Grouping Packet Scheduling for Virtual Networks by Genetic Algorithm

Heru SUKOCO[*] [†]
Department of Informatics,
Graduate School of ISEE, Kyushu University,
744 Motooka, Nishi-ku
Fukuoka-shi, JAPAN 819-0395
heru@itslab.csce.kyushu-u.ac.jp

Koji OKAMURA[‡]
Department of Advanced IT,
Graduate School of ISEE, Kyushu University
Hakozaki 6-10-1, Higashi-ku,
Fukuoka-shi, JAPAN 812-8581
oka@ec.kyushu-u.ac.jp

ABSTRACT

This paper proposed a design of group-based packet scheduling to maintain virtual network resources such as queue, bandwidth, and process time on routers. We attempt to improve our previous approach, a Packet and Slot Size-based, by creating network slicing and flow priority level done by OpenFlow Protocol to define parameters needed by Genetic Algorithm. We desire to get better results compared to our previous approach that only involved 2 parameters, namely packet size and the number of slots.

The problem will arise when the networks connected to other types of networks that do not support packet priority mechanisms. We expect that this problem can be accommodated by using virtual switch that integrated to virtual OpenFlow switch.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Sequencing and scheduling, Sorting and searching*

[*]This work is supported in part by the Ph.D Award Scholarship of the Directorate General of Higher Education (DGHE), Department of National Education, Indonesia under Grant No.1724.2/D4.4/2008.

[†]H. SUKOCO is also with the Department of Computer Science, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Bogor, INDONESIA 16680 e-mail: hsrkom@ipb.ac.id.

[‡]K. OKAMURA is a professor in the Multimedia and Network Laboratory. He is interested in Future Internet Technology, Next Generation Internet, Multimedia Communication, QoS, and Active Networks

CFI '11 June 13-15, 2011, Seoul, Korea

General Terms

Design, Experimentation, Management

Keywords

Network Virtualization, Resource Management, Queuing Management, Genetic Algorithm, OpenFlow Protocol, Slicing

1. INTRODUCTION

We present a grouping scheme used by genetic algorithm to schedule packets in network virtualization (NV) technology. This paper proposed one of solutions to achieve optimization on priority level selection of packets arrived in the virtual networks. The paper contains six sections organized as follows: section Introduction introduces the research background, motivation, related works, problem statements and contribution.

We introduce a brief review of OpenFlow protocol and Genetic Algorithm. In section Case Study and section Result and Discussion, we describe our grouping schemes on network slicing and prority level used by Genetic algorithm and its results, respectively. Finally, some conclusions and the next works are drawn in section Conclusion and Future Works.

1.1 Background

Computer networks and the Internet have become part of the critical infrastructure of our daily live and businesses. Now, we are living in the midst of information technology (IT) revolution involving the generation, processing, and transmission of information. Currently, there is almost no practical way to experiment with new network protocols such as new routing protocols or IP replacement in sufficiently realistic settings to gain the confidence needed for their widespread deployment. The most new ideas from the networking research community work without implementation and testing [1].

Eventhough, Internet as the most popular network technology in the world grows exponentially and widely used by users around the world in their daily activities [2]. Any organization possessing and connected to such networks would like to be able to rely on orderly deployment, use, and upgrade of the networking equipments, both hardware and software.

Since its born, it becomes more complex networks with very huge members serving many applications. The complexity of the Internet could be a problem for network providers when adopting a new architecture. Due to its multi-provider nature on the Internet technologies, implementing a new technology or modification of the existing infrastructure requires consensus among them. Alteration to the Internet architecture and deployment new network technology has become restricted and more difficult [3, 4].

NV has been proposed as a diversifying attribute of the Future Internet (FI) [5] and focused on developing new services to increase the virtual networks more effective [6, 7, 8]. Therefore, providing quality of service (QoS) guarantees such as bandwidth allocation and packet loss in virtual networks are imperative [6]. Beside NV technology, OpenFlow [9] has been implemented as a new experimental technology for researchers and practitioners in their production networks.

