38),(39),(40), G_3^1=(41),(42),(43),(44), G_3^2=(45),(46),(47),(48).

(4) From definition 4: (a)$\overline{N}_0^0$=N_1, $\overline{N}_0^1$=N_2, $\overline{N}_0^2$=N_3; (b)$\overline{N}_1^0$=N_0, $\overline{N}_1^1$=N_2, $\overline{N}_1^2$=N_3; (c)$\overline{N}_2^0$=N_0, $\overline{N}_2^1$=N_1, $\overline{N}_2^2$=N_3; (d)$\overline{N}_3^0$=N_0, $\overline{N}_3^1$=N_1, $\overline{N}_3^2$=N_2;

(5) From definition 5:(a) $G_0 \to \overline{N}_0^0 = G_0 \to N_1 = \{G_0^0, G_0^1\} \to N_1$, $G_0 \to \overline{N}_0^1 = G_0 \to N_2 = \{G_0^1, G_0^2\} \to N_2$, $G_0 \to \overline{N}_0^2 = G_0 \to N_3 = \{G_0^2, G_0^0\} \to N_3$; (b) $G_1 \to \overline{N}_1^0 = G_1 \to N_0 = \{G_1^0, G_1^1\} \to N_0$, $G_1 \to \overline{N}_1^1 = G_1 \to N_2 = \{G_1^1, G_1^2\} \to N_2$, $G_1 \to \overline{N}_1^2 = G_0 \to N_3 = \{G_1^2, G_1^0\} \to N_3$; (c)(d) Omitted

Solution B:

From definition 7: $SSUR = \frac{(1+P)}{k} = \frac{1+2}{4} = 0.75$. From the calculus above we can know each node stores 36 shares of data, and 4 nodes in all. So all the shares of data are 36*4, 36*4/(48*4)=0.75. the theoretical value equals to the actual value, that is, SSUR=0.75.

4 Parallel Scheduler

4.1 The Basic Idea for Scheduler

Definition 10. (Local Data Remainder: S_i) Let x_i represent the ideal data downloaded from node N_i, where $x_i = V_i/(\sum_{j=0}^{k-1} V_j)$. Let $S_i = x_i - LND_i$, it is called Local Data Remainder.

This indicator reflects the node load. If S_i tends to or equals to 0, then it shows that each node can roughly finish downloading at the same time. If S_i is positive, then it shows that the download speed of the current node is fast and it can share the load of other nodes whose S_i is negative. From definition 9 we can see that under ideal circumstances, there is $\sum S_i = 0$.

Examples (1) Given k=4, p=1, metasum=3, V_1=12, V_2=10, V_3=4, V_4=28. Scheduling objective: As far as possible to ensure that each node finishes downloading at the same time, and the data is non-repeated. We first give final results of scheduling algorithm as shown in table 1. Download the gray data blocks from node N_i. The ideal download data blocks from each node is (8, 6.7, 2.7, 18.7), the scheduling result is (8, 7, 3, 18). The load(S_i) of each node happens (-1, -2, -6, 10) both before (I-A) after scheduling , that is, (0, -0.3, -0.3, 0.7), the load is balanced more or less.

As the sum of data blocks is an integer, so the actual scheduling result is rounded off. Here, SSUR=50%, Compared with the full copy deployment, only 50% storage space is used.

Table 1 Example (1)

Node	*V*	S_i	*Ideal (I)*	*Actual (A)*	*I-A*	*LND data*	*OND data*
N_1	12	-1	8	8	0	(1)(2)(3)(4)(5)(6)(7)(8)(9)	(10)(11)(12)(19)(20)(21)(28)(29)(30)
N_2	10	-2	6.7	7	-0.3	(10)(11)(12)(13)(14)(15)(16)(17)(18)	(1)(2)(3)(22)(23)(24)(31)(32)(33)
N_3	4	-6	2.7	3	-0.3	(19)(20)(21)(22)(23)(24)(25)(26)(27)	(4)(5)(6)(13)(14)(15)(34)(35)(36)
N_4	28	10	18.7	18	0.7	(28)(29)(30)(31)(32)(33)(34)(35)(36)	(7)(8)(9)(16)(17)(18)(25)(26)(27)

From Table 1 we can see that for the fast nodes they not only complete LND download mission, but they also share simultaneously the download mission of the slow nodes from their OND. For example, the data blocks (25)(26)(27) lie in LND of node N_3, but they are downloaded from the OND of node N_4.

4.2 Description of Scheduler

Objective: Optimize the S_i of each node, make them close to or tend to 0, to make load balance for each node.

Explanation: For any two nodes ZN and FN, if the S_i of ZN is positive and there are remainder parts in LND of FN that are not shared by the OND of ZN, then ZN can share the load of FN. When a block is shared, do ZN.S_i–, FN.S_i++. If a data block is shared, then the flag which points out download position is marked, which contains the same data block stored at other nodes. So the same block at other nodes will not be operated.

Input: k, p, V_i , and the data distributed according to storage model.

Output: Program for how to download data blocks.

(1) Build list ZSi_list consisted with the nodes whose Si are positive, and sort ZSi_list in descending order.

(2) Build list FSi_list consisted with the nodes whose Si are negative, and sort ZSi_list in ascending order.

/* Deal with each node in FSi_list. */

(3) For every node FN in FSi_list deal from first

/* Let the nodes in ZSi_list share the load of nodes in FSi_list*/

(4) For every node ZN in ZSi_list deal from first

(5) IF FN.$S_i \geq 0$ THEN FSi_list.Move_to_Next, go to(3)

(6) IF ZN.$S_i > 0$ THEN

(a)Let the data blocks in OND of ZN share the load in LND of FN. According to the sharing sum, update ZN.Si. Share one block each time, and judge the

below two conditions timely: (b)IF FN.$S_i \geq 0$ THEN FSi_list.Move_to_Next, go to(3); (c) IF ZN.$S_i \leq 0$ THEN ZSi_list.Move_to_Next, go to(4).

(7) In stages (1)-(6) above, let the fast node share the load of slow node. If either list has reached the tail, then exit.

(8) If there are still some nodes(FN) whose Si are still negative in FSi_list, then deal with ZSi_list from first. If the intersection for OND(non-processed) of ZN and LND(non-processed) of FN is not null, then let ZN share the load of FN until the intersection is null, in this process regardless of whether the Si of ZN is already negative.

(9) There are two circumstances for the nodes in FSi_list, their Si is positive or non-positive. Let the nodes whose Si are positive share the load of negative ones until they can not share.

Now the nodes in ZSi_list exist in two cases, their Si is positive or non-positive. For every node in ZSi_list, repeat (1)-(6). Then output the scheduling sequence.

5 Parallel Tranfer Algorithm: PTABM

If the transmission speed is stable, then the scheduler can be used directly. As network transmission speed changes dynamically, a parallel transmission algorithm, which is based on scheduler and can meet the dynamic speed, is proposed in this chapter.

5.1 Model-based storage policy

Assuming there are four nodes. As shown in Figure 3, let the data meet sub-block model storage, that is, each data block meets Distributed storage Model separately. In fig.1 data groups $< B_i^1, B_i^2, B_i^3, B_i^4 >$ $(1 \leq i \leq 4)$ and $B_i = B_i^1 \cup B_i^2 \cup B_i^3 \cup B_i^4$ meet storage model separately(every group meets the model separately).

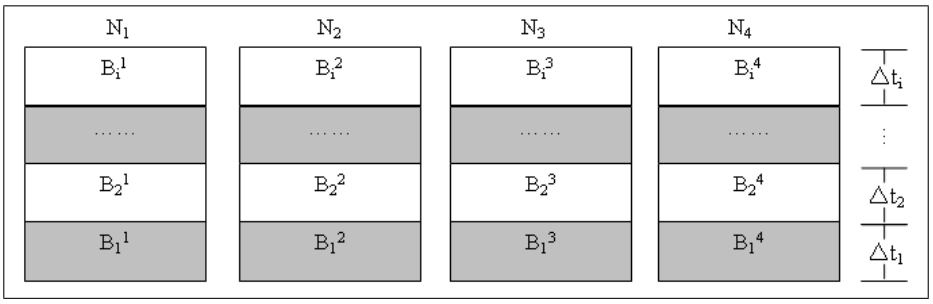


Figure 3: Storage policy sample

5.2 Parallel Transmission Algorithm Based on Model (PTABM)

In fig.1, as any B_i^j is only part of B_i, so if download B_i from four nodes in parallel, the following restriction must be met: (a) Uninterrupted download data from the four nodes at the same time; (b) Non-repeated download.

As the speed changes dynamicly, let Δt_i represent the time interval from the time of first finishing downloading B_{i-1} to the time of first finishing downloading B_i. For example, the time of completion of B_{i-1} downloaded by N_1 is t_{i-1}, the time of completion of B_i downloaded by N_3 is t_i, so the time interval $\Delta t_i = t_i - t_{i-1}$. As the speed changes dynamically, the size of each time interval may also be different.

The amount of B_i data (x_i) $(0 \leq i \leq k-1)$ that each node should commit is determined as follows:(a)Take the V_i at t_{i-1} $(0 \leq i \leq k-1)$ as the input speed of scheduler; (b)The remainder data that every node did not complete is m_i ; (c)Formula (2) must be met in theory, that is, as far as possible to ensure that each node fulfils its download mission at the same time.

$$\frac{m_0 + x_0}{V_0} = \frac{m_1 + x_1}{V_1} = ... = \frac{m_{k-1} + x_{k-1}}{V_{k-1}} \tag{4}$$

Formula (2) is changed into formula (3) through Identity Theorem. From formula (3), x_i is deduced as formula (4). Take x_i as the input value of scheduler to calculate S_i.

$$\left(\frac{m_i + x_i}{V_i}\right)_{i=0}^{k-1} = \frac{\sum_{i=0}^{k-1}(m_i + x_i)}{\sum_{i=0}^{k-1} V_i} = \frac{\sum_{i=0}^{k-1} m_i + \sum_{i=0}^{k-1} x_i}{\sum_{i=0}^{k-1} V_i} = \frac{\sum_{i=0}^{k-1} m_i + M}{\sum_{i=0}^{k-1} V_i} = \lambda \tag{5}$$

$$x_i = \lambda V_i - m_i \ (0 \le i \le k-1) \tag{6}$$

According to formulas and ristriction in this section, the schematic diagram of parallel transmission architecture is shown in Figure 4 The client side consists of two parts. At the end of every Δt Download Unit passes V_i and m_i to Scheduler Unit. Scheduler Unit calculates x_i and the data block B_i^j, and then output them to Download Unit. Download Unit downloads data from the nodes according to B_i^j, and real-time observe whether there is a new generation of Δt, and then repeat the process above.

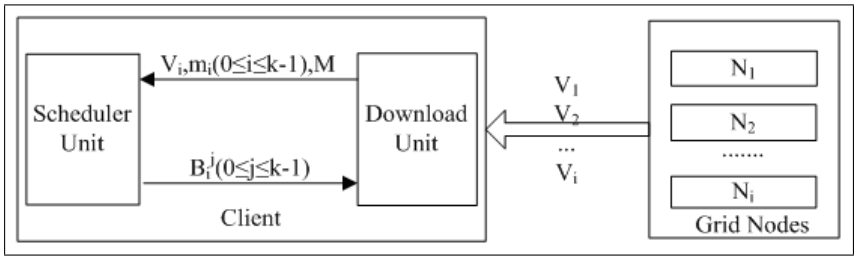


Figure 4: Schematic diagram of parallel transmission

5.3 Reliability analysis

The reliability of algorithm includes two aspects, which are scheduler reliability and algorithm reliability.

(1) From theorem 1, if there are random p nodes fail, then the data ,the other k-p nodes contain, is integrated. (2) If there are nodes fail during the download period, then (a) get V_i of current normal nodes; (b) Let the speed of failure nodes be 0; (c) Mark the downloaded blocks of B_i, and the undownloaded blocks are $B_i^{'}$; (d) Take the amount of data(M') of $\sum B_i^{'}$, V_i and m_i as the input values of scheduler; (e)Use Scheduler to recompute the scheduling sequences.

5.4 Verification System (VPG-Torrent)

Verification System contains four parts, as shown in Figure 5: Data Dispatcher, RLS server, virtual node (Vpeer), Peer node.

Data Dispatcher: Data Dispatcher take p, k, list of servers and original data as input, through model computing unit, original data was divided and deployed into servers. Meanwhile, the storage information of data blocks(M-table) was sent to RLS server.

RLS Server: the Replica Location Service (RLS) keeps track of the physical locations of files. The information stored in the Replica Location Service defines the number of replicas for each file as well as the physical location of the actual data. The physical location is identified by a unique physical file name (PFN) such as a GridFTP URL. To enable the use of the GridTorrent protocol we introduce the GridTorrent URL.

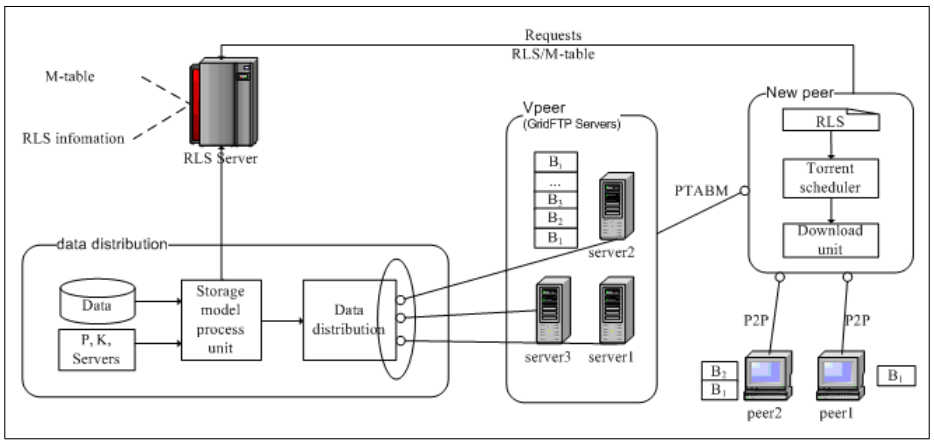


Figure 5: Data dispatcher and download

Virtual peer (Vpeer): In the process of data download, GridFTP servers are regarded as a virtual peer node (Vpeer). So Vpeer has a fast transmission speed. The other peers regard Vpeer as a general peer to schedule. Of course, the transmission of Vpeer is based on PTABM.

Peer node: Each peer contains scheduling algorithm of BitTorrent and a download unit. Scheduling algorithm treat Vpeer as a normal peer. Download unit responsible for data transmission from other peers or Vpeer.

Each new peer must first request access to RLS server to get RLS and M-talbe information, and upload the information of data blocks they owned to RLS server periodically. As Vpeer has fast speed, so when the number of peers is small, its contribution to transmission will be greatest. When download from Vpeer PTABM must be used. Verification system consists of two major experiments: Real experiment 5-3 is to verify PTABM, and compare with previous algorithm. Simulation experiments 5-4, 5-5 are to verify the performance of large-scale concurrent access, so experiment is carried out in oversim.

6 Experiments

6.1 Purpose

From Figure 5, (a)if there are no peers willing to upload, then VPG-Torrent will degenerate into the most common parallel transmission system based on GridFTP and multi-copy, this transmission mode plays an irreplaceable role in some applications in data grid. Therefore, the performance between VPG-Torrent and traditional DADC must compared; (b) For large-scale network where BitTorrent plays an important role, the performance of VPG-Torrent must be verified as the number of concurrent peers increase from small to large, and should be compared with GridTorrent.

There is a natural experiment. If storage model is not used, but the data is directly divided into uniform shares, and then stored them into multiple nodes, so the bandwidth is also increased. If so, Vpeer is not exist, accordingly PTABM can not be used too, but only BitTorrent can be used. As bittorrent algorithms do not rely on global data storage knowledge to schedule, so it leads to the service capacity of servers can not be maximized. This experiment is carried out in 5-4(2).

Simulation experiments (1) (2) (3) are made to verify the performance of scheduler; real experiments (4) (5) are made to verify the performance of Parallel Transmission Algorithm based on Storage Model & full copies and compare with the performance, experiment (6) (7) are

made to verify the performance VPG-Torrent. Experiment 5.5 is made to verify the reliability of sheduler and algorithm. In experiment (4) (5) the client host lies in education network, and the other four grid nodes are all telecommunication network access. GridFTP is used as transmission protocol.

In experiment the concurrent connections to Vpeer is limited to 20. The maximum connection speed of every server in Vpeer is 200k, and rated bandwidth is 5M. Each peer can start up 15 connections at the same time, the upload rate is up to 20kb.

6.2 Basic Definitions and Explanations

If the speed is determined, the ideal amount of data downloaded from each node can be: $M * V_i/(\sum_{j=0}^{k-1} V_j)$ $(0 \leq i \leq k-1)$. The result that scheduler can give and the ideal amount of download data are compared during experiments (1) (2) (3).

Literature [11] combines basic parallel transmission mode of GridFTP and P2P technologies, but the prerequisite is that there are multiple clients requesting; the research of parallel transmission algorithm in literature [9] [10] is based on GridFTP only, its dynamic collaborative algorithm (DCDA) achieved good performance. The performances of PTABM and DCDA are compared in experiments (4) (5).

Definition 11. (The percentage of time that scheduling results exceeds ideal value:TRER) Given the ideal download time is T_{id}, the download time outputted by scheduler is T_{re}. Let $TRER = (100(T_{re} - T_{id})/T_i)\%$. This parameter is used to detect the performance of scheduler.

Definition 12. (Unit Speed Variance:D(V)) Variance divided by the sum of the speed is called Unit Speed Variance, that is:

$$E(V) = \left(\sum_{i=0}^{k-1} V_i\right)/k \; and \; D(V) = \left(\sum_{i=0}^{k-1} (V_i - E(V))^2\right) / \left(k\left(\sum_{i=0}^{k-1} V_i\right)\right) \tag{7}$$

From definition 9 we can see that when k is fixed and for the same group V_i, V_{top} increases by $\left(\sum_{i=0}^{k-1} V_i\right)/k$ when p increases by 1. From definition 7 we can see that the Storage Space Usage Rate increases by 1/k.

6.3 Scheduler Performance Test

(1) This experiment is to verify the impact of D(V), V_{top}, V_{ratio} on TRER. Let p=1, k=4, metasum=15, the range for the speed distribution is (1,100), then the theoratical V_{ratio}=90. Consecutively record the results of algorithm for 20 times, the impact of D (V), V_{ratio}, and V_{top} on TRER is listed in Table 2.

Calculate the average of D (V), that is, ED(V). Classified the observational results that meet D(V)>ED(V) into a group, and D(V)<ED(V) as another group. In table 2 the experiments numbered by 1-9 are of D (V)>ED (V). TRER is 4.40.

The experiments 1-9 and 10-20 calculate the average TRER respectively, the value are 7.40, 1.87 and they are greater or less than the overall average of 4.40.

Experiment (1) shows that if the distribution of node speed is very uneven and the difference between maximum and minimum speed is large, TRER may be increased. If Vtop exceeds the theoratical maximum value, then TRER will have a substantial increase.

(2) This experiment is made to verify the impact of p and speed range on TRER.

Let k=10, and change the range of p & speed, as shown in Table 3. There are nine combinations, observe ten times for every combination. Calculate the TRER average of every combination, the histogram is shown in figure 6.

Let k=2,p=2, observe the impact of speed range on TRER. Observing ten times of algorithm results in every speed range, and calculates TRER average. The curve is shown in Figure 7.

Table 2 Impact of D(V), V_{ratio}, V_{top} on *TRER*

No.	*>D(V) Average?*	*D(V)*	*V.*	*Actual* $V_{...}$	*Theoretical* V_{xy}	*Actual* V_{xy}	*TRER*	**TRER Average**
1	Y	5.34	37,10,88,93	9.30	114.00	93.00	2.70	
2	**Y**	**4.47**	**12,23,55, 1**	**55.00**	**45.50**	**55.00**	**18.03**	
3	Y	3.07	85,18,89,76	4.94	134.00	89.00	0.00	
4	Y	4.48	1,77,67,75	77.00	110.00	77.00	0.00	
5	Y	4.48	95, 9,72,49	10.56	112.50	95.00	3.22	**7.40**
6	Y	4.59	7,85,57,91	13.00	120.00	91.00	5.33	**(>4.40)**
7	Y	3.92	55,95,70,12	7.92	116.00	95.00	6.33	
8	**Y**	**5.89**	**12,24,76,10**	**7.60**	**61.00**	**76.00**	**29.44**	
9	Y	4.00	67,34, 1,34	67.00	68.00	67.00	2.37	
10	N	2.16	58,66,22,83	3.77	114.50	83.00	2.16	
11	N	0.46	36,28,41,21	1.95	63.00	41.00	2.50	
12	N	2.03	22,26,50,67	3.05	82.50	67.00	2.24	
13	N	1.99	33,94,89,73	2.85	144.50	94.00	2.48	
14	N	1.09	35,31,45,67	2.16	89.00	67.00	1.71	
15	N	0.75	19,10,32,24	3.20	42.50	32.00	0.00	**1.87**
16	N	2.35	19,13,46,55	4.23	66.50	55.00	5.00	**(<4.40)**
17	N	2.47	64,22,66,90	4.09	121.00	90.00	0.00	
18	N	2.62	70,68,44,14	5.00	98.00	70.00	1.46	
19	N	1.94	85,29,46,64	2.93	112.00	85.00	2.99	
20	N	1.82	51,19,16,45	3.19	65.50	51.00	0.00	
Average		**3.00**					**4.40**	

Table 3 Experimental Methods

P \ V_i	5-25	5-50	5-75
1	10 Times	10 Times	10 Times
2	10 Times	10 Times	10 Times
3	10 Times	10 Times	10 Times

From Fig.6 we can see that when p=1, the change of speed has greater impact on TRER. But when p=2,3 as the speed range increases, TRER are almost unaffected. Scheduling results are approximately equivalent to the ideal download speed.

It can be seen from Figure 7, TRER gradually increases as the speed range increases. However, when the speed range is in 5-150 and TRER are in the vicinity of one, the effect is good.

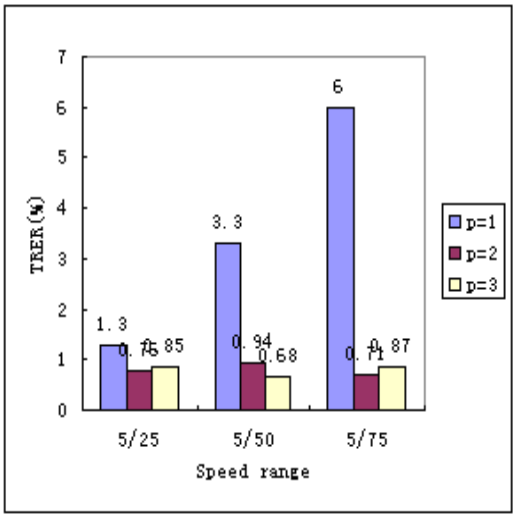


Figure 6: mpact of p and speed range on *TRER* when k=10

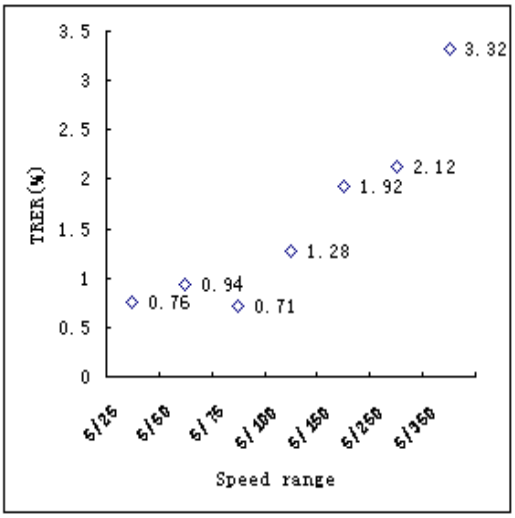


Figure 7: Impact of speed range on *TRER* when k=10, p=2

Experiment (2) shows that in a random speed range, if p has already guaranteed that TRER is maintained at an ideal value. Even when p continues to increase, TRER will not decrease.

(3) This experiment is done to verify the impact of k and speed range on TRER.

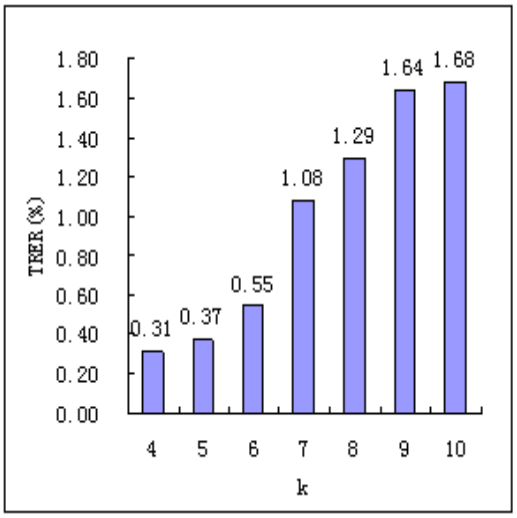


Figure 8: Impact of k on *TRER* when P=2, $V_i \in(1,50)$

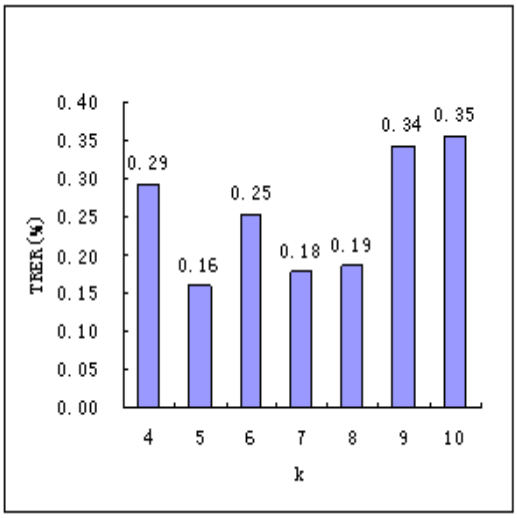


Figure 9: Impact of k on *TRER* when P=2,$V_i \in(5,50)$

Let the amount of data at every node is 300. In order to ensure the metadata will not be affected when k and the overall amount of data change, do as follows: When k=4 and metasum=100, from formula (1) metasum of other nodes can be calculated, as shown in table 4.

Because the sum of nodes increases, if the amount of data doesn't increase, then the download time will decrease, so the TRER will become inaccurate. In order to guarantee experimental parameters consistency, a special treatment is needed.

Table 4 Value of k, M and metasum

K	M	metasum
4	1200	100
5	1500	75
6	1800	60
7	2100	50
8	2400	43
9	2700	38
10	3000	33

Let p=2, $V_i \in(1,50)$, observe the effect of changes of k on TRER. For every k, observe 10 times, then calculate the average of TRER. From Figure 8 we can see that TRER increases gradually as k increases.

Let p=2, $V_i \in(1,50)$, observe the effect of the change of k on TRER. For every k, observe 10 times, then calculate the average of TRER. From Figure 9 we can see that as k increases TRER changes little.

When $V_i \in(1,50)$, the difference among the speeds of nodes is larger, and the maximum theoretical V_{top}=50; while $V_i \in(5,50)$, there is little difference among the speeds of nodes, and the maximum theoretical V_{top}=10 only.

Experiment (3) shows that the larger speed range affects TRER greatly. If the speed range is not large, then k has little effect on TRER.

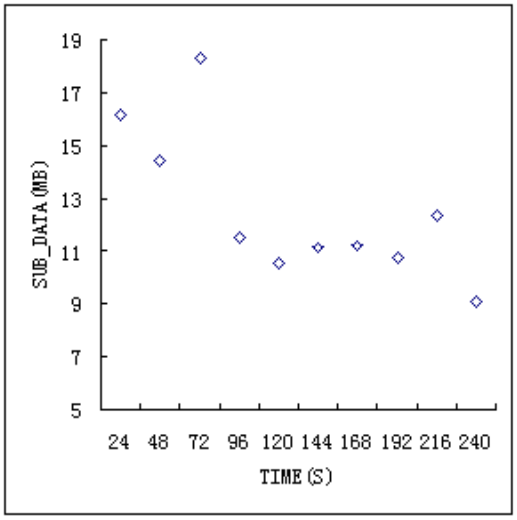


Figure 10: Distribution of SUB_DATA points when matadata=5M)

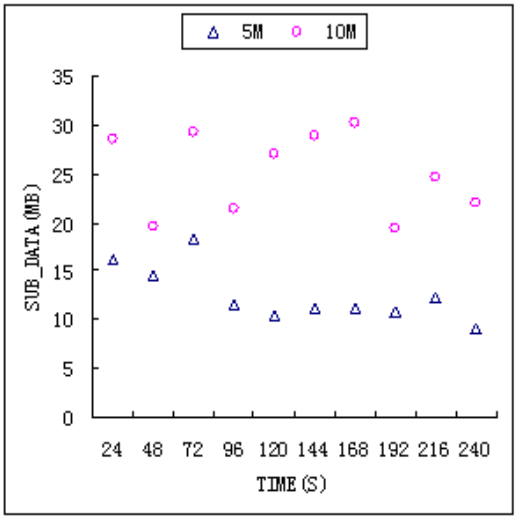


Figure 11: Distribution of SUB_DATA points when matadata=5M,10M

It is found through experiments that the value of TRER near one is caused by the sub-block. The ideal download data of each node is based on an arbitrary partition for the overall data, and the error generated by the scheduler can be reduced by increasing the metasum appropriately. Therefore for the value of TRER near one we can consider that the scheduler has achieved the ideal scheduling result.

6.4 PTABM Performance Test

(4) If there are no peers willing to upload, then VPG-Torrent will degenerate into the most common parallel transmission system based on GridFTP and multi-copy. This experiment is made to compare the transmission speed between PTABM and DCDA, as well as compare the amount of download data at the same time of the two algorithms. Let metadata be 5M and 10M, analyze the performance of PTABM.

Let matadata=5M, observe the amount of data downloaded by DCDA and PTABM separately ten times at different time. Let SUB_DATA equal the amount of download data by SUB_DATA minus PTABM's. The point distribution is shown in Figure 10.

Let matadata=5M,10M, observe the amount of data that downloaded by DCDA, PTABM separately ten times at different time. Let SUB_DATA equal the amount of data downloaded by DCDA minus PTABM's. Two type point distribution is shown in Figure 11.

Figure 10 in experiment (4) shows that SUB_DATA does not increase as the time increases, and shows the oscillation effects. It shows that SUB_DATA is non-cumulative. So as the amount of download data increases, the ratio of SUB_DATA and overall data tends to zero, so the performance of PTABM tends to DCDA. Fig.8. shows that when matadata is larger, SUB_DATA is also larger. But as time increases, it also shows the oscillation effects.

(5) This experiment is made to compare the performance between PTABM and DCDA in download time and amount.

Let one time unit contain 10 seconds and matadata=15M. Observe the amount of download data and calculate the ratio of data downloaded by DCDA and PTABM, take the percentage (DR). The curve for the relationship between DR and time units is shown in Figure 12.

Let metadata=5M,10M,20M, p=1. Separately observe the time that taken by DCDA and PTABM to download 960M and 1920M data. Then calculate the ratio of PTABM finishing time and DCDA's, take the percentage (TR). The TR bar graph is shown in Figure 113.

Figure 12 in experiment 5 shows that as the download time increases the speed of PTABM tends to DCDA. At 40 50s their speeds are similar and are approximately equivalent at 100s. From Figure 13 we can see that smaller matadata can achieve better effects than larger one. When matadata=5 transfer 1920M data the ratio of the time used by PTABM and DCDA is 1.05. So for vast data the two algorithms can achieve similar performances.

6.5 VPG-Torrent Verification

(6) This experiment is made to verify performance of VGP-Torrent when p=1, P2P is allowed or not allowed (download from GridFTP servers), and compare performance with GridTorrent.

Let original data size be 1G, and the amount of data permitted to be deployed is 2G. Within 20 minutes specific amount of peers are generated randomly. Each peer will continue to permit to be downloaded by other peers for 40 minutes from its completion.

Firstly, deploy two copies for GridTorrent system; secondly, given p=1, k=5,6,7 as the input of data dispatcher of VPG-Torrent, so the information of data storage is stored into RLS server later.

When P2P is allowed, from Figure 14, in the performance curves of different peers, the curve GT of GridTorrent is located in the bottom. When k=5,6,7, the curves of VPG-Torrent increased gradually. And the curves are all concave. It is because that when the number of peers is small, parallel transmission only based on GridFTP servers play an important role, as the number increases gradually, then BitTorrent plays an important role.

When P2P is not allowed, i.e., the transmission is only based on GridfTP. From Figure 15, VPG-Torrent has a great advantage over BitTorrent in average speed of peers. Most importantly, the storage space used by them is the same.

(7) This experiment is made to test how impact of storage method on GridTorrent and compare performance with VPG-Torrent.

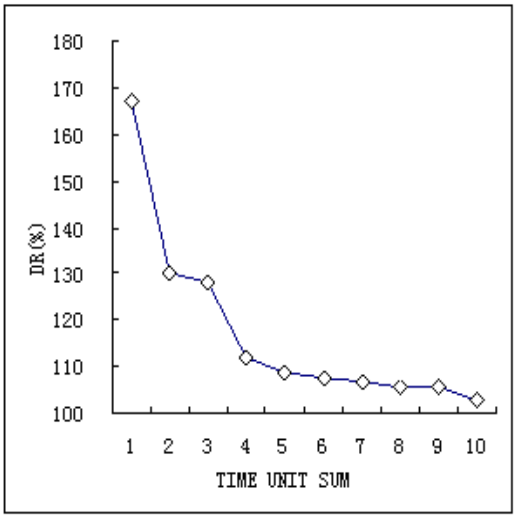


Figure 12: Download data ratio at the same time when matadata=15M

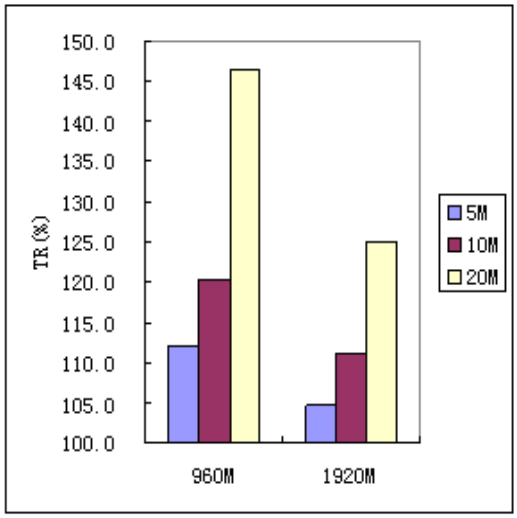


Figure 13: Impact of matadata on download speed

For GridTorrent we process the original data as follows: let original data size be 1G, and the amount of data permitted to be deployed is 2G. The original data were evenly divided into 3 shares, and are distributed to 6 nodes, so that there are two nodes have the same data. These six nodes will treated as normal peers for any new peers. Experiments were carried out in two conditions: P2P is allowed, and not allowed. When P2P is not allowed, there will be six threads downloading data concurrently, and the completion time depends on the node which complete transmission task at last.

In Figure 16, k=6 is the performance curve of VPG-Torrent, AS is the curve of GridTorrent when storage method is adopted, and GT is the curve of GridTorrent obtained in experiment 6-5-(6). Obviously AS is better than GT. So for BitTorrent to increase performance, the better way is to increase the bandwidth by deploying original data to multiple nodes under the condition of using the same size of storage space. However, the performance of AS is a little worse than VPG-Torrent's. It is because that BitTorrent will not utilize the storage knowledge to schedule, accordingly it can not make full use of server bandwidth.

In Figure 17, when P2P is not allowed, 'M' is the column of VPG-Torrent, and 'A' is the column of GridTorrent. Obviously the transmission time of VPG-Torrent is less than GridTorrent's. In this experiment VPG-Torrent can ensure that all the download tasks can be finished almost at the same time, therefore, it can make the load balance. Instead, BitTorrent can not. Although the storage space used is the same, VPG-Torrent exceeds BitTorrent in transmission time.

6.6 Reliability Test

This experiment is made to verify the performance of scheduler at normal conditions and when node failed, verify the performance of PTABM when one node failed, and compare the performance of PTABM and DCDA at this condition.

Let p=(1,2), V_i ∈(5,25), k=(4,5,6,7), observe the performance of scheduler in normal and failure circumstances. In Figure 18 'F=1' represents that one node is failed. Draw the curve of k and TRER . From Figure 18 we can see that when p=1, there is one failure node and the performance of scheduler is a little worse than in normal circumstances. While p=2, there is also one failure node, but the performance of scheduler is good, with TRER kept near 1.

Let p=1, V_i ∈(5,25), metasum=5, observe the performance of PTABM and DCDA in normal

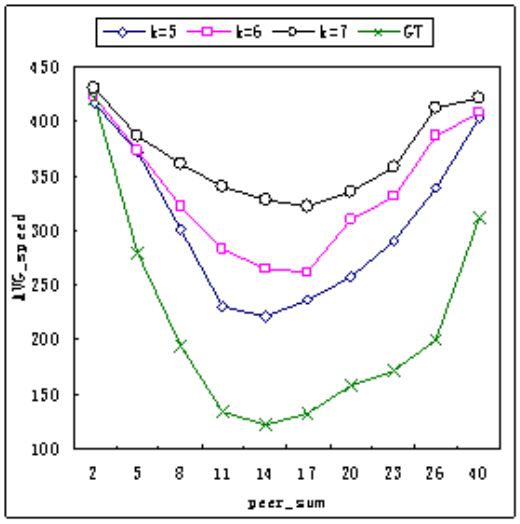


Figure 14: Impact of peers on average speed when BitTorrent is allowed

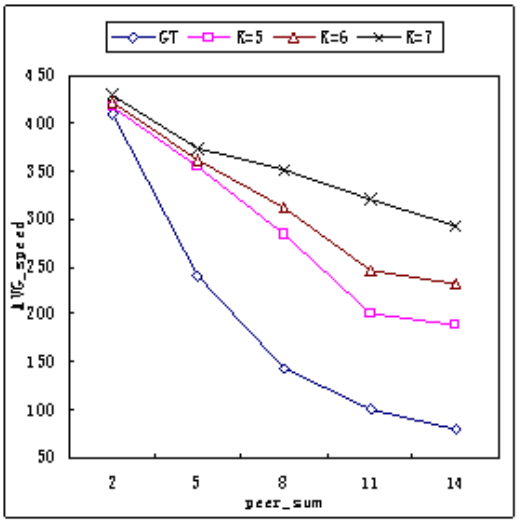


Figure 15: Impact of peers on average speed when BitTorrent is not allowed

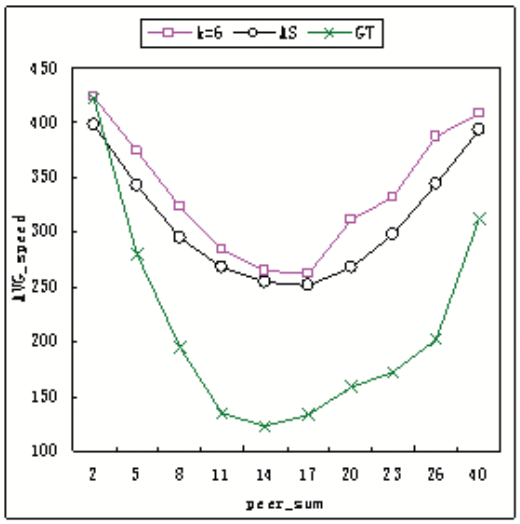


Figure 16: Performance compares for different storage methods when P2P is allowed

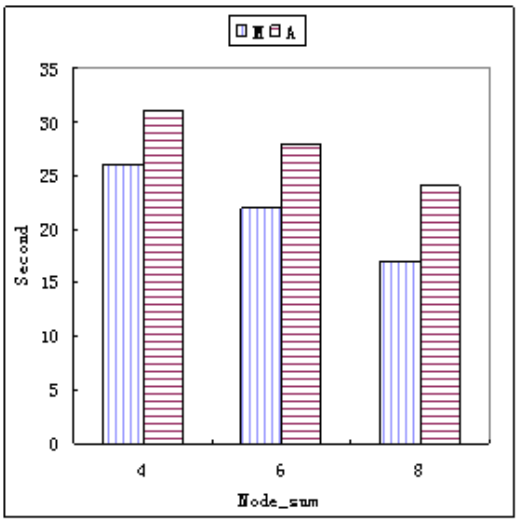


Figure 17: Performance compares for different storage methods when P2P is not allowed

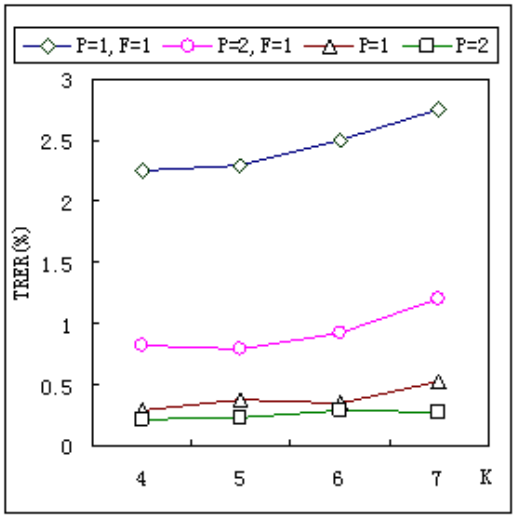


Figure 18: Comparison of scheduler performance between normal and failure curcumstances

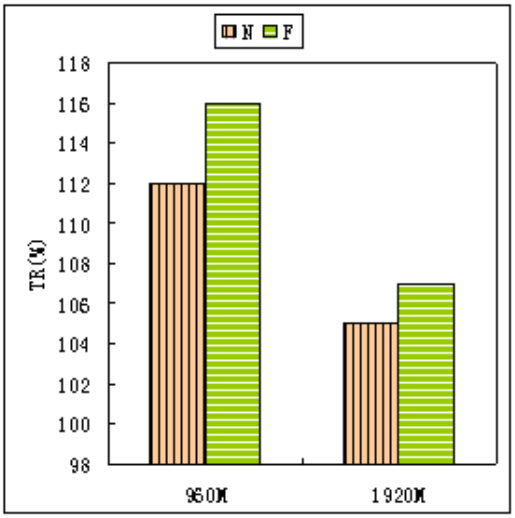


Figure 19: Comparison of algorithm performance between normal and failure curcumstances

and failure circumstances, make the ratio(TR) of PTABM and DCDA. In Figure 19 'N' represents normal circumstance while 'F' represents failure circumstance. From the fig we can see when the amount of data is 1920M the download time of the two algorithms is similar. It shows that PTABM has good reliability.

