

A fuzzy approach for network bandwidth management

Julija Asmuss¹ Gunars Lauks¹

¹Institute of Telecommunications, Riga Technical University
Azenes str. 12, LV-1048, Riga, Latvia, julija.asmuss@rtu.lv, gunars.lauks@rtu.lv

Abstract

The paper deals with the problem of resource allocation in a substrate network with DaVinci architecture. We present a methodology of decision making on network bandwidth allocation based on fuzzification and defuzzification principles and the expert knowledge database of fuzzy rules and we describe a fuzzy logic based modification of the adaptive bandwidth allocation mechanism to optimize decision making under uncertain network conditions. We describe a simulation scheme used for the estimation of performance of this modification for two nodes network topology.

Keywords: Bandwidth allocation problem, Fuzzy rule, Decision system, Simulation

1. Introduction

The current Internet carries different types of services (voice, video, music, web pages, e-mail). Some services need low delay mechanisms while other services need high throughput mechanisms, and their requirements may conflict with each other. Network virtualization principles [1] can be used for building experimental platforms that run multiple virtual networks. Network virtualization is a widely applied technique discussed in the networking research community. From network point of view network virtualization divides a network into a set of virtual networks. These virtual networks give us the opportunity to classify and separate traffic. Each virtual network is logically separated and can be customized for a particular traffic class. Resources offered by the substrate network are shared between all virtual networks. Finding the proper bandwidth allocation to the virtual networks is one of the key problems of network virtualization [3], [15], [16].

DaVinci approach (Dynamically Adaptive Virtual Networks for a Customized Internet) describes a technique of network virtualization, according to which all virtual networks are constructed over a physical substrate network by subdividing each physical node and each physical link into multiple virtual nodes and virtual links [4]. We consider the problem of bandwidth resource management in a substrate network on the basis of DaVinci architecture. In this context it is a maximization problem for the aggregate utility of all virtual networks [9], which effective solution depends on the design of dynamically adaptive bandwidth allocation protocols. When different types of traffic coexist over the same network substrate, each virtual network could control a subset of resources at each node and link. At a

smaller timescale each virtual network maximizes its own utility. The question is whether optimization of virtual networks together with the bandwidth share adaptation scheme performed by the substrate network actually maximizes the aggregate utility.

Classical bandwidth allocation mechanisms are not capable of making decisions in uncertain conditions, which are prevail in the modern networks. The dynamic traffic demand in the fast changing environment practically eliminates the possibility of fast online reasoning. Taking into account, that fuzzy logic serves as the excellent tool to cope with uncertain and multivariable data, this giving the flexibility and robustness for decision making while using fuzzy rules, in this paper we propose a fuzzy approach for network bandwidth management. For the estimation of performance of the proposed fuzzy logic based network bandwidth management policy for two types of traffic we present the design and simulation scheme of dynamically adaptive bandwidth allocation using Coloured Petri Nets.

This paper is organized as follows. Section 2 describes DaVinci approach for modelling of substrate networks accordingly to network virtualization principles. The problem of network bandwidth allocation is considered in Section 3. A fuzzy approach to decision making on bandwidth allocation for delay sensitive traffic and throughput sensitive traffic is described in Section 4. Section 5 presents a short introduction on Coloured Petri Nets used in Section 6 for modelling and simulating the proposed bandwidth allocation mechanisms. A conclusion is drawn in Section 7.

2. Description of DaVinci architecture

DaVinci architecture [4] allows us to describe how a single substrate network (SN) can support multiple traffic classes, each with a different performance objective. The problem of bandwidth allocation in SN is a maximization problem for the aggregate objective of multiple applications with diverse requirements. According to DaVinci approach, each traffic class is carried on its own virtual network (VN) with customized traffic-management protocols. The substrate runs schedulers that arbitrate access to the shared node and link resources, to give each virtual network the illusion that it runs on a dedicated physical infrastructure.

Let the topology of a substrate network SN be described by a graph $G_s = \{V_s, E_s\}$, given by a set V_s of nodes (or vertices) and a set E_s of links (or edges). We suppose that links of E_s are with finite capacities C_l (links are denoted by $l: l \in E_s$). Correspondingly to

$G_s = \{V_s, E_s\}$ we consider DaVinci model with N virtual networks, indexed by k , where $k = 1, 2, \dots, N$. Let the key notations be the following:

- $y^{(k)}$ bandwidth of virtual network k , $k = 1, 2, \dots, N$;
- $z^{(k)}$ path rates for virtual network k , $k = 1, 2, \dots, N$;
- $\lambda^{(k)}$ satisfaction level degree of virtual network k , $k = 1, 2, \dots, N$;
- $O^{(k)}$ performance objective for virtual network k , $k = 1, 2, \dots, N$.