1.2 Motivation

NV has been popular not only in implementing new technology over the Internet but also in research topic that many researchers around the world are interested with it [10]. An OpenFlow Protocol (OFP) is presented as a new flow forwarding technique that allows researchers to run experiments on heterogeneous switches. An appropriate packet management and scheduling algorithms are needed to manage virtual network resources effectively. In [6, 11, 12] traditional scheduling algorithms such as weighted fair queuing (WFQ) and recursive round robin (RRR) are not capable of providing service in a virtual network since traffic of a number of sessions in a virtual network is aggregated and tunneled through a physical network which cannot isolate the traffics.

Genetic algorithm (GA) has been applied to solve many problems in variety fields, including computer networks. It works best in a wide range of low to high epistasis [13]. GA offers an effective and optimal technique to scheduling problems, route selection, and queuing management. It is a stochastic process and uses a learning based (heuristic method) loosely on principles of biological evolution [14, 15, 16].

Combining OFP and NV technologies are tremendous scheme to manage virtual network resources. This scheme will support GA as scheduling algorithm to simplify its works achieving our objective function. Objective function is a function indicating the optimal values of our solution to solve our problems.

1.3 Related Works

NV is a technique for isolating computational and network resources through virtualization to allocate them to a logical network for accommodating multiple independent and programmable virtual networks [10]. It comprises a collection of virtual nodes and virtual links overlaid on a physical network infrastructure [5, 6] which all of arriving packets will be forwarded. The research related to QoS in managing network virtual resources has been limited.

Another isolating technique has been introduced by OpenFlow Consortium called OpenFlow Protocol [9]. OpenFlow enforces isolation between each slice on OpenFlow networks by using a special OpenFlow controller named FlowVisor. [17] introduced sliced programmable networks established by Virtual Local Area Network (VLAN) technology and OpenFlow architectures. They presented a slice monitoring program that graphically displays flows in real time. An OpenFlow switch that proposed by [1] comprises a flow table, a secure channel, and the OpenFlow Protocol.

[18, 19] have been proposed the concept of the link sharing and packet scheduling algorithm on a physical link. The methods enable a physical link into several logical links to transmit traffics with different priority. GA has been used as a technique for constrained optimization and it can be improved with enclosing information about the scheduling problem represented by the use of heuristics. Scheduling problems introduced by [20, 21, 22] showed that the GA produced much better results in terms of quality of solutions in a multiprocessors and parallel system. The population size and the number of generations depend on the number of tasks.

The scheduling problems in virtual networks arrived when the packets received from many nodes in aggregated or tunneled packets. The scheduling of integrated best effort and real-time traffics in a virtual network and the complex real-time scheduling based on Genetic Algorithms have been discussed in [6] and [16], respectively. [23] proposed packet-size-based priority selection to manage network resource in virtual networks. The research result an alternative method to maintain the packets and to arrange the packets arrived on a router.

1.4 Problem Statement

Today many providers use NV technologies to support their business and deploy their services to the customers. If the providers could not guarantee QoS, they may miss critical customers who could result in lost revenue, service level agreement (SLA) penalties, lost business, and further damage to the quality reputation [2].

A router which is specialized in forwarding and routing packets from source to destination has a particular queue length and capacity. These packets have varying packet sizes according to the protocol used by them. We want all of packets on virtual networks could reach the destination host by using network virtualization technologies. Since aggregated and tunneled traffics come from virtual nodes to a router, typical queuing disciplines such as first-in first out (FIFO), weighted fair queuing (WFQ), and weighted round robin (WRR) have some problems [6, 11, 12].

We will use GA and OFP in maintaining network resources by arranging network slicing mechanism to manage and improve our virtual network resources.

1.5 Contribution

Maintaining network resources on virtual network infrastructure are very important to reduce cost. Our research proposed a case study of managing network resources on virtual networks based on GA and OpenFlow Slicing scheme. In addition, we also defined a WFQ-like model on implementing GA-based scheduling and network slicing technique.

Finally, we evaluated our GA-based queuing model and performance to get an optimal arrangement of packets in slots of the queue and a minimal number of queue length on typical router technology. It is crucial to use minimal network resources as possible such as memory and process time on our network infrastructure to reduce operational costs.

2. OPENFLOW PROTOCOL

OpenFlow Protocol (OFP) is a great tool as an alternative flow forwarding technique that can communicate within basic function of the switch and it can manage an OpenFlow switch (OFS) from a remote controller as shown in Figure 1. A scheme called FlowVisor uses in an OFS as transparent proxy that has a special purpose to control flows between OPS and multiple OpenFlow controllers. FlowVisor creates many slices of network resources and delegates control of each slice to a different controller and enforces isolation between each slice. A slice can be defined by any combination of switch ports, ethernet address, IP address, ICMP type, and TCP/UDP port [1, 9].