6.7 Supplementary Analysis of Experimental Results

The large speed range in experiment 6.3 is to verify the performance of scheduler in the extreme distribution of speed. So in practice as long as taking appropriate p value according to the speed range, the smaller TRER can be achieved. At the same time, given a reasonable matadata value, the performance of PTABM can be close to or approximately equal to the performance of DCDA. The most important is saving storage space and reducing the network traffic caused by the creation of multi-copies. Fortunately, the Distributed storage Model and PTABM meet those requirements.

In experiment 6.4 p=1, matadata=5M, the effects are very good. Derived from definition 7: when p=1, the SSUR and k are calculated as in table 5. When k=4, the SSUR is 50%; but when k=10, the SSUR is only 20%. As achieved, the objectives of rapid data transmission, the storage space and network traffic are also significantly optimized at the same time.

Table 5 Relationship between p=2, k and SSUR

k	*4*	*5*	*6*	*7*	*8*	*9*	**10**
SSUR	0.5	0.4	0.33	0.29	0.25	0.22	0.2

From experiment 6.5, whether P2P is allowed or not, the performance of VPG-Torrent is higher than GridTorrent's, even if increasing the bandwidth by adopting uniform storage method.

7 Conclusions

Parallel transmission algorithms based on multi-copy cause a waste of storage space seriously and are difficult to adapt to a wide range of network access. Instead, BitTorrent for wide range of file-sharing has a unique advantage. Although GridTorrent combined the two above,

the performance is poor when the peers are few. In this paper we achieved multiple optimization objectives: storage space saving, suitable for two kinds of application modes (i.e. parallel transfer based on GridFTP and BitTorrent), adaptability for wide range of network and higher performance when there are fewer peers. The proposed storage model, scheduler, parallel transmission algorithm, virtual peer and VPG-Torrent verification system achieved very good result, and the system proposed has certain advantages compared with the previous algorithms.

Bibliography

[1] CHEN Lei, LI San-li. A Calking Dynamic Replication Distribution Algorithm in Data Grid. *ACTA ELECTRONICA SINICA*, 34(11):1-4, 2006

[2] XIE Xiao-lan, LIU Yu, ZHOU De-jian. Research on Manufacturing Grid Data Access and Integration Key Technology. *JOURNAL OF WUHAN UNIVERSITY OF TECHNOLOGY*, 31(6):1-4, 2009

[3] ZHANG Guangzhi, HE Jieyue. Application Research on Biological Data Grid. *Computer Engineering*,(2):1-4, 2004

[4] QIN Xin, LUO Ze, NAN Kai etal. Design and Implementation of Problem Solving Environment for Astronomy Application Based on Science Data Grid. *Application Research of Computers*,(4):1-4, 2009

[5] H.A. James, K.A. Hawick. Scientific Data Management in a Grid Environment. *Journal of Grid Computing*,3: 39-51, 2005

[6] Mingwei Wang, Shusheng Zhang, Jingtao Zhou etal. An Architecture of Semantic Desktop Data Grid. *Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design*,IEEE Computer Society ,1-6, 2006

[7] S. Fiore, M. Mirto, Cafaro. A GRelC based Data Grid Management Environment. *21st IEEE International Symposium on Computer-Based Medical Systems*, IEEE Computer Society, 355-360,2008

[8] Richard McClatchey, Ashiq Anjum etal. Data Intensive and Network Aware (DIANA) Grid Scheduling. *Journal of Grid Computing*,5:43-64, 2007

[9] H. Liu, et al., Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, *Future Generation Computer Systems*:1-8,2009

[10] Xiangang Zhao, Bai Wang, Nan Du. Qos-based Algorithm for Job Allocation and Scheduling in Data Grid. *Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops (GCCW'06)*, IEEE Computer Society:1-7,2006

[11] Nhan Nguyen Dang, Soonwook Hwang, Sang Boem Lim. Improvement of Data Grid's Performance by Combining Job Scheduling with Dynamic Replication Strategy. *The Sixth International Conference on Grid and Cooperative Computing(GCC 2007)*, IEEE Computer Society:1-8,2007

[12] Esther Pacitti. Patrick Valduriez. Marta Mattoso. Grid Data Management: Open Problems and New Issues. *Journal of Grid Computing*,5:273-281, 2007

[13] Jiang Jianjin, Yang Guangwen. Replication Strategies in Data Grid Systems with Clustered Demands. *JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT*,46(2):1-8,2009

[14] W u Chang-ze, Chen Shu-yu, Ti an Dong. The strategy of creating replica based on cost shared in data grid. *Huazhong Univ. of Sci. & Tech. (Nature Science Edition)*,35(2):1-4, 2007

[15] Pangfeng Liu. Jan-Jan Wu, Optimal Replica Placement Strategy for Hierarchical Data Grid Systems. *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*:IEEE Computer Society: 1-4, 2006

[16] Tim Ho, David Abramson. A Unified Data Grid Replication Framework. *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*: IEEE Computer Society: 1-8, 2006

[17] Ingmar Baumgart, Bernhard Heep, Stephan Krause, OverSim: A scalable and flexible overlay framework for simulation and real network applications, *Proceedings of the 9th International Conference on Peer-to-Peer Computing (IEEE P2P'09)*, pp. 87-88, Seattle, WA, USA, Sep 2009

[18] Ingmar Baumgart, Bernhard Heep, Stephan Krause, OverSim: A Flexible Overlay Network Simulation Framework, *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007*, p. 79-84, Anchorage, AK, USA, May 2007

[19] R.S.Bhuvaneswaran, Yoshiaki Katayama, Naohisa Takahashi. Dynamic Co-allocation Scheme for Parallel Data Transmission in Grid Environment. *Proceedings of the First International Conference on Semantics, Knowledge, and Grid, IEEE Computer Society*: 1-6, 2006

[20] Sudharshan, Vazhkudai. Distributed Downloads of Bulk, Replicated Grid Data. *Journal of Grid Computing*,2:31-42, 2005

[21] Gaurav Khanna, Umit Catalyurek, Tahsin Kurc, et al. A Dynamic Scheduling Approach for Coordinated Wide-Area Data Transfers using GridFTP. *The 22nd International Parallel and Distributed Processing Symposium (IPDPS '08)*. IEEE Computer Society, 2008,1-12

[22] Liu Dongmei, Liu Dongmei. Multi-path parallel transmission scheme for optical grid systems. *Chinese High Technology Letters*,5:1-4,2008

[23] A. Zissimos, K. Doka, A. Chazapis and N. Koziris. GridTorrent: Optimizing data transfers in the Grid with collaborative sharing. in *Proceedings of the 11th Panhellenic Conference on Informatics (PCI2007)*, Patras, Greece, May 2007:1-12

[24] Athanasia Asiki, Katerina Doka, Ioannis Konstantinou, et al. A Distributed Architecture for Multi-Dimensional Indexing and Data Retrieval in Grid Environments. *In Proceedings of the Cracow 2007 Grid Workshop (CGW'07)*, Krakow, Polland, October 16-17, 2007:1-8

[25] A. Kaplan, G.C. Fox and G. von Laszewski, GridTorrent Framework: A High-performance Data Transfer and Data Sharing Framework for Scientific Computing. *Proc Grid Computing Environments, Supercomputing Workshops*, Reno, NV, USA, November 2007:1-10

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 701-712

Application Plugins for Distributed Simulations on the Grid

I.L. Muntean, A.I. Badiu

Ioan Lucian Muntean
Technical University of Cluj-Napoca
Department of Computer Science
Romania, 400027 Cluj-Napoca, 28 G. Baritiu
E-mail: ioan.lucian.muntean@cs.utcluj.ro

Alexandra Ioana Badiu
Technical University of Cluj-Napoca
Data Communication Center "Pusztai Kalman"
Romania, 400027 Cluj-Napoca, 28 G. Baritiu
E-mail: badiu@net.utcluj.ro

Abstract: Computing grids are today still underexploited by scientific computing communities. The main reasons for this are, on the one hand, the complexity and variety of tools and services existent in the grid middleware ecosystem, and, on the other hand, the complexity of the development of applications capable to exploit the grids. We address in this work the challenge of developing grid applications that keep pace with the rapid evolution of grid middleware. For that, we propose an approach based on plugins for grid applications that encapsulate a set of commonly used type of grid operations. We further propose more complex high-level functionalities, such as the plugins for remote exploration of simulation scenarios and for monitoring of the behavior of end-user applications in grids. We provide an example of a grid application constructed with these software components and evaluate based on it the performance of our approach in the context of the simulation of biological neurons. The results obtained on test and production grids demonstrate the usefulness of the proposed plugins, with a small performance overhead compared to traditional grid tools.
Keywords: Grid computing, application plugins, simulation services, GridSFEA.

1 Introduction

Simulation is acknowledged today as the third pillar of modern sciences, in addition to theory and experiments. Computational sciences are driven by simulation and provide the widest majority of applications for high-performance computing resources available in grid environments. Today, user communities of grid computing, having scientific computing background, are bounded to a large extent to a middleware traditionally employed within the respective community. Such examples are the communities formed around the projects: TeraGrid, supported by the Globus Toolkit [12], Enabling Grids for e-Science (EGEE/EGEE2) supported by the g-Lite middleware [8], or NorduGrid, where the Advanced Resource Connector [3] has emerged from. A special place in this sense is taken by the user community formed around the project Distributed European Infrastructure for Supercomputing Applications (DEISA/DEISA-2). In DEISA, the user has typically the choice between the Globus Toolkit and UNICORE [5].

The co-existence of these middleware imposes big challenges to the end-users of grids. Basically, these middleware are rather incompatible in many aspects (security, job and resource

management, etc). Therefore, users have to accommodate with different tools, languages, and operating procedures. This makes the adoption of a middleware other than the one used within his community a challenging task for the computational scientist. A great effort, supervised by the group Grid Interoperability Now of the Open Grid Forum (OGF-GIN) [20] is currently invested by the international grid community in the interoperability of these various middleware. Such examples in Europe are international consortia such as the European Grid Infrastructure (EGI) [11] or the Initiative for Globus in Europe (IGE) [12] aim at providing grid middleware components that are capable of operating in an unified grid middleware (UMD [9]). In this context, the development of grid applications that keep pace with the continuous evolution of the different middleware still remains a big challenge that complements the approach to the integration of the middleware.

In this paper, we address this latter challenge: the development of grid applications. For that, we employ the Grid Simulation Framework for Engineering Applications (GridSFEA) [15], a software package for the gridification of numerical simulation programs. Our approach is to offer out-of-the-box high-level functionalities, packed as plugins, commonly needed in such applications. The plugin mechanism available with the Eclipse Platform has been successfully used in our work. GridSFEA plugins cover for example job submission, job monitoring, file transfer, simulation explorer etc. Interactions with different middleware are possible due to low-level plugins that pack client-side functionality of these middleware. The major benefit of our work is that new applications requiring grid interactions or extensions to existing software packages can rapidly be developed without any grid-knowledge, by solely incorporating these plugins. Furthermore, end-users get a unified way of working with the grid, regardless the underlying middleware in place. In [16] we proposed a first version of the plugin-based architecture of client-side tools of GridSFEA. The evaluated prototype included only one high-level plugin, the parameter generator functionality. In the current work, we further refine the proposed architecture with additional low- and high-level plugins. The evaluation is carried out on development and production grids.

The remaining of this paper is organized as follows: next section outlines the related work. In Sec. 3, the GridSFEA framework is introduced. Eclipse plugins are briefly introduced in Sec.4. Section 5 presents the plugins for grid applications available with GridSFEA. New high-level plugins are discussed in Sec. 6. The results of this work in the context of a scenario from computational neuroscience are the topic of Sec. 7, whereas Sec. 8 outlines our conclusions.

2 Related Work

For the development of grid applications, middleware specific client-side libraries and toolkits are available in the grid community. Such examples are CoG Kit [23] for Globus Toolkit [2], GLUE [4] and CREAM [1] for g-Lite. The use of these libraries requires a thorough understanding of many technical and architectural details of each middleware stack. A great simplification in this sense is brought by the Grid Application Toolkit (GAT) [2] and its successor Simple API for Grid Application (SAGA) [14]. They provide a common programming interface to different grid operations, hiding the implementation details of each middleware. The grid know-how required for the application developer is rather low. Thus, for low-level grid operations, GAT and SAGA are a very good option. Nevertheless, they are not meant to provide high-level functionalities. Low level plugins of GridSFEA are based on community libraries such as GAT or CoG Kit. In addition to these, the framework comes with ready-to-use out-of-the-box application components providing high-level functionalities. Thus, both the application developer and the end-user of the grid need almost no knowledge about the grid.

More evolved tools for the development of grid applications are the grid frameworks such as g-Eclipse [14], Gridbus [22] , WS-GAF [19] or GFac [13]. These programs come with their

own strengths and weaknesses. Gridbus has evolved more in the direction of grid resource broker and cloud, having a rich support for parameter sweep use cases. Composition of scientific applications from Web services is the main feature of WS-GAF and GFac, being more a server-side development. A tool that focuses on the end-users is g-Eclipse. It allows operations on more than one middleware (e.g. g-Lite, GRIA [21]), and, to some extent, the development of new grid applications. The enabling technology of g-Eclipse is the plugin mechanism of the Eclipse Platform. GridSFEA considers both server-side and client-side developments. In the approach proposed in this paper, the plugins of GridSFEA are designed to be interoperable with the ones of g-Eclipse and to complement them with high-level functionalities such as remote exploration of simulation scenarios. Furthermore, the GridSFEA framework comes with its own support for simulation migration on grid computing environments.

The adoption of the grid standards is essential in the realization of grid developments. With this respect, we mention here the JSDL [18] specification of the Open Grid Forum. The end-user requests for jobs and resources can be specified independently of the grid middleware on which the jobs will run. Client tools become, to a large extent, responsible for translating the requests to the specifics of the target middleware. At the moment of writing, only a part of the JSDL specification is implemented by GridSFEA. The JSDL support complements the existing full support to GT4 resource specification language available with GridSFEA.

3 An Overview of the GridSFEA Framework

GridSFEA is a framework that enables computational scientists and engineers to gridify numerical simulation programs and to run them comfortably on the grid. The framework handles simulation scenarios on the grid and assists the user with typical tasks such as formulation of a computing job, submission of the job to the grid, retrieval of resulted simulation data, investigation of simulation results remotely and locally. Two major types of simulation jobs are especially supported by GridSFEA on the grid: parameter sweep and long running jobs.

The framework is modular and distributed, having components located on the end-user side (client-side applications), on grid nodes (simulation services), and on the grid systems where the user jobs run (the checkpoint migration tool). In Fig. 1 is depicted the architecture of GridSFEA. Simulation scenarios are annotated by the framework; at the beginning of the job execution, computation metadata of the scenario is collected by the application wrappers and provided to the simulation services. The services organize the metadata and provide it to consumer applications. Grid data exploration tools or the migration tool are examples of such applications. Based on information recorded in the metadata, the consumers can interact with the simulation data located on the grid.

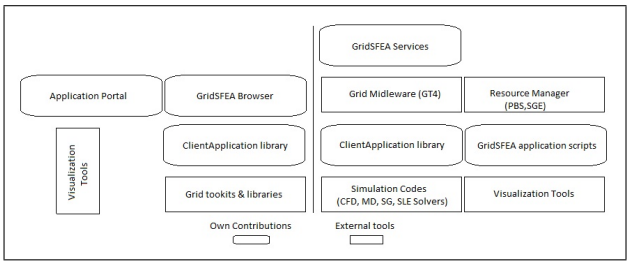


Figure 1: Modules of GridSFEA located in the user space (left) and in the grid environment (right) [15].

We briefly introduce below the role of each of the major elements of the framework. More

details about GridSFEA can be found in [15]. The simulation services are the core of GridSFEA. Their role is to store, organize, and provide scenario metadata to all modules of the framework. The Client Application Library (CAL) acts on the one hand as a client to the simulation services. On the other hand, it handles all the specific grid interactions typically required by client applications such as authentication in grid, file transfer etc. Thus, this module acts as the glue that binds the framework components and integrates the third-party grid libraries. On top of CAL client-side applications are built, such as shell or GUI-based user tools for interacting with the simulations on the grid. Such examples are portal applications, data browser programs, and command-line tool when shell interactions are required by the scientist. Gathering the metadata of the scenarios and handling of simulation checkpoints (checkpoint transfer, simulation resume, and registration of new checkpoints) are the main responsibilities of the Checkpoint Migration Tool. State-of-the-art grid libraries available in the scientific community such as CoG Kit, GAT (Globus Toolkit), DESHL (UNICORE), and CREAM (g-Lite) are employed by the framework for the implementation of grid interactions. This way, our framework allows for client-side interoperability between grids based on these middleware.

Features such as the migration of simulation scenarios or the preview of results in grid make our framework very valuable to computational scientists and engineers and give it a uniqueness touch in the field of grid applications.

4 Eclipse Plugins

In the field of software development, the Eclipse platform is widely employed, on the one hand due to its rich set of supportive tools such as compilers, debuggers, editors, and, on the other hand, due to the large number of contributions from the software community that bring many additional features to the platform. The big success of the Eclipse platform was possible mainly because of the powerful plugin mechanism it employs, which is a good example of state-of-the-art software engineering [5]. Basically, the way that the Eclipse IDE can be extended by its contributors is specified in terms of extension points. The plugins contribute with appropriate extensions to these points, without that the source code of the development environment gets changed. At runtime, these plugins are located, identified, and loaded mostly only when their functionality is required.

In [16] we have presented a detailed list of features that are important to our approach. We summarize below the most relevant ones:

- By means of extension points and extensions, plugins interact, without being familiar with each others implementation.
- The extension points define the public interface the plugin publishes and that can be implemented by further applications.
- Any number of extensions can be used for contributing with functionality to an application.
- The software life-cycle of these plugins is defined in such a way, that a plugin can be installed, updated, debugged, removed independently of the rest of the platform/hosting application.

A further advantage of the Eclipse plugin model and mechanisms is that plugins can be easily grouped together, packed and deployed as extensions of the Eclipse platform, of an existing plugin, or of a plugin-based application. The last two deployment targets are especially of interest in the context of our work on the GridSFEA framework.

5 GridSFEA Plugins for Grid Computing Applications

The mechanisms of GridSFEA are applicable to a wide range of simulation scenarios and can also be used with relatively small customization effort by further grid applications not necessarily related to computational sciences and engineering. To foster the reuse of the know-how we gathered with GridSFEA, we proposed in [16] a new plugin-based organization of the client-side modules of our framework. As enabling technology have been used the Eclipse plugins. The plugin mechanism allows for modular organization of software, for clean and rapid integration of new functionalities, for the extension of existing implementations, for the reuse of framework's features by other applications, all these without altering the existing source code of the software.

The plugins available so far in our framework provide functionalities such as:

- low-level interactions with the grid (such as authentication, job submission, file transfer);
- high-level end-user operations (such as parameter sweep, exploration of results).

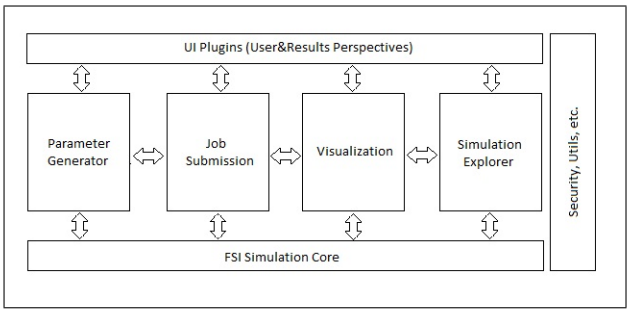


Figure 2: GridSFEA plugins for the development of grid applications [16]

Figure 2 depicts the components of GridSFEA realized with plugins, the arrows representing extension point-extension relationship between components. Several client-side functionalities of GridSFEA such as job submission, job monitoring and exploration of results have been refactored as plugins. In this paper, we further extend this architecture with two groups of plugins: First, we have designed plugins for checking the behavior of applications and services deployed on various grids. Secondly, we provide a plugin for the remote exploration of simulation results located on the grid. These new components of our framework allow the user to tackle various cases studies, such as benchmarking a grid site from an application perspective or deciding on the early stage of a long simulation weather to continue or not the respective grid computation.

Based on the proposed plugins, an application with basic grid capabilities can be rapidly created. In Fig. 3 we show the architecture of such an application. GridSFEA plugins can make contributions to all layers of classical three-layer software architectures, as long as the integration of the grid components is done using the plugin mechanism discussed in the previous section. This way, the developer can focus entirely on the functionality required by the application domain. In the upcoming sections, we give two usage scenarios for the plugins of GridSFEA: The first one is the client-side monitoring of applications and services deployed on grid sites of potential interest for a given end-user and the second one is the exploration of simulation results located on grid.

6 Novel High-level Plugins

This section introduces two novel high-level plugins of GridSFEA. Both are focused on the end-user, the computational scientist: With the first plugin, the user can formulate job requests

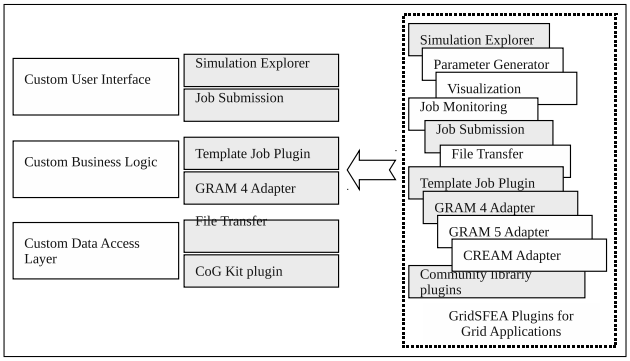


Figure 3: Enhancing a custom application (right) with grid computing capabilities by means of GridSFEA plugins (left): formulation of computation tasks, their submission to GT4-based grids, investigation and download of computation results (gray boxes)

trimmed to the needs of its favorite simulation applications and submit them to different grid sites. The simulation results are accessed with the second plugin, the Simulation Explorer, without that the user knows the actual location of the data in grid.

6.1 Plugins for monitoring grid applications

The goal of these plugins is to provide a simple yet effective mean to check to what extent jobs involving specific applications can run properly one or more grid sites. Such functionality is very useful to users that have access to several grid sites and have one or more applications to run on those sites. A set of pre-defined grid jobs is made available to the application hosting the monitoring plugins. These jobs exercise the targeted application, can use grid services such as data transfer, job submission, or information services, and are submitted to the known grid sites. The plugin collects the execution results for each test job and updates the status of the respective application and services of the grid sites.

The computational scientist, as an end-user of these plugins, can create test jobs by providing job scripts in the JSDL format. The plugins implement a subset of the JSDL specification that allows for most common operations with jobs on grids. In addition to that, the framework accepts also native WS-GRAM job scripts. By offering support for the JSDL specification of the Open Grid Forum, the framework is not bound a single grid middleware or user community. Furthermore, in order to employ these plugins, the end-users and client application of different grid middleware only need to be able to phrase JSDL requests.

The incoming JSLD requests are transformed into middleware specific job requests by means of adapter plugins, as depicted in Fig. 4. The submission of the adapted jobs to the targeted grid sites is carried out by middleware-specific libraries (e.g. CREAM for gLite, CoG Kit for Globus Toolkit 4), that become available to the framework in the form of contributing plugins. The adaptor plugins contribute to the Template Job Plugin by providing implementations of the extension point of the latter one. Community libraries such as GAT, CoG Kit are packed as plugins that export high level packages.

Our approach is lightweight and flexible for several reasons. First, the contributor plugin corresponding to a grid middleware, such as the GRAM 4, encapsulates properly the client-side specifics of that middleware, here Globus Toolkit 4. The sole interaction between the plugins is done by means of the defined extension points. Second, the monitoring plugin may be used for one or more grid sites without any changes. Furthermore, the customized test jobs can be used to benchmark the grid sites from an application perspective. The end-user of the grid can write

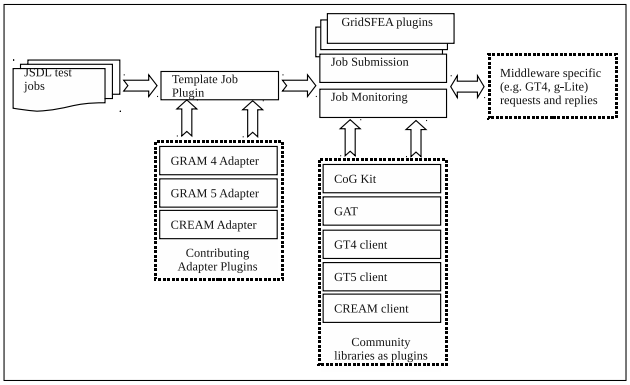


Figure 4: JSDL incoming job requests are adapted on-the-fly to middleware-specific outgoing job requests by means of plugins.

test jobs that reflect the way she uses a given application for her scenarios, the response of the monitoring plugins being thus trimmed to the user needs.

By providing access to different types of grid middleware, our plugin bundle provides client-side interoperability of grids. In a seamless way, files can be transferred between heterogeneous sites (such as from GT4 to UNICORE) and jobs can be submitted to grid sites running any of the supported grid middleware. The beneficiaries of our approach are, on the one hand, the end-users with access to multiple grid sites (for operations such as checking the existence of installed programs, running application trimmed test jobs, benchmarking sites from an application perspective). On the other hand, these plugins can be easily integrated in any Eclipse-based plugin or RCP (Rich Client Platform) application with the purpose of interacting with grid installations (such as monitoring tools).

6.2 Plugin-based exploration of simulation scenarios

A second example of usage scenario of the GridSFEA plugins contributes to the interaction of end-users with remote data located in grids. For that, a plugin for data exploration was designed based on components of GridSFEA. All data belonging to a simulation run i.e. input files, input parameters, result files, post processing results, and checkpoint files are grouped logically in a simulation scenario. Scenarios can be downloaded to the end-user space, transferred between grid sites, or used to reset/continue a simulation at the same location or at a different one.

The access to the simulation scenarios is made in a way transparent to the end user. He needs not to know the exact location of the remote data nor the way (protocol and service) to access it. With our plugin running on the user system, metadata of scenarios is retrieved from simulation services. The plugin lists all instances of a scenario, instances created during its migration for several times.

All the movements and changes of a scenario are recorded with the Simulation Services and can be displayed by tools such as the Simulation Explorer plugin (Eclipse-based plugin), the GridSFEA browser (standalone desktop application), or a thin client for mobile devices. A common feature of these tools is that the scenario data can be explored on the client side in a preview-like mode. Excerpts from scenario files can be displayed in the end-user application, as well as results of the post processing steps (images generated by VisIt, MATLAB, GNU Plot etc.).

Thus, the preview of scenarios enables the user to comfortably investigate/explore the results of the simulation without downloading the files to the end-user system. This feature gives the user the needed instrument to assist her in deciding if the results worth further investigations

or the simulation needs to be recalculated, even with changes in the input configuration of the simulation. Complete result files can also be downloaded to the end-user's machine and further explored locally with scientific visualization tools, e.g.

7 Results

The results discussed in this section have been obtained on three grid sites: one operated with GT4 (`labacal.utcluj.ro`, 48 cores), one with GT5 (`acalgrid01t.utcluj.ro`, 48 cores), and one again with GT4 (`a01.hlrb2.lrz-muenchen.de`, +8000 cores) as well. The first two sites are for the development and testing of grid applications, the third one is a production site from the DEISA infrastructure. The aim of the experiments was to show how the plugins introduced in Sec. 6 were employed to build a custom application. This evaluation was done with respect to the requirements of a simulator based on the Neocortex program [17] for investigating the dynamic properties of microcircuits of spiking neurons.

7.1 Scenario: simulation of neural microcircuits

Understanding the behavior of biological neurons, in particular of the human one, is a challenging task for the entire neuroscience community. This is an interdisciplinary area, with major contributions from biology, physics, chemistry, psychology. Since the experimental and observational means are in this particular area rather limited, the simulation becomes a very valuable instrument for gathering knowledge, along with theory and experiments. *Neurosim* is a spiking neural simulator based on the Neocortex tool, at the moment of writing still under development, having the goal to compute on HPC systems realistic models of neural microcircuits based on biological neuron models. The general aim of the investigations where *Neurosim* will be used is the evaluation of the impact the different parallelization strategies on the dynamics of neural microcircuits. In-depth descriptions of this simulation program and of the mathematical models it employs in simulations are beyond the scope of this paper. Nevertheless, we outline a few typical computational requirements of *Neurosim*: batch mode (headless operation), support for input/output files and program arguments; post-processing operations with Gnuplot; parameter-based investigations; parallel runs using hybrid parallelization techniques (MPI combined with OpenMP). The list above comprises requirements common to typical applications running on HPC systems. The execution of such applications on computing grids with HPC resources is widely supported by GridSFEA.

7.2 Custom application based on high-level plugins: GridSFEA Trotter

From the pool of plugins available with GridSFEA we have built a rich client application named here *GridSFEA Trotter*, having the kern functionalities: incoming requests for simulation jobs are formulated in JSDL, these requests are transformed into GRAM4 or GRAM5 specific job descriptions, jobs are submitted to any grid site running the middleware GT4 or GT5, results of the jobs are monitored and measured, computation results are presented to users in terms of simulation scenarios, results and post-processing files are pre-viewed remotely and further processed locally. The application comes with a set of pre-defined job templates that can be used to benchmark an application/service hosted by a grid site. These templates have been customized to the needs of *Neurosim* and have been used for describing simulations of neural microcircuits. The template jobs for Neurosim address different ways of working with this program such as performing sequential and parallel runs, staging in input files (network descriptions, initial values etc), staging out results, trigger post-processing operations. GridSFEA Trotter comprises the

high-level plugins for benchmarking grid applications and for remote exploration of simulation scenarios, and a suite of low-level plugins, such as the Template Job Plugin, GRAM 4 and GRAM 5 adaptors, File Transfer, plugins packing community libraries (CoG Kit, GT4 client, and GT5 client). By means of extension points, further adaptors can be loaded automatically at runtime by the Trotter.

7.3 Results and measurements

The goal of our experiments was to quantify the overhead introduced by our tool in comparison to the local execution of the simulation programs and to corresponding tools from the grid middleware. The job scripts perform the operations: submission of jobs without file staging, submission of jobs with file staging (in and out), and submission of jobs that require a checkpoint file handled by GridSFEA. The computation tasks are described in the JSDL format. The implementations of the JSDL specification and of its POSIX extension are partial. The suite of templates was executed on different grid sites using both our plugin-based application and the client tools of Globus Toolkit. In Table 1 is displayed the execution time of job scripts for different Linux commands contained by the templates. These commands have very small execution times, of the order of mili-seconds.

Table 1: Execution time for the JSDL job templates with GridSFEA Trotter. The size of the input and of the output file is approximately the same for one measurement.

Template	labacal[s]	acalgrid01t[s]	hlrb2[s]
/bin/ls, /bin/date, /bin/hostname	3.8	3.7	3.9
Neurosim (no processing)	6.7	6.9	7.1
Batch job	2.5	2.6	2.8

Considering the small duration of the execution of the commands employed in these tests, the results indicated in Table 1 show actually the overhead introduced by the grid middleware deployed on any of the three sites. This overhead has the order of seconds and clearly dominates the execution time of the grid job. Each site has a slightly different response time, depending on its setup. When comparing time results obtained on any of these sites, one should fairly consider the response time characteristic to the respective site. The job scripts for executing *Neurosim* require file staging operations. Table 2 shows the results related to such file operations (upload of input and checkpoint files, download of result, standard output/error, and checkpoint files). These experiments have been conducted with the GridSFEA Trotter application on the grid site labacal. The file dimensions employed here are typical for different scenarios of *Neurosim*, corresponding to neural microcircuits with different physical connections (ranging from O(1) to O (10^4) synapses for one neuron). The execution time of the simulation program has been subtracted from these time results to allow a fair comparison of the scenarios.

The time results listed in the second column of Table 2 correspond to *Neurosim* jobs where input and output files have been staged in and staged out, resp. The third column contains execution time of simulations that need to be restarted from a checkpoint and include the transfer time of that file. These values represent the total transfer time, including the contribution of GridSFEA Trotter and of the grid middleware. Results in the last column represent the time needed for the Trotter to find the actual location of a checkpoint file, information provided by the simulation services of GridSFEA. This information is extracted from the metadata about the scenario collected by the framework. The localization time must be added to the values in the third column to get the entire duration of such a job from the user perspective. The time needed

Table 2: Time measurement for executing *Neurosim* jobs with GridSFEA Trotter. The size of the input, output and checkpoint file is approximately the same for one measurement.

Scenario Size[MB]	Stage In/Out Time [s]	Checkpoint Time[s]	Localization Time[s]
0.01	6.9	5.5	9.92
0.10	5.9	5.4	9.89
1.0	5.3	5.4	9.90
10	10.2	14.1	8.72
100	53.5	98.1	9.93

to get the localization of the checkpoint does not depend on the size of the scenario (given in the first column) and is a characteristic of the framework. The job files in the GRAM 4 format corresponding to the JSDL jobs used for these measurements have been submitted to the grid with the client tool of the Globus Toolkit middleware (see Table 3).

Table 3: Time measurement for the execution of *Neurosim* jobs with client tools of GT. The size of the input, output and checkpoint file is approximately the same for one measurement.

Scenario Size[MB]	Stage In/Out Job Time [s]	Checkpoint Job Time[s]
0.01	4.4	5.5
0.10	3.9	4.6
1.0	4.2	4.8
10	6.1	9.2
100	31.3	63.8

The localization of the checkpoint has been carried out by the user, as well as the altering of the job file. Thus, these have not been considered in the measurements. We can see that the duration of the jobs is shorter than the execution of the same scenarios with GridSFEA Trotter. This is mainly due to the parallel file transfers implemented in the client tools of the Globus Toolkit. This feature is currently missing from the CoG Kit library that is employed by our application. Nevertheless, the advantages of our approach are threefold: automatic localization of checkpoint files, automatic generation of the job script that relied on checkpoints, and the use of the middleware independent job description language JSDL. The evaluation of the Simulation Explorer plugin, hosted by GridSFEA Trotter, was carried out against the tools of the GT middleware. The results were similar to the ones above.

Our conclusion reads that for small data size, below 10 MB, the overhead of our approach was small. For larger files, the download with the plugin took longer. The preview data is typically much smaller than the raw data produced by the simulation. Thus, it is expected that in real-word usage, GridSFEA Trotter will have almost the same response time as the middleware specific tools. An advantage of our approach is the integration of the simulation explorer with the rest of the plugins being capable, thus, to automatically obtain form the simulation services localization information about the scenario data, regardless where the jobs have been computed. Therefore, the investigation of the scenarios was carried out with the Trotter in a seamless way.

The performance of this approach was to some extent better than in [15], without plugins. Nevertheless, the localization time was higher here due to generally longer responses from the grid servers. The experiments conducted here showed that the plugins we provide are useful to users from the scientific computing area. We have created a sample application, GridSFEA

Trotter, which demonstrates the way our plugins are integrated and are employed for high-level operations on the grid. A strong limitation of our approach occurs when large files are involved in the operations of the end-user, the time overhead introduced by our approach being significantly high. This could be reduced by exploiting parallel file transfers, for example. A further drawback is that applications need to support the Eclipse plugin mechanism in order to host our plugins. Nevertheless, our plugin-based approach brings out-of-the-box high-level functionalities very useful to the development of grid applications.

8 Conclusions and lessons learnt

We have presented in our work on a plugin-based approach for the development of grid applications. With our approach, the realization of both high-level (focused on the grid user) and low level (oriented towards the grid) operations is possible, leading to out-of-the-box functionalities reusable in further applications. This was demonstrated here with GridSFEA Trotter. Comfortable for the grid user, the operations on grid introduce an overhead in some cases, when large data files are involved. In other cases, the performance of our approach is similar to the one of the native grid client tools, such as the ones of the Globus Toolkit, but enriches their usage within more complex scenarios such as the exploration of simulation data located on grid. Specifics of the grid middleware are packed as contributing plugins, thus, making the adoption of a further middleware a straightforward task. Moreover, the use of JSDL for formulating job requests opens the use of applications based on our plugins to scientific communities relying of different grid middleware than the Globus Toolkit. In the global context of the unification of the grid middleware, our work makes a step forward from the grid application development perspective.

Acknowledgements

This work was partly supported by National Council of Scientific Research in High Education within the PNII Human Resources program, project number 10/2009, and by the POSDRU program, financing contract POSDRU/89/1.5/S/62557.

Bibliography

[1] C. Aiftimiei, P. Andreetto, S. Bertocco S. Dalla Fina, A. Dorigo, E. Frizziero A. Gianelle, M. Marzolla, M. Mazzucato, M. Sgaravatto, S. Traldi, and L. Zangrando. Design and implementation of the gLite CREAM job. *Elsevier*, 26, 2010.

[2] G. Allen, K. Davis, T. Goodale, A. Hutanu, H. Kaiser, T. Kielmann, A. Merzky, R. van Nieuwpoort, A. Reinefeld, F. Schintke, T. Schott, E. Seidel, and B. Ullmer. The grid application toolkit: Toward generic and easy application programming interfaces for the grid. *Proceedings of the IEEE*, 93:534–550, March 2005.

[3] ARC. Advanced resource connector resources. http://www.nordugrid.org/arc, 2010.

[4] S. Burke, S. Andreozzi, and L. Field. Experiences with the GLUE information schema in the LCG/EGEE production grid. *Journal of Physics: Conference Series*, 119, 2008.

[5] E. Clayberg and R. Dan. *Eclipse: Building Commercial- Quality Plug-ins (2nd Edition).* Addison-Wesley Professional, 2006.

[6] L. Clementi, M. Rambadt, R. Menday, and J. Reetz. UNICORE deployment within the DEISA supercomputing grid infrastructure. In W. Lehner, N. Meyer, A. Streit, and C. Stewart, editors, *Euro-Par Workshops*, volume 4375 of *LNCS*, pages 264–273. Springer, 2006.

[7] EGI. Unified middleware distribution. Technical report, EGI, 2009.

[8] EGI. European Grid Infrastructure resources. http://ww.egi.eu, 2010.

[9] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputer Applications*, 11(2):115–128, 1997.

[10] gLite. Lightweight middleware for grid computing. http://glite.web.cern.ch/glite, 2010.

[11] T. Goodale, S. Jha, H. Kaiser, T. Kielmann, P. Kleijer, G. von Laszewski, C. Lee, A. Merzky, H. Rajic, and J. Shalf. SAGA: A simple api for grid applications. high-level application programming on the grid. *Computational Methods in Science and Technology*, 12(1):7–20, 2006.

[12] IGE. Initiative for Globus in Europe. http://www.ige-project.eu, 2010.

[13] G. Kandaswamy, L. Fang, Y. Huang, S. Shirasuna, S. Marru, and D. Gannon. Building web services for scientific grid applications. *IBM J. Res. Dev.*, 50(2/3):249–260, 2006.

[14] H. Kornmayer, M. Stümpert, H. Gjermundrød, and P. Wolniewicz. g-Eclipse – a contextualised framework for grid users, grid resource providers and grid application developers. In M. Bubak et al., editors, *Computational Science – ICCS 2008*, volume 5103 of *LNCS*, pages 399–408. Springer, 2008.

[15] I.L. Muntean. *Efficient Distributed Numerical Simulation on the Grid*. PhD thesis, Institut für Informatik, Technische Universität München, 2008.

[16] I.L. Muntean. Plugins for numerical simulation with GridSFEA on computing grid. In *Procs. of the 8th Intl. RoEduNet Conference Galati*, pages 51–56, 2009.

[17] R.C. Muresan and I. Ignat. The Neocortex neural simulator, a modern design. In *International Conference on Intelligent Engeneering Systems*, 2004.

[18] Open Grid Forum. Job submission description language resources. http://www.gridforum.org/documents/GFD.56.pdf, 2010.

[19] S. Parastatidis, J. Webber, P. Watson, and T. Rischbeck. WS-GAF: a framework for building grid applications using web services: Research articles. *Concurrency Computat.:Pract. Exper.*, 17(2-4):391–417, 2005.

[20] M. Riedel and et al. Interoperation of world-wide production e-science infrastructures. *Concurr. Comput.: Pract. Exper.*, 21:961–990, 2009.