Bandwidth values $y^{(k)} = (y_l^{(k)})_{l \in E_s}$ for each substrate link $l \in E_s$ are assigned by the substrate network, taking into account such local information as current satisfaction indicators and performance objectives (Fig. 1). The substrate network periodically reassigns bandwidth shares $y^{(k)}$ for each substrate link between its virtual links. Thus, values $\lambda^{(k)} = (\lambda_l^{(k)})_{l \in E_s}$ and $O^{(k)}$ are periodically updated by the substrate network and used to compute virtual link capacity $y^{(k)}$.

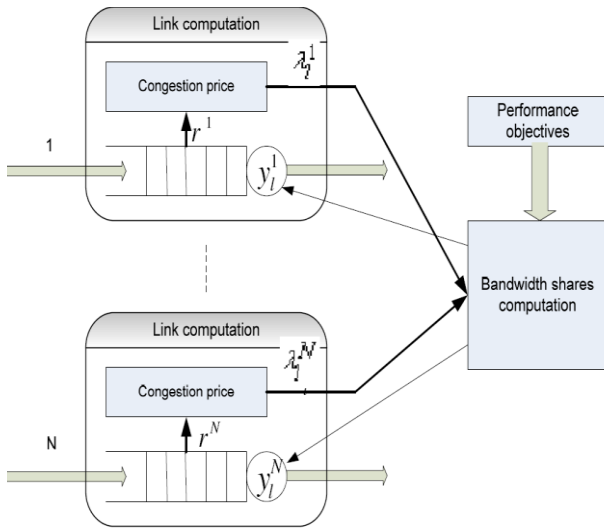


Fig. 1: Bandwidth shares computation scheme.

At a smaller timescale, each virtual network runs according to a distributed protocol that maximizes its own performance objective independently. Under such combined conditions in a dynamically changing virtual network environment a fundamental problem of resource allocation is the design of dynamically adaptive bandwidth allocation protocols.

3. Description of bandwidth allocation problem

The goal of the substrate network is to optimize the aggregate utility of all virtual networks

$$\sum_{k=1}^N w^{(k)} O^{(k)}(z^{(k)}, y^{(k)}),$$

where $w^{(k)}$ is the weight the substrate assigns to represent the importance of virtual network k . If the substrate wants to give virtual network k strict priority, then to $w^{(k)}$ can be assigned a value several orders of magnitudes larger than the other weights.

First we consider an optimization problem for the performance objective of each virtual network, which represents also constraints of each virtual network k :

$$\begin{aligned} & \text{maximize} && O^{(k)}(z^{(k)}, y^{(k)}) \\ & \text{subject to} && H^{(k)} z^{(k)} \leq y^{(k)}, \\ & && g^{(k)}(z^{(k)}) \leq 0, \\ & && z^{(k)} \geq 0, \\ & \text{variables} && z^{(k)}. \end{aligned}$$

Usually it is supposed that the objective function $O^{(k)}$ depends on both virtual link rates $z^{(k)}$ and virtual link capacity $y^{(k)}$. The objective is subject to a capacity constraint and possibly other constraints described in terms of $g^{(k)}(z^{(k)})$. The capacity constraint requires the link load

$$r^{(k)} = H^{(k)} z^{(k)}$$

to be no more than the allocated bandwidth. To compute $r^{(k)}$ we use routing indexes

$$H_{lj}^{(k)i} = \begin{cases} 1, & \text{if path } j \text{ of source } i \text{ in virtual} \\ & \text{network } k \text{ uses link } l, \\ 0, & \text{otherwise,} \end{cases}$$

and path rates $z_j^{(k)i}$ that determine for source i the amount of traffic directed over path j .

Now we formulate the optimization problem for the aggregate utility:

$$\begin{aligned} & \text{maximize} && \sum_{k=1}^N w^{(k)} O^{(k)}(z^{(k)}, y^{(k)}) \\ & \text{subject to} && H^{(k)} z^{(k)} \leq y^{(k)}, \quad k = 1, 2, \dots, N, \\ & && \sum_{k=1}^N y^{(k)} \leq C, \\ & && g^{(k)}(z^{(k)}) \leq 0, \quad k = 1, 2, \dots, N, \\ & && z^{(k)} \geq 0, \quad k = 1, 2, \dots, N, \\ & \text{variables} && z^{(k)}, y^{(k)}, \quad k = 1, 2, \dots, N. \end{aligned}$$

The optimization scheme (Fig. 1) follows directly from DaVinci principles. First, the substrate network determines how satisfied each VN is with its allocated bandwidth. Satisfaction level degree $\lambda_l^{(k)}$ (for link l of VN k) is one indicator that a virtual network may want more resources. In congestion control the link conges-

tion prices are summed up over each path and interpreted as end-to-end packet loss or delay.