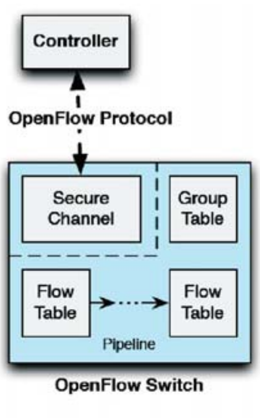


Figure 1: An OpenFlow Switch Communicates with a Controller over a Secure Connection using the OpenFlow Protocol [9]

OFP is an open protocol that can communicate to other protocol and technology such as IEEE 802.1Q, IEEE 802.3, and IEEE 802.2 standards. Its header provides basic Layer 2 (L2) header information as shown in Table 1 and uses in OFS components. OFS comprises many components such as a flow table, a secure channel, and the OpenFlow protocol [1]. A Flow table is a set of flow information which will be used by the switch to process the flows. A secure channel allows commands and packets to be sent between a controller and the switch and the OpenFlow protocol, which provides an open and standard way for a controller to communicate with a switch.

Table 1: OpenFlow Header Format [9]

```
/* Header on all OpenFlow packets. */
struct ofp_header {
uint8_t version; /* OFP_VERSION. */
uint8_t type; /* One of the OFPT_ constants. */
 uint16_t length; /* Length including this ofp_header. */
uint32_t xid; /* Transaction id associated with this packet. */
};
OFP_ASSERT(sizeof(struct ofp_header) == 8);
```

The OFS avoids the need for researchers to program the switch if we spesified a standard interface. It is useful to categorize switches into dedicated OFS that do not support normal Layer 2 and Layer 3 processing.

3. GENETIC ALGORITHM

The usual form of GAs was described by Goldberg [24]. GA is a method for solving both constrained and unconstrained optimization problems that is based on a mechanism of natural selection on natural genetics. Among the evolutionary techniques, GAs are the most extended group of methods representing the application of evolutionary tools that rely on the use of a selection, crossover, mutation operators, and replacement by generations of new individuals [25].

GA is one of the heuristic methods based on the fittest individual surviving in a population. It is a tremendous tool for solving a variety of optimization problems that are not well suited for standard optimization algorithms.

The GA uses three main types of operations, i.e. selection, crossover, and mutation at each step to create the next generation from the current population. The flowchart of GA procedure is shown in Figure 2 while the following terminologies describing the GA's concepts are shown below:

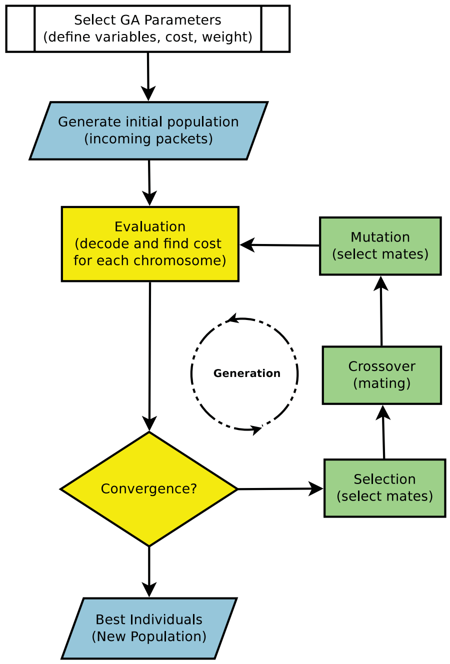
1. **Population** is a set of chromosomes (individuals).
2. **Chromosome** expresses a possible solution to the problem as a string (a set of bits).
3. **Fitness function** results a fitness value from a chromosome as an input and returns a higher value for better solutions.
4. **Selection method** determines how the individuals, called parents, are selected for breeding from the initial population that contribute to the new population at the next generation.
5. **Crossover operation** determines how parents combine to produce offspring (children) for the next generation.
6. **Mutation operation** apply random changes to individual parents to form children.