[21] M. Surridge, S. Taylor, D. De Roure, and E. Zaluska. Experiences with GRIA — industrial applications on a web services grid. In *E-SCIENCE '05: Proceedings of the First International Conference on e-Science and Grid Computing*, pages 98–105, Washington, DC, USA, 2005. IEEE Computer Society.

[22] S. Venugopal, R. Buyya, and L. Winton. A grid service broker for scheduling distributed data-oriented applications on global grids. In *MGC '04: Proceedings of the 2nd workshop on Middleware for grid computing*, pages 75–80, New York, NY, USA, 2004. ACM.

[23] G. von Laszewski and M. Hategan. Workflow concepts of the Java CoG Kit. *J. Grid Computing*, 3(3-4):239–258, 2005.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 713-730

Quality of Service Control for WLAN-based Converged Personal Network Service

E.-C. Park, I.-H. Kim, G.-M. Jeong, B. Moon

Eun-Chan Park, Bongkyo Moon
Department of Information and Communication Engineering,
Dongguk University-Seoul, Korea
E-mail: ecpark@dongguk.edu, bkmoon@dongguk.edu

In-Hwan Kim
Mobile Device Development Team,
SK Telecom, Seoul, Korea
E-mail: ihkim@sktelecom.com

Gu-Min Jeong
School of Electrical Engineering,
Kookmin University, Seoul, Korea
E-mail: gm1004@kookmin.ac.kr

Abstract: This paper proposes a framework for quality of service (QoS) control in WLAN-based converged personal network service (CPNS). First, we show that the CPNS devices in WLANs occupy the shared wireless channel in an unfair manner; and thus, QoS is degraded. The reasons of such problem are analyzed from two viewpoints of (*i*) channel access mechanism according to carrier sensing multiple access protocol of WLAN and (*ii*) TCP congestion control mechanism in response to packet loss. To improve QoS and assure fair channel sharing, we propose an integrated QoS control framework consisting of admission control and rate control. Based on the available capacity, the admission control determines whether or not a specific QoS service can be admitted. The rate control is implemented using congestion window control or token bucket algorithm. The proposed mechanism differentiates QoS service from best-effort (BE) service such that the QoS service is preferentially served to satisfy its QoS requirements and the BE service is served to share the remaining resource in a fair manner. The extensive simulation results confirm that the proposed scheme assures QoS and fair channel sharing for WLAN-based CPNS.
Keywords: CPNS, QoS, fairness, WLAN, TCP

1 Introduction

Recently, many new portable devices equipped with Wi-Fi interface, e.g., smart phone, tablet PC, mobile internet device (MID), portable media player, and digital camera/camcorder, have been introduced. They are connected to the Internet or with each other and enable many new emerging services to come into wide use, such as real-time multimedia streaming, video telephony, on-line game, image/video sharing. In order to provide converged service for the portable devices in short-range personal networks such as home networks or in-car networks, Open Mobile Alliance (OMA) makes a standard for converged personal network service (CPNS) [1], [2] by identifying the overall service architecture, service scenario, and functional requirements. The CPNS system consists of CPNS server, personal network gateway (PN GW), and personal network element (PNE); the PNE accesses to the PN GW via wireless link and is provided with

various services from the CPNS server. We consider in this paper that the CPNS system is implemented with IEEE 802.11 wireless local area networks (WLANs) [3]. WLAN is widely deployed in home networks and installed in many devices, i.e., WLAN is a *de facto* standard for wireless networking for portable devices in small area, and it is the most suitable wireless access technology considering the service scenario, coverage, and data rate for CPNS.

This paper deals with the issue of QoS assurance in WLAN-based CPNS. The various services of CPNS (e.g., real-time multimedia streaming, Internet telephony, file uploading/downloading, remote surveillance) have different QoS requirements in terms of delay, loss, or throughput; however it is a challenging problem to satisfy these diverse QoS requirements because the MAC protocol of WLAN works in a distributed contention-based manner and the quality and capacity of wireless channel are time-varying and location-dependent. First, we show that the PNEs[1] access the channel in an unfair way, i.e., the uploading (UL) PNEs dominate the use of shared channel while the downloading (DL) PNEs cannot occupy their fair shares; and thus, QoS can be severely degraded especially for DL PNEs. We analyze the reasons of this problem from two viewpoints; (*i*) unfair channel access mechanism of MAC and (*ii*) asymmetric congestion control mechanism of TCP.

To resolve this problem, we propose an integrated QoS control framework, which consists of admission control and rate control. We classify the service into two categories; service with and without QoS requirements, each of which is referred to as *QoS service* and *best-effort (BE) service*, respectively. The proposed scheme differentiates the QoS service from the BE service. The admission control is applied to the QoS service in order to prevent an excessive request of QoS service from being allowed. The admission control estimates the available channel capacity and converts the required data rate of a QoS service to the equivalent channel capacity. Only if the requirement is less than the available capacity, the QoS service can be admitted and the required capacity is virtually reserved. On the other hand, the rate control is applied to both QoS service and BE service so that the required data rate of QoS service can be assured and the BE services are enforced to share the channel in a fair manner. The rate control is implemented by employing TCP congestion window control mechanism or token bucket mechanism. Furthermore, the proposed framework can be extended to minimize the delay of QoS service and to support weighted fairness for BE services. Consequently, QoS can be assured for QoS services and fairness can be attained among BE services, which is confirmed through ns-2 [4] simulations.

There have been many proposals for assuring QoS and fairness in IEEE 802.11 WLANs in the literature. The enhanced distributed channel access (EDCA) mechanism has been standardized in IEEE 802.11e [5] to improve QoS by differentiating the channel access delay between different access categories. Similar QoS enhancement mechanisms have also been proposed for prioritized service differentiation [6–8]; they modify the basic MAC protocol and use differentiated MAC parameters, e.g., inter-frame space, contention window, or transmission opportunity, depending on the service priority. Moreover, the fairness issue of WLAN has been actively studied. Many MAC protocols have been proposed to assure MAC-layer fair access opportunity between competing stations or between DL and UL stations by employing a proper backoff mechanism or by controlling contention window size [9–12]. Recently, the issue of TCP fairness in WLAN has been addressed in [13–15]. They aim to assure fairness between TCP flows in WLAN by limiting TCP congestion window size [13], by employing per-flow scheduling in the access point (AP) [14], or by differentiating channel access delay based on IEEE 802.11e [15]. Compared to these previous studies, this paper makes the following contributions;

- This paper proposes a *generalized* framework to assure QoS and fairness simultaneously, it is applicable regardless of packet size, transmission rate, service direction (DL or UL), and

[1] In this paper, we interchangeably use the terms, PNE, device, and station.

transport-layer protocol (TCP or UDP).

- The proposed mechanism provides QoS service with the *absolute* assurance in terms of data rate. At the same time, it provides BE service with the *weighted* fairness without decreasing the overall channel utilization.
- The proposed mechanism is *practical and simple* for CPNS; it defines a service establishment signaling but does not modify the standard IEEE 802.11 MAC protocol, and the major function of the proposed scheme needs to be implemented in the PN GW while the PNEs require a minor modification.

The rest of this paper is organized as follows. In Section 2, we identify the problem of unfair channel sharing and QoS degradation arising in WLAN-based CPNS and analyze its reasons. The QoS control framework and algorithm are proposed in Section 3, and the performance is evaluated through extensive simulations in Section 4. The conclusion follows in Section 5.

2 Problem statement

In this section, the problem of QoS degradation and unfairness is shown via the preliminary simulation, and the reasons of such problem are analyzed.

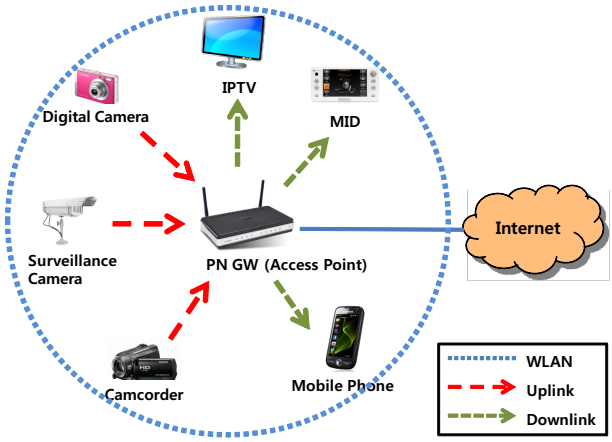


Figure 1: Service model of WLAN-based CPNS.

2.1 QoS degradation and unfairness in WLAN-based CPNS

We consider the service model of WLAN-based CPNS, especially focusing on home network, as shown in Fig. 1. The PN GW plays a role of AP in WLAN and it is connected to the Internet or CPNS server through a wired link and communicates with the PNEs. We consider the following typical service scenario for CPNS;

- the digital camera and camcorder upload their photos and moving pictures to the PN GW so that the contents can be played in the IPTV, smart TV, or digital photo frame. Also, they upload the contents to the PC or remote storage server to free up its built-in memory, or they upload the contents to social community sites (e.g., YouTube or flickr) to share them with others;
- the surveillance camera uploads the real-time streaming video to the server for the purpose of security;

Table 1: Service scenario used in the simulation.

device	service	direction	transmission protocol	required data rate	start time
IPTV	streaming	DL	TCP	1 Mb/s	0 sec
mobile phone	file transfer	DL	TCP	N/A	0 sec
camcorder	file transfer	UL	TCP	N/A	50 sec
surveillance camera	security	UL	UDP	1 Mb/s	100 sec

- the MID and mobile phone download multimedia contents from the CPNS server or the Internet to store the contents into the built-in memory and play them later. Moreover, they can play the multimedia contents in the PC or remote storage server as streaming.

We perform preliminary ns-2 simulation with the specific service scenario given in Tab. 1. Note that the service without the requirement of data rate, i.e., file transfer of mobile phone and camcorder, is considered as BE service. The MAC and PHY parameters are set according to IEEE 802.11b specification, the PHY transmission rate and packet size are set to 11 Mb/s and 1 Kbyte, respectively, the buffer size in the PN GW is set to 100 Kbyte [2], and the TCP window size is set to 64 Kbyte (default value in Microsoft Windows XP). The file size is assumed to be large enough for the service to last till the end of simulation. The wireless channel error is modeled as a two-state Markov-modulated process, where the packet error rate in the wireless channel is 1%.

We observe per-device throughput to evaluate the performance in terms of QoS and fairness from Fig. 2. Before the camcorder uploads a file at $t = 50$ sec, the required data rate for IPTV is satisfied and the remaining capacity is utilized by the mobile phone, and the throughputs of both devices are stable. When t = [50 - 100] sec, however, the average throughput of IPTV is about 770 Kb/s, i.e., the required data rate cannot be assured, moreover its throughput fluctuates severely, which results in the significant degradation of QoS due to the increase of delay and jitter. Also, the throughput of mobile phone is decreased suddenly because of the UL file transfer of camcorder. The wireless channel capacity is not allocated in a fair way; during the time interval between 50 sec and 100 sec, the average throughput of camcorder is higher than that of IPTV and mobile phone by about 2.10 and 1.87 times, respectively. After $t = 100$ sec, the throughput of surveillance camera is stable and satisfies its required data rate of 1Mb/s and it is little affected by the other services. However, the throughputs of IPTV and mobile phone are further decreased. Even in this case, the throughput of camcorder is higher than those of IPTV and mobile phone by about 3.10 and 2.22 times, respectively. These simulation results confirm that (*i*) the QoS of DL service is highly vulnerable to the UL service while the QoS of UL service is little affected by the DL service, (*ii*) the channel capacity is unfairly shared among BE services; the UL service is favored over the DL service.

2.2 Causes of the problem

MAC: Unfair channel access

In infrastructure WLANs, all the devices are associated with the AP. The UL devices transmit packets to the AP while the DL devices receive packets from the AP. The MAC-layer contention for channel access takes place among transmitters, AP and UL devices, according to the carrier sensing multiple access (CSMA) mechanism, i.e., IEEE 802.11 distributed coordination function

[2]The buffer size is determined for bounded queuing delay to support the multimedia streaming services whose QoS is sensitive to delay and/or delay variation.

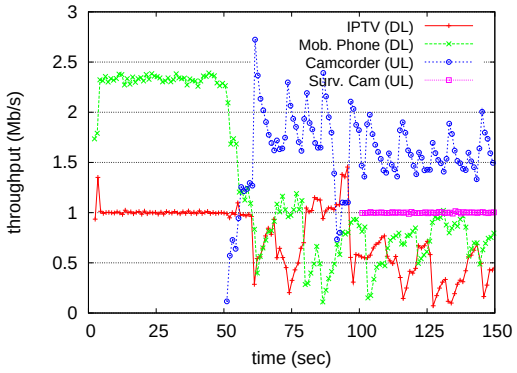


Figure 2: Per-device throughput in the WLAN-based CPNS without QoS control.

(DCF) [3]. If the AP wins the contention, the DL devices can be served, otherwise if the UL device wins, it gets its own transmission opportunity. Defining the number of UL and DL devices as N_u and N_d, respectively, the number of contending devices are $N_u + 1$, independent of N_d. The DCF mechanism provides all the contending devices with the equitable chances of channel access in the long term. Assume that all the UL devices always make attempt to send packets at the same PHY-layer transmission rate and the data generation rates for all the DL devices are same. Then, the steady-state throughputs of UL and DL devices, denoted as th_u and th_d respectively, become

$$\begin{aligned} th_u &= \frac{C}{N_u + 1}, \\ th_d &= \frac{C}{(N_u + 1)N_d}. \end{aligned} \tag{1}$$

Here, C denotes the effective capacity of WLAN. Defining a fairness index, γ, as the ratio of the average throughput achieved by the DL devices to that by the UL devices, it becomes

$$\gamma \triangleq \frac{th_d}{th_u} = \frac{1}{N_d} \tag{2}$$

from (1). It is important to note that γ depends on N_d, but it is irrespective of N_u.

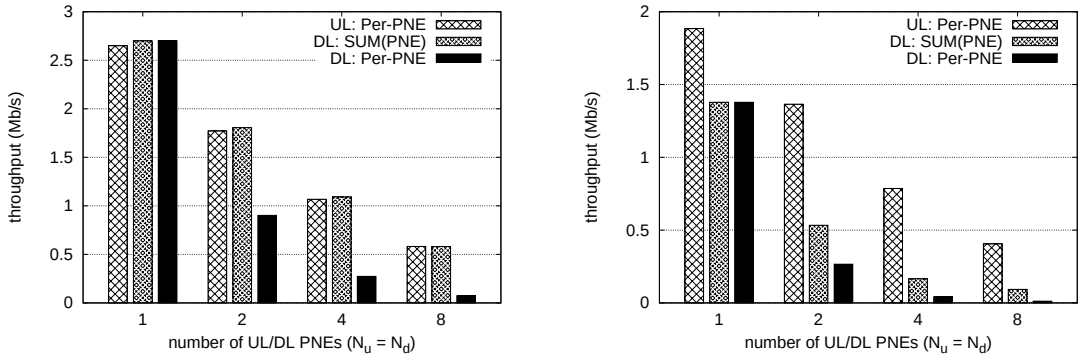
We perform ns-2 simulation to validate this analysis. Figure 3(a) shows the average per-device throughput with various values of $N_d (= N_u)$ when devices send/receive UDP traffic. The discrepancy between th_u and th_d increases, i.e., γ decreases, as N_d increases. Moreover, we observe that the average per-device throughput of an individual UL device is nearly equal to the sum of per-device throughputs of all DL devices, which agrees with (1).

TCP: Asymmetric congestion control

The problem of unfair channel access becomes exacerbated when devices send/receive TCP traffic. As shown in Fig. 3(b), even when only one DL device coexists with one UL device (i.e., $N_d = N_u = 1$), the throughput of DL device is lower than that of UL device by about 26%, i.e., $\gamma = 0.74$. As N_u increases from 2 to 8, the value of γ remarkably decreases from 0.19 to 0.03.

The aggravation of unfair channel sharing results from the asymmetric congestion control mechanism of TCP in response to packet loss. In general, the capacity of WLAN is smaller than that of wired link connected to the AP. According to TCP congestion control, TCP sender increases its transmission rate until it detects a packet loss. Consequently, the downlink buffer of

AP easily becomes congested and packets are dropped due to buffer overflow. It is important to note that two different types of packets are buffered in the AP's downlink buffer; TCP data packet for DL device and TCP ACK packet for UL device. When TCP data packet for DL device is dropped in the buffer, the TCP sender recognizes this as an indication of congestion and decreases the congestion window by half to relieve the congestion; and thus the throughput of DL device is halved. On the other hand, the TCP in the UL device can be unresponsive to the TCP ACK packet loss in the AP's downlink buffer. Due to the cumulative ACK mechanism of TCP, the loss of an ACK packet for UL device does not necessarily decrease the congestion window as long as the next ACK packet with a higher sequence number is delivered before the retransmission timer expires. Therefore, the cumulative ACK mechanism lets the UL device tolerate the loss of ACK packets, and the TCP in the UL device increases the congestion window gradually until it reaches the maximum size, increasing the throughput of UL device. Consequently, the throughput of UL device becomes higher than that of DL device, and the unfairness problem occurs more severely, compared to the case of UDP traffic.



(a) UDP (b) TCP

Figure 3: Per-device throughput with different numbers of UL/DL devices.

3 QoS control mechanism

We propose QoS control mechanism that provides QoS service and BE service with the quantitative assurance for QoS and the relative fairness in terms of throughput, respectively. We describe the overall structure and signaling of the proposed mechanism in the first subsection, and the rate control and admission control, two main components of the proposed mechanism, in the subsequent subsections. Lastly, we discuss several issues to elaborate and extend the proposed mechanism.

3.1 Overall structure and signaling

Figure 4 shows the operational flow chart of the proposed QoS control mechanism. To establish a service flow (SF) in advance of service start, a PNE transmits a *service request* (`S-REQ`) message to the PN GW, where several properties of the service (e.g., required data rate, packet size, and protocol) are described. If the required data rate is not specified in the `S-REQ` message, we consider such a service as BE service, otherwise the service is considered as QoS service. On receiving the service request, the PN GW classifies the service as QoS service or BE service. For QoS service, the admission control is applied so that the service can be admitted only if its required data rate can be assured. For BE service, the service is admitted without admission

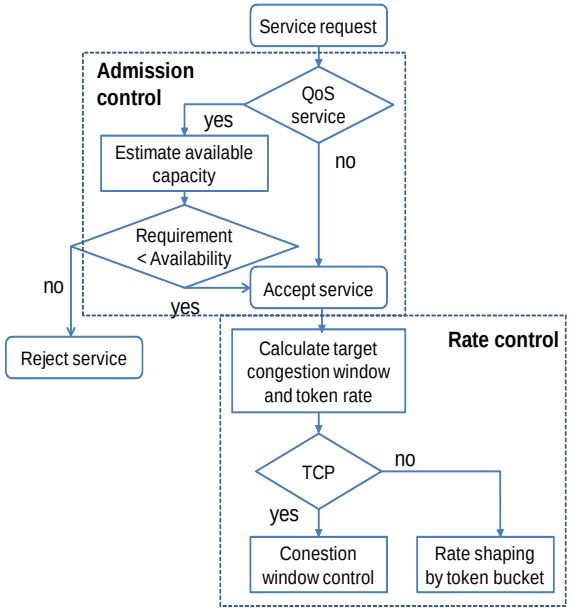


Figure 4: Operational flow chart of the proposed QoS control mechanism.

control and the remaining resource is evenly allocated to BE services under the rationale that the priority of QoS service is strictly higher than that of BE service. Once the service is admitted, the PN GW broadcasts a *service response* (`S-RSP`) message to the PNEs, to acknowledge the resource reservation for the QoS service and to inform the PNEs with the on-going BE services of the change in the available resource. The `S-RSP` message includes target values of TCP congestion window (CGW) and token generation rate for both QoS and BE services, which are calculated by the PN GW based on the QoS requirement, available channel capacity, and the number of BE SFs. The rate control is implemented by congestion window control for TCP flows or token bucket algorithm for UDP flows. Receiving `S-RSP` message, the PNE sets the advertised window size (i.e., the maximum congestion window size) for the TCP flow, as the value in the `S-RSP` message, to control its transmission rate. For the UDP flow, PN GW or PNE sets the token generation rate for the flow as the value in the `S-RSP` message, so that its transmission rate can be regulated around the target rate. When a service is terminated, a *service termination* (`S-END`) message is delivered to the PN GW so as to release the resource occupied by the terminating service and redistribute it to the PNEs with existing BE services. The PN GW recalculates the target values of congestion window and token generation rate and informs PNEs of the updated values by broadcasting the `S-RSP` message.

3.2 Rate control

We propose two mechanisms for rate control; congestion window control for TCP flows and rate shaping for UDP flows. First, we focus on the rate control for TCP flows. It is shown from [13] that by controlling the size of congestion window, the packet loss due to buffer-overflow in the AP's buffer can be avoided and the unfair channel sharing among TCP flows can be alleviated. We extend this idea to QoS service and BE service delivered by TCP flows. To assure the resource reservation for QoS service and fair resource allocation among BE services, we determine the value of the maximum congestion window for service i, denoted as $CGW_{max,i}$, as;

$$CGW_{max,i} = w_i B, \tag{3}$$

where

$$w_i = \begin{cases} \frac{R_{QoS,i}}{\widetilde{C}} & \text{for QoS service,} \\ \left(1 - \frac{\sum_j^{N_{QoS}} R_{QoS,j}}{\widetilde{C}}\right) \frac{1}{N_{BE}} & \text{for BE service.} \end{cases} \tag{4}$$

Here, N_{QoS} and N_{BE} are the number of QoS and BE service flows, respectively, $R_{QoS,i}$ denotes the required data rate for the ith QoS service, and $\widetilde{C}$ is the estimated capacity of WLAN. Note that the size of CGW limits the number of on-the-fly packets, i.e., the number of packets that can be transmitted before receiving TCP ACK. From (3) and (4), we can see that a dedicated portion of buffer is reserved for each QoS service for assuring its desired data rate and the remaining portion of buffer is evenly distributed to BE services for fairness.

Proposition 1. *The CGW control in (3) and (4) prevents packet loss due to buffer-overflow in the AP.*

Proof: From (3) and (4), it can be shown that

$$\sum_{i}^{N_{QoS}+N_{BE}} CGW_{max,i} = B. \tag{5}$$

Since the number of on-the-fly packets is limited by the CGW and the packets may be stored in the buffer except the AP, they are not dropped in the AP's downlink buffer due to buffer-overflow. Therefore, the proposed CGW control can remove one cause of unfair channel sharing, i.e., asymmetric congestion control in response of packet loss. □

Proposition 2. *Assume that the wireless link is a bottleneck link so that the queuing delay in the downlink buffer of AP is a dominant factor of round trip time (RTT) and that the packet loss rate in the wireless link is negligible due to link-layer collision avoidance and retransmission mechanisms. Then, the CGW control in (3) and (4) assures the desired data rate for QoS service and achieves fair and full channel utilization for BE service.*

Proof: From Proposition 1 and the assumptions of Proposition 2, we can consider that the packet loss, which may occur due to buffer-overflow, collision among contending stations, and the poor quality of wireless channel, is insignificant. Then, the size of congestion window increases up to its maximum value according to TCP congestion control, and the steady-state throughput for the ith service flow, th_i^*, becomes

$$th_i^* = \frac{CGW_i}{T_{RTT,i}^*}, \tag{6}$$

where $T_{RTT,i}^*$ is the value of RTT in steady state for the ith flow. Decomposing $T_{RTT,i}^*$ as queueing delay in the AP's downlink buffer ($T_{q,i}$) and the other delay ($T_{o,i}$) (e.g., processing delay at a node and link propagation delay), $T_{RTT,i}$ can be represented as

$$T_{RTT,i}^* = T_{q,i} + T_{o,i} \approx \frac{B}{\widetilde{C}}, \tag{7}$$

from the assumptions of Proposition 2, i.e., $T_{q,i} \approx B/\widetilde{C} \gg T_{o,i}$, where $\widetilde{C}$ is the effective capacity of WLAN in steady state. From (3) and (7), we can rewrite (6) as

$$\begin{aligned} th_i^* &\approx \frac{w_i B}{B/\widetilde{C}} = w_i \widetilde{C} \\ &= \begin{cases} R_{QoS,i} & \text{for QoS service,} \\ \left(\widetilde{C} - \sum_j^{N_{QoS}} R_{QoS,j}\right) \frac{1}{N_{BE}} & \text{for BE service.} \end{cases} \end{aligned} \tag{8}$$

Also, the total throughput in steady state becomes from (8)

$$\sum_{i}^{N_{QoS}+N_{BE}} th_i^* = \widetilde{C}. \tag{9}$$

Consequently, it is shown from (8) and (9) that the proposed CGW control mechanism assures the desired data rate for QoS service and allows BE services to fully share the channel in a fair manner. □

Next, we focus on the rate shaping for UDP flows. Whenever a new service is requested or terminated, the PN GW calculates the token rate (R_{token}) for each service as the value in the rightmost of (8) and broadcasts this value to the PNEs via `S-RSP` message. Receiving the `S-RSP` message, the UL PNE sets the token generation rate for UL UDP flow as the value received. Similarly, the PN GW sets the token generation rate for DL UDP flows. When a packet arrives, the corresponding amount of tokens are removed from the bucket and the packet is served. If the remaining token is less than the size of a new arriving packet, its service is deferred until the amount of tokens becomes larger than the packet size. In this way, the transmission rate of UDP flow can be regulated around the target rate and the proposed mechanism still works even when TCP and UDP flows coexist.

3.3 Admission control

Consider that a new QoS service requests the data rate of $R_{QoS,i}$ and there are N_{QoS} existing QoS services. The new QoS service will be admitted only if the following condition is satisfied,

$$R_{QoS,i} < \left(\alpha\widetilde{C} - \sum_{j}^{N_{QoS}} R_{QoS,j} \right) \tag{10}$$

where $\alpha(0 < \alpha < 1)$ is a control parameter.

The key point of the proposed admission control is estimating the capacity accurately, which is described below. Consider that a PNE sends or receives l-byte (excluding TCP/IP header) TCP data packet to or from the PN GW at the PHY rate of r Mb/s according to the channel access mechanism of IEEE 802.11 DCF. Let us define $T_{data}(r,l)$ and $T_{ack}(r)$ as the average time in the unit of μs required to successfully serve the TCP data packet and TCP ACK packet, respectively; they can be represented as

$$\begin{aligned} T_{data}(r,l) &= t_{DIFS} + \overline{t_{bo}} + t_{MSDU}(r,l) + t_{SIFS} + t_{MACK} \\ T_{ack}(r) &= t_{DIFS} + \overline{t_{bo}} + t_{MSDU}(r,0) + t_{SIFS} + t_{MACK}. \end{aligned} \tag{11}$$

In (11), t_{DIFS} and t_{SIFS} are DCF inter frame space (DIFS) and short inter frame space (SIFS), respectively, t_{MACK} is the time required to send MAC-layer ACK frame, which is constant, $\overline{t_{bo}}$ is the average backoff time before sending a MAC frame. Moreover, $t_{MSDU}(r,l)$ is the time required to send a MAC service data unit (MSDU) at the rate of r Mb/s, which can be given as;

$$t_{MSDU}(r,l) = t_{PHY} + \frac{(l+l_h)8}{r}, \tag{12}$$

where t_{PHY} is the time required to send PHY layer preamble and header and l_h is the size of TCP/IP/MAC header in byte. Note that r is selected among pre-defined rates (e.g., 11, 5.5, 2, and 1 Mb/s for 802.11b) depending the channel quality and the link adaptation mechanism adopted.

Next, we calculate $\overline{t_{bo}}$. Let us define p_c as the probability of packet collision in steady state, and $CTW(n)$ as the size of contention window under the condition that the transmission succeeds after n consecutive failures. On the detection of transmission failure, the contention window size is doubled according to the binary exponential backoff (BEB) mechanism of IEEE 802.11, it can increase up to the maximum value of CTW_{max} and returns to the minimum value of CTW_{min} if the transmission succeeds, i.e.,

$$CTW(n) = \begin{cases} 2^n CTW_{min} & \text{for } n \leq m, \\ CTW_{max} & \text{for } n > m. \end{cases} \tag{13}$$

where $m = log_2 (CTW_{max}/CTW_{min})$. Then, $\overline{CTW}$ becomes[3]

$$\begin{aligned} \overline{CTW} &= \sum_{n=0}^{m} p_c^n (1-p_c) 2^n CTW_{min} + \sum_{n=m+1}^{\infty} p_c^n (1-p) CTW_{max} \\ &= CTW_{min} \frac{1 - p_c - p_c (2p_c)^m}{1 - 2p_c}. \end{aligned} \tag{14}$$

On the other hand, the probability of packet collision, p_c can be represented with $\overline{CTW}$. According to the *p-persistent* model [17], which closely approximates the standard IEEE 802.11 protocol, each node makes a transmission attempt with the probability of $p_a = 2/\overline{CTW}$ in steady state. Then, the probability that a station collides with any of the other stations becomes

$$p_c = 1 - (1 - p_a)^{M-1} = 1 - \left(1 - 2/\overline{CTW}\right)^{M-1}, \tag{15}$$

where M is the number of active contending stations. We can get p_c and $\overline{CTW}$ by solving two equations (14) and (15) numerically [4]. A station selects a random backoff counter uniformly distributed between zero and $\overline{CTW}$-1. If a station decrements its backoff counter in the backoff stage, the other (M-1) active contending stations also decrement their own backoff counters. Therefore, the effective backoff time can be approximated as

$$\overline{t_{bo}} = \frac{\overline{CTW} - 1}{2} \frac{1}{M} t_{slot}, \tag{16}$$

where t_{slot} is the slot time. From (11) – (16), $T_{data}(r, l)$ and $T_{ack}(r)$ are obtained. Finally, we can get the capacity of WLAN, $C(r, l)$ (Mb/s), when stations send/receive l-byte TCP data packet at the rate of r Mb/s;

$$C(r, l) = \frac{8l}{T_{data}(r, l) + T_{ack}(r)}. \tag{17}$$

Table 2 lists several parameters of IEEE 802.11b used in the analysis model.

We can extend the analytical capacity model to the case where the link adaptation mechanism is applied so that the PHY-layer transmission rate changes depending on the channel quality. When the auto rate fallback (ARF) algorithm [18], which is the most common link adaptation algorithm, is used, the distribution of PHY-layer transmission rate can be obtained from the analysis result in [19]. The rate distribution can also be obtained by measuring signal to noise ratio (SNR) experimentally when a close-loop SNR-based link adaption algorithm is used. Let

[3]Here, we assume that the number of retransmissions is infinite even though it is limited to a certain value in the IEEE 802.11 standard. However, this assumption is not invalid since the probability that the number of retransmissions reaches the maximum value is very low.

[4]The simulation study in Section 4.1 shows that the actual capacity under the proposed mechanism is little deviated from the analytical capacity derived with the condition of $M = 1$, i.e., $p_c = 0$ from (15).

Table 2: Parameters of IEEE 802.11 in the analysis model.

parameter	value
slot time (t_{slot})	20 μs
SIFS (t_{SIFS})	10 μs
DIFS (t_{DIFS})	50 μs
ACK (t_{MACK})	248 μs
PHY header (t_{PHY})	192 μs
TCP/IP/MAC header (l_h)	74 byte
minimum contention window (CTW_{min})	32
maximum contention window (CTW_{max})	1024

us define $p(r_i)$ as the probability that the PHY-layer rate is $r_i \in \mathbb{R}$. The effective capacity can be obtained as

$$\widetilde{C}(l) = \sum_{r_i \in \mathbb{R}} C(r_i, l) p(r_i). \tag{18}$$

3.4 Discussion

Adjustment of the required data for admission control

The proposed admission control makes the decision on the admission of QoS service by comparing the required data rate and the available capacity, as shown in (10). However, the total capacity, $\widetilde{C}$, is not constant but depends on several factors such as packet size, transport-layer protocol, number of contending nodes, and channel quality. For example, the capacity of WLAN decreases as the packet size decreases, because the MAC/PHY overhead required to transmit a frame becomes relatively large as the packet size decreases. Recall that the analytical capacity in Section 3.3 is derived under the condition that the stations send/receive l-byte TCP packets. Also note that the maximum size of Ethernet frame is 1500 bytes and the size of VoIP packet is typically a few hundreds bytes, and that the data service is usually served with TCP while some real-time service may be served with UDP. We need to cope with the variation of capacity due to difference in the packet size, transport-layer protocol, and channel quality. For this purpose, we propose an approach of adjusting the required data rate of QoS service, instead of adjusting the total capacity.

Consider that a QoS service is requested with the required data rate of $R_{QoS,i}$ and the average TCP packet size of l byte, which is different from the standard packet size of l_o used in the capacity estimation. Also, consider that the transmission rate for the service is r Mb/s [5]. In order to deal with the capacity variation due to different packet size, the required data rate for the QoS service with the packet size of l is adjusted to the equivalent data rate, $R'_{QoS,i}$;

$$R'_{QoS,i} = \frac{\widetilde{C}(l_o)}{C(r,l)} R_{QoS,i}. \tag{19}$$

From (19), $R'_{QoS,i}$ increases as the packet size decreases to compensate for the decrease of capacity. Similarly, $R'_{QoS,i}$ decreases if the channel quality for the service is relatively good because the time required to send a packet with the higher transmission rate is smaller than that with the lower transmission rate, i.e., a station in good channel quality consumes less resource to serve a

[5]The transmission rate may change during service time due to varying channel quality. However, the channel quality is mostly depends on the path-loss, which is determined by the distance between transmitter and receiver; and thus the transmission rate does not change severely as long as the station does not move.

packet. Note that the adjusted rate requirement, $R'_{QoS,i}$ is used not only in the admission control (10), but also in the TCP window control (4).

In the same way, the required data rate for a QoS service with UDP can be adjusted. In contrast to TCP, UDP does not require transport-layer ACK transmission from the receiver, resulting in less overhead and higher capacity. The capacity with UDP traffic, denoted as $C_{udp}(r,l)$ can be estimated by replacing (17) with (20);

$$C_{udp}(r,l) = \frac{8l}{T_{data}(r,l)}. \tag{20}$$

Then, the equivalent required data rate for a UDP QoS service, $R'_{QoS,i}$, becomes

$$R'_{QoS,i} = \frac{\widetilde{C}(l_o)}{C_{udp}(r,l)} R_{QoS,i}. \tag{21}$$

Extension to support weighted fairness for BE services

The proposed scheme can be easily extended to support weighted fairness for BE services. Consider that a BE service has the corresponding service weight $w_{BE}(\geq 1)$, whose value is integer and determined by a service operator of CPNS depending on the relative service priority. Also, we consider the service weight is delivered to the PN GW in the `S-REQ` message. Then, the PN GW calculates the maximum congestion window (CGW_{max}) for TCP BE service and the token generation rate (R_{token}) for UDP BE service as

$$\begin{aligned} CGW_{max} &= \left(1 - \frac{\sum_j^{N_{QoS}} R_{QoS,j}}{\widetilde{C}}\right) \frac{B}{\sum_k^{N_{BE}} w_{BE,k}}, \\ R_{token} &= \left(\widetilde{C} - \sum_j^{N_{QoS}} R_{QoS,j}\right) \frac{1}{\sum_k^{N_{BE}} w_{BE,k}}. \end{aligned} \tag{22}$$

These values of CGW_{max} and R_{token} are delivered to PNEs in the `S-RSP` message. Receiving the `S-RSP` message, each PNE sets the congestion window size and token generation rate for its BE service that has the service weight of $w_{BE,k}$, each of which is denoted as $CGW'_{max,k}$ and $R'_{token,k}$, respectively, as;

$$\begin{aligned} CGW'_{max,k} &= w_{BE,k} CGW_{max}, \\ R'_{token,k} &= w_{BE,k} R_{token}. \end{aligned} \tag{23}$$

Note that $CGW'_{max,k} = CGW_{max}$ and $R'_{token,k} = R_{token}$ for a normal BE service with $w_{BE,k} = 1$. From (22) and (23), we observe that BE services share the buffer and capacity in a weighted manner and this extension for weighted service does not degrade QoS service since

$$\begin{aligned} \sum_k^{N_{BE}} CGW'_{max,k} &= \left(1 - \frac{\sum_j^{N_{QoS}} R_{QoS,j}}{\widetilde{C}}\right) B < B, \\ \sum_k^{N_{BE}} R'_{token,k} &= \widetilde{C} - \sum_j^{N_{QoS}} R_{QoS,j} < \widetilde{C}. \end{aligned} \tag{24}$$

Further service differentiation for QoS

The proposed mechanism can further differentiate service to enhance QoS. The 802.11e EDCA can be straightforwardly incorporated with the proposed mechanism so that the channel access for the QoS service can be made preferentially over the BE service. Moreover, such service differentiation can be simply realized without employing 802.11e EDCA. There is an interface queue between IP layer and MAC layer, which typically operates in a first-come-first-serve (FCFS) manner. By implementing dual queue, one for QoS service and the other for BE service, and adopting the priority scheduling algorithm for the dual queue, the packets of BE service is not served as long as there is the packets of QoS service waiting for the service. In this way, the delay for QoS service can be minimized.

4 Performance evaluation

In this section, we perform the extensive ns-2 simulations to evaluate the performance of the proposed QoS control mechanism in terms of QoS assurance and fair channel sharing. The simulation configuration is same to that in Section 2 and 802.11 MAC/PHY parameters used in the simulations are listed in Tab. 2.

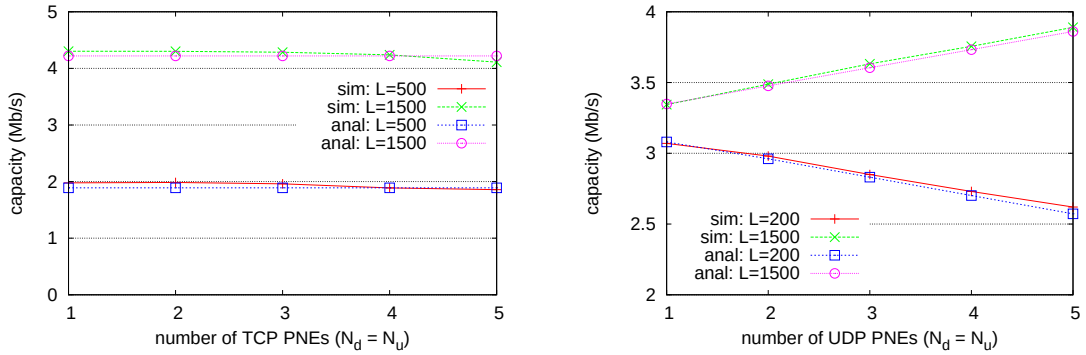
4.1 Validation for capacity estimation

First, we validate the methodology for capacity estimation by comparing the simulation results with the analysis results. The capacity is affected by the packet size or transport-layer protocol as described in Section 3.3, and the estimated capacity is used in the admission control and rate control; therefore the accurate estimation of capacity plays a key role in the proposed mechanism. We consider the following two scenarios;

- TCP-only: We observe the network capacity with different TCP packet size ($l = 500$ and 1500 bytes) and different number of UL and DL PNEs ($N_u = N_d = 1 \sim 5$).
- Mixed: In this scenario, each two PNEs upload and download 1 Kbyte TCP packets, while each N_{udp} PNEs upload and download UDP packets, respectively. The size and interval of UDP packet are set to 1500 bytes and 30 ms (i.e., 360 Kb/s), and 200 bytes and 20 ms (i.e., 32 Kb/s), each of which is set to consider the real-time video and audio service for mobile device, respectively.

Fig. 5 compares the capacity obtained from simulation with that obtained from analysis, for the above two cases. From Fig. 5(a), we observe that the simulation result is almost immune to the number of PNEs. This result seems not to agree with the well-known WLAN capacity model [17], [20], where the capacity mostly decreases as the number of stations increases due to the increased collision probability. These previous models make an assumption of saturated traffic, i.e., every station always has packets to send and it always participates in the MAC-layer contention. However, this explanation does not hold in our mechanism because the admission control along with the rate control assures the sum of transmission rates for all the service is regulated not to exceed the available capacity so that every station cannot always take part in the MAC-layer contention. Note that the analysis result shown in Fig. 5 is derived under the condition that the number of active contending stations is one, i.e., $M = 1$ and $p_c = 0$ accordingly from (15), resulting in the constant capacity, regardless of N_d and N_u. On the other hand, as was expected, the capacity decreases with the smaller size of packet. However, Fig. 5(b) shows that the capacity linearly increases or decreases with respect to the number of PNEs with UDP traffic in the mixed traffic case. From the analysis in Section 3.3, the estimated capacity in the

case of 1500-byte and 200-byte UDP packet is 5.00 and 1.57 Mb/s, respectively, and that in the case of 1-Kbyte TCP packet is 3.22 Mb/s.



(a) TCP-only (b) Mixed

Figure 5: Comparison of capacity obtained from simulation and capacity derived from analysis.