Next, the substrate network determines how much bandwidth virtual network k should have on link l : the substrate network increases or decreases value $y_l^{(k)}$ in dependence on the satisfaction level $\lambda_l^{(k)}$ on link l and the relative importance $w^{(k)}$ of virtual network k .

Given that each virtual network is acting independently, the question is whether virtual networks together with the bandwidth share adaptation performed by the substrate network actually maximize the overall performance objective.

4. A fuzzy approach for decision making on network bandwidth allocation

Our work focuses on two traffic types: delay sensitive (the objective is to minimize the average delay) and throughput sensitive (the objective is to maximize the average link rate). Accordingly to DaVinci principles we consider two virtual nets ($N = 2$) for two types of traffic.

We denote by t_j , where $j = 1, 2, \dots$, moments of system adaptation, i.e. moments of decision on bandwidth shares $y_l^{(1)}(t_j)$ and $y_l^{(2)}(t_j)$. Such decision is based on the results of monitoring the performance of the system during interval $[t_{j-1}, t_j]$ and values $y_l^{(1)}(t_{j-1})$ and $y_l^{(2)}(t_{j-1})$. We suppose that values $y_l^{(1)}(t_0)$ and $y_l^{(2)}(t_0)$ are given and consider the behavior of the system for $t \geq t_0$.

The first step of fuzzy decision making system design is to define fuzzy variables. The proposed fuzzy solution requires three input variables for each link l and each traffic class k :

- $U_l^{(k)}(t_j)$ – the average utilization of the virtual link,
- $L_l^{(k)}(t_j)$ – the average length of queue,
- $D_l^{(k)}(t_j)$ – the average delay of packets.

These variables are evaluated for the period $[t_{j-1}, t_j]$ and have linguistic values according to their membership functions. Input variables are used to describe the system state and the starting point of adaptation decision.

The introduced fuzzy solution requires one output variable for each link l , which determines a fuzzy logic decision at t_j :

$F_l(t_j)$ – a fuzzy response, which is a defuzzified output value of the fuzzy inference module.

Taking into account the above defined input and output parameters, the decision process on link level (considering the link level we omit index l) can be described by the following pseudo-code:

START

READ $y^{(1)}(t_0), y^{(2)}(t_0)$

FOR $j = 1$ TO J STEP 1

CALCULATE t_j

READ $U^{(1)}(t_j), U^{(2)}(t_j)$

READ $L^{(1)}(t_j), L^{(2)}(t_j)$

READ $D^{(1)}(t_j), D^{(2)}(t_j)$

CALCULATE $F(t_j)$

IF $d_1 \leq F(t_j) \leq d_2$ THEN $FL(t_j) = \text{„Not reassign”}$

AND $y^{(k)}(t_j) = y^{(k)}(t_{j-1})$;

ELSE IF $F(t_j) < d_1$ THEN $FL(t_j) = \text{„Reassign 1”}$

AND CALCULATE $y^{(1)}(t_j), y^{(2)}(t_j)$;

ELSE $FL(t_j) = \text{„Reassign 2”}$

AND CALCULATE $y^{(1)}(t_j), y^{(2)}(t_j)$;

END IF

AND FOR

END

We use the denotation FL for the linguistic values of F .

Now we determine the following parameters of decision making in order to achieve the advanced goals (see e.g. [12], [14]):

- the number of linguistic values and the membership functions of linguistic values for each input and output variable,
- the base of if-then rules,
- the method of defuzzification of output parameters,
- the type of decision making system.

The COG (center of gravity) defuzzification method was used in this investigation. We apply Mamdani fuzzy inference technique, introduced in [10], [11] (see also [12], [14]).

For all input variables we consider three linguistic values „Low”, „Medium” and „High” (we use the denotation L , M and H as indexes). But for the output variable linguistic values are: „Reassign 1” (meaning that bandwidth share increases for the first type of traffic and decreases for the second type, correspondingly), „Not reassign” and „Reassign 2”.

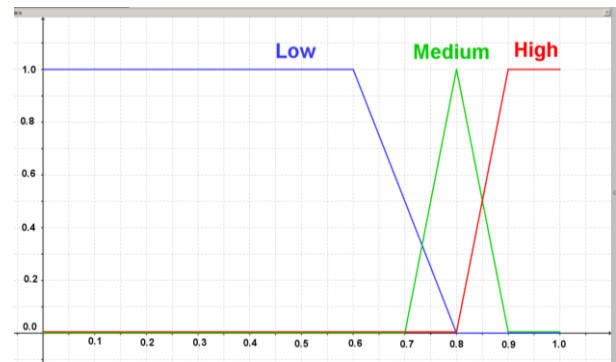


Fig. 2: Membership functions of the linguistic values U_L, U_M, U_H .