The GA commonly starts with an *initial population*, i.e. an initial set of random solutions, which are represented by chromosomes and repeatedly modifies a population of individual solutions. The chromosomes evolve through successive iterations, called *generations*. During each generation, the solutions represented by the chromosomes of the population are evaluated using some measures of fitness. At each step, the GA selects individuals at random from the current population to be parents and uses them to produce the offspring (children) for the next generation towards an optimal solution as a better new population.

4. CASE STUDY

We are interested in network virtualization technology used in Future Internet, the replacement of current Internet technology. A research on implementation of GA as one of the alternative algorithms for managing virtual network resources is a challenging issue. We are proposing a new queue management for optimizing and managing packets arrived on a router queue. Our research uses a simulation to examine our model.

We use common models to describe our design both on queue model and some well-known terms of the GA compared to other classical optimization algorithms as shown in Figure 3 and Table 2, respectively.



Note: GA represents solutions as binary strings (called chromosomes) and creates a random population of candidate solutions. Afterwards, it selects two parents from the population and crosses them to create two offsprings and sometimes mutate the children. The process will be repeated for the next selections, crossover, and mutation, until GA reaches a solution based on a desired level of fitness.

Figure 2: Flowchart of Generation Algorithm

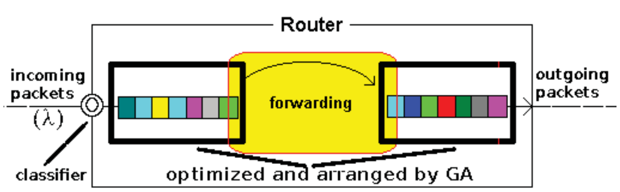


Figure 3: Forwarding packets and queue management in a router

Table 2: Comparison of Natural Evolution and Genetic Algorithm Terminology [25]

Natural Evolution	Genetic algorithm
Chromosome	String
Gene	Character (or Feature)
Allele	Feature value
Locus	String position
Genotype	Structure or coded string
Phenotype	Parameter set, a decoded structure

A virtual network has virtual nodes overlaid on the physical nodes. Packets transmitted by any sources will be classified and processed by GA. It defines and arranges the packets to get an optimal condition based on particular fitness function. Fitness function is our objective to solve the problems and yields fitness values needed to decide whether the condition has reached an optimal condition or not.

In this research, we attempt to propose a design of grouping packet scheduling that will be used by GA as parameters in our objective function due to our previous work [23] did not involve network slicing technique for packet scheduling. We use the GA's pseudo-code, as shown in Algorithm 1, to impelement our case study. The algorithm consists of network slicing done by FlowVisor, initialization, iterative variation and selection by evaluating the fitness function. Over iterations of random variation and selection, the population can be made to converge to optimal solutions by examining fitness values. The effectiveness of GA depends on the variation and selection operators as applied to a chosen representation and initialization.

Algorithm 1 A Pseudo-code for Group-based Packet Scheduling using Genetic Algorithm

```
Require: t ← 0; initialize P(0); evaluate P(0)
{comment: Done by OFP}
{comment: Slice is a group of nodes/vnodes/TOS}
if NOTPacketsToOFNetworks then
    ForwardPacketsToVSwitch
    ForwardPacketsToVOFS
    CreateSlices()
else
    CreateSlices()
end if
{comment: Get a new better population}
while N ≠ 0 (not convergence) do
    {comment: Get a generation}
    t ← X | 1
    select P(t) from P(t − 1)
    alter P(t)
    evaluate P(t)
end while
```

5. RESULT AND DISCUSSION

In the previous work, we only used 2 parameters to define our objective function namely packet size and maximum slot capacity. The candidate solution of our problems are by evaluating the most minimum fitness value in each generation as shown in Equation 1 and Equation 2, respectively. We performed our simulation by using Java-based simulator to evaluate our problems and used the crossover probability and mutation probability are set to 0.90 and 0.05, respectively.

$$\mathbf{V_i} = \left(\sum_{i=1}^{n} w_i\right) \times S_i \tag{1}$$

where:
V_i = fitness value of the i-th chromosome
w_i = wasted capacity of the i-th slot
S_i = the number of slot needed by a router

$$\mathbf{CandidateSolution} = \min\left(V_i\right) \tag{2}$$

Figure 4 illustrates our previous approach when arranging the packets in queue iteratively. GA will evaluate each packet arrived on queues based on packet size (s) and slot size (w). Afterwards, it will calculate the fitness value of each arrangement as a candidate solution of the problem iteratively until the fitness value reaches the most minimum value and convergence. We simulated 10 trials with various values of each parameters and result as reported in [23] and shown in Table 3. By using these techniques, it was not very easy to evaluate each packet arrived at the routers and the result still have high standard deviation both fitness values and the number of slots by 139,575 and 83, respectively.