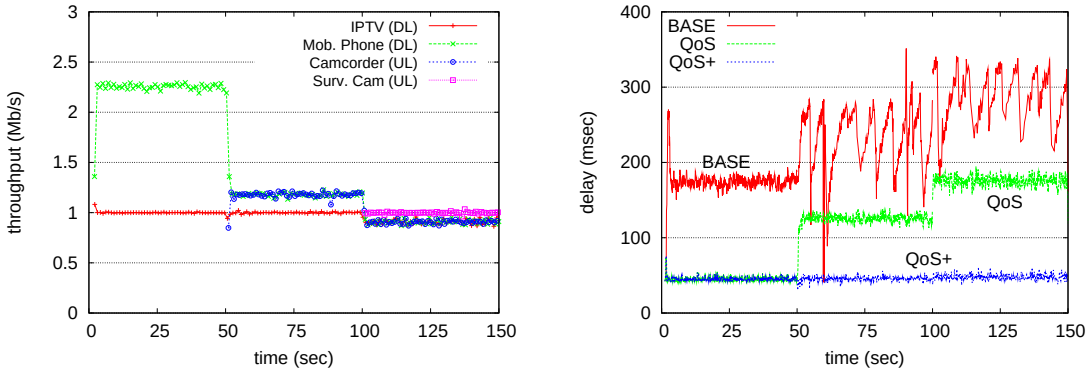
Fig. 5 shows that there is little difference in the simulation results and analysis results for both cases. It is important to note that the capacity ($\widetilde{C}$) with the standard packet size of 1KB is used as decision criteria of admission control (see (10)), and the required data rate is scaled if the packet size or transport-layer protocol is different from those used to estimate the capacity as shown in (19) and (21). The results in Fig. 5 confirm that the estimated capacity is accurate, regardless of packet size and/or transport-layer protocol, and that the capacity can be easily estimated under the assumption of negligible collision probability, regardless of the number of stations. Hereafter, we set the packet size of QoS and BE service as 1 Kbyte to concentrate on evaluating performance without the effect of packet size.

4.2 QoS assurance and fair channel sharing

The objective of this simulation is to evaluate the performance of the proposed mechanism in terms of QoS assurance and fair channel sharing. Figure 6(a) shows per-PNE throughput under the simulation scenario in Section 2.1 (see Tab. 1). By comparing the case without QoS control (see Fig. 2), we observe that the proposed mechanism strictly assures the desired data rate for QoS services, e.g, 1Mb/s for IPTV (DL) and 1Mb/s for surveillance camera (UL), and the assurance is hardly affected by other services. The proposed mechanism completely removes the unfairness between DL and UL BE services; the throughputs of the DL mobile phone and UL camcorder are almost equal during the whole simulation time when both coexist. Moreover, the severe fluctuation in the throughput is significantly reduced, compared to the case without QoS control. Table 3 compares the per-PNE throughput with the proposed QoS control and that without QoS control. During t = [50-100]s and [100-150]s, the fairness index is increased from 0.53 and 0.45 to 1.01 and 1.00, respectively, due to the proposed QoS control mechanism. Also, the total throughput of the proposed mechanism is not decreased to assure QoS and fairness, rather it is slightly increased in some cases, compared to the case without QoS control.

Next, we focus on the delay of streaming service (IPTV) for three cases;

- `BASE`: No QoS control mechanism is implemented.
- `QoS`: The proposed QoS control mechanism is implemented and the PN GW serves packets with the FCFS scheduling discipline.



(a) Per-PNE throughput

(b) Delay of streaming service (IPTV)

Figure 6: Throughput and delay with the proposed QoS control mechanism for the simulation scenario in Tab. 1.

Table 3: Comparison of per-PNE throughput in each time interval.

interval (sec)	device	direction	throughput (Mb/s)	
			w/o control	with control
0 ∼ 50	IPTV	DL	1.00	1.00
	Mob. phone	DL	2.25	2.24
50 ∼ 100	IPTV	DL	0.78	1.00
	Mob. phone	DL	0.87	1.18
	Camcorder	UL	1.63	1.17
100 ∼ 150	IPTV	DL	0.51	1.00
	Mob. phone	DL	0.71	0.91
	Camcorder	UL	1.58	0.91
	Surv. Cam	UL	1.00	1.00

- `QOS+`: This enhances the proposed QoS control mechanism by implementing dual queue in the PN GW and applying the strict priority scheduling discipline to the dual queue, as discussed in Section 3.4.

As shown in Fig. 6(b), the delay of `BASE` is higher than that of `QoS+` up to 7.5 times and it oscillates highly after a UL service starts at $t = 50$ s, while the delay of `QoS+` is maintained at the smallest value (about 45 ms) and it does not increase due to other services. In the case of `QoS`, the variation of delay is small as `QoS+`, but the delay increases once other service starts; the average delay is about 45, 125, and 170 ms, in the time interval of [0-50]s, [50-100]s, and [100-150]s, respectively. We also measure the jitter of streaming service, defined as standard deviation of delay [6]. In the case of `BASE`, the jitter increases from 16 ms to 77 ms in the time interval of [0-50]s and [100-150]s, respectively. The jitter of `QoS` during t = [100-150]s increases from 3.64 ms to 7.13m by about 2 times, compared to the time interval of t = [0-50]s. However, as can be expected from 6(b), the jitter of `QoS+` is not affected by other service and is maintained below 3.9 ms for the whole simulation time, reconfirming that the proposed mechanism can minimize delay of QoS service, as well as assuring its desired data rate.

[6]The jitter can be defined in several ways to quantify the variation in the delay of successive packets. It is an important performance index for streaming service, as well as delay.

Table 4: Fairness and capacity with different number of devices.

(N_u, N_d)	(1,1)	(1,2)	(1,4)	(2,1)	(2,2)	(2,4)	(4,1)	(4,2)	(4,4)
fairness index	0.987	0.988	1.004	1.000	1.014	1.000	0.992	0.999	0.982
capacity (Mb/s)	3.27	3.35	3.34	3.26	3.32	3.34	3.31	3.26	3.27

4.3 Fairness with different number of devices, weights, and transmission rates

Here, we focus on evaluating the performance of the proposed mechanism from the viewpoints of fairness and capacity for TCP BE services. Table 4 shows fairness index and total throughput with different number of UL/DL devices, N_u and N_d. The fairness index is not much deviated from the ideal value of one, it ranges between 0.982 and 1.014 in all the cases of N_d and N_u. Also, the capacity is little affected by the number of devices, it ranges between 3.26 Mb/s and 3.35 Mb/s and close to the analysis result of 3.23 Mb/s. These results validate that the proposed algorithm assures fair channel sharing among BE services without debasing the total throughput, regardless of the number of devices.

The next simulation evaluates how the proposed mechanism can support the weighted fairness for BE services, which is discussed in Section 3.4. Figure 7 shows throughput for six PNEs (three DL PNEs and another three UL PNEs) that have different service weights of 1, 2, or 4. We observe that the throughput is almost in proportion to the service weight, the throughput of PNE5/PNE6 is higher than that of PNE1/PNE2 by 3.96 times and that of PNE3/PNE4 is higher than that of PNE1/PNE2 by 1.97 times. Moreover, there is no notable difference between throughput of DL PNE and UL PNE that have the same service weight. Since the proposed mechanism assures weighted fairness, the CPNS operator can assign different service weight based on the service priority, service fee, or operational policy, to increase its revenue.

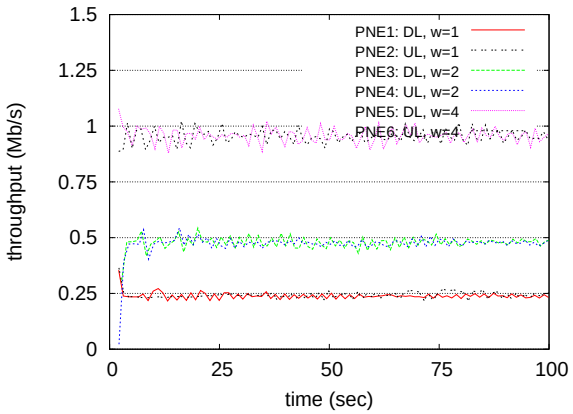


Figure 7: Per-PNE throughput with different service weights.

Finally, we observe the performance of fairness in multi-rate WLAN configuration. Table 5 shows per-PNE throughput, total throughput, and fairness index, as well as indicating the direction and transmission rate (`HI` and `LO` denote the transmission rate of 11 Mb/s and 2 Mb/s, respectively.). The difference in the per-PNE throughput is insignificant, i.e., the fairness index is almost one, regardless of transmission direction and transmission rate. However, the total throughput decreases as the number of low-rate PNEs increases, because the low-rate PNE occupies the channel longer than the high-rate PNE to transmit a packet. The proposed mechanism assures *throughput-fairness*, regardless of transmission rate of PNEs. Moreover, it is expected from the results of Fig. 7 and Tab. 5 that the proposed mechanism can be extended

Table 5: Fairness and capacity with different transmission rates.

Per-PNE throughput (Mb/s)				capacity	fairness
DL: HI	UL: HI	DL: LO	UL: LO	(Mb/s)	index
1.626	1.645	-	-	3.271	0.988
0.863	0.880	0.871	-	2.614	0.985
0.574	0.582	0.561	0.558	2.275	0.996

to support *time-fairness*, i.e., allocate channel occupation time fairly, by assigning high weight to the high-rate PNE, to improve the total throughput without impairing fairness.

5 Conclusion

In this paper, we proposed a generalized framework to assure QoS and fairness for WLAN-based CPNS. We identified the problem of QoS degradation and unfair channel sharing that occurs in the existing WLAN-based CPNS, and analyzed its cause from the viewpoints of unfair channel access of MAC and asymmetric congestion control of TCP. By employing admission control and rate control, the proposed mechanism provides absolute guarantee for QoS service in terms of data rate, at the same time, it provides weighted fairness for BE service without decreasing the efficiency of channel sharing. Moreover, the assurance of QoS and fairness is effective, regardless of service direction, packet size, transmission rate, and transport-layer protocol. Through analysis and simulation, it was confirmed that the proposed mechanism effectively achieve QoS assurance for QoS service and fair channel access for BE service.

Acknowledgement

This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (KRF), funded by the Ministry of Education, Science, and Technology [Grant No. 2010-0008711], and by Information Technology Research Center (ITRC) support program through the National IT Industry Promotion Agency (NIPA), funded by the Ministry of Knowledge Economy, Korea [Grant No. NIPA-2011-C1090-1121-0005]. Corresponding author is G.-M. Jeong.

Bibliography

[1] Open Mobile Alliance, Converged Personal Network Service Architecture Candidate Version 1.0, OMA-AD-CPNS-V1_0-20100615-C, Jun 2010.

[2] Open Mobile Alliance, Converged Personal Network Service Requirements, Candidate Version 1.0, OMA-RD-CPNS-V1_0-20091117-C, Nov. 2009.

[3] IEEE 802.11 Working Group, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Std 802.11-2007, June 2007.

[4] The Network Simulator, ns-2, http://www.isi.edu/nsnam/ns/

[5] IEEE 802.11 Working Group, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification, Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements IEEE Std 802.11e-2005, Nov. 2005.

[6] I. Aad and C. Castelluccia, Differentiation Mechanisms for IEEE 802.11, in *Proc. of IEEE INFOCOM*, 209-218, 2001.

[7] Q. Ni, L. Romdhani, and T. Turletti, A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN, *Wiley Journal of Wireless Communication and Mobile Computing*, 4(5):547-566, 2004.

[8] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, A survey of Quality of Service in IEEE 802.11 Networks, *IEEE Wireless Communications*, 11(4):6-14, 2004.

[9] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, Achieving MAC Layer Fairness in Wireless Packet Networks, in *Proc. of ACM MobiCom*, 87-98, 2000.

[10] D. Qiao and K.G. Shin, Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802 WLAN under the DCF, in *Proc. of Int. Workshop Quality of Service (IWQoS)*, 227-236, 2002.

[11] M. Bottigliengo, C. Casetti, C.-F. Chiasserini, and M. Meo, Smart Traffic Scheduling in 802.11 WLANs with Access Point, in *Proc. IEEE Vehicular Technology Conf.-Fall (VTC03)*, 2227-2231, 2003.

[12] S. Kim, B.-S. Kim, and Y. Fang, Downlink and Uplink Resource Allocation in IEEE 802.11 Wireless LANs, *IEEE Trans. Vehicular Technology*, vol. 54, no. 1, pp. 320-327, Jan. 2005.

[13] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, Understanding TCP Fairness over Wireless LAN, in *Proc. of IEEE INFOCOM*, pp. 863-872, 2003.

[14] Y. Wu, Z. Niu, and J. Zheng, Study of the TCP Upstream/Downstream Unfairness Issue with Per-Flow Queuing over Infrastructure-Mode WLANs, *Wireless Comm. and Mobile Computing*, vol. 5, no. 4, pp. 459-471, 2005.

[15] D.J. Leith, P. Clifford, D. Malone, and A. Ng, TCP Fairness in 802.11e WLANs, *IEEE Communications Letters*, vol. 9, no. 11, pp. 964-966, 2005.

[16] S. Choi, K. Park, and C. Kim, Performance Impact of Inter-Layer Dependence in Infrastructure WLANs, *IEEE Trans. Mobile Computing*, vol. 5, no. 7, pp. 829-845, 2006.

[17] F. Calì, M. Conti, E. Gregori, Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit, *IEEE Trans. on Networking*, vol. 8, no. 6, pp. 785-799, Dec. 2000.

[18] A. Kamerman and L. Monteban, WaveLAN-II: A High-performance wireless LAN for the unlicensed band, *Bell Lab Technical Journal*, pp. 118-133, 1997.

[19] J. Choi, K. Park, and C.-K. Kim, Analysis of Cross-Layer Interaction in Multirate 802.11 WLANs, *IEEE Trans. on Mobile Computing*, vol. 8, no. 5, pp. 682-693, May 2009.

[20] G. Bianchi, Performance Analysis of the IEEE 802.11 Distributed Coordination Function, *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, March 2000.

Sensitive Ants in Solving the Generalized Vehicle Routing Problem

C.-M.Pintea, C.Chira, D.Dumitrescu, P.C. Pop

Camelia-M. Pintea
G. Cosbuc N.College
Romania, A. Iancu, 400083, Cluj-Napoca
E-mail: cmpintea@yahoo.com

Camelia Chira, D.Dumitrescu
"Babes-Bolyai" University
Romania, M. Kogalniceanu, 400084, Cluj-Napoca
E-mail: cchira@cs.ubbcluj.ro, ddumitr@cs.ubbcluj.ro

Petrică C. Pop
North University of Baia Mare
Romania, V. Babeş , 430083, Baia Mare
E-mail: pop_petrica@yahoo.com

Abstract: The idea of sensitivity in ant colony systems has been exploited in hybrid ant-based models with promising results for many combinatorial optimization problems. Heterogeneity is induced in the ant population by endowing individual ants with a certain level of sensitivity to the pheromone trail. The variable pheromone sensitivity within the same population of ants can potentially intensify the search while in the same time inducing diversity for the exploration of the environment. The performance of sensitive ant models is investigated for solving the generalized vehicle routing problem. Numerical results and comparisons are discussed and analysed with a focus on emphasizing any particular aspects and potential benefits related to hybrid ant-based models.
Keywords: ant-based models, optimization, sensitivity, complex problems

1 Introduction

The potential of ant-based models [2,3,9] in solving difficult optimization problems has been well emphasized by successful results obtained in many and varied fields including transportation optimization, quadratic assignment, scheduling, vehicle routing and protein folding. Inspired by the real-world collective behaviour of social insects, *Ant Colony System (ACS)* algorithms [2] rely on the stigmergic interactions between many identical artificial ants to find solutions to a given problem. Each ant generates a complete tour (associated to a problem solution) by probabilistically choosing the next node at each path intersection based on the cost and the amount of pheromone on the connecting edge. Stronger pheromone trails are preferred and the most promising tours build up higher amounts of pheromone in time.

Inducing heterogeneity in the population by enabling each artificial ant to react in a different way to the same environment [10] represents a promising approach to the application of ant-based models for solving complex real-world problems possibly with a dynamic character. Each individual ant can be endowed with a certain level of sensitivity to the pheromone trail triggering various types of reactions to a changing environment. The variable pheromone sensitivity within the same population of ants can potentially intensify the search (normally through high sensitivity levels) while in the same time inducing diversity for the exploration of the environment. The

decision of a low-level sensitive ant regarding the action to be performed crucially contributes to the quality of the search process and solutions.

The Generalized Vehicle Routing Problem (GVRP) is an extension of the Vehicle Routing Problem (VRP) and was introduced by Ghiani and Improta [6]. The GVRP is the problem of designing optimal delivery or collection routes, subject to capacity restrictions, from a given depot to a number of predefined, mutually exclusive and exhaustive node-sets (clusters).

The GVRP belongs to the class of generalized combinatorial optimization problems, which are natural extensions of combinatorial optimization problems by considering a related problem relative to a given partition of the nodes of the graph into node sets, while the feasibility constraints are expressed in terms of the clusters. In the literature we can find several generalized problems such as the generalized minimum spanning tree problem (see [13]), the generalized traveling salesman problem, the generalized vehicle routing problem, the generalized (subset) assignment problem, etc. These generalized problems belong to the class of NP-complete problems, are harder than the classical ones and nowadays are intensively studied due to the interesting properties and applications in the real world, even though many practitioners are reluctant to use them for practical modeling problems because of the complexity of finding optimal or near-optimal solutions.

Ghiani and Improta [6] showed that the problem can be transformed into a capacitated arc routing problem (CARP) and Baldacci et al. [1] proved that the reverse transformation is valid. Recently, Pop [12] provided a new efficient transformation of the GVRP into the classical vehicle routing problem (VRP). As far as we know, the only specific algorithm for solving the GVRP was developed by Pop et al. [11] and was based on ant colony optimization.

The aim of this paper is to investigate the performance of the *Sensitive Ant Model (SAM)* [10] in solving the *Generalized Vehicle Routing Problem (GVRP)* [6]. We report numerical results of the *SAM* model for several *GVRP* benchmark problems and discuss the performance of *SAM* compared to the standard *ACS* technique.

2 Definition and Complexity of the GVRP

Let $G = (V, A)$ be a directed graph with $V = \{0, 1, 2, ..., n\}$ as the set of vertices and the set of arcs $A = \{(i, j) \mid i, j \in V, i \neq j\}$. A nonnegative cost c_{ij} associated with each arc $(i, j) \in A$. The set of vertices (nodes) is partitioned into $k + 1$ mutually exclusive nonempty subsets, called clusters, $V_0, V_1, ..., V_k$ (i.e. $V = V_0 \cup V_1 \cup ... \cup V_k$ and $V_l \cap V_p = \emptyset$ for all $l, p \in \{0, 1, ..., k\}$ and $l \neq p$). The cluster V_0 has only one vertex 0, which represents the depot, and remaining n nodes belonging to the remaining k clusters represent geographically dispersed customers. Each customer has a certain amount of demand and the total demand of each cluster can be satisfied via any of its nodes. There exist m identical vehicles, each with a capacity Q.

The generalized vehicle routing problem (GVRP) consists in finding the minimum total cost tours of starting and ending at the depot, such that each cluster should be visited exactly once, the entering and leaving nodes of each cluster is the same and the sum of all the demands of any tour (route) does not exceed the capacity of the vehicle Q. An illustrative scheme of the GVRP and a feasible tour is shown in the Figure 1.

The GVRP reduces to the classical Vehicle Routing Problem (VRP) when all the clusters are singletons and to the Generalized Traveling Salesman Problem (GTSP) when $m = 1$ and $Q = \infty$.

The GVRP is NP-hard because it includes the generalized traveling salesman problem as a special case when $m = 1$ and $Q = \infty$.

Several real-world situations can be modeled as a GVRP. The post-box collection problem

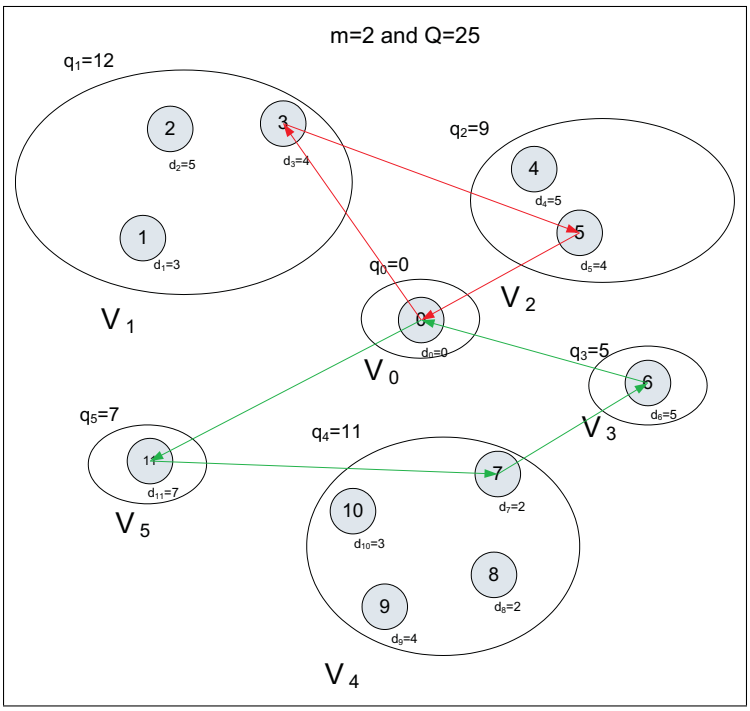


Figure 1: An example of a feasible solution of the GVRP

described in Laporte et al. [7] becomes an asymmetric GVRP if more than one vehicle is required. Furthermore, the GVRP is able to model the distribution of goods by sea to a number of customers situated in an archipelago as in Philippines, New Zeeland, Indonesia, Italy, Greece and Croatia. In this application, a number of potential harbours is selected for every island and a fleet of ships is required to visit exactly one harbour for every island.

Several applications of the GTSP (Laporte et al. [8]) may be extended naturally to GVRP. In addition, several other situations can be modeled as a GVRP, these include:

- the Traveling Salesman Problem (TSP) with profits (Feillet et al. [4]);
- a number of Vehicle Routing Problem (VRP) extensions: the VRP with selective backhauls, the covering VRP, the periodic VRP, the capacitated general windy routing problem, etc.;
- the design of tandem configurations for automated guided vehicles (Baldacci et al. [1]).

3 The ACS algorithm for solving GVRP

The ACS-based algorithm for GVRP [11] uses artificial ants in order to construct vehicle routes by successively choosing exactly one node from each cluster. This task continues until each cluster has been visited. Whenever the choice of another node from a cluster would lead to an infeasible solution because of vehicles capacity, the depot is chosen and a new route is started.

Initially, the Nearest Neighbor (NN) algorithm - with the rule always go to the nearest as-yet-unvisited location - is considered. The best solution of Nearest Neighbor (NN) algorithm (L^+) is used for ACS-based algorithm start.

The number of ants corresponds to the number of GVRP customers m. At the beginning of an iteration, an ant is placed at each node (customer). After initializing the basic ant system

algorithm, the two steps: (i) construction of vehicle routes and (ii) trail update are repeated for a given number of iterations.

To favor the selection of an edge with a high pheromone level and high visibility, a probability function p_k^{ij} is defined as follows:

$$p_k^{ij}(t) = \frac{\tau_{ij}^k(t)[\eta_{ij}^k(t)]^\beta}{\sum_{o \in J_i^k} \tau_{io}^k(t)[\eta_{io}^k(t)]^\beta} \tag{1}$$

where J_i^k is the set of unvisited neighbors of node i by ant k, $j \in J_i^k$ and β is a parameter used for tuning the relative importance of visibility. After an artificial ant has constructed a feasible solution, the pheromone trails are laid depending on the objective value L_k. For each edge that was used by ant k, the pheromone trail is updated according to the following rule:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\frac{1}{L^k} \tag{2}$$

where $\rho \in (0,1)$ is an evaporation rate parameter.

A tabu list prevents ants visiting clusters they have previously visited. The ant tabu list is cleared after each completed tour.

The global update rule, applied by the elitist ants, as in ACS [3] is:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\frac{1}{L^+}, \tag{3}$$

where L^+ is the so far best solution.

4 Sensitive Ant-based Model for GVRP

The *Sensitive Ant Model (SAM)* technique proposed in [10] is engaged for solving the *GVRP*. The general approach to solving GVRP using SAM is the same with the ACS approach presented in the previous section except that the transition probabilities defined by SAM are used. The initialization of the algorithm, the update rules and the maintainance of a tabu list are kept the same in SAM for GVRP.

The *SAM* algorithm involves several ants able to communicate in a stigmergic manner (influenced by pheromone trails) for solving complex search problems. Within the *SAM* model, each ant is characterized by a pheromone sensitivity level (*PSL*). The *PSL* value is expressed by a real number in the unit interval [0, 1]. When *PSL* is null the ant completely ignores stigmergic information and when *PSL* is one the agent has maximum pheromone sensitivity. The ants with a low *PSL* value are more independent and are considered environment explorers. They have the potential to autonomously discover new promising regions of the solution space. The ants with high *PSL* values are very sensitive to pheromone traces. They are influenced by stigmergic information and therefore intensively exploit the promising search regions already identified.

SAM introduces a measure of randomness proportional to the level of individual *PSL* in the decisions of ants regarding the path to follow. This is achieved by modifying the transition probabilities using the *PSL* values in a renormalization process [10]. The SAM renormalized transition probability for ant k (influenced by *PSL*) is denoted by $sp_k^{ij}(t)$ and is given by the following equation:

$$sp_k^{ij}(t) = p_k^{ij}(t) \cdot PSL_k(t), \tag{4}$$

where $p_k^{ij}(t)$ is the probability for ant k to choose the next node j from current node i (as given in *ACS* - see Equation 1) and $PSL_k(t)$ represents the *PSL* value of ant k at time t.

Table 1: Problem characteristics for the ant-based algorithms for *GVRP*

Problem	*VR*	*Q*	*Q'*	*No.vehicles*	*No.Routes*
11eil51	2	160	320	6	3
16eil76A	2	140	280	10	5
16eil76B	3	100	300	15	5
16eil76C	2	180	360	8	4
16eil76D	2	220	440	6	3
21eil101A	2	200	400	8	4
21eil101B	2	112	224	14	7

It can be noticed that the SAM probability of selecting the next node is the same with the ACS one when PSL value is one. In order to associate a standard probability distribution to the system, the SAM virtual state corresponding to the 'lost' probability of $(1 - PSL_k(t))$ has to be defined. The associated virtual state decision rule specifies the action to be taken when the virtual state is selected using the renormalized transition mechanism. The following rule is used in the current paper: the ant randomly chooses an available node with uniform probability if the virtual state is selected. This approach favors the increasing of randomness in the selection process with the decreasing of sensitivity level to pheromone.

5 Computational results

The performance of the SAM and ACS for solving *GVRP* is investigated. Numerical experiments focus on seven benchmark problems from the *TSPLIB* library [14]. These problems contain between 51 and 101 customers (nodes), which are partitioned into a given number of clusters, and in addition the depot.

Originally the set of nodes in these problems is not divided into clusters. The CLUSTERING procedure proposed by Fischetti *et al.* [5] is used to divide data into node-sets. This procedure sets the number of clusters $m = [\frac{n}{5}]$, identifies the m farthest nodes from each other and assigns each remaining node to its nearest center.

Table 1 contains the description of the GVRP instances addressed in this paper.

The meaning associated with the columns in Table 1 is as follows:

- Problem: The name of the test problem contains the number of clusters (first digits in the problem name) and the number of nodes (last digits in the problem name).
- VR: The minimal number of vehicles needed for a route in order to cover even the largest capacity of a cluster (VR=Vehicles/Route)
- Q': the capacity $Q \cdot VR$, where Q is the capacity of a vehicle available at the depot.

The same parameter setting was used in both SAM and ACS algorithms in order to allow a meaningful direct comparison: $\tau_0 = 0.1$ (the initial value of all pheromone trails), $\alpha = 1$, $\beta = 5$, $\rho = 0.0001$ and $q_0 = 0.5$. In the SAM algorithm, the PSL value is randomly generated between 0 and 1 for each ant.

Numerical results indicate a competitive performance of the SAM algorithm. Figure 2 presents SAM results from 20 succesive runs for the considered problem instances.

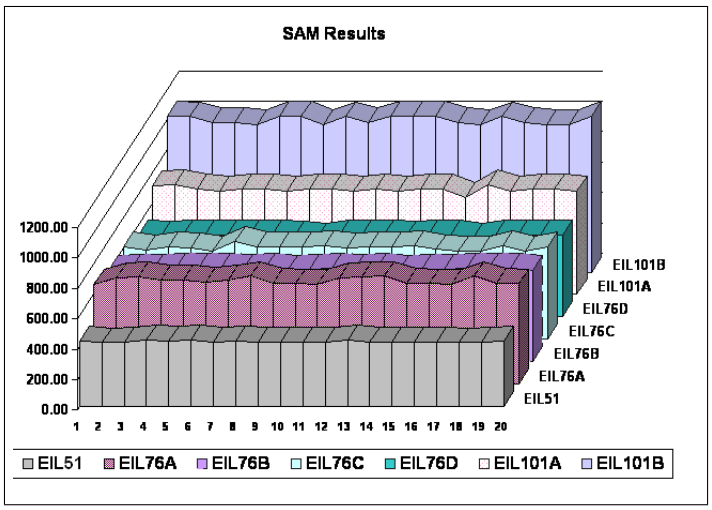


Figure 2: SAM results from 20 runs for the considered problem instances [14]

Table 2: Best Values and Times - *ACS* and *SAM* algorithms for solving *GVRP*

Problem	*ACS*	*Time ACS*	*SAM*	*Time SAM*
11eil51	418.85	**212**	**418.21**	297
16eil76A	668.78	**18**	**651.98**	25
16eil76B	625.83	**64**	**599.23**	166
16eil76C	**553.21**	215.00	577.49	**88**
16eil76D	**508.81**	177.00	515.64	**120**
21eil101A	**634.74**	**72**	**634.74**	111
21eil101B	**875.58**	**8.00**	966.17	52

Tables 2 and 3 present comparative numerical results obtained in 20 runs. The performance of SAM in solving GVRP is compared to that of ACS [11].

The following information is contained in Tables 2 and 3:

- Best length: the minimal length of collection routes;
- Best time: the time of the minimal collection routes;
- Avg. length: the average length for 20 runs;
- Avg. time: the average time (in seconds) for 20 runs.

The computational values are the result of the average of 20 successively runs of both algorithms. Termination criterion is either the maximum number of iterations, $N_{iter} = 250000$ or the maximum running time (five minutes) on a AMD 2600, 1.9Ghz and 1024 MB.

When comparing the best solution reported in 20 runs, the performance of the ACS and SAM algorithms are similar. SAM reports better values for three out seven problems while ACS does better for other three problems. It should be noticed that whenever a method is able to obtain a better solution, it also reports a longer running time compared to the other one.

The average solutions obtained by SAM are clearly improved compared to ACS although the running time has slightly increased for some of the GVRP instances. SAM detects a better average solution (calculated based on the 25 runs) for six out of the seven benchmark problems.

Table 3: Average Values and Times - *ACS* and *SAM* algorithms for solving *GVRP*

Problem	*ACS*	*Time ACS*	*SAM*	*Time SAM*
11eil51	429.85	210.20	**424.05**	**96.40**
16eil76A	706.09	**109.20**	**677.85**	187.60
16eil76B	684.04	**50.7**	**608.62**	173.7
16eil76C	625.87	73.00	**602.06**	**42.25**
16eil76D	566.56	**93.20**	**533.12**	223.45
21eil101A	699.46	**29.00**	**690.39**	124.30
21eil101B	**996.41**	**25.95**	998.71	27.90

Overall better average SAM results are facilitated by a better exploration of the search space and exploitation of new solutions. This is due to the variable sensitivity induced in SAM via random individual PSL values.

6 Conclusions

Sensitive heterogeneous ant-based models facilitate a balanced search process by endowing ants with different pheromone senitivity levels translated into different search strategies. An effective exploration of the search space is performed particularly by ants having low pheromone sensitivity while the exploitation of intermediary solutions is facilitated by highly-sensitive ants.

The performance of hybrid ant-based models is investigated with succesful results for solving the NP-hard Generalized Vehicle Routing Problem. Variable pheromone sensitivity in ant-based models proves to be benefic to the search process leading to better results compared to the ant colony system algorithm. Numerical results encourage the exploration of new ways to induce heterogeneity in ant-based models.

Acknowledgments. This work was supported by CNCSIS-UEFISCSU, project number PNII-IDEI 508/2007 *New Computational Paradigmes for Dynamic Complex Problems.*

Bibliography

[1] R. Baldacci, E. Bartolini and G. Laporte, *Some applications of the Generalized Vehicle Routing Problem*, Le Cahiers du GERAD, G-2008-82, 2008.

[2] M. Dorigo, C. Blum, *Ant Colony Optimization Theory: A Survey*, Theoretical Computer Science, 344, 2-3, pp. 243-278, 2005.

[3] M. Dorigo, L. M. Gambardella. *Ant Colony System: A cooperative learning approach to the Traveling Salesman Problem. IEEE Trans. Evol. Comp.*, 1, pp.53-66, 1997.

[4] D. Feillet, P. Dejax and M. Gendreau, *Traveling salesman problems with profits*, Transportation Science, Vol. 39, pp. 188-205, 2005.

[5] M. Fischetti, J.J. Salazar, P. Toth. *A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Operations Research*, 45:378-394, 1997.

[6] G. Ghiani, G. Improta. *An efficient transformation of the generalized vehicle routing problem. European Journal of Operational Research* 122, 11-17, 2000.

[7] G. Laporte, S. Chapleau, P.E. Landry and H. Mercure, *An algorithm for the design of mail box collection routes in urban areas*, Transportation Research B, Vol. 23, pp. 271-280, 1989.

[8] G. Laporte, A. Asef-Vaziri and C. Sriskandarajah, *Some applications of the generalized traveling salesman problem*, Journal of Operational Research Society, Vol. 47, pp. 1461-1467, 1996.

[9] C.M. Pintea, D. Dumitrescu and P.C. Pop,*Combining heuristics and modifying local information to guide ant-based search*, Carpathian Journal of Mathematics, Vol. 24, No. 1, pp. 94-103, 2008.

[10] C-M. Pintea, C. Chira, D.Dumitrescu, *Sensitive Ants: Inducing Diversity in the Colony*, Nature Inspired Cooperative Strategies for Optimization NICSO 2008, Studies in Computational Intelligence, 236 (Krasnogor, N.; Meliďż˝n-Batista, B.; Moreno-Pďż˝rez, J.A.; Moreno-Vega, J.M.; Pelta, D.; Eds.), Springer-Verlag, 15-24, 2009.

[11] P.C.Pop, C.M.Pintea, I.Zelina, D.Dumitrescu. *Solving the Generalized Vehicle Routing Problem with an ACS-based Algorithm*, American Institute of Physics (AIP) Springer, Conference Proceedings: BICS 2008, Vol.1117, No.1, 157–162, 2009.

[12] P.C. Pop, Efficient Transformations of the Generalized Combinatorial Optimization Problems into Classical Variants, Proceedings of the 9-th Balkan Conference on Operational Research, Constanta, Romania, 2-6 September 2009.

[13] P.C. Pop, *A survey of different integer programming formulations of the generalized minimum spanning tree problem*, Carpathian Journal of Mathematics, Vol. 25, No. 1, pp. 104-118, 2009.

[14] http://www.iwr.uni-heidelberg.de/groups/comopt/software/ TSPLIB95/vrp/

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 739-748

Computing by Folding

D. Sburlan

Dragoş Sburlan
Faculty of Mathematics and Informatics
Ovidius University of Constanta, Romania
E-mail: dsburlan@univ-ovidius.ro

Abstract: The present paper introduces a new computing paradigm based on the idea of string folding. Comparisons between the computational power of the proposed model with the classical families of languages from the Chomsky hierarchy are studied. Some preliminary results are reported and some conjectures are discussed. In this respect, the proposed model is promising not only because of the expected theoretical results, but also because of the possible indirect applications in various fields (as for instance, mathematical linguistics, DNA computing, computing using light, and so on).

1 Introduction

The computing paradigm proposed in this paper has as motivation several completely different domains that naturally converge. Whether we talk about the traditional Japanese art of paper folding (Origami), or about some in-vivo biochemical processes happening at the cellular level (for instance, the protein folding that produces particular 3-dimensional structures which are essential for their "correct" functioning), or about the in-vitro folding of DNA (by which one can arbitrary build two and three dimensional shapes at the nanoscale), a common feature is the folding of the physical support. Aiming to design novel computational techniques, the present paper introduces a new model of computation which can be easily described by the controlled folding of the computational support.

In order to be more eloquent, one may consider the Origami perspective. Assume for the sake of argument that the paper used in Origami is actually a bi-dimensional array of symbols. Hence the paper is not only the physical support for building two or three-dimensional shapes and forms but also it might be considered as the support for computation. In this simple framework a fold of the paper corresponds to a step of computation. Even if in this informal presentation we are not interested to precisely define the outcome of performing such steps of computation, we only want to point out that, for instance, a simple fold along the vertical middle of a page corresponds actually to the simultaneous movement of symbols from one side to another. In this case, if to each symbol one associates a position in the two/three dimensional space, then this folding represents, in a certain sense, a parallel rewriting of symbols. More specifically, this simple procedure moves (rewrites) a lot of space in just one computational step. This is just one argument which indicates that, under certain conditions, the expected computational power for such systems exceeds the semi-linearity (see [1] for an operation that recalls in a certain sense the folding).

In this general framework several issues have to be settled, namely the computational support, the method by which the computational support will be folded, and the way the output is read. Here we only consider a restricted model where both the support of computation and the folding procedure are strings of symbols from some given languages. The model deals with the one-dimensional case and in our construction "computing" represents the generation of a language by means of a folding operation, starting from two "simpler" languages.

Finally, it is worth noting that the folding operation appears in various forms and contexts in formal language and natural computing theories; we quote here only [7] and [9]. In addition, the type of string operation considered in this paper recalls in a certain sense the shuffle operation as defined in [3] and [6].

2 Preliminaries

For the present work, we assume the reader be familiar with the basic notions and results of formal languages and theory of computation (for more details we indicate [2] and [8]). Here, we only present some results related to the current work.

If FL is a family of languages, then NFL denotes the family of length sets of languages in FL.

We denote by REG, CF, CS, REC, and RE the family of regular, context-free, context-sensitive, recursive, and recursively enumerable languages, respectively. It is known that $REG \subsetneq CF \subsetneq CS \subsetneq REC \subsetneq RE$. For regular and context-free languages there are known necessary conditions also called pumping lemmas. In case of regular languages the pumping lemma can be stated as:

Lemma 1. *Let L be an infinite regular language over a finite alphabet Σ. Then there is a positive constant n_L such that if $w \in L$, $|w| \geq n_L$, then w can be decomposed as $w = xyz$, $x, y, z \in \Sigma^*$, with $|y| \geq 1$, $|xy| \leq n_L$, and $w_i = xy^iz \in L$, for any $i \geq 0$.*

In case of context-free languages the pumping lemma can be stated as:

Lemma 2. *Let L be an infinite context-free language over a finite alphabet Σ. Then there is a positive constant n_L such that if $z \in L$, $|z| \geq n_L$, then z can be decomposed as $z = uvwxy$, $u, v, w, x, y \in \Sigma^*$, with $|vx| \geq 1$, $|vwx| \leq n_L$, and $z_i = uv^iwx^iy \in L$, for any $i \geq 0$.*

Let $\mathbb{N}$ be the set of nonnegative integers and k be a positive integer. A set $S \subseteq \mathbb{N}^k$ is a linear set if there are the vectors $v_0, v_1, \ldots, v_n \in \mathbb{N}^k$ such that $S = \{v \mid v = v_0 + a_1v_1 + \cdots + a_nv_n, a_i \in \mathbb{N}, 1 \leq i \leq n\}$. The vectors v_0 (*the constant vector*) and $v_1, v_2, \ldots, v_n$ (*the periods*) are called the *generators* of the linear set S. A set $S \subseteq \mathbb{N}^k$ is semilinear if it is a finite union of linear sets.

Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$ be a finite alphabet. A mapping $\Psi : \Sigma^* \to \mathbb{N}^{|\Sigma|}$ such that $\Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n})$ is called the Parikh mapping. The mapping Ψ can be extended to languages such that if $L \subseteq \Sigma^*$ then $\Psi(L) = \bigcup_{w \in L} \Psi(w)$. Two languages L_1 and L_2 are called *letter equivalent* if $\Psi(L_1) = \Psi(L_2)$. By $length(L)$ we will denote the length set of L.

A language L is called *semilinear* if $\Psi(L)$ is a semilinear set of vectors of numbers.

Semilinear sets are closed under complement, finite intersection and finite union. A permutation of symbols in a word preserves the Parikh image. All context-free languages are semilinear.