The membership function for each linguistic value is given by a triangular or trapezoidal fuzzy number.

For example, Fig.2 shows the graphs of the membership functions of linguistic values U_L, U_M, U_H for the utilization variable. In the form of trapezoidal fuzzy numbers:

$$U_L = (0, 0, 0.6, 0.8), U_M = (0.7, 0.8, 0.8, 0.9),$$

$$U_H = (0.8, 0.9, 1.0, 1.0).$$

These membership functions are considered to be independent on l, k and j . Obviously, such assumption is not true for membership functions of linguistic values for the delay variables. The graphs of the membership functions used for linguistic values $D_L^{(1)}, D_M^{(1)}, D_H^{(1)}$ for delay sensitive traffic are given by Fig.3, but the graphs of $D_L^{(2)}, D_M^{(2)}, D_H^{(2)}$ for throughput sensitive traffic – by Fig.4.

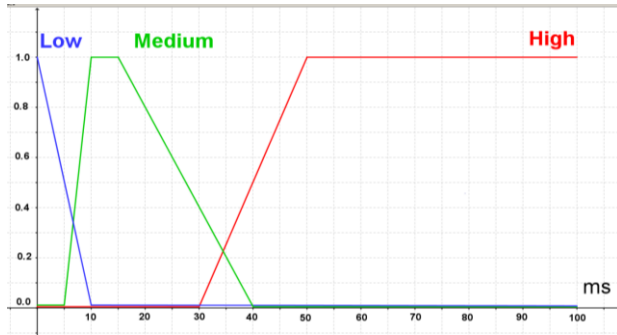


Fig. 3: Membership functions of the linguistic values D_L^1, D_M^1, D_H^1 .

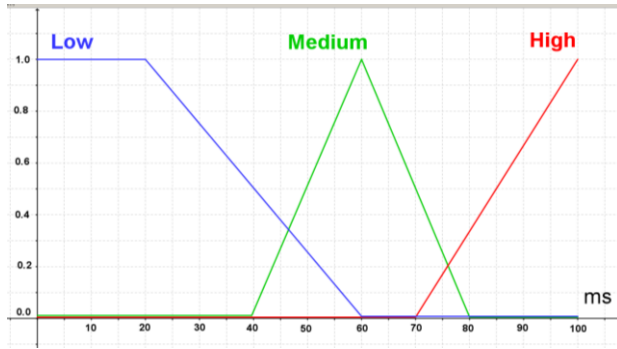


Fig. 4: Membership functions of the linguistic values D_L^2, D_M^2, D_H^2 .

The knowledge base for the current implementation of the algorithm is made as a rule database which is based only on the “expert knowledge”, where the experts are the authors of the paper, and the rules are assumed as logical assumptions, e.g.:

if $UL^{(1)} = \text{"High"} and LL^{(1)} = \text{"Low"} and DL^{(1)} = \text{"Low"} and$
 $UL^{(2)} = \text{"High"} and LL^{(2)} = \text{"Low"} and DL^{(2)} = \text{"Low"},$
 then $FL = \text{"Not reassign"};$

if $UL^{(1)} = \text{"High"} and LL^{(1)} = \text{"Medium"} and DL^{(1)} = \text{"High"} and$
 $UL^{(2)} = \text{"High"} and LL^{(2)} = \text{"Low"} and DL^{(2)} = \text{"Medium"},$
 then $FL = \text{"Re assign 1"};$

if $UL^{(1)} = \text{"Low"} and LL^{(1)} = \text{"Low"} and DL^{(1)} = \text{"Low"} and$
 $UL^{(2)} = \text{"High"} and LL^{(2)} = \text{"Medium"} and DL^{(2)} = \text{"Low"},$
 then $FL = \text{"Re assign 2"};$

if $UL^{(1)} = \text{"High"} and LL^{(1)} = \text{"Low"} and DL^{(1)} = \text{"Medium"} and$
 $UL^{(2)} = \text{"Medium"} and LL^{(2)} = \text{"Low"} and DL^{(2)} = \text{"Low"},$
 then $FL = \text{"Re assign 1"}.$

The rules can be freely modified as well as the definitions of the membership functions. The assumed fuzzy-rules and the membership functions of input and output values were used for the proposed approach performance evaluation by simulation process and the impact of their modification is considered as the field for the future research. In two next Sections of the paper we focus mainly on practical realization of the proposed approach by means of Coloured Petri Nets (CPN) using CPN Tools.