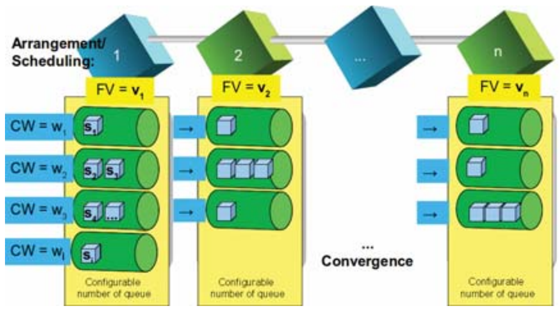


Figure 4: Our Previous Approach

Table 3: The results of our previous work

Tr	Ev	Pop	Pckt	SS	FV	NS
1	2,250	50	125	4	306	49
2	3,750	50	150	6	4,570	94
3	1,750	100	150	4	503	58
4	750	100	150	4	430	47
5	4,750	100	300	6	1,188	94
6	9,750	100	300	10	135,124	201
7	$\geq$ 10,000	300	300	4	436,993	298
8	$\geq$ 10,000	300	300	6	657	79
9	5	300	300	15	576	32
10	2750	300	1000	15	1,953	109

Note:
Tr = i-th Trial
Ev = Evolution
Pop = Population
Pckt = Number of Packets
SS = Slot Size
FV = Fitness value
NS = Number of slots

In current research, we attempt to propose a new technique to optimize our previous problems by grouping packets into slices done by FlowVisor both in OpenFlow networks and Virtual OpenFlow networks (VOFN) as shown in Figure 5. These scheme will improve the performance of GA by evaluating the slices that have flow priority level created by OFP.

All packets from source nodes will be sliced by OFP based on a set of virtual nodes or type of services. GA will use the flow priority level created on slices to arrange and schedule the group of packets. Those parameters will be used to define our new fitness function and it may use less computation time to schedule the packets compared to previous approach.

6. CONCLUSION AND FUTURE WORKS

This paper presented the next design and implementation to optimize virtual network resources by involving OFP-based network slicing and Genetic Algorithm. Our previous research showed unstable and high computation due to evaluating each packets arrived, rather than a group/slice, on a router.

We proposed a new scheme to improve our previous approach to solve the problems by involving network slicing mechanism that contains flow priority levels. These parameters will be defined as our new objective function to get a better results. The problem arise when our networks involve other types of networks that do not support packet priority mechanisms. We propose to accommodate the type of these packets by using virtual switch that integrated to virtual OpenFlow switch.

In the future, we will continue the research by implementing the design and the scenarios on well-defined testbed that supports network virtualization technology and OpenFlow protocol such as Corelab, PlanetLab, or NVLab established in Japan. Afterwards, we will compare our approach to the existing resource allocations and managements.

Acknowledgment

The authors would like to thank the anonymous editors and reviewers for constructive comments and suggestions, the members of Okamura Laboratory, Department of Advanced Information Technology, Kyushu University, Japan and the members of NCC Laboratory, Department of Computer Science, Bogor Agricultural University (IPB), Indonesia for valuable help and support in discussion.

7. REFERENCES

[1] McKeown, N., et al. *OpenFlow: Enabling Innovation in Campus Networks*, In Proceedings of the ACM SIGCOMM Computer Communication Review Vol.38 No.2, pp.69-74, DOI: 10.1145/1355734.1355746, ISSN:0146-4833, April 2008.

[2] Sukoco, H., Y. Hori, Hendrawan, and K. Sakurai, *Towards a Fairness Multimedia Transmission Using Layered-Based Multicast Protocol*, IEICE TRANSACTIONS on Information and Systems Vol.E93-D No.11, pp.2953-2961, Special Section on Architectures, Protocols, and Applications for the Future Internet, ISSN: 1745-1361, 0916-8532, Nov. 2010.