3 Folding Systems

Let Σ be a finite alphabet and $\Gamma = \{u, d\}$. Let $f : \Sigma^* \times \Sigma \times \Gamma \to \Sigma^*$ such that

$$\begin{aligned} f(w, a, u) &= aw, \\ f(w, a, d) &= wa. \end{aligned}$$

We define the *folding function* $h : \Sigma^* \times \Gamma^* \to \Sigma^*$ such that

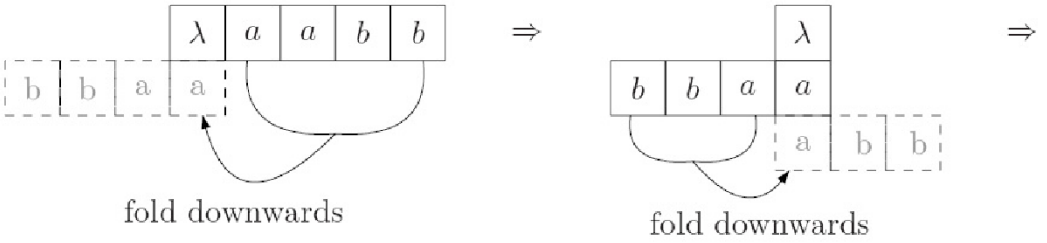
$$h(w_1, w_2) = \begin{cases} f(\ldots f(f(\lambda, a_1, b_1), a_2, b_2) \ldots, a_k, b_k), \text{if} \\ |w_1| = |w_2|, w_1 = a_1 a_2 \ldots a_k, w_2 = b_1 b_2 \ldots b_k; \\ \text{undefined, if } |w_1| \neq |w_2|. \end{cases}$$

Remark 3.1. If $w_1 \in \Sigma^*$, $w_2 \in \Gamma^*$, and $|w_1| = |w_2|$, then by applying h we mean that w_1 is folded symbol by symbol according with the folding procedure described by the symbols of w_2 (d means to fold the "remaining" of w_1 downwards and u upwards); the result can be seen as a stack of symbols that by definition we read from the top to the bottom (see Figure 1).

Example 3. *Let $w_1 = aabb$ and $w_2 = dduu$. Then we formally have*

$$\begin{aligned} h(w_1, w_2) &= f(f(f(f(\lambda, a, d), a, d), b, u), b, u) \\ &= f(f(f(a, a, d), b, u)b, u) \\ &= f(f(aa, b, u), b, u) \\ &= f(baa, b, u) \\ &= bbaa \end{aligned}$$

or graphically, the recurrent application of f is given in Figure 1.



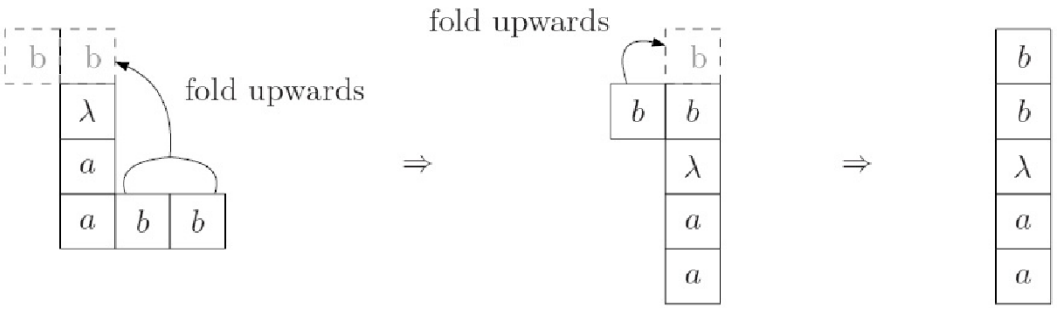


Figure 1: The computation of $h(aabb, dduu)$.

A *folding system* (in short an F-system) is a construct $\Phi = (L_1, L_2)$ where L_1 is an arbitrary language over an arbitrary alphabet Σ (called the *core language*) and L_2 is an arbitrary language over $\Gamma = \{u, d\}$ (called the *folding procedures set*). The language generated by $\Phi(L_1, L_2)$ is

$$L(\Phi) = \bigcup_{\substack{w_1 \in L_1, w_2 \in L_2 \\ |w_1| = |w_2|}} h(w_1, w_2),$$

where $h : \Sigma^* \times \Gamma^* \to \Sigma^*$ is the folding function.

For a class $\mathcal{C}$ of core languages and a class $\mathcal{H}$ of folding procedures sets, we denote by $\mathcal{F}(\mathcal{C}, \mathcal{H})$ the family of all languages generated by F-systems, using components from the respective classes, that is

$$\mathcal{F}(\mathcal{C}, \mathcal{H}) = \{L(\Phi) \mid \Phi = (L_1, L_2), L_1 \in \mathcal{C}, L_2 \in \mathcal{H}\}.$$

The following result shows the relation between the family of regular languages and the family of all languages generated by F-systems using regular languages as components.

Proposition 4. $\mathcal{F}(REG, REG) \supsetneq REG$.

Proof: Let L_1 be an arbitrary regular language and the regular language $L_2 = \{d\}^* \subseteq \Gamma^*$. Then we have that for any word $w_1 \in L_1$ there exists $w_2 \in L_2$ such that $|w_1| = |w_2|$. Since w_1 is "folded" always down (w_2 consists only of d symbols), then it follows that $h(w_1) = w_1$, hence $h(L_1) = L_1$ and further more $\mathcal{F}(REG, REG) \supseteq REG$.

In order to prove the strict inclusion one can construct the following F-system:

$$\Phi(L_1, L_2) \text{ where } L_1 = (ab)^* \text{ and } L_2 = (ud)^*.$$

It follows that for any even number $i = 2k$, $k > 0$, there exist exactly one word $w_1 = ab \dots ab \in L_1$ and exactly one word $w_2 = ud \dots ud \in L_2$ such that $|w_1| = |w_2| = i$. Hence $h(w_1, w_2) = a^k b^k$ and consequently $L(\Phi) = \{a^n b^n \mid n > 0\}$, the well-known non-regular but context-free language. □

By using context-free languages as core languages while preserving the regularity of folding procedure sets, one can surpass the class of context-free languages.

Proposition 5. $\mathcal{F}(CF, REG) \supsetneq CF$.

Proof: One can use the same argument as given in the proof of Proposition 4 to show that $\mathcal{F}(CF, REG) \supseteq CF$. In addition, consider the context-free language $L_1 = \{b^n(ac)^n \mid n \geq 0\}$ (as being generated for instance by the context-free grammar $G = (N, T, P, S)$, where $N = \{S\}$, $T = \{a, b, c\}$, and $P = \{S \to bSac, S \to \lambda\}$) and the regular language $L_2 = \{(ud)^n \mid n \geq 0\}$ indicated by the regular expression $(ud)^*$. The length of any word in L_1 is a multiple of 3, while any word from L_2 has a length which is a multiple of 2. Consequently, any word from $\Phi(L_1, L_2)$ must have a length multiple of the least common multiple of 2 and 3. It follows that $\Phi(L_1, L_2) = \{a^{2n}b^{2n}c^{2n} \mid n \geq 0\}$, which by using the pumping lemma for context-free languages can be shown not to be a context-free language. Consequently, we have that $\mathcal{F}(CF, REG) \supsetneq CF$. □

Proposition 6. $\mathcal{F}(CF, CF) \supsetneq CF$.

Proof: Since obviously $\mathcal{F}(CF, CF) \supseteq \mathcal{F}(CF, REG)$ then to prove the strict inclusion one can construct the following F-system, whose generated language does not belong to context-free languages family.

Let $\Phi = (L_1, L_2)$, where $L_1 = \{ww^r \mid w \in \{a, b\}^*\}$ is the context-free language of even palindromes and $L_2 = \{u^n d^n \mid n \geq 0\}$ is a context-free language as well. It follows that $L(\Phi) = \{ww \mid w \in \{a, b\}^*\}$ – a language that is not context-free.

Another interesting example which proves the strict inclusion regards the generation of the language $L = \{a^n b^n c^n \mid n \geq 0\}$. This can be achieved by an F-system $\Phi(L_1, L_2)$ where $L_1 = \{b^n(ac)^n \mid n \geq 0\}$ and $L_2 = \{u^n(ud)^n \mid n \geq 0\}$ are also context-free languages. □

Using a similar technique as the one presented in the proofs of Proposition 4 and Proposition 5 one can actually prove a more general result (that in addition suggests some interesting open problems regarding the usage of subfamilies of REG as folding procedures sets)

Lemma 7. *Let FL be a family of languages. Then $FL \subseteq \mathcal{F}(FL, REG)$.*

Moreover, one can state the following result:

Lemma 8. *Let FL_i, $1 \leq i \leq 4$, some families of languages such that $FL_1 \subseteq FL_2$ and $FL_3 \subseteq FL_4$. Then $\mathcal{F}(FL_1, FL_3) \subseteq \mathcal{F}(FL_2, FL_4)$.*

The following result states that all F-systems using context-free languages as components are equivalent to F-systems with a particular relation between the core languages and the folding procedures sets.

Lemma 9. *(Normal Form) Let $L \in \mathcal{F}(CF, CF)$. Then there exists an F-system $\Phi(L_1, L_2)$, $L_1, L_2 \in CF$, with $length(L_1) = length(L_2)$ and such that $L(\Phi) = L$.*

Proof: Let $L = L(\overline{\Phi}(\overline{L_1}, \overline{L_2}))$ such that $\overline{L_1}, \overline{L_2} \in CF$ and $length(\overline{L_1}) \neq length(\overline{L_2})$. We construct an equivalent F-system $\Phi(L_1, L_2)$, $L_1, L_2 \in CF$, such that $length(L_1) = length(L_2)$ as follows.

It is known from [5] that $NREG = NCF$. This means that there exist the languages $L_B, L_C \in REG$ such that $length(L_B) = length(\overline{L_1})$ and $length(L_C) = lenght(\overline{L_2})$. Let $\Sigma_{L_1} = \{a_1, a_2, \ldots, a_k\}$ be the alphabet of L_1, $\Sigma_B = \{b_1, b_2, \ldots, b_r\}$ be the alphabet of L_B, and $\Sigma_C = \{c_1, c_2, \ldots c_s\}$ be the alphabet of L_C. In addition, let us consider the finite substitution given by

$$\Delta_1 : \Sigma_C \to \mathcal{P}(\Sigma_B^*) \text{ such that } \Delta_1(c_i) = \Sigma_B,\ 1 \leq i \leq s.$$

Because regular languages are closed under finite substitutions and intersection then it follows that $\overline{L_B} = L_B \cap \Delta(L_C) \in REG$ ($\overline{L_B}$ contains all the words of length $t \geq 0$ from L_B for which there exists at least one word in L_C with the same length). Moreover, we have that $length(\overline{L_B}) = length(L_B) \cap length(L_C) = length(L_1) \cap length(L_2)$.

Let us consider now the substitutions:

$$\Delta_2 : \Sigma_B \to \mathcal{P}(\Sigma_A^*) \text{ such that } \Delta_2(b_i) = \Sigma_A,\ 1 \leq i \leq r,$$

and

$$\Delta_3 : \Sigma_B \to \mathcal{P}(\{u, d\}^*) \text{ such that } \Delta_3(b_i) = \{u, d\},\ 1 \leq i \leq r.$$

Then, if we take $L_1 = \overline{L_1} \cap \Delta_2(\overline{L_B})$ and $L_2 = \overline{L_2} \cap \Delta_3(\overline{L_B})$ it follows that $L_1, L_2 \in CF$ (as being the results of intersection of context-free languages with regular languages) and moreover $length(L_1) = length(L_2)$ (both substitutions Δ_2 and Δ_3 do not change the length sets of the languages on which they are applied; in particular $length(\overline{L_B}) = length(\Delta_2(\overline{L_B})) = length(\Delta_3(\overline{L_B}))$).

Similarly as before, in order to construct L_1 we get from $\overline{L_1}$ only those words which match as length at least one word in $\overline{L_B}$.

As a consequence, we have that $L(\Phi(L_1, L_2)) = L(\Phi(\overline{L_1}, \overline{L_2})) = L$. □

Remark 3.2. The language generated by an F-system $\Phi(L_1, L_2)$, where L_1 and L_2 are semilinear sets, is also semilinear.

Let us prove now a necessary condition for a language generated by a folding system using as components regular languages. Let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \{u, d\}^*$ be two arbitrary infinite regular languages and assume that $L(\Phi(L_1, L_2))$ (the corresponding generated language) is also infinite. Let n_{L_1} and n_{L_2} be the constants from the pumping lemma for regular languages applied to L_1

and L_2, respectively. Let $n = max(n_{L_1}, n_{L_2})$. Because $L(\Phi(L_1, L_2))$ is infinite, then there are the words $r \in L_1$ and $s \in L_2$ such that $|r| = |s| \geq n$. According with the pumping lemma for regular languages, it follows that r can be written as $r = xyz$, where $x, y, z \in \Sigma^*$ and such that $|xy| \leq n$, $y \geq 1$. Similarly, $s = uvt$ where $u, v, t \in \{u, d\}^*$, and such that $|uv| \leq n$, $v \geq 1$. In both cases, the pumping lemma states that $r_i = xy^i z \in L_1$ and $s_i = uv^i t \in L_2$, for any $i \geq 0$. Although in general $|r_i| \neq |s_i|$, for all $i \geq 0$, one can remark that, based on the lowest common multiple between y and v, one can define the sub-sequences of words

$$\overline{r}_i = xy^{\frac{lcm(|y|,|v|)}{|y|} \cdot i} z, \text{ for any } i \geq 0,$$

and

$$\overline{s}_i = uv^{\frac{lcm(|y|,|v|)}{|v|} \cdot i} t, \text{ for any } i \geq 0,$$

such that $|\overline{r}_i| = |\overline{s}_i|$, for any $i \geq 0$.

For simplicity of argument, let us consider now the double stranded structures $\left[\frac{\overline{r}_i}{\overline{s}_i}\right]$, for any $i \geq 0$. First of all, remark that x, z, u, t have constant lengths. Moreover, the above sub-sequences are infinite (because $|y| \geq 1$ and $|v| \geq 1$, one can remark that $|\overline{r}_{i+1}| > |\overline{r}_i|$, $|\overline{s}_{i+1}| > |\overline{s}_i|$, for any $i \geq 0$). It follows that, starting from a certain positive constant $\widetilde{n}$ (that depends on $|x|$, $|y|$, $|u|$, $|v|$, hence on L_1 and L_2), in the double stranded structure $\left[\frac{\overline{r}_j}{\overline{s}_j}\right]$, with $j \geq \widetilde{n}$, the substring from $\overline{r}_i$ composed by multiple concatenations of y overlaps (partially) the one from $\overline{s}_i$ composed by multiple concatenations of v. Moreover, assume in addition that $\widetilde{n}$ is the first rank in the sequence of double stranded structures such that the overlapping part is equal or longer than $min(\frac{lcm(|y|,|v|)}{|y|}, \frac{lcm(|y|,|v|)}{|v|})$; this is necessary for having the same "context" repeated at least once.

Then, it follows that any term in the sequence $\left[\frac{\overline{r}_i}{\overline{s}_i}\right]$, for any $i \geq \widetilde{n}$, can be written as $\left[\frac{k_1 k_2^j k_3}{q_1 q_2^j q_3}\right]$, where $|k_l| = |q_l|$, $1 \leq l \leq 3$, and $j \geq 1$ (see Figure 2 for a particular case).

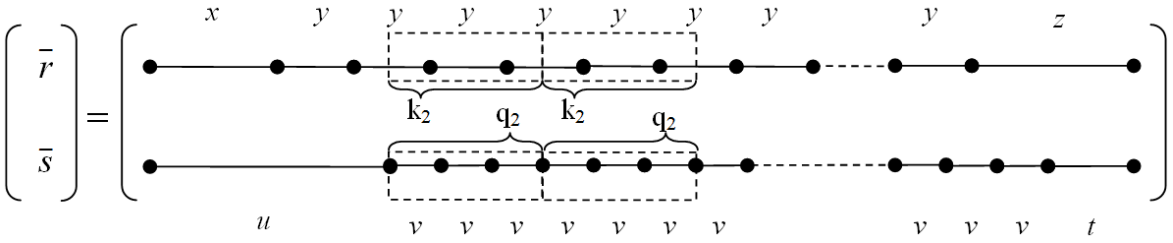


Figure 2: Here is presented a particular case of a double stranded structure (when $|u| > |x|$). For instance, here we have that $q_1 = u$ and $q_2 = v^3$.

Finally, by folding the string $k_1 k_2^j k3$ with respect to $q_1 q_2^j q_3$, one gets $m_1 m_2^j m_3 m_4^j m_5$. The string k_1 folded according with q_1 produces m_3. The string k_2 folded according with q_2 may produce a string m_2 ahead of m_3 and a string m_4 behind of m_3, but such that $|m_2 m_4| \geq 1$ (because $|k_2| \geq 1$). By repeating the above procedure for all strings k_2 and q_2, one gets $m_2^j m_3 m_4^j$. Eventually, by folding k_3 with respect to q_3, one obtains $m_1 m_2^j m_3 m_4^j m_5$.

Based on the above arguments, the following result holds true.

Lemma 10. *Let $L \in \mathcal{F}(REG, REG)$ be an infinite language. Then there exists a positive constant n such that if $w \in L$, $|w| \geq n$ then w can be decomposed as $w = uvwxy$ such that $|vx| \geq 1$, $|vwx| \leq n$, and $w_i = uv^i wx^i y \in L$, for any $i \geq 0$.*

Using Lemma 10, one can easily prove that $L = \{a^{2n}b^{2n}c^{2n} | n \geq 0\} \notin \mathcal{F}(REG, REG)$. Consequently, one can prove that

Proposition 11. $\mathcal{F}(REG, REG) \subset \mathcal{F}(CF, REG)$.

A similar result stands also in case of a infinite language $L \in \mathcal{F}(CF, REG)$. More precisely, let $L_1 \subseteq \Sigma^*$ be a infinite context-free language and $L_2 \subseteq \{u, d\}^*$ be a infinite regular languages and assume that $L = L(\Phi(L_1, L_2))$. Using the same type of arguments as in the proof of Lemma 10 it follows that there exists $r \in L_1$ and $s \in L_2$ such that $|r| = |s| \geq n = max(n_{L1}, n_{L_2})$, where n_{L_1} is the constant for L_1 given by the pumping lemma for context-free languages and n_{L_2} is the constant for L_1 given by the pumping lemma for regular languages. According to these lemmas we have

- r can be decomposed into $r = x_1x_2x_3x_4x_5$, where $x_i \in \Sigma^*$, $1 \leq i \leq 5$, such that $|x_2x_4| \geq 1$, $|x_2x_3x_4| \leq n_{L_1} \leq n$, and $r_i = x_1x_2^ix_3x_4^ix_5 \in L_1$, for $i \geq 0$;
- s can be decomposed into $s = uvt$, where $u, v, t \in \{u, d\}^*$, such that $|uv| \leq n_{L_2} \leq n$, $v \geq 1$, and $s_i = uv^it \in L_2$, for $i \geq 0$.

Based on the above relations one can define the sequences $(a_n)_n \subseteq I\!N$ and $(b_n)_n \subseteq I\!N$ such that $|r_{a_i}| = |s_{b_i}|$, for $i \in I\!N$. More precisely if we denote by $c_1 = |x_1| + |x_3| + |x_5|$, $c_2 = |x_2| + |x_4|$, and $c_3 = |u| + |t|$, $c_4 = v$, then the following relations have to be true.

$$c_1 + a_ic_2 = c_3 + b_ic_4 \text{ , hence } b_i = \frac{c_1 - c_3 + a_ic_2}{c_4}$$

Taking now

$$a_i = \begin{cases} (c_4i - 1)(c_1 - c_3) & \text{, if } c_1 \geq c_3 \\ (c_4i - 1)(c_3 - c_1) & \text{, if } c_3 > c_1 \end{cases} \quad i \in I\!N, i > 0$$

it follows that the subsequences $\overline{r}_i = x_1x_2^{a_i}x_3x_4^{a_i}x_5$, for $i > 0$, and $\overline{s}_i = uv^{b_i}t$, for $i > 0$, have equal lengths, that is $|\overline{r}_i| = |\overline{s}_i|$, for $i > 0$.

Considering the double stranded structures $\left[\frac{\overline{r}_i}{\overline{s}_i}\right]$, for any $i \geq 0$, then it follows that, starting from a certain positive constant $\widetilde{n}$ (that depends on L_1 and L_2), in the double stranded structure $\left[\frac{\overline{r}_j}{\overline{s}_j}\right]$, with $j \geq \widetilde{n}$, the substring from $\overline{s}_j$ composed by multiple concatenations of v underlays (partially) the one from $\overline{r}_j$ composed (in order) by multiple concatenations of x_2, x_3, and multiple concatenations of x_5 (because the substrings x_1 and x_3 have constant lengths). Moreover, assume that $\widetilde{n}$ is the first rank in the double stranded sequence such that the following conditions are satisfied:

- the substring from r_j composed by multiple concatenation of x_2 overlaps the substring from s_j composed by multiple concatenations of v; assume that the overlapping part is equal or longer than $min(\frac{lcm(|x_2|,|v|)}{|x_2|}, \frac{lcm(|x_2|,|v|)}{|v|})$;
- the substring from r_j composed by multiple concatenation of x_4 overlaps the substring from s_j composed by multiple concatenations of v; assume that the overlapping part is equal or longer than $min(\frac{lcm(|x_4|,|v|)}{|x_4|}, \frac{lcm(|x_4|,|v|)}{|v|})$.

Using similar arguments as in the proof of Lemma 10, it follows that any term in the sequence $\left[\frac{\overline{r}_i}{\overline{s}_i}\right]$, for any $i \geq \widetilde{n}$, can be written as $\left[\frac{k_1k_2^hk_3k_4^jk_5}{q_1q_2^hq_3q_4^jq_5}\right]$, where k_l, q_l, $1 \leq l \leq 5$, are strings (of finite lengths), $|k_l| = |q_l|$, $1 \leq l \leq 5$, and $h, j \geq 1$. Moreover, we have that $|k_2k_4| \geq 1$ (because k_2k_4

contains at least one symbol from x_2x_4 and $|x_2x_4| \geq 1$). Although in general $h \neq j$, one can decompose $\overline{r}_i$ and $\overline{s}_i$ as

$$\left[\frac{k_1k_2^hk_3k_4^{j-h}k_4^hk_5}{q_1q_2^hq_3q_4^{j-h}q_4^hq_5}\right], \text{ if } j \geq h, \text{ or } \left[\frac{k_1k_2^jk_2^{h-j}k_3k_4^jk_5}{q_1q_2^jq_2^{h-j}q_3q_4^jq_5}\right], \text{ if } h \geq j.$$

Furthermore, based on the above construction, one can actually build an infinite sequence of type

$$\left[\frac{\alpha_1\alpha_2^i\alpha_3\alpha_4^i\alpha_5}{\beta_1\beta_2^i\beta_3\beta_4^i\beta_5}\right], \text{ for } i \geq 0, \text{ where } |\alpha_l| = |\beta_l|, 1 \leq l \leq 5, |\alpha_2\alpha_4| \geq 1 \text{ and } |\alpha_2\alpha_3\alpha_4| \leq \widetilde{n}.$$

Consequently, by folding $\alpha_1\alpha_2^i\alpha_3\alpha_4^i\alpha_5$ according with $\beta_1\beta_2^i\beta_3\beta_4^i\beta_5$, for $i \geq 0$, one gets a string $w_i = w_1w_2^iw_3w_4^iw_5w_6^iw_7w_8^iw_9$, hence the following result is true.

Lemma 12. *Let $L \in \mathcal{F}(CF, REG)$ be an infinite language. Then there exists a positive constant n such that if $w \in L$, $|w| \geq n$ then w can be decomposed as $w = w_1w_2w_3w_4w_5w_6w_7w_8w_9$ such that*

$$|w_2w_4w_6w_8| \geq 1$$
$$|w_2w_3w_4w_5w_6w_7w_8| \leq n$$

and $w_i = w_1w_2^iw_3w_4^iw_5w_6^iw_7w_8^iw_9 \in L$, for any $i \geq 0$.

Using Lemma 12 one can prove that the language $L = \{ww \mid w \in \{a, b\}^*\} \notin \mathcal{F}(CF, REG)$. Consequently, the following result holds true.

Proposition 13. $\mathcal{F}(CF, REG) \subset \mathcal{F}(CF, CF)$.

Furthermore, one can show the following result.

Proposition 14. $\mathcal{F}(CS, CS) = CS$.

Proof: Assume that L_1 and L_2 are arbitrary context-sensitive languages. Then one can construct a linear bounded automaton M such that $L(M) = L(\Phi(L_1, L_2))$ and which works as follows. M uses one read only input tape and three auxiliary tapes. A word w is placed on the input tape. The first auxiliary tape is used by M to nondeterministically generate a word w_1 of the same length as the one present on the input tape, and then to check if $w_1 \in L_1$. The second auxiliary tape is used by M to nondeterministically generate a word w_2 of the same length as the one present on the input tape, and then to check if $w_2 \in L_1$. Finally, if $w_1 \in L_1$ and $w_2 \in L_2$, then M uses the third tape to simulate the folding of w_1 according with w_2 and to check whether or not the result is equal to w. Because all the above mentioned tasks can be performed in linear space with respect to the length of the input, then it follows that $L(\Phi(L_1, L_2)) \in CS$. Moreover, one can notice that all context-sensitive languages can be generated (to prove this, one can use a similar construction as in the proof of Proposition 4). Consequently, we have that $\mathcal{F}(CS, CS) = CS$. □

Because the class o context-sensitive languages CS contains non semilinear languages, then using Remark 3.2 it follows that

Proposition 15. $\mathcal{F}(CF, CF) \subset \mathcal{F}(CS, CS) = CS$.

Using a similar construction as in the proof of Proposition 14 one can prove that

Proposition 16. $\mathcal{F}(REC, REC) = REC$.

Proposition 17. $\mathcal{F}(RE, RE) = RE$.

Finally we can sum up all the results in the following hierarchy:

Theorem 18.

$$\begin{array}{lllllllll} REG & \subset & \mathcal{F}(REG, REG) & \subseteq & CF & \subset & \mathcal{F}(CF, REG) & \subset \\ \mathcal{F}(CF, CF) & \subset & \mathcal{F}(CS, CS) & = & CS & \subset & \mathcal{F}(REC, REC) & = \\ REC & \subset & \mathcal{F}(RE, RE) & = & RE. & & & \end{array}$$

4 Conclusion

In this paper we introduced a new computing paradigm called *folding systems*. We investigated their computational power under certain constraints and we studied some of their properties.

During our endeavor several open problems arose. One of them regards the existence of a pumping lemma for folding systems where both the core language and the folding procedure set are context-free languages. Another open problem regards the existence of a polynomial time parsing algorithm for folding systems which use regular/context-free languages as core/folding procedure set. Future investigations will focus on variants of folding systems where other families of languages will be used (for example those generated by Lindenmayer systems), but also on defining more complex and realistic methods for performing the string folding.

We conclude with the belief that many interesting problems can be pursued in this line of research.

Acknowledgment

The work of the author was supported by the CNCSIS IDEI-PCE grant, no. 551/2009, Romanian Ministry of Education, Research and Innovation.

Bibliography

[1] T. Head, Parallel Computing by Xeroxing on Transparencies, *Algorithmic Bioprocesses*, 9, pp. 631–637, Springer-Verlag, Berlin, 2009.

[2] J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley Publishing, Reading Massachusetts, 1979.

[3] A. Mateescu, G. Rozenberg, A. Salomaa, Shuffle on Trajectories: Syntactic Constraints, *Theoretical Computer Science*, 197, 1-2, pp. 1–56, 1998.

[4] M. Minsky, *Computation – Finite and Infinite Machines*. Prentice Hall, Englewood Cliffs, NJ, 1967.

[5] R. J. Parikh. On Context–free languages. *Journal of the Association for Computing Machinery*, 13 (4), pp. 570–581, 1966.

[6] G. Păun, G. Rozenberg, A. Salomaa, Grammars Based on the Shuffle Operation, *Journal of Universal Computer Science*, 1 (1), pp. 67–82, 1995.

[7] P.W.K. Rothemund, Folding DNA to create nanoscale shapes and patterns, *Nature* (440), pp. 297–302, 2006.

[8] G. Rozenberg, A. Salomaa, eds., *Handbook of Formal Languages*, 3 volumes. Springer-Verlag, Berlin, 1997.

[9] G. Rozenberg, Gene Assembly in Ciliates: Computing by Folding. *A Half-Century of Automata Theory*, World Scientific Publishing Company, pp. 93–130, 2000.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 749-760

On Fuzzy Sequences, Fixed Points and Periodicity in Iterated Fuzzy Maps

H.N. Teodorescu

Horia Nicolai Teodorescu
Gheorghe Asachi Technical University,
Iasi, Romania
Institute for Computer Science,
Romanian Academy, Iasi Branch
E-mail: hteodor@etc.tuiasi.ro

Abstract: I exemplify various elementary cases of fuzzy sequences and results related to the iteration of fuzzy mappings and to fuzzy logic systems (FLS). Several types of fuzzy logic system iterations are exemplified in relationship with oscillations in FLS and with the problem of stability in fuzzy logic control. I establish several conditions for fixed points and periodicity of the iterations based on fuzzy systems.
Keywords: modeling, control, fuzzy map iteration, fuzzy sequence, behavior.

1 Introduction

In this paper, I provide simple examples of iteration of fuzzy sets and I discuss the asymptotic behavior of their successive iterations. I discuss several cases of iterated fuzzy logic and arithmetic systems with and without defuzzification and determine conditions for fuzzy sets that are fixed points. The topic was dealt with in the literature in the early approaches by Kloeden [1] and in [2]- [9], but attracted little interest during the last two decades. However, sequences of fuzzy numbers and various generalizations constituted recently the subject of a large number of papers [10]- [21], while many applications indirectly referred to fuzzy sequences [22]- [24]. However, few recent papers, like [24], addressed fuzzy sequences and the problem of fixed points in fuzzy mappings with relation to fuzzy systems and control.

In this paper, a fuzzy map is any application from a set of fuzzy sets to itself. A fuzzy system is a function with fuzzy set-valued variables, also named inputs, and fuzzy set values, named output. An iterated fuzzy system can be seen as a fuzzy system, in the sense used in control, but without defuzzification, that has a connection from the fuzzy output to the fuzzy input through a unitary delay loop. Analyzing fuzzy iterations has therefore applications in control, in decision making, and in modeling, among others.

The connection between iterations and sequences is straightforward: while iterations are defined by $f^{[n+1]} = f(f^{[n]})$, many sequences are obtained as $a_{n+1} = f(a_n)$, $a_{n+1} = f(a_{n-1}, a_n)$, $a_{n+1} = f(g(a_{n-1}), a_n)$, or by similar formulas.

A fixed point of a specified fuzzy function (system) iteration is a fuzzy set, $\tilde{\mathrm{A}}$, (membership function, $\mu_{\tilde{A}}$) that is preserved under the iteration; in other words, when presented to the input of the fuzzy system, it is found unchanged at its output.A periodical fuzzy system will recover after a fixed number of iterations - the period - the initial input membership function as its output. Assume that a metric space has been defined on a space of fuzzy sets. Subsequently, when convergence of sequences is considered in the respective metric space, I mean that $\tilde{A}_n \to \tilde{A}_\infty$ iff $d(\tilde{A}_n, \tilde{A}_\infty) \longrightarrow^{n\to\infty} 0$. Whenever absolute convergence will be meant, the following definition applies: $\tilde{A}_n \to \tilde{A}_\infty$ iff $|\mu_{\tilde{A}_n}(x) - \mu_{\tilde{A}_\infty}| \longrightarrow^{n\to\infty} 0 \ \forall x$. In this paper, the primary interest is in absolute convergence.

A basin of attraction of a specified input fuzzy set, which is a fixed point for a specified fuzzy system, is the set of membership functions that determine the output of the fuzzy system to tend during the iteration to the specified fuzzy set. The applications of the above concepts in the real space are well known to control engineers, biologists, and virtually in all disciplines.

I denote by $[0,1]^R$ the set of functions from R to [0,1]. Membership functions used at the input and at the output of the fuzzy logic systems belong to $[0,1]^R$. Being given an application $\varphi : [0,1]^R \to [0,1]^R$, the successive iteration of φ with initial condition $\tilde{A}_0$ is $\tilde{A}_0 \to \tilde{A}_1 = \varphi(\tilde{A}_0), \tilde{A}_2 = \varphi(\tilde{A}_1) = \varphi^{[2]}(\tilde{A}_0), \tilde{A}_3 = \varphi(\tilde{A}_2) = \varphi^{[3]}(\tilde{A}_0)$, ...

A fuzzy logic system (FLS) is basically an application $\varphi : [0,1]^R \to [0,1]^R$, which is defined in a specific manner, using rules. When a discrete-time feedback loop is added to a fuzzy logic system, the equivalent operation is an iteration of the function φ starting with an initial input membership function $\tilde{A}_0$. The operation of the loop is expressed by the successive iterations $\varphi^{[n]}(\tilde{A}_0)$, n=1,2,3,... . From the application point of view, an iteration of a FLS can be seen as a FLS with a discrete time, constant loop, like in Fig.1.

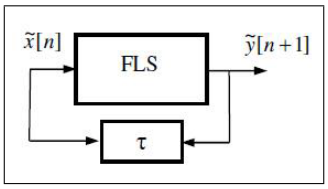


Figure 1: Iteration of a FLS represented as a FLS with a delay loop. The loop "gain" is unitary. The operation is in discrete time.

The set $[0,1]^R$, when endowed with one of the usual definitions of distance between functions, becomes a metric space and has the corresponding induced topologic structure. Therefore, we are able to talk about limit points in that topologic space and about the asymptotic behavior of the successive iteration of an application represented by a FLS. When a metric is needed, I use $d(\tilde{A}, \tilde{B}) = \int_{-\infty}^{\infty} (\mu_{\tilde{A}}(x) - \mu_{\tilde{B}}(x))^2 dx$, where $\tilde{A}, \tilde{B}$ stand for the output of the FLSs, assuming integrability.

In the next section, which is preparatory, I exemplify various ways for building sequences of fuzzy sets. In section 3, I discuss sequences generated by FLSs. Conclusions are derived in Section 4. Throughout this paper, $\Diamond$ denotes the end of a proof, example, or a line of reasoning.

2 Elementary examples of fuzzy sequences

2.1 Games based on fuzzy sequences produced by logic operations over fuzzy sets

I start illustrating the topic of fuzzy sequence by a simple example based on logic operations with fuzzy sets. Consider the following game: Three school colleagues have a high spirit of contradiction. When they start a discussion, they typically start with two disparate opinions; then, each of them is ready to contradict what the last speaker told and partly agree and partly disagree with the ante-precedent speaker; to do so, they answer with what at the same time contradicts the predecessor, yet agrees with the ante-predecessor. They take turns in expressing their opinions in that way. Can they ever come to a conclusion? If yes, after how many steps of mutual contradictions? (I would say that the game is not far from what seldom happens in real life, so this example is not devoid of interest. The problem is if a group of three skillful contradictors still can agree under fuzzy logic.)

This "game" is not possible in binary logic, but makes sense in the frame of fuzzy logic. Consider two fuzzy sets, $\tilde{A}_0$ and $\tilde{A}_1$ representing the first two opinions. The third colleague's and the subsequent responses define a sequence of fuzzy sets by $\tilde{A}_{n+1} = \neg\tilde{A}_n \cap \tilde{A}_{n-1}, n \geq 1$, where $\neg\tilde{A}_n$ denotes the complement of the fuzzy set $\tilde{A}_n$. The above recurrence, together with the initial conditions $\tilde{A}_0$ and $\tilde{A}_1$, defines the fuzzy set sequence. It is easy to see that the sequence is $\tilde{A}_0, \tilde{A}_1, \tilde{A}_2 = \neg\tilde{A}_1 \cap \tilde{A}_0, \tilde{A}_3 = \neg\tilde{A}_2 \cap \tilde{A}_1$,... . The membership functions of $\tilde{A}_3$ is $\mu_3(x) = min(\mu_1(x), 1 - \mu_2(x))$, or $\mu_3(x) = min(\mu_1(x), 1 - min(\mu_0(x), 1 - \mu_1(x)))$.
(i) Assume $\mu_0(x) < 1 - \mu_1(x)$. Then $\mu_3(x) = min(\mu_1(x), 1 - \mu_0(x))$. Because $\mu_0(x) < 1 - \mu_1(x)$, $\mu_1(x) < 1 - \mu_0(x)$, thus $\mu_3(x) = min(\mu_1(x), 1 - \mu_0(x)) = \mu_1(x)$.
(ii) Assume $\mu_0(x) \geq 1 - \mu_1(x)$. Then $\mu_3(x) = min(\mu_1(x), 1 - (1 - \mu_1(x))) = \mu_1(x)$. Therefore, $\tilde{A}_3 = \tilde{A}_1$. Next, $\tilde{A}_4 = \neg\tilde{A}_3 \cap \tilde{A}_2 = \neg\tilde{A}_1 \cap \tilde{A}_2$ and $\mu_4(x) = min(\mu_2(x), 1 - \mu_1(x))$, or $\mu_4(x) = min(min(1 - \mu_1(x); \mu_0(x)), 1 - \mu_1(x))$.
(i) Assume $\mu_0(x) < 1 - \mu_1(x)$. Then $\mu_4(x) = min(\mu_0(x), 1 - \mu_1(x)) = \mu_0(x)$.
(ii) Assume $\mu_0(x) \geq 1 - \mu_1(x)$. Then $\mu_4(x) = min(1 - \mu_1(x), 1 - \mu_1(x)) = 1 - \mu_1(x)$. Thus, $\tilde{A}_4 = \neg\tilde{A}_1$. Next, $\tilde{A}_5 = \neg\tilde{A}_4 \cap \tilde{A}_3 = \tilde{A}_1 \cap \tilde{A}_1 = \tilde{A}_1$.

If the response strategy is changed, that is $\tilde{A}_{n+1} = \tilde{A}_n \cap \neg\tilde{A}_{n-1}, n \geq 1$, the game is less simple. Indeed, the membership functions of $\tilde{A}_2$ and $\tilde{A}_3$ are $\mu_2 = min(\mu_1, 1 - \mu_0), \mu_3 = min(min(\mu_1, 1 - \mu_0), 1 - \mu_1)$. Whenever $\mu_1(x) < 1 - \mu_0(x)$, $\mu_3(x) = min(\mu_1(x), 1 - \mu_1(x))$, $\tilde{A}_3 = \tilde{A}_1 \cap \neg\tilde{A}_1$, else $\tilde{A}_3 = \neg\tilde{A}_0 \cap \neg\tilde{A}_1$. The next step is $\tilde{A}_4, \mu_4 = min(\mu_3, 1 - \mu_2)$.

2.2 Fuzzy sequences produced by arithmetic operations over values of the membership functions

Arithmetic operations with fuzzy numbers have been used to build chaotic processes by Kloeden [1]. Here I exemplify how, based on sequences of fuzzy sets and operating them through logic operations and using arithmetic manipulations of the elements of the sequences, one can obtain new sequences with the desired properties. Consider the fuzzy set $\tilde{A}_0$ with

$$\mu_0(x) = \begin{cases} x^2 & x \in [0,1] \\ 0 & elsewhere. \end{cases}$$

and the sequence of singletons with the membership functions

$$\mu_n(x) = \begin{cases} 1 & x = \frac{1}{n} \\ 0 & elsewhere. \end{cases}$$

Define $\tilde{C}_n$ by $\tilde{C}_0 = \tilde{A}_0$ and $\tilde{C}_n = \tilde{C}_{n-1} \cup \tilde{B}_n$ for $n > 0$. The graph of the fuzzy set $\tilde{C}_n$ looks like in Figure 2, where, for convenience, I have drawn a line to show where a singleton occurs. Recall that a small full circle on the graph conventionally denotes the actual value at a point of discontinuity of a function, while a small empty circle denotes the lack of value on the graph of a continuous function. The number of singularities (discontinuities) of the membership function of the sets $\tilde{C}_n$ is equal to n.$\Diamond$

Define a sequence of fuzzy sets, $\{\tilde{u}_n, n = 1, 2, ...\}$, based on the fuzzy numbers ν_0, ν_1,

$$\mu_{\tilde{\nu}_0}(x) = \begin{cases} 0 \;\; if \;\; x < 0 \;\; or \;\; x > 3 \\ 1 - x/3 \qquad else \end{cases},$$

$$\mu_{\tilde{\nu}_1}(x) = \begin{cases} 0 \;\; if \;\; x \neq 1 \\ 1 \qquad else \end{cases},$$

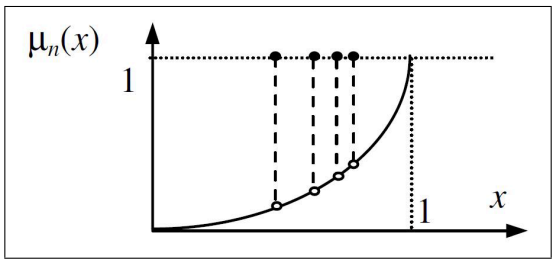


Figure 2: .

$$as \quad \tilde{u}_n = \tilde{\nu}_0 \cap \neg(3 \cdot \tilde{\nu}_1)^n.$$

I propose to graphically and analytically find the first two sets in the series, then to write their expressions, determining if there is a limit of the sequence. According to the operations with fuzzy numbers in the framework of the fuzzy arithmetic,

$$\mu_{(3-\nu_1)^n}(x) = \begin{cases} 1 & if \quad x = 3^n \\ 0 & elsewhere. \end{cases}$$

This expression is obtained by applying the definition of the product of fuzzy numbers:

$$\tilde{m} \cdot \tilde{n} = \tilde{q}, \mu_q(w) = \max_{all \ y,z,s,t;y\cdot z=w} min(\mu_m(y), \mu_n(z)).$$

For singletons $\tilde{\nu}$, the expression $min(\mu_m(y), \mu_n(z)) \neq 0$ only if z corresponds to the value of the singleton. From the defining relations, $\tilde{u}_1 = \tilde{\nu}_0 \cap \neg(3 \cdot \tilde{\nu}_1)^1$ and

$$\mu_{3\nu_1}(x) = \begin{cases} 1 & if \quad x = 3 \\ 0 & elsewhere, \end{cases}$$

$$\mu_{-3\nu_1}(x) = \begin{cases} 1 & if \quad x \neq 3 \\ 0 & if \quad x = 3, \end{cases}$$

$$\mu_{u_1}(x) = \begin{cases} 0 & if \quad x < 3 \quad or \quad x > 3 \\ 1 - x/3 & if \quad x \in [0,3]|\{3\} \\ 0 & if \quad x = 3. \end{cases}$$

Therefore, $\mu_{u_1} = \mu_{\nu_0}$. Similarly,

$$\tilde{u}_2 = \tilde{\nu}_0 \cap \neg(3 \cdot \tilde{\nu}_1)^2,$$

$$\mu_{u_1}(x) = \begin{cases} 0 & if \quad x < 0 \quad or \quad x > 3 \\ 1 - x/3 & else \end{cases}$$

Thus, $\mu_{u_1} = \mu_{\nu_0}$. Moreover, for all $n > 2, \mu_{u_n} = \mu_{\nu_0}$ (see Figure 3). Therefore, $\tilde{\nu}_0$ is the limit of the constant sequence $\{\tilde{u}_n\}$.