[3] Turner, J.S. and D.E. Tylor, *Diversifying the Internet*, In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'05), Vol.2, DOI:10.1109/GLOCOM.2005.1577741, ISBN: 0-7803-9414-3, Dec. 2005.

[4] Anderson, T., L. Peterson, S. Shenker, and J. Turner, *Overcoming the Internet impasse through virtualization*, IEEE on Computer, Vol.38 No.4, pp.34-41, DOI 10.1109/MC.2005.136, ISSN: 0018-9162, April 2005.

[5] Chowdhury, N.M.M.K and R. Boutaba, *A Survey of Network Virtualization*, Computer Networks, Elsevier, Vol.54 pp.862-876, DOI:10.1016/j.comnet.2009.10.017, 2010.

[6] Garg, R. and H. Saran, *Scheduling Algorithms for Bounded Delay Service in Virtual Networks*, In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'99), Vol.2, pp.1318-1322,

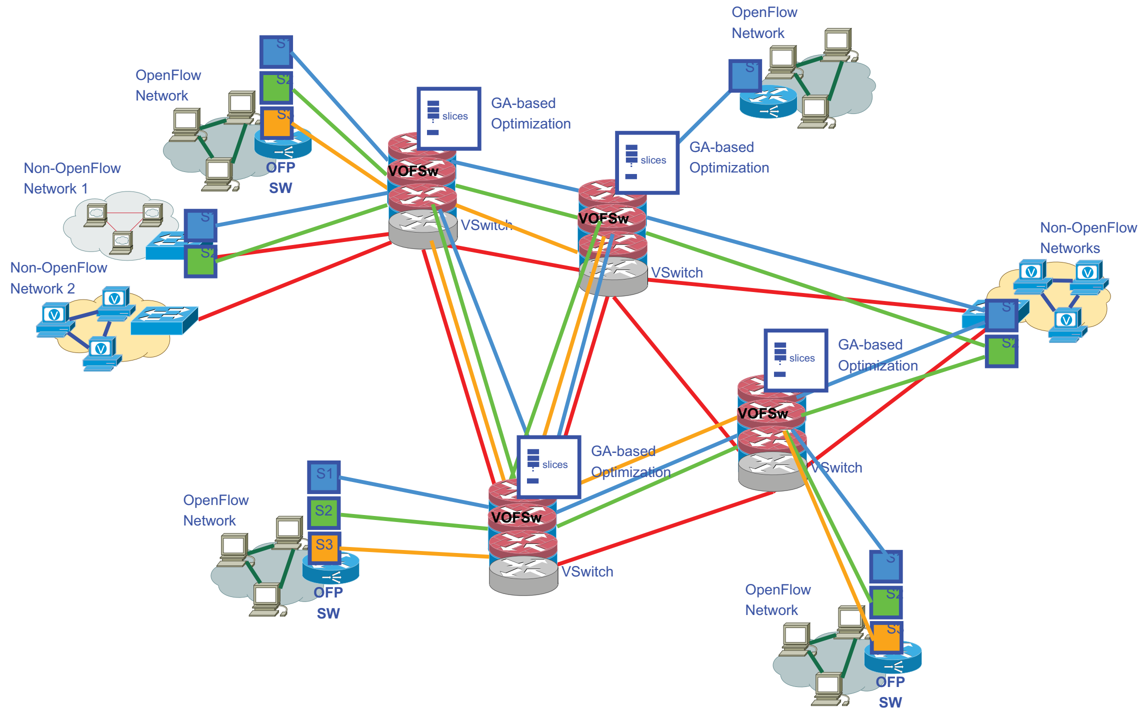


Figure 5: Group-based Packet Scheduling in Virtual Networks

DOI:10.1109/GLOCOM.1999.829986, ISBN: 0-7803-5796-5, August 2002.

[7] Vicente, J., *Genesis: Toward Programmable Virtual Networking*, In Proceedings of OPENSIG'98 Workshop on Open Signaling for ATM, Internet and Mobile Networks, DOI:10.1.1.70.8838, October 1998.

[8] Florissi, D., *The NetScript Approach to Programmable Networks*, In Proceedings of OPENSIG'98 Workshop on Open Signaling for ATM, Internet and Mobile Networks, October 1998.

[9] The OpenFlow Switch Consortium, *OpenFlow Switch Specification* version 1.1.0 Implemented (Wire Protocol 0x02), February 2011, Available at http://www.openflow.org

[10] Nakao, A. *Network Virtualization as Foundation for Enabling New Network Architectures and Applications*, Invited Letter on Special Edition on New Generation Network towards Innovative Future Society, IEICE Transactions on Communication, Vol.E93-B No.3, pp.454-457, DOI:10.1587/transcom.E93.B.454, March 2010.

[11] Demers, A., S. Keshav, and S. Shenker, *Analysis and Simulation of a Fair Queueing Algorithm*, In Proceedings of ACM Symposium on Communications Architectures and Protocols (SIGCOMM'89), pp.3-12, DOI:10.1145/75246.75248, ISBN:0-89791-332-9, October 1990.

[12] Garg, R. and X. Chen, *RRR: Recursive Round Robin Scheduler*, In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'98), Vol.1, pp. 422-432, DOI: 10.1109/GLOCOM.1998.775767, ISBN: 0-7803-4984-9, 1998.

[13] Haupt, R.L. and S.E. Haupt, *Practical Genetic Algorithm*, 2nd Ed. ISBN 0-471-45565-2, John Wiley & Sons, Inc, 2004.

[14] Sheta, A., M. Salamah, H. Turabieh, and H. Heyasat, *Selection of Appropriate Traffic of a Computer Network with Fixed Topology Using GAs*, IAENG International Journal of Computer Science (IJCS), Vol.34 Issue:1, ISSN: 1819-9224 and 1819-656X, August 2007.

[15] Ahn, C. W. and R.S. Ramakrishna, *A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations*, IEEE Transactions on Evolutionary Computation, Vol.6 No.6, pp.566-579, DOI:10.1109/TEVC.202.804323, ISSN: 1089-778X, December 2002.

[16] Montana, D., M. Brinn, S. Moore, and G. Bidwell, *Genetic algorithms for Complex, Real-Time Scheduling*, In Proceedings of IEEE on Systems, Man, and Cybernetics, Vol.3 pp.2213-2218, DOI:10.1109/ICSMC.1998.724984, ISBN: 0-7803-4778-1, October 1998.

[17] Sherwood, R., G. Gibb, and M. Kobayashi, *Carving Research Slices out of Your Production Networks with OpenFlow*, In Proceedings of the ACM SIGCOMM Computer Communication Review, vol.40, No.1, DOI: 10.1145/1672308.1672333, ISSN:0146-4833, January 2010.

[18] Floyd, S. and V. Jacobson, *Link Sharing and Resource Management Models for Packet Networks*, IEEE/ACM Transactions on Networking, Vol.3 No.4, August 1995.

[19] Zhang, H. and J.C.R. Bennett, *Hierarchical Packet Fair Queueing Algorithms*, In Proceedings of ACM SIGCOMM'97, pp.143-156, September 1997.

[20] Ahmad, I., Y.K. Kwok, and M.Y. Wu, *Analysis,*

Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors, In Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, pp.207, ISBN: 0-8186-7460-1, IEEE Computer Society Washington, DC, USA, 1996.

[21] Kwok, Y.K., and I. Ahmad, *Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors*, ACM Computing Surveys, Vol. 31, Issue. 4, December 1999, ISSN: 0360-0300, ACM New York, NY, USA, 1999.

[22] Kaur, K., A. Chhabra, and G. Singh, *Heuristics Based Genetic Algorithm for Scheduling Static Tasks in Homogeneous Parallel System*, the International Journal of Computer Science and Security (IJCSS), Vol.4, Issue 2, pp.183 - 198, May 2010. ISSN (Online): 1985-1553, CSC Journals, 2010.

[23] Sukoco, H., and K. Okamura, *Genetic Algorithm for Optimizing and Arrangement of Queue in Virtual Networks*, In Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS) 2011, Vol.1 pp.670-675, ISBN: 978-988-18210-3-4 and ISSN: 2078-0958, March 2011.

[24] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, ISBN-10: 0201157675, ISBN-13: 9780201157673, Addison-Wesley, 1989.

[25] Sivanandam, S.N. and S.N. Deepa, *Introduction to Genetic Algorithms*, Chap.2 pp.15-27, ISBN 978-3-540-73189-4, Springer-Verlag, 2008.