Further sequences are obtained by varying the manipulation of the membership functions in the precedent sequence. For example, changing the definition condition $\tilde{u}_n = \tilde{\nu}_0 \cap \neg(3 \cdot \tilde{\nu}_1)^n$ with the condition $\tilde{u}_n = \tilde{\nu}_0 \cap \neg \left(\frac{2}{3} \cdot \tilde{\nu}_1\right)^n$, a new sequence is obtained.

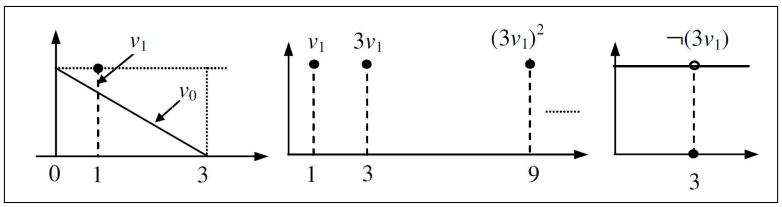


Figure 3: Explanatory graphs .

2.3 Sequences defined by conditions on the membership functions

Sequences of fuzzy sets can be conveniently defined based on sequences of real numbers standing for α-cuts. Another fuzzy sequence with limit is exemplified subsequently. Consider a fuzzy set $\tilde{A}_0$ satisfying the condition $\mu_0(x) > 0$ if x $\in$ (0,1) and $\mu(x) = 0$ elsewhere. I denoted $\mu_{\tilde{A}_0}$ by μ_0. Apply the following iterative procedure to generate a sequence of fuzzy sets:

$$\mu_n(x) = \begin{cases} \mu_{n-1}(x) & if \quad \mu_{n-1}(x) < \left(\frac{2}{3}\right)^n, \quad n \geq 1, \\ 0 & elsewhere, \end{cases}$$

where μ_n represents the membership function of the fuzzy set obtained at the n-th iteration. It is easy to check that $\mu_n \longrightarrow_{n\to\infty} \emptyset$ (see Figure 4.)

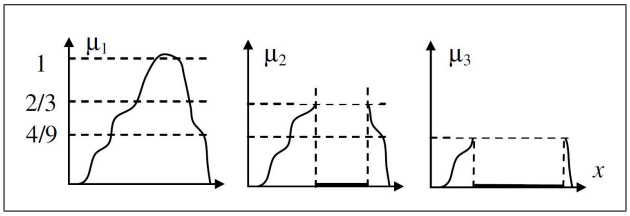


Figure 4: A sequence of fuzzy sets tending to the null fuzzy set, $\mu(x) = 0 \; \forall$ x $\in \mathbf{R}$.

Periodical fuzzy sequences are defined as a direct extension of periodicity for real valued sequences. I define a sequence of fuzzy sets as follows:

$$\tilde{A}_0 : \qquad \mu_{\tilde{A}_0}(x) = \begin{cases} 0 & if \quad x < 0 \quad or \quad x > 3 \\ 1 - x/3 & else, \end{cases}$$

$$\tilde{A}_1 : \qquad \mu_{\tilde{A}_1}(x) = \begin{cases} 0 & if \quad x < 0 \quad or \quad x > 3 \\ x/3 & if \quad x \geq 0 \quad and \quad x \leq 3, \end{cases}$$

$$\tilde{A}_{n+1} = \tilde{A}_{n-1} \cap \neg\tilde{A}_n.$$

The first two sets in the sequence are shown in Fig.5. I start by drawing the graphs of the membership functions of the first two sets in the sequence (see Figure 5.) From $\tilde{A}_{n+1} = \tilde{A}_{n-1} \cap \neg\tilde{A}_n$, derive $\tilde{A}_2 = \tilde{A}_0 \cap \neg\tilde{A}_1$. Notice that on the interval [0,3], $\neg\tilde{A}_1 = \tilde{A}_0, \neg\tilde{A}_0 = \tilde{A}_1$. Moreover, for $x < 0$ and for $x > 3$, $\mu_{\tilde{A}_0 \cap \neg\tilde{A}_0}(x) = 0$, $\mu_{\tilde{A}_1 \cap \neg\tilde{A}_1}(x) = 0$ and $\mu_{\tilde{A}_0 \cap \neg\tilde{A}_1}(x) = 0$. Therefore, $\tilde{A}_2 = \tilde{A}_0 \cap \tilde{A}_0 = \tilde{A}_0$.$\diamondsuit$

Next, write the expression of the membership functions of $\tilde{A}_2$ and $\tilde{A}_3$ like

$$\tilde{A}_2 : \mu_{\tilde{A}_2}(x) = \begin{cases} 0 & if \quad x < 0 \quad or \quad x > 3 \\ 1 - x/3 & if \quad x \geq 0 \quad and \quad x \leq 3; \end{cases}$$

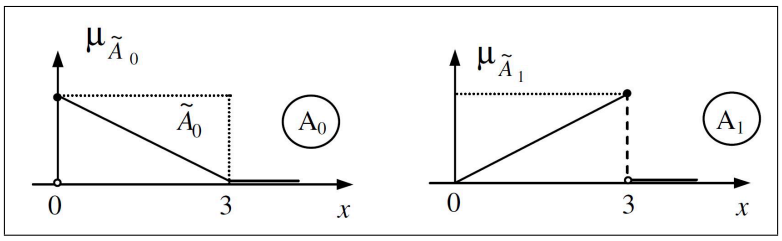


Figure 5: Explanatory graphs.

also, $\tilde{A}_3 = \tilde{A}_1 \cap \neg\tilde{A}_2, \tilde{A}_3 = \tilde{A}_1$, where

$$\tilde{A}_3 : \begin{cases} 0 \;\; if \;\; x < 0 \;\; or \;\; x > 3 \\ x/3 \;\; if \;\; x \geq 0 \;\; and \;\; x \leq 3; \end{cases}$$

There is no limit for this sequence. Indeed, $\tilde{A}_n = \tilde{A}_{n-2} \; \forall \; n$. Concluding, this sequence of fuzzy sets has the form $\tilde{A}_0, \tilde{A}_1, \tilde{A}_0, \tilde{A}_1$,..., that is, it oscillates with period 2.

2.4 Fuzzification-defuzzification-based sequences

Consider the set of fuzzy sets $\Phi = \{\tilde{A} | \mu_A : \mathbf{R} \to [0,1]\}$. Define an operation $\circ : \mathbf{R} \times \Phi \times \Phi \to \Phi$, where $\mathbf{R}$ is the set of real numbers. An example of such operation is the fuzzification followed by an inference (applying rules and truncation). The operation $\circ$ is defined by $(x \in R, (\tilde{A}, \tilde{B}) \in \Phi \times \Phi) \to \Phi$, the result being the fuzzy set with the membership function $\mu(u; (\tilde{A}, \tilde{B}, x)) = min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(u))$. Then, define a "seed" $x_0 \in \mathbf{R}$ and the sequence $\tilde{A}_0, \tilde{A}_1 = x_0 \circ \neg\tilde{A}_0, \; \tilde{A}_2 = x_0 \circ \neg\tilde{A}_1$,... .

Consider, as above, the sets $\mathbf{R}$ and Φ, and the operation $\circ : \mathbf{R} \times \Phi \to \Phi$. Moreover, consider a map $\hat{\circ} : \Phi \to \mathbf{R}$. An example of such a map is the defuzzification, but a more interesting case would be the defuzzification followed by a map from $\mathbf{R}$ to $\mathbf{R}$. Define a "seed" $x_0 \in \mathbf{R}$, then the sequence $\tilde{A}_0, \; x_1 = \hat{\circ}(\circ(x_0, \tilde{A}_0, \tilde{A}_0)), \; \tilde{A}_1 = (\circ(x_1, \tilde{A}_0, \tilde{A}_0)), \; x_2 = \hat{\circ}(\circ(x_1, \tilde{A}_0, \tilde{A}_0))$,... The generation of this sequence relates to the main subject of study in this topic. Indeed, the procedure above corresponds to an "identity (Mamdani-type) fuzzy logic system", that is, a system based on a single rule "If input is $\tilde{A}$, then output is $\tilde{A}$, followed by a defuzzifier, the defuzzified output value being fed back at the input for fuzzification. Both the defuzzification and the feedback loop are included in the same $\circ$ operation. This amounts to a simplified fuzzy system in a discrete-time feedback loop configuration. As a matter of example, consider a Mamdani-type fuzzy system with a single rule, $\tilde{A} \to \tilde{A}$, where the graph of the membership function of $\tilde{A}$ is shown in Figure 6 a) or b).

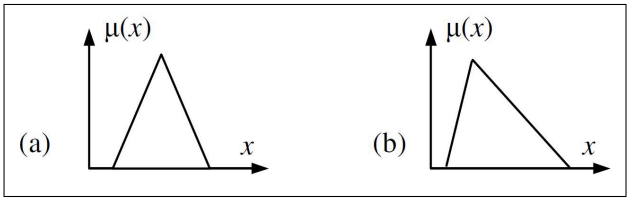


Figure 6: .

For case (a), the sequence $\tilde{A}_0, \; x_1 = \hat{\circ}(\circ(x_0, \tilde{A}_0, \tilde{A}_0)), \; \tilde{A}_1 = (\circ(x_1, \tilde{A}_0, \tilde{A}_0)), \; x_2 = \hat{\circ}(\circ(x_1, \tilde{A}_0, \tilde{A}_0))$,... becomes constant $(\tilde{A})$ after two steps, for whatever x_0 such that $\mu_{\tilde{A}}(x_0) > 0$.

For case (b), there is a limit when $x = \hat{\circ}(\circ(x, \tilde{A}, \tilde{A}))$ has a solution, that is, the application $\hat{\circ}(\circ(\cdot, \tilde{A}, \tilde{A}))$ has a fixed point.

3 Iterated fuzzy logic systems

Essentially, a fuzzy logic system without defuzzification operation at the output is a function from a set of fuzzy sets (the "input" fuzzy sets) to another set of fuzzy sets (the "output" fuzzy sets). A FLS is described by:

i) A finite set of "input" fuzzy sets; I denote these sets by $\tilde{A}_i$, $i = 1, ..., p$.

ii) A finite set of "output" fuzzy sets; I denote these sets by $\tilde{B}_j$, $j = 1, ..., q$.

iii) An index function $\imath : \{1, ..., p\} \to \{1, ..., q\}$.

iv) An actual "input" fuzzy set (equivalently, membership function), denoted by $\tilde{X}$; $\tilde{X}$ is the value of the variable of the function represented by the fuzzy logic system.

v) The equations defining the "output" (the fuzzy set value of the function) corresponding to the input are:

$$w_{j=\imath(i)} = \max_x \min\lfloor \mu_{\tilde{X}}(x), \mu_{\tilde{A}_i}(x) \rfloor,$$

$$\mu_{\tilde{Y}}(x) = \max_j \lfloor \min(w_1, \mu_{\tilde{B}1}(x)), ..., (w_2, \mu_{\tilde{B}2}(x)), ..., (w_q, \mu_{\tilde{B}q}(x)) \rfloor.$$

In applications, fuzzy logic systems are described by rules, in the form IF "input" $\tilde{X}$ is $\tilde{A}_i$ THEN "output" $\tilde{Y}$ is $\tilde{B}_j$.
where $\tilde{A}_i$ corresponds to an input fuzzy set and $\tilde{B}_j$ to an output fuzzy set, according to the rule. In principle, nothing prevents that two rules, with the same antecedent have different consequents. Also, rules in the form:

IF "input" $\tilde{X}$ is $\tilde{A}$ THEN "output" Y is $\tilde{B}$ AND $\tilde{Y}$ is $\tilde{B}\prime$;

are allowed. However, in the discussion in this section, I do not include such cases. With the above definition, introduce iterated fuzzy systems as follows:

i) Assume the sets of the input and output fuzzy sets are defined on the same universe of discourse. This condition allows us to perform the iteration. While it is not required, we may require that the input fuzzy sets are identical to the output ones; that is, the function represented by the fuzzy logic system maps a set of fuzzy sets into itself.

ii) At time moment "0", present the fuzzy set $\tilde{X}[0]$ at the input of the fuzzy system.

iii) Assign to the actual input fuzzy set $\tilde{X}[n]$, which is presented at the input of the fuzzy system at the time moment n, the output at time moment n-1, $\tilde{Y}[n-1]$, for $n > 0$.

This procedure produces the repeated iteration φ^n of the mapping represented by the fuzzy system $\varphi : M_U \to M_U$, where M_U is the set of fuzzy sets on the universe of discourse U.

In what follows, I present some examples related to typical applications of FLSs. I first address the general case of Sugeno iterated system; next, I address an example involving Sugeno systems; finally, I address an example involving an iterated Mamdani system.

Consider a directed graph G=(V,E), where V is the set of vertices and E is the set of directed edges. The graph can be interpreted as the pictorial description (graph) of the index function used in the definition of the input-output mapping represented by the FLS. When the input membership functions differ from the output membership functions, the graph is bipartite. Intuition may induce us to believe that the iteration of a FLS system can have a fixed point only if the graph has a node with a self-loop. This is not true. Even FLSs with bipartite graphs may have a fixed point.

Iterated fuzzy logic systems with singleton-type output. General case.

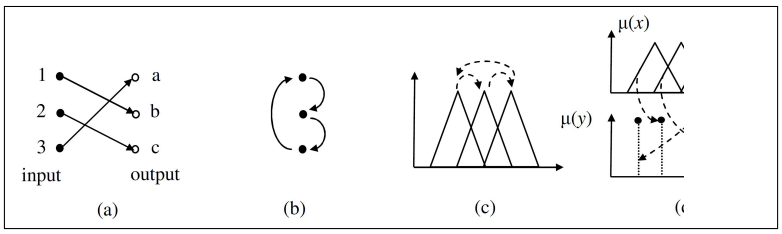


Figure 7: (a) Bipartite graph and loop for a fuzzy logic system with the index functions $1 \to b, 2 \to c, 3 \to a$. (b) respectively $1 \to 2, 2 \to 3, 3 \to 1$. (c) Example of membership functions for the case (b). Arrows indicate the input to output correspondence. Mamdani-type system. (d) Sugeno-type system.

Consider the set of membership functions $M_R = \{\mu : \mathbf{R} \to [0,1]\}$ and the set of weighted singletons on $\mathbf{R}$, denoted by $S_R = \{(\beta, w) | \beta \in \mathbf{R}, w \in [0,1]\}$. In (β, w), β are the values where the singletons occur, that is, $x \neq \beta \Rightarrow \mu_\beta(x) = 0$, and w ∈ [0,1] are the corresponding weights (truth-values), that is $x = \beta \Rightarrow \mu_\beta(x) = w$.

Subsequently, formalize the definition of fuzzy logic systems with singleton-type output, which is similar to the classic 0-order Sugeno system, but "deprived" of the defuzzification of the output. Such a system always produces at the output a set of singletons. Thus, the system is a fuzzy-input, fuzzy-output system. When iterated, such a system may have as fixed point a set of singletons that, when presented at the input of the system, produces the same set of singletons at the output.

First, define:

i) A finite set of "input" fuzzy sets; denote these sets by $\tilde{A}_i, i = 1, ..., p$.

ii) A finite set of "output" singleton-type fuzzy sets; denote these sets by $(B_j, w_j = 1), j = 1, ..., q$.

iii) An index function $\imath : \{1, ..., p\} \to \{1, ..., q\}$.

vi) An actual "input" fuzzy set (equivalently, membership function), denoted by $\tilde{X}$; $\tilde{X}$ is the value of the variable of the function represented by the fuzzy logic system.

iv) The equation defining the truncation of the "output" singletons corresponding to the input $\tilde{X}$ is $w_{j=\imath(i)}(\tilde{X}) = \max_x \min\lfloor \mu_{\tilde{X}}(x), \mu_{\tilde{A}_i}(x) \rfloor$. Here, $w_{j=\imath(i)}(\tilde{X})$ is the truncation value of the singleton with the current number j, produced by the rule which for an antecedent $\tilde{A}_i$ has consequent β_j .

Notice that the above relation determines the weights of the output singletons, based on the intersection of the fuzzy sets $\tilde{X}$ and $\tilde{A}_i$. FLSs as above will be named Sugeno-type.

Proposition. The iteration of a FLS as above defined may have a fixed point.

The proof is based on an example. Consider the FLS defined as in Fig. 8. Assume the membership functions and the rules are:

$$\tilde{A}_1 \to \beta_2 = 2.5,\ \tilde{A}_2 \to \beta_1 = 1.5,\ \mu_{\tilde{A}_1}(x) = \begin{cases} x-1 & x \in [1,2] \\ 3-x & x \in [2,3], \\ 0 & else \end{cases} \quad \mu_{\tilde{A}_2}(x) = \begin{cases} x-2 & x \in [2,3] \\ 4-x & x \in [3,4]. \\ 0 & else \end{cases}$$

Then, $\tilde{X}$, where $\mu_{\tilde{X}}(x) = \begin{cases} 0.5 & x = 1.5 \\ 0.5 & x = 2.5 \ is\ a\ fixed\ point\ of\ the\ FLS. \diamondsuit \\ 0 & else \end{cases}$

It is not difficult to see that:

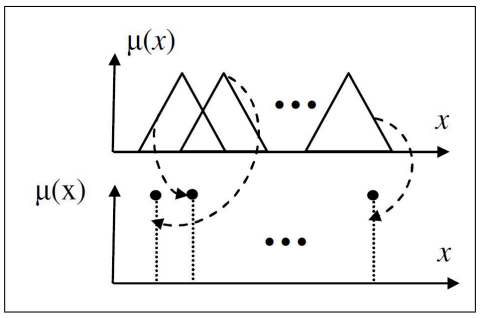


Figure 8: .

Consider a Sugeno-type FLS as above, with continuous input membership functions and with two rules $\tilde{A}_i \to \beta_j$ $\tilde{A}_{i+1} \to \beta_k$ where $\mu_{\tilde{A}_i}(\beta_j) \neq 0, \mu_{\tilde{A}_{i+1}}(\beta_k) \neq 0$ and $\forall$ h such that $h \neq i \& h \neq i+1$ $\mu_{\tilde{A}_h}(\beta_j) = 0$ and $\mu_{\tilde{A}_h}(\beta_k) = 0$. Then, the FLS has the fixed point $(\beta_j, \mu_{\tilde{A}_i}(\beta_j)) \cup (\beta_k, \mu_{\tilde{A}_{i+1}}(\beta_k))$.

The condition of continuity is needed only to prevent undefined values.

Subsequently, I consider Mamdani-type FLS, that is, systems having "complete" (not singletons) fuzzy sets at the input and the output. Assume that the input and the output membership functions are the same and are isosceles triangular, overlapping two by two (typical case in applications, see figures 7,8). Also assume that every rule has a single consequent, that is rules have the form $\tilde{A}_i \to \tilde{A}_j$, while rules like $\tilde{A}_i \to \tilde{A}_j \vee \tilde{A}_k$ are not allowed. Also assume the output of the FLS is defuzzified by the center of gravity (c.o.g.) method. Then,

Proposition. If there is a cycle in the graph of the above described FLS, then there is an initial state of the system that produces an oscillatory iteration.

Indeed, there is an input real value where only one input membership function is non-null and thus selects a single output membership function. Defuzzification of the output by c.o.g. produces a real value; by the assumed conditions, for that value only one input membership function is non-null. Continuing, the cycle in the graph is thus completed.$\diamondsuit$

As a last case, I deal with Sugeno-type FLS. The equation defining the output fuzzy (overall) set (the fuzzy set value of the function represented by the fuzzy system so defined) is:

$$\mu_Y(y, \tilde{X}) = \max_j \left[\min(w_1(\tilde{X}), \mu_{\beta 1}(y), ..., (w_j(\tilde{X}), \mu_{\beta j}(y), ..., (w_q(\tilde{X}), \mu_{\beta q}(y)\right]$$

Notice that the last relation represents the union of the weighted output singletons.

The iteration is obtained by imposing $\mu_X[n](x) = \mu_Y[n-1](y)$. For allowing the iteration, the fuzzy system must satisfy the condition that the input and output membership functions must be the defined and take values on the same space. It is also apparent that, whatever the initial input membership function is, after the first iteration, the input membership function can be but a union of singletons. Consequently, with no loss of generality, I assume that the initial input in the iteration is a set of singletons with the same positions as the output singletons, (β_j, w_j), j=1,...,q. These singletons determine a set of p×q values of the input membership functions, $\mu_j(\beta_j)$, some of them possibly null.

Therefore, at iteration $n > 1, \mu_i(\beta_j[n-1])$. The input truncation values become $\max(\mu_i(\beta_j[n-1]), w_j[n-1])$. Thus, the new values of (truth-degree of) the singletons are:

$$w_h[n] = \max_{j \ s.t. h=\imath(j)} \max_i(\mu_i(\beta_j[n-1]), w_j[n-1]);$$

The fixed point condition becomes:

$$w_h = \max_{j \ s.t. h=\imath(j)} \max_i(\mu_i(\beta_j), w_j);$$

because the equality must hold for whatever time moment is considered at the two sides. The p-period condition is obtained by requesting that

$$w_h[n] = \max_{j \ s.t.h=\imath(j)} \max_i(\mu_i(\beta_j[n-p-1], w_j[n-p-1]).$$

Example. In this example I deal with a fuzzy logic system as above, with two fuzzy sets defining the input and two singletons defining the output. Consider the two input functions

$$\mu_{A1}(x) = \begin{cases} 4x(1-x) & if \quad x \in [0,1] \\ 0 & elsewhere \end{cases},$$

$$\mu_{A2}(x) = \begin{cases} 3x & if \quad x \in [0,1/3] \\ 1-(3/2)(x-1/3) & if \ x \in [1/3,1]. \\ 0 & elsewhere \end{cases}$$

The index function is described by 1→2, 2→1. The output singletons are at $\beta_1 = 0.27$ and $\beta_2 = 0.6$ (see Figure 9).

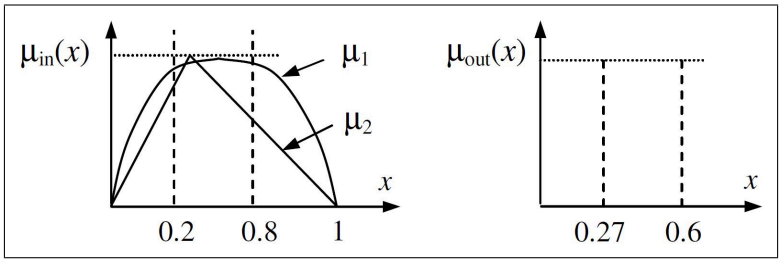


Figure 9: Explanatory graphs for the iteration.

The initial input is $\mu_X(0) = \begin{cases} 1 & if \quad x = 0.2 \\ 1 & if \quad x = 0.8. \\ 0 & elsewhere \end{cases}$

The first value of the output is $(0.27, w_1[0])(0.6, w_2[0])$ where:

$w_1[0] = \max[\min(\mu_{A2}(0.2), 1), \min(\mu_{A2}(0.8), 1)] = \max(0.6, 1-.7) = 0.6,$
$w_2[0] = \max[\min(\mu_{A1}(0.2), 1), \min(\mu_{A1}(0.8), 1)] = \max(0.64, 0.64) = 0.64.$

The output becomes the new value of the variable of the function represented by the fuzzy system. The second value of the output is $(\beta_1 = 0.27, w_1[1] = 0.6) \cup (\beta_2 = 0.6, w_2[1] = 0.64)$ where:

$$w_1[1] = \max[\min(\mu_{A2}(0.27), w_1[0]), \min(\mu_{A2}(0.6), w_2[0])] =$$
$$\max[\min(0.81, 0.6), \min(0.6, 0.64)] = \max(0.6, 0.6) = 0.6,$$

$$w_2[1] = \max[\min(\mu_{A1}(0.27), w_1[0]), \min(\mu_{A1}(0.6), w_2[0])] =$$
$$\max(0.64, 0.64) = 0.64.$$

Therefore, the output for the second iteration is identical to the output of the first, $(\beta_1 = 0.27, w_1[1] = 0.6) \cup (\beta_2 = 0.6, w_2[1] = 0.64)$; thus, the iteration result is constant. The result is a fixed point of the system.

4 Conclusions

In this paper, I focused on preparatory topics, namely fuzzy sequences, the iteration of FLS, which are functions defined on sets of (membership) functions, and on their fixed points and their periodicity. Whatever narrow is this topic, it is central in the theory of dynamic fuzzy systems, because both fuzzy logic systems without defuzzifier, and fuzzy algebraic systems apply sets of functions to sets of functions, while fuzzy logic systems with defuzzifier apply sets of functions to real numbers.

The iteration may be convergent (either in a specified metric, or uniformly in the space of the output, that is in the space of the membership functions values) to a limit fuzzy set, or may be periodical, as it has been illustrated using simple examples. More complex behaviors can appear when the output of the FLS is defuzzified. I primarily dealt with fuzzy systems without defuzzifier at the output. The systems with defuzzified output will be dealt with in subsequent papers.

Acknowledgment. I thank several colleagues and the referees for their comments and suggestions. The paper was not supported by, but is related to research performed in the frame of the Romanian Academy.

Bibliography

[1] P.E. Kloeden, *Chaotic iterations of fuzzy sets*. Fuzzy Sets and Systems 42 (1991) Volume 42, Issue 1, 5 July 1991, Pages 37-42

[2] H.N. Teodorescu: *Fuzzy oscillators*. In: H.N. Teodorescu (Editor): Fuzzy Signals and Systems (Proc. AMSE Int. Symposium on Fuzzy Signals and Systems, Cetinje, Yugoslavia, 3-5 Sept. 1990). AMSE Press, France, 1990

[3] H.N. Teodorescu: *Controlled Fuzzy Oscillators*. in vol: Engineering (Editors: R. Lowen, M. Roubens). Proc 4th IFSA Congress, 1991, Brussels, pp. 211-214

[4] H.N. Teodorescu: *Chaotic fuzzy systems: a survey*. In: H.N. Teodorescu (Ed): Fuzzy Systems and Signals, vol. 3 (Proc. AMSE Int. Symp. Fuzzy Signals and Systems, Warsaw, June 11-13, 1991). AMSE Press (France)

[5] H.N. Teodorescu: *Chaos in fuzzy systems and signals*. Vol. Proceedings of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks. Vol. 1., pp. 21-50 (Jono Printing Co., 1992, Iizuka, Japan)

[6] H.N. Teodorescu: *Verhulst-type fuzzy growth processes and chaos*. Vol. Fuzzy Systems. Proc. ISKIT'92, Iizuka, 1992. pp. 21-28

[7] T. Yamakawa, H.N. Teodorescu, E. Sofron, S. Pavel: *Fuzzy models of mechanical phenomena exhibit chaotic behavior*. Vol. Fuzzy Systems. Proc. ISKIT'92, Iizuka, 1992. pp. 29-32

[8] H.N. Teodorescu, T. Yamakawa, F. Grigoras, S. Pavel: *A hardware implementation of a chaotic fuzzy logic system*. Vol. Fuzzy Systems. Proc. ISKIT'92, Iizuka, 1992. pp. 33-38

[9] H.N. Teodorescu, T. Yamakawa, V. Belous, St. Suceveanu: *Interpretation of neuro-fuzzy systems in models in management and creativity*. Chaos generation. Fuzzy Economic Review. Nov. 1995 (No. 1), pp. 25-42

[10] S. Tamilselvan, K. Vairamanickam, K. Chandrasekhara Rao, *Generalized Fuzzy Sequence Spaces*. Int. Journal of Math. Analysis, Vol. 4, 2010, no. 45, 2235 - 2242

[11] B.C. Tripathy and A. J. Dutta, *On fuzzy real-valued double sequence spaces*, Soochow J. of Mathematics, Volume 32, No. 4, pp. 509-520, October 2006

[12] B.C. Tripathy and A. J. Dutta, *On fuzzy real-valued double sequence space* . Mathematical and Computer Modelling 46 (2007) 1294-1299

[13] K. Chandrasekhara Rao, V. Karunakaran and K. Vairamanickam, *Some Theorems on Fuzzy Matrix Transformation*, Int. J. of Math. Analysis, Vol. 4, 2010, no. 5, 243 - 248

[14] B.C. Tripathy, A.J. Dutta, *Bounded variation double sequence space of fuzzy real numbers*. Computers and Mathematics with Applications. Vol. 59, Issue 2, Jan. 2010, pp. 1031-1037

[15] B.C. Tripathy, B. Sarma, *Statistically Convergent Difference Double Sequence Spaces*. Acta Mathematica Sinica, English Series, May, 2008, Vol. 24, No. 5, pp. 737-742 (Springer-Verlag 2008)

[16] B.C. Tripathy, Sabita Mahanta, *On a class of difference sequences related to the space defined by Orlicz functions*. Math. Slovaca 57 (2007), No. 2, 171-178

[17] M. Et, A. Gökhan, and H. Altinok. *On statistical convergence of vector-valued sequences associated with multiplier sequences*. Ukrainian Mathematical Journal, Vol. 58, No. 1, 2006

[18] L. Cheng, G. Lin, Y. Lan and H. Liu, *Measure theory of statistical convergence*. Science in China Series A: Mathematics, Vol. 51, Number 12, 2285-2303, DOI: 10.1007/s11425-008-0017-z

[19] N. Subramanian, M. Basarir, *The Orlicz Space of Entire Sequence of Fuzzy Numbers*. Tamsui Oxford J. of Mathematical Sciences 24(1) (2008) 109-122

[20] D. Li, A. Laurent, and P. Poncelet, *Discovery of Unexpected Fuzzy Recurrence Behaviors in Sequence Databases*. Int. J. of Computer Information Systems and Industrial Management Applications (IJCISIM). Vol. 2 (2010), pp. 279-288

[21] Zhi-Qiang Liu, Leonard T. Bruton, James C. Bezdek, Fellow, IEEE, James M. Keller, Sandy Dance, Norman R. Bartley, and Cishen Zhang, *Dynamic Image Sequence Analysis Using Fuzzy Measures*. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 31, No. 4, August 2001 557

[22] M. Gholizadeh, M. M. Pedram, J. Shanbehzadeh, *Fuzzy Sequence Mining for Similar Mental Concepts*. Proc. Int. Multi-Conf. IMECS 2011, March 16-18, Hong-Kong, pp. 362-367

[23] Bill C.H. Chang, Saman K. Halgamuge, *Approximate symbolic pattern matching for protein sequence data*. International Journal of Approximate Reasoning 32 (2003) 171-186

[24] J.-Y. Dieulot, P. Borne, *Inverse Fuzzy Sum-product Composition and its Application to Fuzzy Linguistic Modelling*. Studies in Informatics and Control, Vol. 14, No. 2, 2005, pp. 73-78

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 761-778

A Multi-Agent System Architecture for Coordination of the Real-Time Control Functions in Complex Industrial Systems

J. Wu, R. Tzoneva

Jiang Wu, Raynitchka Tzoneva
Cape Peninsula University of Technology
South Africa, 7350 Bellville, Symphony way
Email: wolfgang.jw@gmail.com, tzonevar@cput.ac.za

Abstract: Multi-agent system architecture for coordination of the real-time control functions in complex industrial systems is presented. The problem which must be solved out is how efficiently to organize the interactions between tasks in order to satisfy the functionality and the time restriction of the system. In order to solve this problem, the treatment of the task interactions is separated from the tasks and is implemented by the proposed multi-agent system. A general three level multi-agent system is introduced to manage the interactions and schedule of tasks. A framework of building of the schedule of the tasks is also presented. Finally, the benefits of the proposed architecture are discussed.
Keywords: Middleware, Multi-agent system architecture, Real-time task, Coordination.

1 Introduction

Real-time systems are usually more complex than other systems because they are based on concurrency, have to deal with multiple independent streams of input events and to produce multiple outputs [1]. Real-time systems are not means to get the results as soon as possible, the requirements towards their operation is to get the results at a prescribed point of time within defined time tolerances [2]. Real-time systems are divided into two classes on the basis of their time tolerances (see [3] and [4]): hard real-time systems and soft real-time systems. Hard real-time systems are those whose deadlines for implementation of the tasks must absolutely be met or the system will be considered to have failed (and this failure might be catastrophic), while soft real-time systems allow for some deadlines, at least occasionally, to be missed with only a degradation in performance but not a complete failure. On the basis of the said above it can be concluded that not only the functions of the systems must be completed, but also the time restriction has to be achieved in order to obtain the goals of the real-time system. The functions of the real-time system are represented by different software tasks having a lot of interactions between themselves. In order to achieve the time requirements of the real-time system a schedule of system's tasks and coordination of their interactions must be organized efficiently.

It is supposed in the paper that every real-time function is realized by a task and each task is coded in one individual software program. This means the interactions between the functions or tasks only may be implemented via ports which are virtual/logical data connections that can be used by the programs to exchange data directly, instead of going through a file or other temporary storage location [5].

Agent theory is applied to build a multi-agent system which coordinates the execution of real-time tasks efficiently using the following capabilities of the agents: autonomy [6], social relations [7], reactivity [8] and pro-activeness [9]. The presented in the paper multi-agent system architecture consists of three agents which represent three levels of processes for interactions and schedule manipulation. The interaction agent, the lowest agent, is used to receive requisitions

for interactions from the tasks; the coordinating agent, the middle level agent, is applied to do the coordination of the tasks; the mapping agent, the highest level agent, takes charge of storing the information of how to deal with the interactions of tasks.

The development and running of tasks which realize the functionality of the real-time system are simplified by adopting this multi-agent architecture. The tasks may only focus on how to achieve the aim of the functionality and how to send and receive the interaction information from the multi-agent system. All interactions are treated by the multi-agent system.

2 The Problem of Real-Time Control Functions

The real-time system is a system in which not only functionality but also the time restriction for implementation of these functions (tasks) must be satisfied. Interaction packets between tasks and the order of execution of these tasks are two main factors which impact the period of time for completion of a given function of the system.

Traditionally, the interaction packets between two tasks are implemented directly and it is not necessary to be managed. Thus, task Ta must know how to connect to task Tb and what is the format of the message which Tb has to receive if Ta wants to send a message to Tb. This obviously increases the burden of the engineering work during the development, running and maintenance of the system.

Usually the order of execution of tasks is scheduled by a high level task which calls some type of scheduler. The scheduler evaluates the time restriction of each task and assigns a priority to the task by a specific arithmetic, and then the task possesses the computation resource (CPU and Memory) ordered according to this priority. Different strategies of scheduling have different arithmetic, but all of them consider that the task is the smallest unit to be processed. The process of determination of priority of the tasks is also necessary to be dynamic one because the environment of the real-time system is open and dynamic one.

Some methods such as Giotto [10], LET [11], TDL (see [12] and [13]) and HTL [14] try to solve the problem for scheduling by division of the task into several subtasks which have no internal interaction. These methods dodge the problem that the priority of tasks should be dynamic but this is still no an ideal solution for the problem for interaction packets management.

3 Method for Coordination of the Real-Time Control Functions

To implement real-time system efficiently, the interaction packets between tasks must be managed and the scheduling of tasks must be dynamic. The method provides a Middleware to supply the function of the interaction packet management and the dynamic scheduling control. The method is not supposed to stop a task which is executed, it schedules tasks by controlling the time which invokes and continues execution of the tasks.

3.1 Time points of the task

There is no concurrency within a task, but the tasks in the real-time structure could be executed in parallel [1]. The tasks often execute asynchronously and are relatively independent of each other for significant periods of time. The tasks need to communicate and synchronize their operations with each other from time to time. Therefore, the tasks need to be paused some times to wait for the feedback of communications and synchronization. A task is also possible to be paused by another task which has higher priority and to be re-executed again after the higher priority task releases the computation resource.

Start time point and end time point are essential time points for every task. A task is invoked with parameters and then executed, see Figure 1. That means the necessary parameters are prepared before the start time point. A task releases the computation resource which is used for execution after it completes the execution and output results of the execution at the end time point.

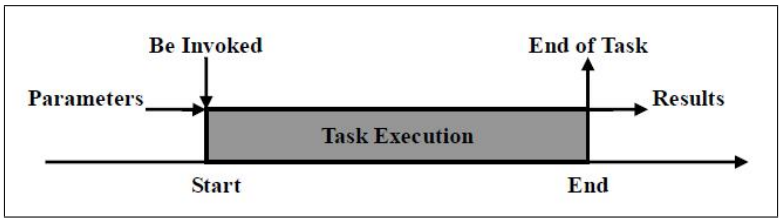


Figure 1: Task with Start and End Time Point of Execution

Almost all of the interaction packets between the tasks (exclude Loosely Coupled Message Communication) make task to pause and continue its execution when the condition is satisfied, see Figure 2. The pause can happen zero or several times during the execution of the task.

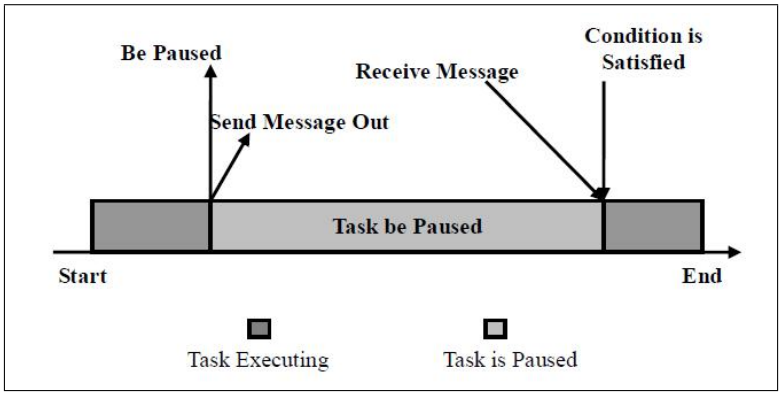


Figure 2: Task to be paused

Loosely Coupled (Asynchronous) Message Communication is a special interaction of the task, it only sends a message out and does not need to wait for any feedback from the other task, see Figure 3. As well as other interactions, it can happen zero or several times during the execution of the task. This point could be called IwF Point (Interaction without Feedback Point).

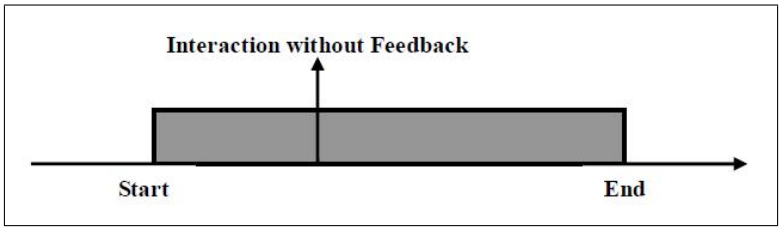


Figure 3: Task's Interaction without Feedback

As mention above, the time points of a task could be Start Point, End Point, Paused Point, Continue Point and IwF Point. Start Point and End Point happen one and only one time during a completed execution of task. Other three time points may happen zero or several times during a completed execution of a given task, and must be noted that Paused Point and Continue Point always happened in couples.

Thus, a task may have five type points which determine the run-able situation of the execution, they are start point and end point, paused point and continue point, and IwF point. The IwF point does not affect the execution of the task. Before start point and after end point, as well as for the time from paused to re-executed points, the task is not able to execute, and the computation resources (at least CPU) are able to be assigned to other tasks.

3.2 Separation of the interactions between real-time tasks

The traditional way of interaction is changed in the new method, the interaction does not directly happen between two tasks in which one is sender and other is receiver. A requisition which contains the information of an interaction packet is sent to the Middleware and then this information is sent to the target task from the Middleware after the requisition is treated by the Middleware. Thus, operation of an interaction packet from task Ta to task Tb is divided into three phases which are: interaction packet sending (Ta to Middleware), interaction packet treatment (Middleware), and interaction packet receiving (Middleware to Tb).

To make the execution of the real-time system more efficient, all possible types of interaction must be registered in the Middleware. And then, the task which sends an interaction packet does not need to know which particular task takes charge to receive the interaction packet information, the Middleware will appoint a task which has capability and also is available to receive and treat the interaction packet. This aspect makes the real-time system more flexible during the development, runtime and even maintenance. Engineers don't need to know which task must receive the interaction packet during development, any interaction packet is simply just sent to the Middleware. There are multi choices of tasks which may treat the interaction packet; the Middleware will appoint one or more tasks to treat the interaction packet by evaluating the status of execution of the real-time system at runtime. This aspect also reduces the workload and limits the effect range when a function is added or removed because the tasks which realize other functions don't need to do any modification to fit the change. The only one requirement is to register the interaction which the new task may send and receive.

A task may receive an interaction packet only when it is not started and then be invoked or it is paused and waiting to receive an interaction packet to continue execution. A task does not possess the computation resource (CPU) before the task starts and continues. The Middleware may schedule the tasks by controlling the time at which it sends the interaction packet to the target task. All changes of the status of the tasks must be sent to the Middleware to monitor the status of the real-time system and schedule the tasks.

Figures 4 and 5 compare the treatment of interaction packets of the traditional way and the new method. One interaction packet is separated into two parts, interaction request sending and interaction packet receiving. Interaction 1 in Figure 4 is separated according to the proposed method into interaction 1A and interaction 1B as can be seen in Figure 5; interaction 2 in Figure 4 is separated into interaction 2A and interaction 2B in Figure 5; interaction 3 in Figure 4 is separated into interaction 3A and interaction 3B in Figure 5. All the interactions in the section A and the section B are connected by the Middleware. The target task of section B is selected by the Middleware and the time at which the Middleware sends interaction packet to the target task is decided by the Middleware by evaluating the situation of the real-time system.

3.3 The mechanism of the coordination

The Middleware takes charge to receive interaction packet and send interaction packet to the target task. An interaction packet must be processed by several phases in the Middleware and then could be sent to the target task. There are six phases to treat an interaction packet in the Middleware. In Figure 6, each ellipse which is located in the Middleware represents a phase of

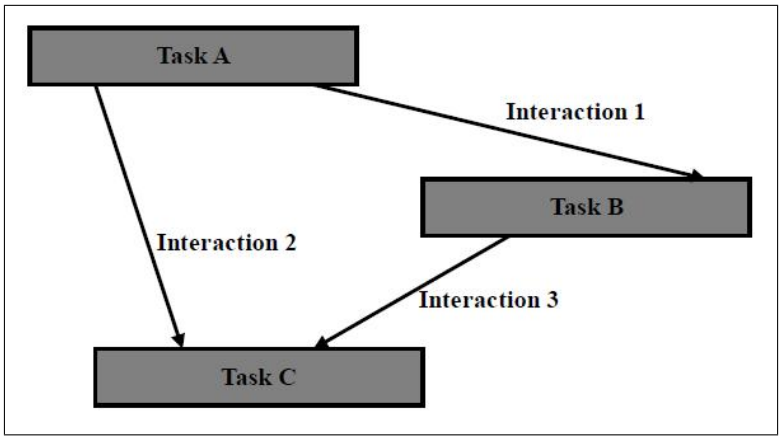


Figure 4: Tasks interact directly

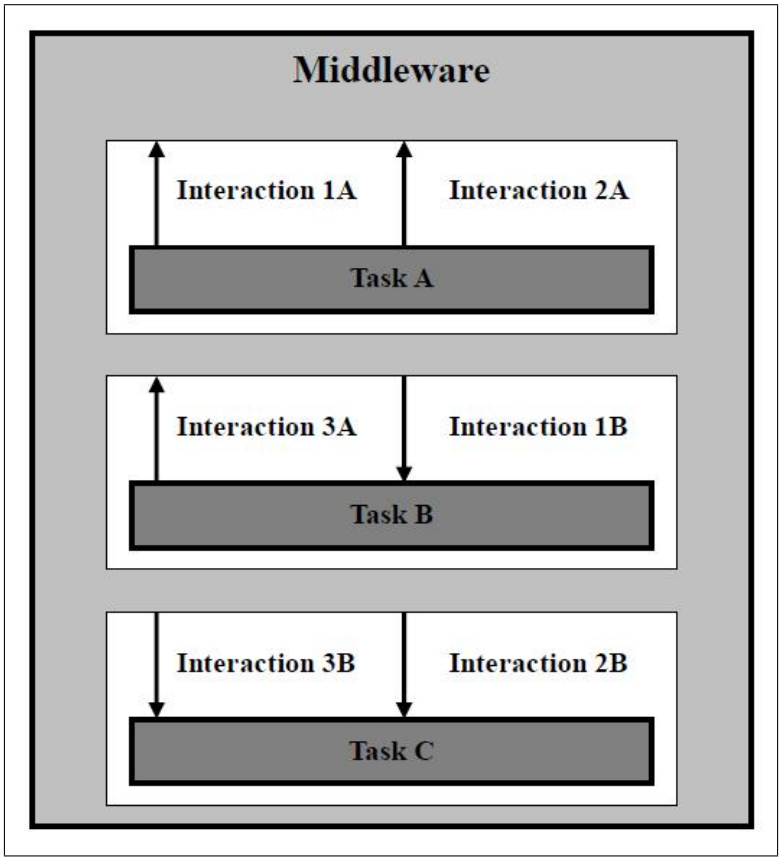


Figure 5: Tasks Interact via Middleware

treatment of interaction packet. By the order of treatment, they are: receive, change status of sending task, find the treatment plan, schedule the tasks, send and change the status of receiving task.

Receiving a requisition of an interaction packet is the first phase. The Middleware must supply a common port to receive the interaction packet. The interaction packet should have several parts. First at all, the sender of the interaction packet must be indicated. The second is the type of the interaction packet. These two parts of the interaction packet is used to find the treatment plan. The third part is the main body of the interaction packet, the message. The last part indicates the time restriction of the interaction packet. This part may not be attached with the interaction packet because some interactions may have no time restriction. After the interaction packet is sent, the task may continue or pause its execution depending on if the interaction needs reply or not. The second phase is changing the status of the task which sends the interaction packet. Any change of the status of tasks must be recorded to monitor the status of the real-time system. The status of the real-time system is foundation for scheduling of the tasks in the Middleware of the real-time system.

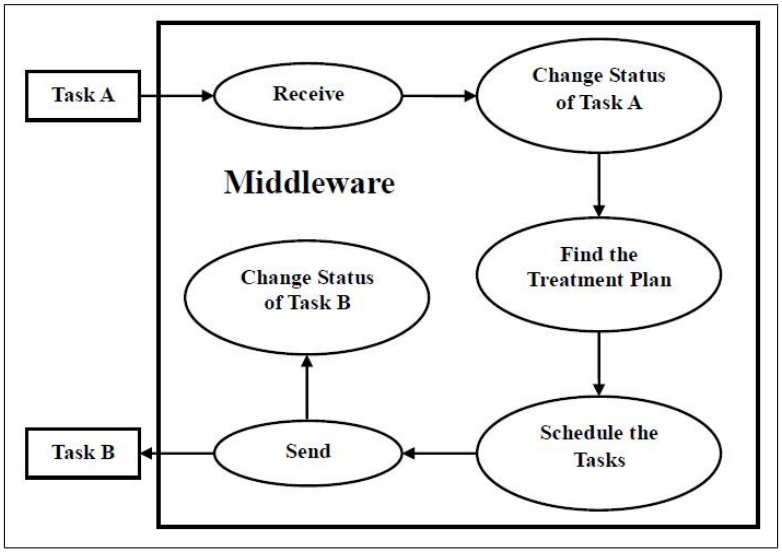


Figure 6: Treatment of Interaction Packet in the Middleware

Finding the treatment plan is the third phase to treat the interaction packet. The Middleware searches the treatment plan of the interaction packet in a database which stores the information of how to treat the interaction packets according to the sender indication and the type of the interaction packet. This search may find several plans, all of these plans must be sent to the next phase. The fourth phase is scheduling the tasks of the real-time system. The Middleware decides which treatment plan will be performed and what time for performing of the plan to be selected based on evaluating the status of the real-time system and the time restriction from the information of the treatment plan and the interaction packet. If the interaction packet is not sent out immediately, it should be stored in a waiting queue, and be sent in time to the target task. The time when the interaction packet will be sent is changeable, every time the Middleware makes the decision affects the waiting queue and the sending time of all the interaction packets which are waiting for sending could be reassigned.

There are two statuses of tasks at which they could receive the interaction packet, one is before the task starts, and other one is the tasks is paused. When a task is executing, it could not be possible to receive any information from outside of the task because the task has no internal concurrency. In addition the interaction packet could contain only the parameters to invoke a

task which is not started or could be a message to make the task to continue the execution. The status of the task which receives the interaction packet is changed when the interaction packet arrives. Thus, the receiving task's status is necessary to be changed at the Middleware for next evaluation of the real-time system. The graphic description of the treatment of an interaction packet is presented in Figure 6. The arrows represent the work flow of treatment of the interaction packet. The Middleware takes charge to find the target task to receive the interaction packet and schedule the tasks by controlling the interaction packet sending time. There are two times where the status of the real-time system changes during treating an interaction packet and reassigning the sending time of the interaction packets waiting in the queue.

4 General Multi-Agent System Architecture for Coordination of the Real-Time Control Functions

The main function of the Middleware is for a specific time to receive the interaction packet from a task, to analyze it, and to send the packet to other tasks which should do further treatment of the packet. To achieve this target, the Middleware should have some information for the system:

* Knowledge of the relationship of the tasks except the tasks of Middleware;
* The time restriction of each interaction;
* A map of the status of tasks which are running except the tasks of Middleware.

The knowledge of the relationship of the tasks is a static global view of the system. It records how to treat an interaction packet and supply the information for further treatment. The time restriction is specified in the relationship of the tasks, it is a guide to indicate when to do the communication.

4.1 Analysis of the Middleware

The objectives of the Middleware are achieved by the cooperation of the three agents.

Objectives of the Middleware

The Middleware does not perform any real function which is required for real-time system from the outside of the system, but it plays an important role in the system to coordinate the tasks which perform the real functions by controlling the communications between them.

The Middleware is also employed to reduce the complexity of the real-time system and makes the system to be developed, maintained and run easily. To achieve this aim, the new method is applied into the Middleware. Therefore, the task which performs some function in the real-time system is required to focus on completing this function only. The task communicates only with the Middleware as follows:

* Sends interactions to the Middleware
* Receives interactions from the Middleware.

The interactions between the tasks are treated by the Middleware to reveal the relationship of the tasks in the system.

Thus, the main function of the Middleware is treating the interactions between the tasks of the real-time system to make the system running correctly and in the logic way. The treatment of the interactions also has to satisfy the time restriction because it is working for a real-time control system. The correctness of the real-time system is based on the correctness of both of these two factors. So, the objectives of the Middleware are:

* Treat the interactions between the tasks to reveal the relationship of the tasks in the system

* Treat the interactions on time to satisfy the time restriction of the real-time system.

Middleware's Agents

It is proposed the Middleware functions to be implemented by three agents: an Interacting Agent, a Coordinating Agent, and a Mapping Agent. Each agent plays a distinct role to complete a specified function. These agents are self-existent, they work together by exchanging information to achieve the objectives of the Middleware. Each agent performs a specified function by itself and cooperates with others by exchange of information.

The Mapping Agent is an agent which supplies a static, global description of the real-time system. The description is expressed by the view of the relationships of the tasks of the system. This view is detailed through the records of the interactions between the tasks of the system. In fact, the rules for the running of the real-time system (for the interactions and their corresponding treatment) are predefined in the Mapping Agent only, and no undefined rules could be executed in the system.

The Coordinating Agent is an agent which supplies a dynamic, real-time status description of the real-time system. The description of the running system is expressed by the view of the status of the tasks. The status of tasks is changed in real-time; it indicates each task's working condition and some details. Usually, once a task starts, the most details are not changed except the working condition. The Coordinating Agent also schedules the tasks by scheduling the sending time of the orders based on the information which is supplied by the Mapping Agent, the view of the status of the tasks of the system and the schedule arithmetic which is predefined. The Coordinating Agent is the core agent in the Middleware. It executes the schedule arithmetic and the rules of the system by obeying the real-time status of the system.

The Interacting Agent is an agent which temporary stores the main body of the interaction information, which is the real content that should be treated by other tasks. The content of the interaction is only sent to the target tasks and no more treatment is done in the Middleware. The content should be kept until it is not useful. The Interacting Agent separates the interaction into the content and the metadata which is information for the interaction. The Interacting Agent also combines the content and the orders from the Coordinating Agent to form a complete order. The receiving and sending of the interactions and the orders are other two important functions of the Interacting Agent. The Interacting Agent is the only agent which communicates directly with the other tasks of the real-time system.

Thus, the Middleware consist of three agents and each agent represents a control level of the Middleware, a rough view of the structure of the Middleware is given in Figure 7.

Environment of the Middleware in the Real-Time System

The Middleware coordinates the tasks by controlling the interactions between the tasks in the real-time system.

In a real-time system which does not employ the Middleware to coordinate the tasks, a task has to spend a lot of time to treat the interactions with other tasks. In this process the developer

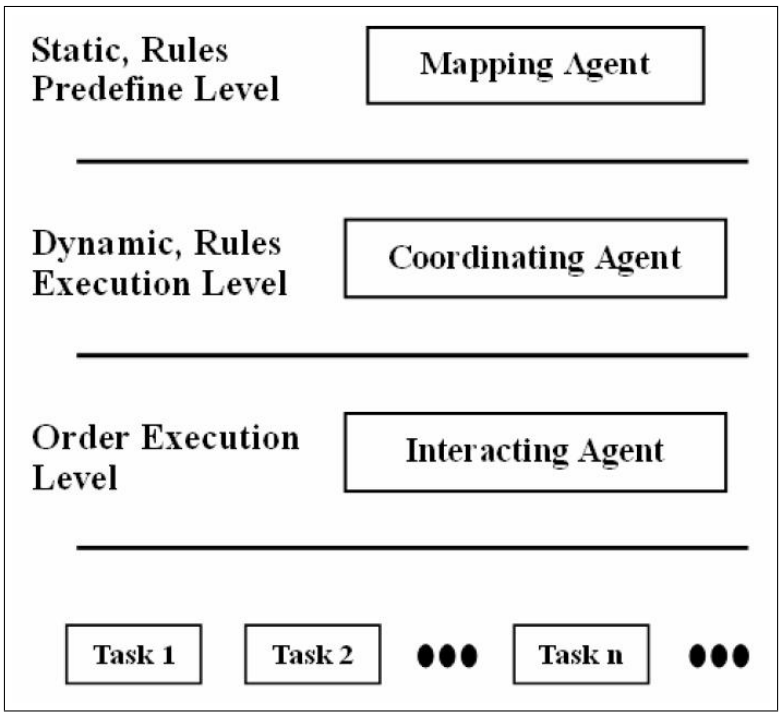


Figure 7: A rough view of the structure of the Middleware

has to know all the details of the tasks which are supposed to be interacted. Additionally various tasks may be forced to do modification when one task is modified. It is also difficult to check the running status of the tasks and difficult to add or remove calculating resources from the system.

The development, maintenance and running of the tasks in the real-time system which employs the Middleware are much simpler than others because they may interact with other tasks by a few steps which are simple and standard. For instance, a sensor task collects value from an electrical sensor. The value then should be sent to a database to be stored and sent to a calculation module to do the further treatment. In a real-time system which does not employ the Middleware, the sensor task should do two interactions with other two tasks to treat one value, and also should have the information of other two tasks. In the real-time system which employs the Middleware, the sensor task should only send the value to the Middleware and only need to have the information of the Middleware. The difference between these two ways of treatment is present in Figure 8.

An interaction usually transfers the information from an initiator (task) to a receiver (task). There are two basic parts of the interaction, one is the content of the information, and other is the order of the information which indicates the receiver how to treat the content of the information. The content of the information could be null when the initiator only wants to inform the receiver that the status of the initiator is changed. The receiver mode is the same in the system with Middleware and in the system without Middleware.

In the system without Middleware, the initiator sends the interactions to the receiver directly and without middle treatment between the initiator and the receiver, so the interaction which is sent by the initiator must exactly be the same as the interaction which is received by the receiver.

In the system with Middleware, the interactions are treated by the Middleware. Therefore, the interaction which is sent from the initiator does not indicate how the interaction will be treated by the receiver, but it indicates which type the interaction is. The type of the interaction

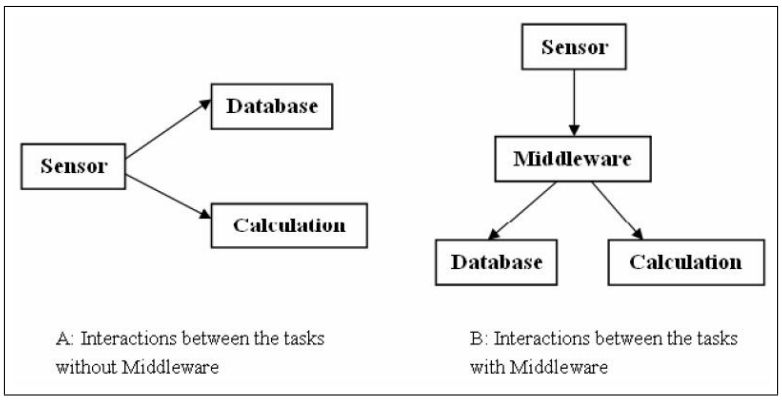


Figure 8: Difference of the interactions between the system with and without Middleware

and the type of the task constitute a unique symbol which could be analyzed in the Middleware. The Middleware may parse the symbol and obtain several orders to the receivers and then send to them these orders with the content of the information.

Figure 9 presents different types of the interactions. The grey area of the two charts are the same, they represent that the orders which the receiver has received are the same. The different is the initiator which only sends one interaction to the Middleware in the system which has Middleware, and the initiator sends two interactions to the receivers in the system without Middleware. Actually, no matter how many tasks receive orders to treat the content of the information, the initiator only needs to send one interaction to the Middleware in the system which is with Middleware. The Middleware parses the task type and the interaction type to obtain the receivers and the orders and then sends the orders to receivers. In the system without Middleware, the initiator has to send separate orders to every receiver by itself. In a complex system, the number of orders is very big, it could be hundreds even thousands orders that have to be sent by the initiator itself.

The workflow of treatment of the interaction

There are three types of interactions, and the different type of interactions should be treated by different steps. The interactions are:

* Task register interaction;

* Task status change interaction;

* The task should communicate with other tasks.

The procedure of treatment of an interaction starts once the Middleware receives the interaction. The whole procedure usually consists of different steps for different type of interactions.

For the first type interaction, the procedure is:

* Assign a task No. to the task;

* Return the task No. to the task;

* Register the task at the Middleware.

For the second type interaction, the procedure is:

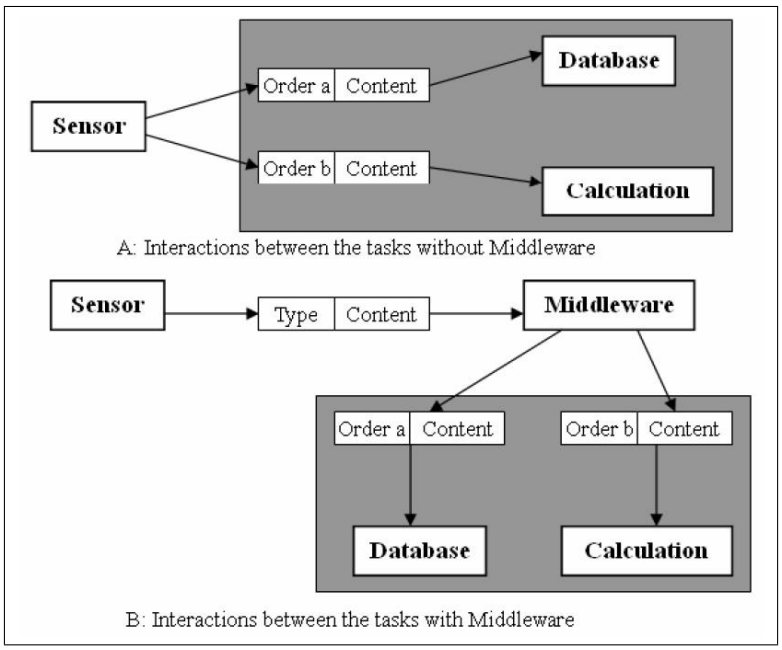


Figure 9: Different type of interaction procedures

* Return a received signal to the task;
* Change the status of the task.

For the third type interaction, the procedure is:

* Return a received signal to the task;
* Separate the metadata from the interaction;
* Change the status of the task;
* Find the order plans for the interaction;
* Select a plan to be executed;
* Combine the order and the content;
* Execute all orders.

The whole workflow of treatment of the interaction is presented in Figure 10.

The interactions between the agents

According the Figure 10, there are two bounds between the agents, they are: the bound between the Mapping Agent and the Coordinating Agent, and the bound between the Coordinating Agent and the Interacting Agent. Each bound has two data transfer directions. In additional of these, the Interacting Agent also has other two data transfer directions to communicate with the outside of the Middleware (receive an interaction and execute the order). Therefore, there are six different data transfers in the Middleware, and they are listed by the order of treatment:

* From outside of the Middleware to the Interacting Agent;

* From the Interacting Agent to the Coordinating Agent;
* From the Coordinating Agent to the Mapping Agent;
* From the Mapping Agent to the Coordinating Agent;
* From the Coordinating Agent to the Interacting Agent;
* From the Interacting Agent to outside of the Middleware

The description below gives the types of data transfer for the third type of the interactions of a task with other tasks.

* In the first data transfer, the intact interaction is transferred from outside of the Middleware to the Interacting Agent. And then the whole interaction is separated into metadata and the content. The content is stored in the Interacting Agent and the metadata is sent to the Coordinating Agent.
* The second data transfer is: the Coordinating Agent receives the metadata from the Interacting Agent. The metadata should indicate the initiator of the interaction, and then the Coordinating Agent is able to change the status of the task which is the initiator.
* The third data transfer is: the Mapping Agent receives the metadata from the Coordinating Agent. The metadata should indicate the unique interaction type, and the interaction type will be used to find the order plans from the database.
* The fourth data transfer is: the Mapping Agent returns the search results which are a set of order plans to the Coordinating Agent. Each plan consists of several orders. The scheduler is able to select a plan to be executed on the bases of the status of the tasks.
* The fifth data transfer is: the Coordinating Agent sends the orders to the Interacting Agent to combine the content which was separated before. All of these orders should be able to obtain the correct content which should be carried to be sent to other tasks.
* The last data transfer is: the Interacting Agent sends the intact orders to the tasks which should receive the orders.

For the other two types of the interactions only the first and the second data transfer are involved.

The data containers in the agents

To supply a static, global description of the real-time system, the Mapping Agent involves a database to store the information of the system. The database should have the information of the task type, its interaction type and how to deal with the interaction. It should be able to return the order plans by supplied task type and the interaction type. To supply a dynamic, real-time status description of the system, all running tasks' status must be stored in a date container and must be changed dynamically. It is the basis for scheduling the tasks. No matter which schedule arithmetic is applied, the real-time status of the system is always useful for the scheduling.

The Interacting Agent keeps the content of the interaction in a data container. The content should be connected to the order must be removed when it will not be used again. The data in the Mapping Agent is static and should only be read. Other two data containers have both

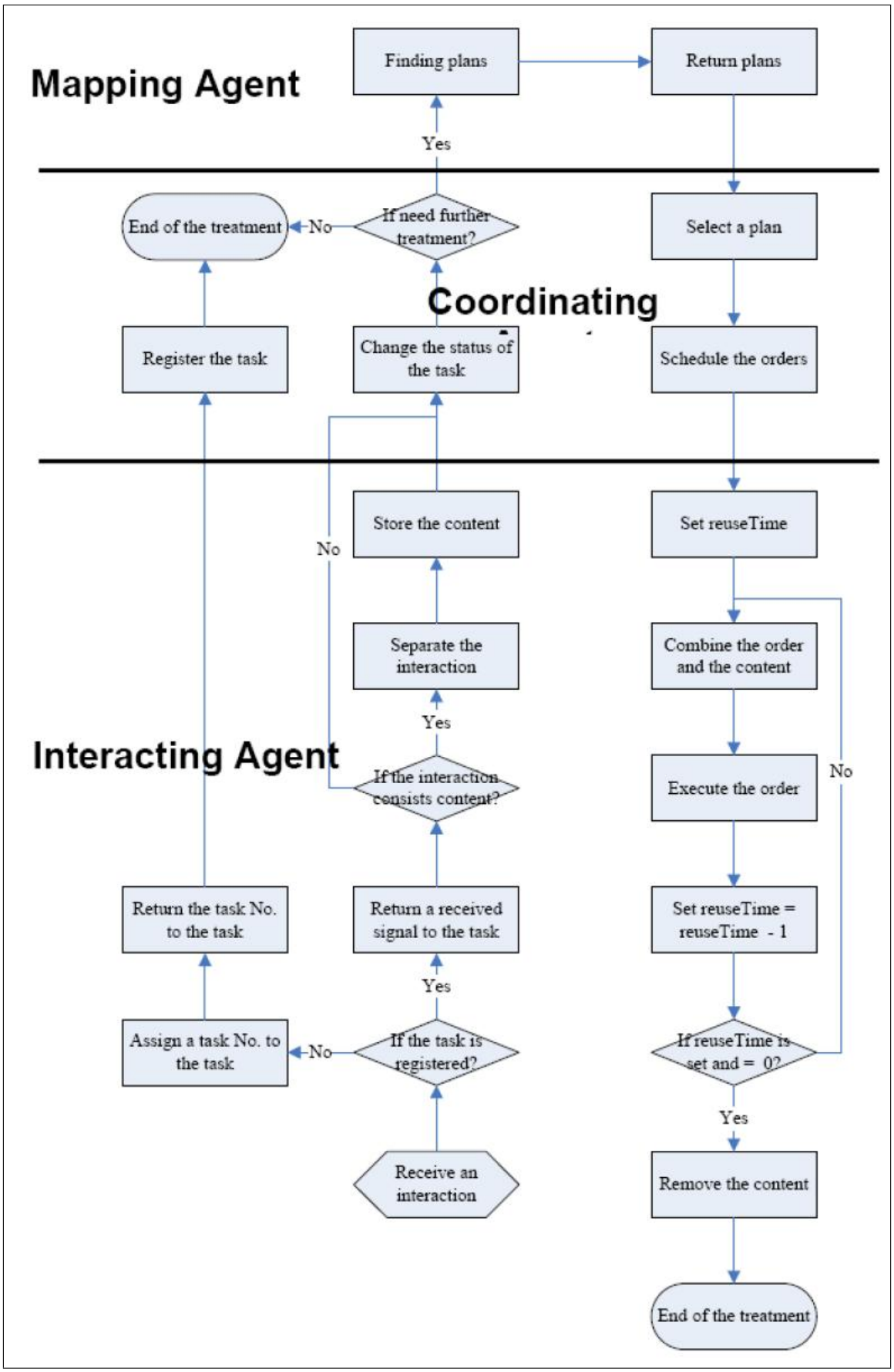


Figure 10: The workflow of treatment of the interaction

read and write (add, remove, change) operations. For each of these, it is possible that two write operation to happen at the same time or a read and a write operation to happen at the same time. This may lead to potential dangerous (unpredictable results) of the data. To avoid this dangerous situation, no other operation may be allowed to be executed when a write operation is happening.

4.2 Architecture of the Middleware

The Middleware is working in a real-time system, and has to communicate with other tasks of the system. Each task is a self-existent program, and they may be distributed in the network and working concurrently. A real-time control system usually is a concurrent system and requests the coordination component to have high response efficiency. Figure 11 presents the detailed architecture of the Middleware.

Data is the core

Three data containers exist to store the information of the system. Each data container is the core of the agent in which it stays. The agents' actions are all around these data containers. The data containers and specific agents are listed below in a Table 1:

Data	Agent
Relationship of the tasks	the Mapping Agent
Running Status	the Coordinating Agent
Content of Interactions	the Interacting Agent

Table 1: Data Containers and Specific Agents

Queues and communicators

The queues and the communicators are the channels of the transfer of the data between the agents.

A queue is a data structure which may store elements and order them in a FIFO (first-in-first-out) manner. The communicators are pair operations on the queue, one is to get the element and other is to put the element. Seven queues exist in the Middleware to transfer data between the agents. They are listed below in a Table 2:

Queue	From (get)	To (put)
qSisiIn	SISI	the Interacting Agent
qInCoor	the Interacting Agent	the Coordinating Agent
qCoorMap	the Coordinating Agent	the Mapping Agent
qMapCoor	the Mapping Agent	the Coordinating Agent
qCoorInA	the Coordinating Agent	the Interacting Agent
qCoorInB	the Coordinating Agent	the Interacting Agent
qInSisi	the Interacting Agent	SISI

Table 2: Queues in the Middleware

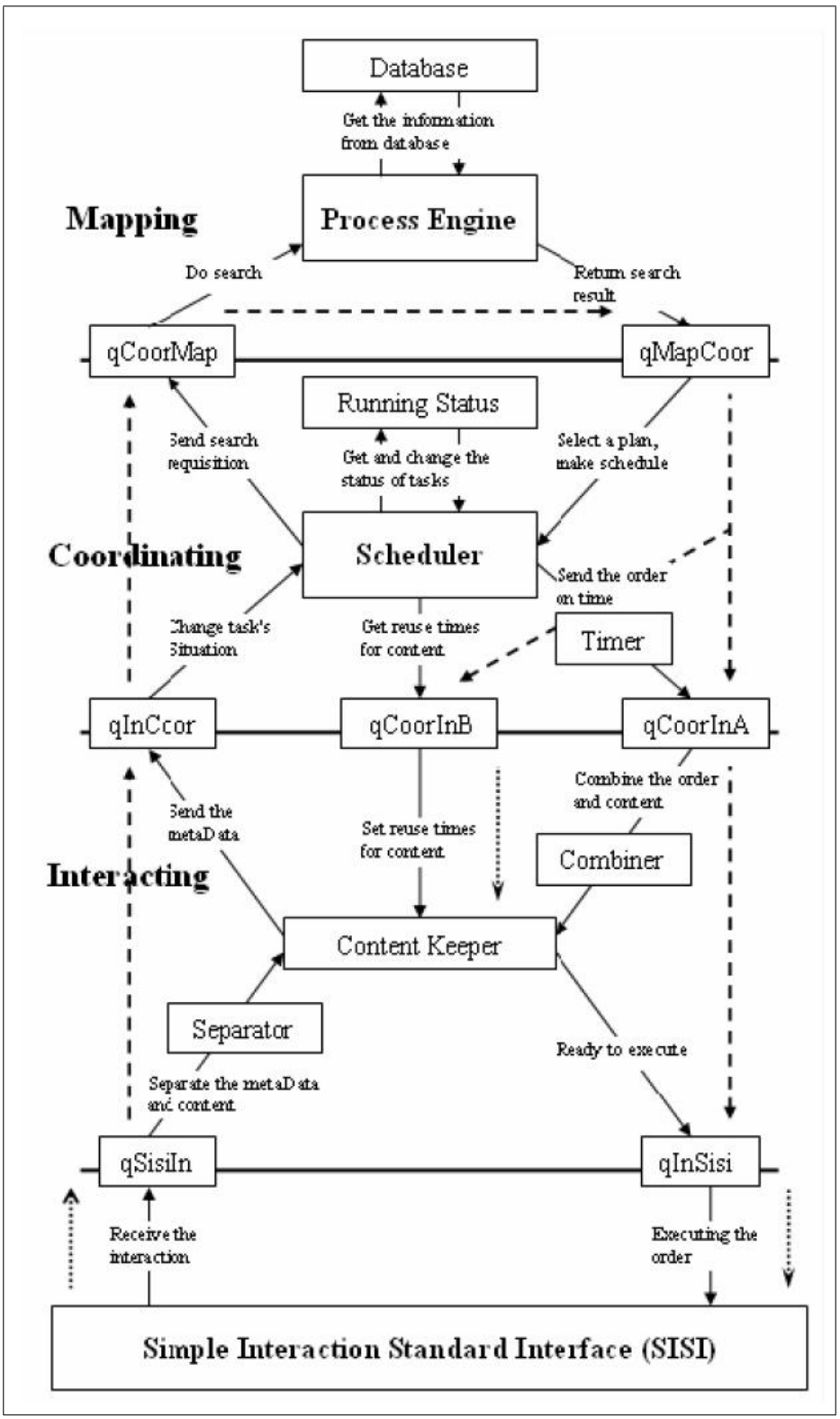


Figure 11: Detailed Architecture of the Middleware

Simple Interaction Standard Interface (SISI)

Simple Interaction Standard Interface (SISI) is a special module of the Interacting agent of the Middleware. It is the only module that communicates with the tasks directly. It receives the interaction packets from the tasks, put them into the qSisiIn and sends the order packets from the qInSisi to the tasks.

Threads

Two types of threads exist. One is running during the lifetime of the Middleware, and the other is started by other thread. The first type of the threads is named "Listen Thread" because they always listen the queues and create a second type threads which are executing the real treatment. So the second type is named "Function Thread".

Listen threads are established when the Middleware starts and they run during the lifetime of the Middleware. Each of them is monitoring a specific queue to check if there is any element in the queue and get the element from the queue by the rule "first-in-first-out". To shorten the time of the response, the listen thread creates a function thread to do the treatment of the element which is gotten from the queue and go back to monitoring the queue immediately. A Process Head which is a special thread listens the port to receive the interaction packet in SISI. It creates a new thread to do the first treatment of the interaction packet and put the result as an element into the queue qSisiIn.

Function threads execute the real treatment of the elements. They usually read or write something from the data container, and also obtain a new element for the further treatment except no further treatment is needed. If the thread obtains a new element which needs further treatment, the function thread then invokes a put function to put the element into the queue which receives the element for next step treatment. A Process End is created by a listen thread QGetInSisi. It sends the order packet to the tasks which should receive the order packet. It is the last function in the treatment cycle.

Timer

Timer is a module in the Coordinating Agent of the Middleware. It executes the specific function in a specific time. The specific function here is put the element into the queue qCoorInA. The element arrives to the queue qCoorInA at the specific time.

The Middleware receives the interactions from the tasks and sends the orders to the tasks. Each agent has a data container to store the information which should be treated. The queues and the listen threads and the put functions provide the capability that the agents may communicate with each other. The real treatments are executed in the threads which are created by listen threads.

5 Scheduling in the Coordinating Agent

Coordinating Agent does real coordination in the Middleware for the whole real-time control system. It schedules tasks in the real-time system and manages the interactions between them. It is placed between the Interacting Agent and the Mapping Agent and consists of a scheduler and running status.

Scheduler is used to organize executions of tasks of the system in the Coordinating Agent. It schedules tasks based on the clock, the status of the system, the meta-data of interaction packets from the Interacting Agent and the sets of treatment plans from the Mapping Agent. Scheduler decides which plan of treatment is used for response to the interaction packet based

on the status of the system. Scheduler also decides at what time to send the order to the Interacting Agent. That is the scheduler decides which order or orders are used to response to the requisition of interaction packet, and at what time the order is sent to which task. Scheduler controls the execution of the system dynamically through control of the execution of the tasks and the interaction packets between the tasks.

Running status is a real-time table to record execution details of the tasks which are running at the time. Any change of the status of the task is recorded in the running status after scheduler receives the meta-data of the interaction packet from the Interacting Agent. Running status supplies the information to the scheduler for scheduling the orders. The order which is sent to the Interacting Agent is also recorded in the running status. Several tasks may run the same code of a program at the same time or a same code of a program may be run several times in the system. A code of the task exists in the running status to indicate which code of the program is run by the task.

6 Conclusions

The Middleware coordinates the tasks in a real-time control system by scheduling them and managing the interaction packets between them. The Middleware could be thought as a post office. The letters (interaction packets) are collected by postman (Interacting Agent) first. Then the post office (Coordinating Agent) sorts the letters based on the yellow page (Mapping Agent). At the end, the postman sends the letter to the addressee.

The whole process of treatment of an interaction packet in the Middleware could be divided into three phases:

* Requisition collection;
* Finding solution;
* Processing of the solution.

Comparing the traditional solutions with the proposed method, the benefit of the method is obvious. First at all, the whole real-time control system is separated into three parts: function realization, schedule execution and system architecture. This makes each part of the system to be developed in more effective and professional way. Each of these three parts is easy to be changed. The table of interaction packets may be changed to realize another purpose. The scheduler may be realized by other type arithmetic for changing the strategy of the scheduling. The tasks may be added, removed or upgraded easily because only some corresponding changes are necessary to be added to the database which is placed in the Mapping Agent.

System status is easy to be monitored during system run, because this status is a snapshot of the system.

Acknowledgements

The investigations are funded by the National Research Foundation of South Africa under the grant 62364 “Substation Automation and Energy Management Systems”.

Bibliography

[1] H. Gomaa, *Designing Concurrent, Distributed, and Real-Time Applications with UML*, Addison-Wesley, 2000.

[2] A. Gambier, Real-time control systems: a tutorial, *The Fifth Asian Control Conference*, Melbourne, Australia, 2004.

[3] C. Liu, J. Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *Association for Computing Machinery*. Vol.20, No. 1, pp. 46-61, 1973.

[4] E. Jensen, C.Locke, H. Tokuda, A time-driven scheduling model for real-time operating systems, *Proceedings of the IEEE Real-Time Systems Symposium*, 1985.

[5] http://en.wikipedia.org/wiki/Computer_port_(software)

[6] C. Castelfranchi, Guarantees for autonomy in cognitive agent architecture. In *M. Wooldridge, N. Jennings, (eds), Intelligent Agents: Theories, Architectures, and Languages*. LNAI Vol.890, pp.56–70,Springer-Verlag: Heidelberg, 1995.

[7] M. Genesereth, S. Ketchpel, Software agents. *Communications of the ACM*, 37(7), pp.48–53, 1994.

[8] H. Chebeane, F. Echalier, Towards the use of a multi-agents event based design to improve reactivity of production systems, *Computers & Industrial Engineering*, Vol.37, No.1-2, pp.9-13, 1999.

[9] M. Wooldridge, N. Jennings, Intelligent agents: theory and practice, *The Knowledge Engineering Review*, 10(2), pp.115-152, 1995.

[10] T. Henzinger, B. Horowitz, C. Kirsch, Giotto: A time-triggered language for embedded programing, *Proceedings of the IEEE 91*, pp.84–99, 2003.

[11] T. Henzinger, C. Kirsch, M. Sanvido, W. Pree, From control models to real-time code using giotto, *IEEE Control Systems Magazine* 23(1), p.50–64, 2003.

[12] E. Farcas, C. Farcas, W. Pree, J. Templ, Transparent distribution of real-time components based on logical execution time, *In Y. Paek, R. Gupta (eds.): LCTES, ACM*, pp.31–39, 2005.

[13] C. Farcas, W. Pree, A deterministic infrastructure for real-time distributed systems, In: *OSPERT 2007 Workshop on Operating Systems Platforms for Embedded Real-Time applications*, 2007.

[14] A. Ghosal, T. Henzinger, D. Iercan, C. Kirsch, A. Sangiovanni-Vincentelli, Hierarchical coordination language for interacting real-time tasks, In: *Proceedings of the 6th ACM International Conference on Embedded software, Seoul, Korea*, ACM (Oct 2006), pp.132–141, 2006.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 4 (December), pp. 779-793

Ubiquitous Containerized Cargo Monitoring System Development based on Wireless Sensor Network Technology

C.M. Yeoh, B.L. Chai, T.H. Kwon, K.O. Yi, T.H. Kim, C.S. Lee, G.H. Kwark, H. Lim

Chee-Min Yeoh, Bee-Lie Chai,Hyoteak Lim
Dongseo University, Churye-2-Dong, Sasang-Gu, Busan, 617-716, South Korea E-mail: yeohcm@dit.dongseo.ac.kr, beelie@dit.dongseo.ac.kr, htlim@dongseo.ac.kr

Tae-Hong Kwon, Ki-One Yi, Tae-Hun Kim, Chang-Sup Lee, Gwang-Hoon Kwark
Media Device Lab, Dong-A University, 840, Hadan-2-Dong, Saha-Gu, Busan, 604-714, South Korea
E-mail: thkwon@dau.ac.kr, koyrim@dau.ac.kr, rider7979@dau.ac.kr, cslee@dau.ac.kr, paxpia@dau.ac.kr

Abstract: Due to globalization, global trade is strongly growing nowadays. The use of containers has significantly increased and bringing the change on the shape of the world economy. Thus, monitoring every single container is a big challenge for port industries. Furthermore, rapid development in embedded computing systems has led to the emergence of Wireless Sensor Network (WSN) technology which enabled us to envision the intelligent containers. This represents the next evolutionary development in logistics industry to increase the efficiency, productivity, security of containerized cargo shipping. In this paper, we present a comprehensive containerized cargo monitoring system based on WSNs. We incorporated tilt/motion sensor to improve the network convergence time of container networks. Moreover, we periodically switch the nodes into sleeping mode to save energy and extend the lifetime of the network. Based on the technical implementation on a real container vessel, we strongly believed that our design which employed WSN technology is viable to be implemented in container logistics to improve port services and provide safe transport of containerized cargo.
Keywords: Containerized cargo monitoring, sensor node, WSN, Low Power Listening

1 Introduction

Containerized shipping made the world smaller and the world economy bigger [1]. Since its introduction in 1960s, containers represent the standard unit-load concept for international freight. Nowadays, the world trade is mostly accomplished by the aid of container using different channel of transportations including railway, airway, maritime, truck and others. These have an extraordinarily large impact that changed the shape of global economy. There had estimated in 2008 more than 500 million Twenty-foot Equivalent Unit (TEU) of containers turnover in the world-wide most important container ports, and, compared with ten years ago, only had around 200 million TEU of containers [2].

The global competition process has been the main external driving force for the new challenging era of the port industry [3]. Ports have to redefine their business process, business strategies as well as service characteristics in order to create and sustain competitive advantage. Port authorities have introduced various add-on services such as safe transport, cargos location and status tracking, rapid customs clearance and so on. In addition, several port authorities have incurred fire damage costs about few hundred million dollars due to weak security assessment and poor cargo monitoring. The damage could be reduced if proper freight monitoring

is deployed and rescue works could be done as quickly as possible. Besides that, a numbers of international security laws have been enforced to detect and take preventive measures against security incidents affecting ships or port facilities. For instance, International Ship and Port Facility Security Code (ISPS), Container Inspection Program (CIP), Container Security Initiative (CSI), Customs-Trade Partnership Against Terrorism (C-TPAT) and so on are a comprehensive set of measures that in response to the perceived threat to ships and port facilities [4–6]. In order to provide a secure transport for containerized cargo which could improves port services at the same time satisfying the international security laws, port authorities should deploy an autonomous cargo monitoring system extensively. However, one of the challenges is container logistics consists of huge number of nodes that need to be monitored. With the old wireless technology, it is not possible to provide coverage in such dense environment. Thus, one of the possible solutions is to adopt Wireless Sensor Networks (WSN) in the containerized shipping for providing an autonomous cargo monitoring system. WSN is formed by a large numbers of nodes which are capable to measure ambient condition related to the environment surrounding the nodes and transform them into electric signal. Each node is able to communicate among each other and collaboratively transmit data through network to a Base Station (BS). The power of WSN lies in the ability to deploy a large number of nodes that could assemble and configure on its own. These advantages make them very promising in wide range of application from military, health, education, commerce and etc.

Sensor nodes life time is limited to the life time of its battery. Sensor nodes may not be charged once their energy is drained, hence the lifetime of the network depends critically on the battery energy. Nodes mainly drain their energy during the communication process rather than in data processing state. In traditional MAC protocol, most of the energy is wasted by idle listening, collision, overhearing and others. Since node does not able to predict when it will receive message from its neighbors, it must keep its radio in receive mode at all the times. Many measurements have shown that idle listening consumes 50% to 100% of the energy required for receiving. For example, Stemm and Katz measure that the idle:receive:send ratios are 1:1.05:1.4 [15]. Besides, for collision, a transmitted packet has to be discarding if it is corrupted and then follow on re-transmissions which will increase energy consumption and latency. A new protocol is needed in order to decrease the initial topology of the network, to reduce the waste of energy from the source and extend the lifetime of the network.

The remainder of this paper is organized as follow: Section 2 describes the basic requirements that need to fulfil when deploying WSN in container networks. Section **??** introduces the overview of our proposed containerized cargo monitoring system and methodology for improving network convergence time. Section 4 explains how the proposed solution deploy the low power listening mechanism. Section **??** provides node and antenna localization in the container. Next, Section 6 presents the experiment and is followed by technical implementation in Section 7. Finally, we conclude this paper in Section 8.

2 Basic Requirements of Container Network

This section discusses about various requirements that needed to deploy WSNs-based autonomous cargo monitoring system. Following are several basic requirements that needed to be fulfilled in order to perform optimally in autonomous logistics network.

- **Robustness and reliability**
  The cargo monitoring systems must equip with the ability to work and cope in rough environments. There are numerous of unexpected conditions in logistics network that could affect the entire system. As a result, autonomous cargo monitoring system should

be robust and reliable enough to continue their task even though system breakdown in the neighborhood.

- **Low power consumption**
  The main source of energy for sensor nodes is battery power. Energy is one of the limitations in sensor nodes. Sensor node with low power consumption could yield longer serving period without maintenance.

- **Minimum human intervention**
  In seaport, there are more than 10,000 containerized cargos that need to be tracked in the monitoring system. Port authorities do not afford to present at every single cargo for maintenance or operation. Hence, cargos monitoring system should be autonomous and minimize the need of human intervention.

- **Sensor network longevity**
  Sensor nodes that run for 9 months from non-rechargeable power source would be preferred for sensor network nowadays. In logistics environment, typically battery life should sustain vary from 9 to 12 months. This is because it is not practical to change battery frequently for every container in such dense environment.

- **Self-organization**
  Autonomous is one of the main features provided by wireless sensor network technology. In logistics network, containerized cargos have high mobility from time to time. A lot of containers can join and leave the network when necessary. Thus, nodes need to have self-governing and self-reconstructed capabilities in this environment.

- **Small monitoring device**
  Monitoring devices that deployed in containerized cargos should not be an obstruction for normal cargo operation. Monitoring devices should be small in size and allow cargo operations run as usual. In addition, a small, tough enclosure should be used to protect the sensor nodes from dust or disruption.

3 System Design

In general, there are two types of transportations that are involved in the whole process of containerized cargo shipping such as land carriage and maritime carriage. For long distance shipping, maritime carriage is the best choice among other ways of transportation due to its economical cost. In maritime carriage, more than 10,000 containerized cargos are shipped simultaneously in one huge vessel to a designated seaport. When the vessel reaches the seaport, truck is used to transfer containerized cargos to its destination. The whole process of shipping might take few weeks or months to reach its final destination. As a result, the condition and safety of containerized cargos should be closely monitored in this long shipping process. In order to monitor the condition of the containers throughout the shipping process, we designed a comprehensive containerized cargos monitoring system. We adopted Wireless Sensor Networks (WSN) technology into our monitoring devices to form an autonomous logistics network. In our design, three main networking devices involve in the monitoring system to form a network. A basic unit of networking device, namely Container Network Device (N/D) is positioned in every single container in the system. Container N/D is the main source to collect container's condition. Container N/D will be discussed in more detail in the following section. Next, a Coordination Point (C/P) is placed on few tall pillars of container terminal or the Wing Bridge of vessel to

communicate with Container N/D as well as to collect the data from Container N/D and then transmit to control room for future processing. In order to link two separate networks, a Container Network Bridge is used in the system. With bridging, traffic from one network is managed and rebroadcasted to adjacent network segments accordingly.

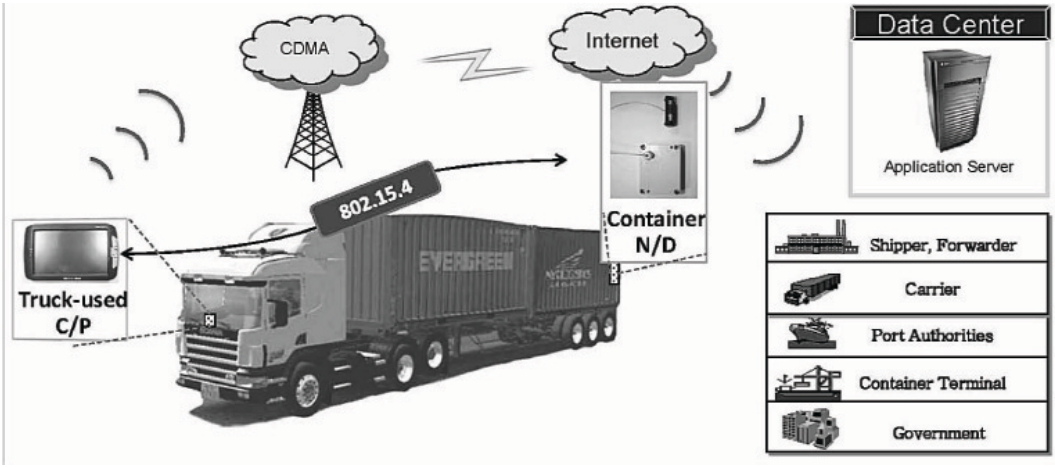


Figure 1: Land Carriage Communication

A secure containerized cargo monitoring system should not stop until seaport. Thus, our cargo monitoring system provides end to end security until cargos final destination such as warehouse or distribution center. We used two core networking devices during the shipment through land carriage as shown in Figure 1. We placed a truck-used Coordination Point in front of the driver seat to communicate with the Container N/D inside the container that is being transported by the truck. Container N/D sends the sensing data to the truck-used C/P with IEEE 802.15.4 complied interface [8]. Through the truck-used C/P, truck driver is able to discover the containers information such as identity of container, Container N/Ds id, temperature and humidity of the container and so on. The truck-used C/P is equipped with Code Division Multiple Access (CDMA) modem to forward the acquired data to the main server through CDMA network for further processing. As a result, the main server will obtained updated data at anytime and at anywhere from the moving truck.

3.1 Container Network Device

Container Network Device (N/D) is the basic unit of networking device which is placed in every single container in the system. The main responsibility of N/D is to sense the containers condition and transmit those sensing data to Coordination Point (C/P). N/D is capable to form a Multi-hop Cluster Tree network automatically among N/Ds and collaboratively pass on the data to C/P when the N/D is situated out of radio range of the C/P. Each cluster has one parent node to coordinate the nodes in the cluster and forward its cluster nodes data to C/P. Like any other commercial sensor nodes such as TelosB, MicaZ and etc [9,10], N/D is powered by two AA batteries and has limited resources in term of computational capability, storage and so on. Thus, low power listening mechanism is deploy onto the N/D which will be discussed in detail in next section. Additionally, application and tasks for N/D are designed with low resource consumption and efficiency in mind to prolong the N/Ds lifetime.

Table 1 shows the basic components of Container N/D. We employed TinyOS 2.1 which is the latest up-to-date version of component based operating system targeting for wireless sensor network in our N/D [11]. This version has improved various features of TinyOS version 1.0 and

Table 1: Basic Components of Container N/D

Parameter	**Values**
MCU	MSP430 Family
Operating System	TinyOS 2.1
Sensor	Tilt Sensor SA1
	Temperature/Humidity
	3.5 volt

supported low power CC2420 RF transceiver extensively which is required in our design. Besides temperature/humidity sensor, we incorporated tilt sensor into N/D. Tilt sensor is used to detect the motion of the container which will be discussed in detail in Section **??**.

3.2 Coordination Point

The main function of Coordination Point (C/P) is to communicate with NDs as Base Station and forwards the data from N/Ds to main server for further processing. Thus, C/P is equipped with wireless interface complies with IEEE 802.15.4 standard to communicate with N/Ds as well as internet connectivity via Ethernet or CDMA network to transmit the acquired data to the application server. Unlike N/D, C/P is powered by unlimited power supply and has superior processor and storage. Generally, there are two types of C/Ps that are used in our system which are seaport-used C/P and truck-used C/P. The differences between these two C/Ps are antenna type, radios range, touch screen option, and so on. The seaport-used C/P which is placed on a tall pillar of container terminal or on Wing Bridge of container vessel. It has higher-end of antenna which could provide wider coverage if compared to truck-used C/P. On the other hand, truck-used C/P which is used by land carriage and is placed in front of drivers seat. This C/P has an interactive 7-inch touch screen that could provide the necessary container's information such as container identity number, N/Ds id, temperature and humidity of the container that being transported by the truck. This C/P uses CDMA interface to forward the acquired data from N/D to application server for further processing. As a result, user could get the containers condition at any time and at anywhere during the shipping.

3.3 Nodes Motion Detection for Improving Network Convergence Time

In the beginning of this paper, we have discussed one of the essential basic requirements to form a logistics network with WSNs is self-organization. Especially in logistics network, thousand of containerized cargos join and leave the network in a short period. Thus, nodes that attached to containers should be self-aware and react according to the current state of the environment. However, it is not practical for nodes to wait for triggering from Status Timer then only broadcast the topology changes when a node is leaving. This is because some nodes might mistakenly send packet to the leaving node as that node still exist in the nodes neighbor table. As a result, real time motion detection is needed for improving network topology reconstruction time when there is a node leaving the network.

As a result, we proposed to incorporate tilt/motion sensor into Container N/D to detect the movement of the container. The tiny tilt sensor or tilt switch has only 3mm x 8mm in size [12]. This sensor has high sensitivity and consumes ultra low power ranging from 0.00025 to 5mA which is suitable for our application.

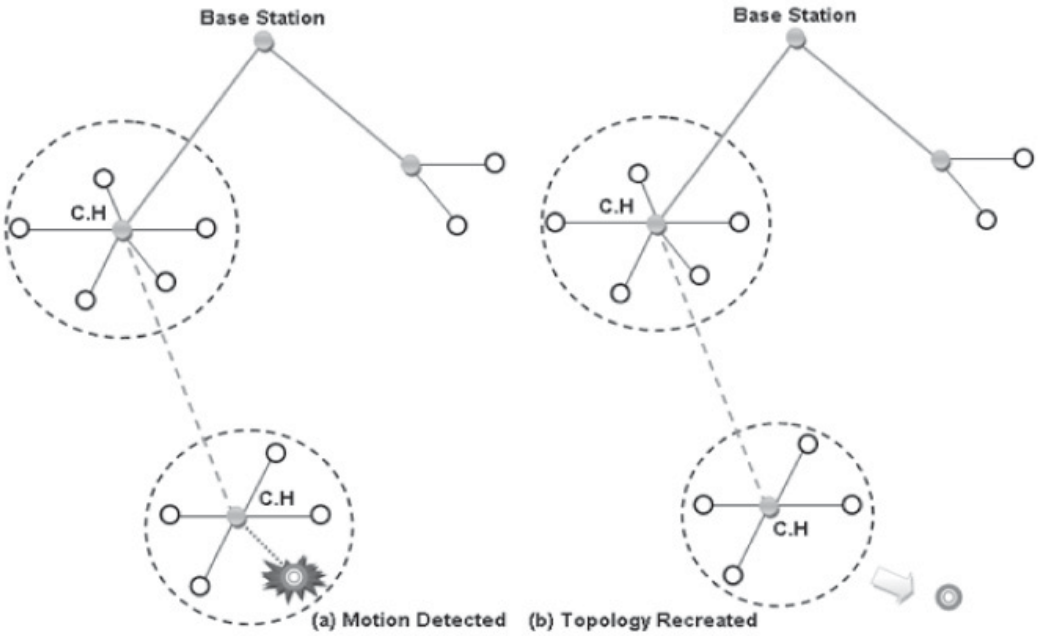


Figure 2: Node Leaving Network Scenario

The main responsibility of the tilt sensor is to detect any movement of the container. If a container is moved from one place to another place, the movement will trigger its tilt state as demonstrated in Figure 2. Once the movement is detected by tilt sensor, N/D immediately broadcasts the leave packet to its 1-hop neighbor. Then, all nodes within 1-hop of the leaving N/D remove its relationship with the leaving N/D. While parent node of the leaving node forwards the leave packet to its parent until the information reaches base station. As a result, the network topology has been reconstructed rapidly without any human intervention. The performance of employing motion sensor into N/D for improving convergence time is presented in Section **??**.

4 Low Power Listening Mechanism

In most of the protocol designed for wireless networks had a common problem, which was the receiver must always be on. In most of the cases that the power consumed during listening to an idle channel same as power consumed when receiving data. This idle listening method was very power-inefficient in wireless sensor network [14]. Moreover, numerous of wireless sensor network researchers found that energy saving can be achieved by duty cycling. Node designed with low duty-cycle MAC protocol should periodically switch its radio interface to active and sleep. The nodes operating time was divided into active-sleep cycles. Therefore, in each cycle consisted of two durations: active and sleep. Node regular transmissions and receptions in active duration, while in sleep duration the node was power off its radio interface and reduce the energy consumption.

In proposed solution, we apply low power listening (LPL) mechanism on N/D but do not apply on C/P. Given that C/P has installed with unlimited power supply, we can utilize this advantage to improve the network performance. As a result, C/P is always in idle listening mode and never switches to sleep mode when it's not transmitting or receiving message. In contrast, N/D is powered by two AA batteries and low power listening mechanism should apply on the N/D to reduce the power consumption and longevity of service. C/P does not apply LPL mechanism,

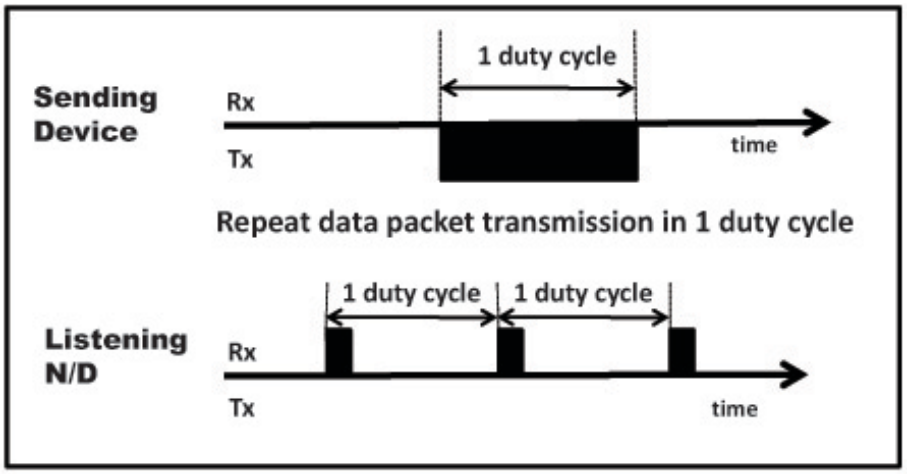


Figure 3: N/D Low Power Listening Model

C/P turns on its radio interface all the time. While N/Ds operating time is divided into low duty cycles as shown in the Figure 3. In each duty cycle, divided into active-sleep intervals, then each active-sleep interval consists of two durations: active and sleep. Active duration is for node turns on its radio interface and sniffing for transmitting data, while in sleep duration the node will power off its radio interface to reduce energy consumptions.

With the low power listening mechanism, when any device in the network would like to send message to any device in the topology, the devices have to consider the role of the receiver. If the receiver is C/P, the sender requires send a single packet as usual is sufficient. However, if the receiver is N/D, the sender has to send the same message repeatedly in one duty cycle similar in Figure 3. If sender would like to broadcast a message or the sender do not know receiver role, sender has to send the message in one duty cycle too.

5 Node and Antenna Localization

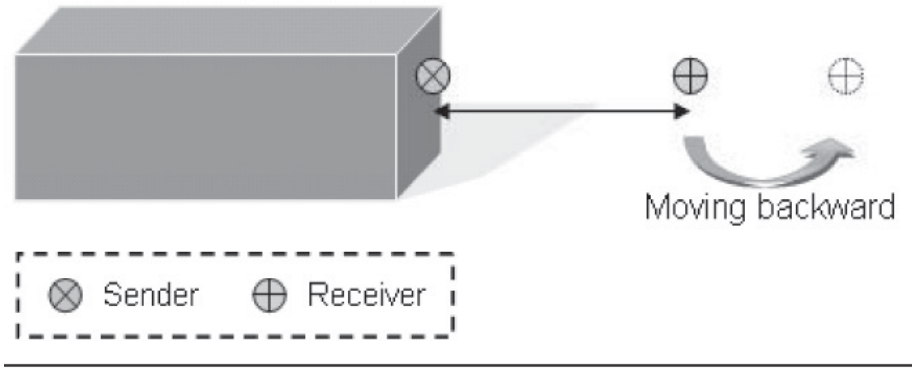


Figure 4: Test Methodology

The placement of node and its antenna is a vital issue in container environment. This is because improper node placement will cause poor performance of data transmission and might affect normal cargo operation.

We proposed and tested with various positions of N/Ds antenna in the container to ensure optimum performance. The performance matrix of this test is measured by the maximum dis-

Table 2: Result of various node placements

Parameter	Maximum Distance (m)
A	14
B	25
C	35

tance that two container N/Ds could stay connected. In this test, one container N/D acts as sender and another container N/D operates as receiver as illustrated in Figure 4. The sender is positioned statically in the container and keeps sending data every 0.5 second to the receiver. Receiver keeps moving apart from the sender until before the receiver could not receive any data packet from the sender. Then, the distance between sender and receiver is measured. All tests were conducted with containers door closed. We used an external PCB antenna with 2dBi and container N/D with RF power about -2dBm in the test. In placement A, we attached the container N/D inside the container upper right and the antenna is placed at container's door ledge as illustrated in Figure 5a. In placement B, container N/D is positioned inside the container upper center as depicted in Figure 5b. Next, in placement C, we attached the container N/D antenna outside container at the hinge while the container N/D seated inside the container to obtain the temperature/humidity of the container as demonstrated in Figure 5c.

Referring to Table 2, placement C which attaching the container N/Ds antenna at the hinge yields great reception if compared to other placements. Besides providing best performance, placement C also provides protection on antenna from human interrupt or any undesired damages. As a result, placement C was selected as an ideal place to setup container N/D and its antenna in the container.

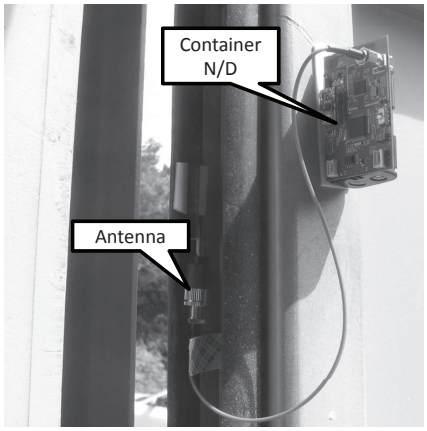
6 Experiment and Analysis

6.1 Network Topology Reconstruction Time

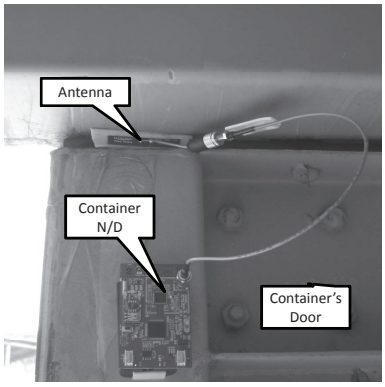
In this section, we evaluate the performance of using motion sensor on N/D that could improve network topology reconstruction time.

We carried out the experiment by using Qualnet version 4.5.1 [13]. Qualnet is the commercial successor to Glomosim which has a widespread set of high fidelity models ranging from the physical layer to application layer that allows modeling a variety of emulated networks. Qualnet's scenario designer provides easy topology configuration while it's graphical Analyzer serves well for performance analysis of the emulated network. Besides that, latest version of Qualnet supports sensor networks library that includes MAC and PHY layer with IEEE 802.15.4 standard compliance which are well-fitted in our system. As discussed previously, short network topology recreation time is favorable in container networks. Fail to update the network topology in a short period of time will cause unwanted broken link and data loss. A state of convergence is achieved once all routing information has been distributed to all nodes participating in the routing protocol process. In this simulation, we use convergence time as our performance metrics which measures how fast a group of nodes could reach the state of convergence.

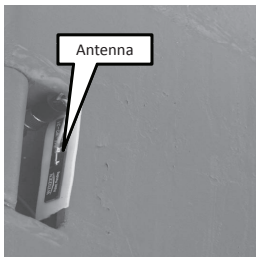
Our simulation consists of 17 nodes across a plain area of 500x500 meters as shown in Figure 6. Node 1 acts as the Coordination Point (C/P) or base station and located at position (141.1, 487.74). Node 6 and Node 8 are the only nodes in the coverage of C/Ps radio range. Thus, these two nodes have 1 hop distance from C/P while the rest of the nodes are required to form multi-



(a) Placement A



(b) Placement B



(c) Placement C

Figure 5: Experiement Placement

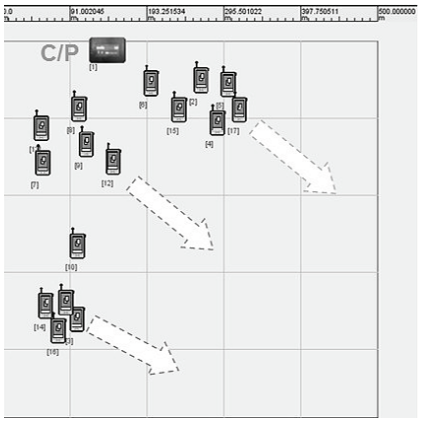


Figure 6: N/D Motion Test Scenario

hop network in order to reach C/P. Once all nodes have joined the parent node and formed a converged network, node 3, node 12 and node 17 leave the network and trigger its virtual motion sensor consecutively. In this simulation, we assumed that whenever node changed its position, the motion sensor is triggered.

Figure 7 shows the simulation result of using tilt detection and without using tilt detection in our system. The result shows that by using the motion sensor in routing portion, the convergence time has improved significantly even though with diverse of status time interval. Averagely, about 22.76 seconds after the node leaved, the network has back to convergence state. On the other hand, the convergence time of nodes without using motion sensor is performed poorly and is with respect to status time interval. This is because the parent node needs to wait for triggering from its status timer then only broadcasts the topology changes after the node has leaved. Thus, the

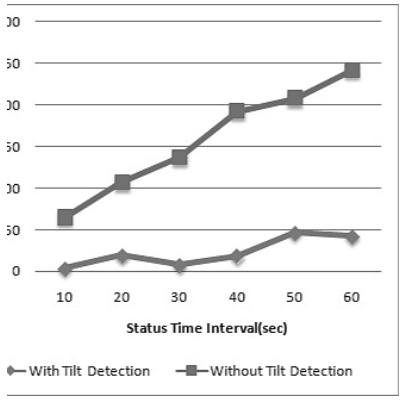


Figure 7: Effect of Using Motion Sensor on Convergence Time

larger the status time interval, nodes take much longer time to reconstruct the network topology. In a nut shell, incorporating motion sensor in our system for enabling real time leave network functionality greatly improved the convergence time of the network which is essential in container networks.

6.2 Evaluation on Energy Consumption

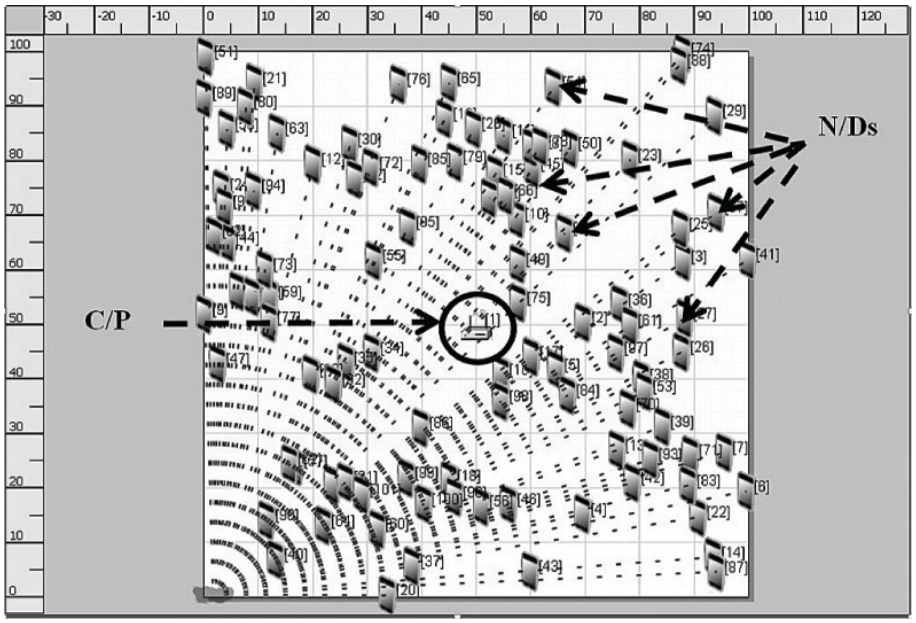


Figure 8: 100 of Static N/Ds Test Scenario

For comparison of the energy efficient of the proposed solution, we carry out the experiment by using Qualnet version 5.0.2 [13]. We implement two types of N/D in Qualnet network simulator: N/D with low power listening mechanism and N/D without low power listening mechanism. N/D with low power listening, the message exchange is implemented as descript in the Section 4. On the other hand, the implementation of N/D without low power listening mechanism, N/D never switch off its radio interface, therefore either unicast or broadcast message in the network always send in single message as usual but not in long one duty cycle. The simulation is running in twenty-four hours with 100 N/Ds and single C/P are created in an open area and C/P at

the middle of the test area as show in Figure 8. The simulation configuration summarizes in Table 3. In the Figure 9 depicts the cumulative of the total energy consumption on 100 of N/DS

Table 3: Simulation Parameter

Variable	**Values**
Duration for 1 duty cycle	1 second
Radio on duration for each duty cycle	0.01 second
Transmitting/Receiving/ Idle Listening Circuitry Power Consumption	70.0 mWatt
Sleep Circuitry Consumption	0.035 mWatt
Power Amplifier Inefficiency Factor	6.5
Supply Voltage (volt)	3.5 volt

respectively between N/Ds with LPL mechanism and N/Ds without LPL mechanism. Since the C/P never switch off its radio interfaces as other N/Ds in both simulation testing, the total energy consumption in both scenarios are almost same. On the other hand, the total energy consumption for 100 of N/Ds deploy with LPL mechanism have dramatically reduce the drain of energy. In the twenty-four hours simulation test has including the energy for building topology in the first hour of the simulation and following hours are working on topology maintenance. During topology maintenance, each N/D will broadcast its current status message to its neighbor every second, especially to its direct parent node and child nodes. According to the test result shows in the Figure 9, after twenty-four hours execution the N/Ds applied LPL mechanism total energy consumption for all 100 N/Ds in the network have consumed 4 Joule and average energy consumption per hour is 1.7 Joule. However, the total energy consumption in the 100 N/Ds does not apply LPL mechanism is 165 Joule and average energy consumption per hour is 7 Joule. It has saved up to 97% of the total of network energy consumption with apply LPL mechanism in the N/Ds. LPL mechanism reduced the drain of energy dramatically.

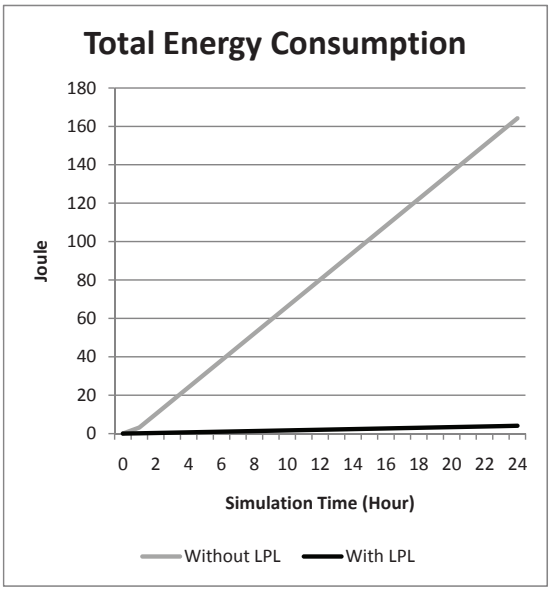


Figure 9: Cumulative of 100 N/Ds Total Energy Consumption

7 Technical Implementation

In order to further study our monitoring system and analyze the feasibility of our proposed system in real logistics environment, we implemented our wireless sensor networks based cargo monitoring system on a real container vessel named Hyundai Jakarta. The vessel has the capability to ship 6800 TEU of containers at a time and its specification is shown in Table 4. This test was carried out to monitor dangerous goods in the vessel during the shipping from Busan, South Korea to Antwerp, Belgium from 26th January 2009 to 20th February 2009. In this

Table 4: Vessel's Specification

Length(m)	**Width(m)**	**Depth(m)**	**Total Ton(G/T)**	**Load (TEU)**
294	40	20.17	75000	6800

test, we installed 10 container N/Ds and 2 seaport-used C/Ps in the vessel. We placed each C/P at the Wing Bridge of the vessel to receive the data from container N/Ds as shown in Figure 10a. On the other hand, 10 dangerous good containers were mounted with Container N/D to monitor the condition inside the container. Figure 10b depicts the position of N/D inside the container. In order to communicate with C/P with closed door, antenna of N/D must not be positioned inside the container. Based on our test on Section **??**, mounting the container N/Ds antenna at the hinge could yield great reception as shown in Figure 10c. Besides providing best performance, this placement also provides protection on antenna from human interrupt or any undesired damages.

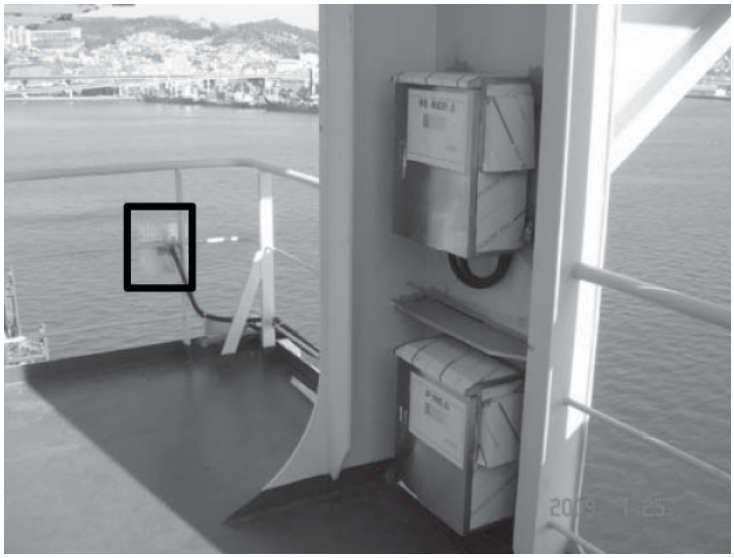
We carried out several tests on the vessel during voyage. Firstly, we tested the communication between C/P and N/D in line-of-sight (LOS) condition by sending data periodically with 0.5 second interval. Secondly, we tested the reliability of our Multi-Hop routing protocol on the vessel. Figure 10d shows the placement of N/Ds where N/D 2 was out of C/Ps radio range which was in Non line-of-sight (NLOS) condition while N/D 1 had LOS with C/P as shown in Figure 10e. The tests show promising result as we successfully received more than 93% of data packets from N/Ds for both tests.

The test was proceeded to verify the reliability of sensing data transmission for few days. In this test, we configured N/Ds to obtain containers temperature and humidity for every hour. A unique node ID was mapped with container identity number accordingly before the test. We logged the data received from container N/Ds for further processing as shown in Table 5.

Table 5: Sample of Logged Data

node ID	**container ID**	**Temp**	**Hum.**	**Time**
9	HDMU2286533	3	70	2009-01-26,00:16:21
2	HDMU2450610	3	72	2009-01-26,00:20:40
7	HDMU2179849	5	70	2009-01-26,00:23:24
2	HDMU2450610	3	72	2009-01-26,01:19:16
7	HDMU2179849	5	69	2009-01-26,01:22:00

Throughout the experiment, we successfully obtained important data such as temperature, humidity, motion and so forth on the ocean environment. Besides that, we also have better understanding of logistics operation flow and concrete concept of container stacking scenario on a

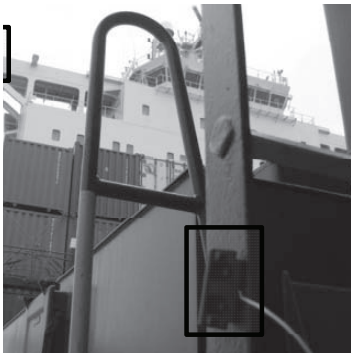


(a) Installation of CP on Wing Bridge

(b) Installation of N/D inside Container

(c) Installation of N/D's Antenna

(d) Installation of N/D for Multi-Hop Test



(e) C/P and N/D with LOS Condition

Figure 10: Installation of C/P and N/D

container vessel. Furthermore, we also have experienced in devices installation and testing during boarding and voyage. As a result, the feasibility of data gathering and container monitoring through wireless sensor networks technology on a container vessel has been studied and verified.

8 Conclusions

This paper addressed the deployment of Wireless Sensor Network in transport and logistics industry. We identified and discussed the basic requirements that are needed to form an autonomous cargo monitoring system with WSN. Network convergence time is one of the important attributes especially in container networks. Therefore, we incorporated tilt/motion sensor to improve the topology re-construction time by immediately broadcast leave message after the containers movement is detected by the sensor. Moreover, we deployed low power listening mechanism into the container communication devices to reduce the energy consumption in the network. Besides that, node placement is an important issue to obtain optimum performance. We conducted the test which positioned the NDs antenna at various places in the container. From the result, we discovered that the antenna at the containers hinge yield the best result in our tests. On the other hand, we also implemented our system on real container vessel. From the result, we strongly believed that our proposed solution is feasible for container networks. Through the test, we have better understanding of logistics operation flow and concrete concept of container stacking scenario on a container vessel that would be useful in the future.

Acknowledgments

This work was supported by the grant No. B0009720 from the Regional Technology Innovation Program of the Ministry of Knowledge Economy(MKE). Corresponding author is Gwang-Hoon Kwark.

Bibliography

[1] M. Levinson, *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger*, Princeton University Press, 2006

[2] Container throughput in the world-wide most important container ports, `http://www.hafen-hamburg.de/content/view/33/33/lang,en/`

[3] Chlomoudis, C.I. and Pallis, A.A., Port Governance and the Smart Port Authority: Key Issues for the Reinforcement of Quality Services in European Ports, *Proc. of the 10th World Conference on Transport Research*, (CDRom), Istanbul, June, 2004

[4] Busan Regional Maritime Affaire and Port Office: `http://www.portbusan.go.kr/Service.do?id=en_sub4-1,`

[5] CSI in Brief: `http://www.cbp.gov/xp/cgov/trade/cargo_security/csi/csi_in_brief.xml`

[6] C-TPAT: Customs-Trade Partnership Against Terrorism: `http://www.cbp.gov/xp/cgov/trade/cargo_security/ctpat`

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, Wireless sensor networks for habitat monitoring, *ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, September 2002.

[8] IEEE 802.15.4 WPAN-LR Task Group: `http://www.ieee802.org/15/pub/TG4.html`

[9] TelosB Datasheet: `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf`

[10] MicaZ Datasheet: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf

[11] TinyOS: http://www.tinyos.net/

[12] Tilt Switch: http://www.das-co.com/bbs_form/Fckeditor/upload/TILT_SW_TS3.pdf

[13] Scalable Network Technologies. Qualnet: http://www.scalable-networks.com

[14] El-Hoiydi A, Aloha with preamble sampling for sporadic traffic in ad-hoc wireless sensor networks, *Proc. of IEEE Int. Conference on Communications*, NY, USA, 5:3418-3423, 2002

[15] M. Stemm and R. H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, *IEICE Trans. on Communications*, E80-B(8): 1125-1131, 1997.

Author index

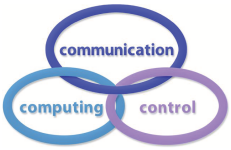
FIRST CALL FOR PAPERS

Dear colleague, we are pleased to announce and invite you to submit a paper to the:

Workshop on Intelligent Decision Support Systems for Crisis Management

May 8-12, 2012, Hotel Termal, Baile Felix, Oradea, Romania
organized by *R & D Agora* and *Agora University*
under the umbrella of

International Conference on Computers, Communications & Control, ICCCC 2012
http://www.iccc.univagora.ro



SCOPE: The goal of this workshop is to bring together researchers interested in Intelligent Computing, Knowledge Discovery in Databases, Decision Support Systems and Crisis Management, in order to exchange ideas, problems, solutions, and to work together in a friendly environment. After this interaction meeting we intend make some teams for design of intelligent decision support systems for management of crisis (economic, financial, energy, food, healthcare, military, manufacturing etc.).

TOPICS: Intelligent Computing, Knowledge Discovery in Databases, Decision Support Systems, Crisis Management.

DEADLINE: March 15, 2012: Extended abstracts submission to section chair (1-2 pages A4 about idea, proposal etc.).
March 30, 2012: Notification of acceptance/rejection of extended abstracts.
April 15, 2012: Registration fee transfer (200 EUR, include all workshop materials, Welcome dinner, Friendly dinner, and refreshments).

PUBLICATION: The accepted extended abstracts will be published in Abstracts of ICCCC, ISSN 1844-4334. The papers written and peer-reviewed after workshop will be published in *International Journal of Computers, Communications & Control*, ISSN 1841-9836. ISI Web of Science/Web of Knowledge, Impact factor in JCR2010=0.650.
http://www.journal.univagora.ro

PROGRAM AT A GLANCE:
1. May 8, Tuesday: Arrival of participants
2. May 9, Wednesday: Official opening, Invited talks, Provocative presentations, Welcome dinner
3. May 10, Thursday: Parallel sessions, Joint meeting, Friendly dinner
4. May 11, Friday: Excursion
5. May 12, Saturday: Departure of participants.

PROGRAM COMMITTEE

Subscription for IJCCC

2011

Subscription for print copy of "International Journal of Computers, Communications and Control", Volume 6/2011, 4 issues/year (800 pages) + Delivery costs
= **560 EUR**.

2012

Subscription for print copy of *International Journal of Computers, Communications & Control*, Volume 7/2012, 5 issues/year (1000 pages) + Delivery costs
= **700 EUR**.

Data for Money Bank Transfer

SUPPLIER:
SC Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Unique Fiscal Code: RO24747462
410526 Oradea, Piata Tineretului, No.8, ROMANIA,
Tel./Fax.: +40 359 101 032
E-mail: ijccc@univagora.ro
IBAN ACCOUNT FOR EURO: RO73 MILB 0000 0000 0093 2235
SWIFT CODE (eq. BIC): MILBROBU
Bank: MILLENNIUM BANK
Bank address: Piata Unirii, Primariei,No.2, Oradea, ROMANIA