5. Coloured Petri Nets based modeling

The concept of Coloured Petri Nets, developed by K. Jensen (see e.g. [5]), is an extended version of classical Petri Nets. In addition to places, transitions and tokens, the concept of types or colour sets is included. This concept enables to involve information (simple or complex) into the tokens and allows the use of tokens that carry data values and can hence be distinguished from each other. Each token could be attached with a colour, indicating the identity of the token. Moreover, each place and each transition has attached a set of colours. A transition can fire with respect to each of its colours. By firing a transition, tokens are removed from the input places and added to the output places in the same way as that in original Petri Nets, except that a functional dependency is specified between the colour of the transition firing and the colours of the involved tokens.

Definition. A Coloured Petri Net is a tuple $CPN = (P, T, V, \Sigma, W, C, G, H, I)$ satisfying the following requirements:

- P is a finite set of places,
- T is a finite set of transitions,
 $P \cap T = \emptyset, P \cup T \neq \emptyset,$
- V is a set of directed arcs, $V \subset (P \times T) \cup (T \times P),$
- Σ is a finite set of types (colour sets), $\Sigma \neq \emptyset,$
- W is a finite set of typed variables, $Type(w) \in \Sigma$ for all $w \in W$, where $Type: W \rightarrow \Sigma$ is a type function assigning types (colour sets) to variables,
- $C: P \rightarrow 2^\Sigma$ is a colour function assigning colour sets to each place, $C(p) \subset \Sigma$ for all $p \in P,$
- $G: T \rightarrow EXPR(W)$ is a guard function assigning a guard $G(t)$ to each transition $t \in T$ (we omit the explanations on the set of expressions $EXPR(W)$), such that $Type(G(t)) = Bool$ for all $t \in T$, where $Bool = \{true, false\},$
- $H: V \rightarrow EXPR(W)$ is an arc expression function assigning an expression $H(v) \in V$ to each arc

$v \in V$, $Type(H(p,t)) \in C(p)$ for all arcs $(p,t) \in V$ and $Type(H(t,p)) \in C(p)$ for all arcs $(t,p) \in V$,

- I is an initialisation function assigning an initial marking to each place, an initial marking can be defined as a multi-set $M_0 \in N^{PLACE}$, where $PLACE = \{(p,c) : p \in P, c \in C(p)\}$.

Triple (P,T,V) constitutes the net structure, pair (Σ,W) describes types and variables, and tuple (C,G,H,I) defines the net inscriptions. Here we omit the explanations on marking iterations and do not discuss how transitions change the marking of places. Due to this iteration scheme, CPN is one of efficient mathematical modeling languages for the description of discrete event systems. CPN combines a well-developed mathematical theory with an excellent graphical representation. This combination is the main reason for the great success of CPN in modeling of the dynamic behavior of systems [5], [6], [7], [8].

Coloured Petri Nets, proposed by K. Jensen, have been developed by the CPN group at Aarhus University, Denmark since 1979. The first version was a part of the PhD Thesis of K. Jensen and was published in 1981. The CPN group has developed and distributed industrial-strength computer tools, such as Design/CPN in 1990 and CPN Tools in 2003. Our simulation scheme is based on CPN Tools [7], [13]. We extend CPN Tools model by adding specially designed bandwidth adaptation module, which realizes the bandwidth allocation mechanism described in the previous Section. As a result, we obtain a discrete event modeling computer tool supporting interactive and automatic simulations, state spaces and performance analysis and combining Coloured Petri Nets and fuzzy logic based decision making system. By making simulations of networks CPN models with this additional module it is possible to investigate different scenarios and explore the behaviors of the system, to use simulation based performance analysis for further decision making and adaptation processes.

6. Coloured Petri Nets based simulation scheme

The aim of our simulation experiment is to get a visualization of virtual network switching due to demand for extra link bandwidth. To simplify the problem we simulate the adaptive bandwidth allocation on link level.

The simulation study focuses on two traffic types: delay sensitive and throughput sensitive. We consider two virtual nets for two types of traffic denoted by A and B . Up to now we experiment with two nodes topology (Fig. 5) and use the following notations: G_A , G_B – packet generators; D_A , D_B – destination nodes. Colours A and B can be effectively used for modeling and simulating such systems [2].

The colours are associated with the tokens which represent packets. According to DaVinci architecture all data packets which are generated by both generators are

handled and transmitted over the virtual links. The queuing model of each virtual link is described independently.

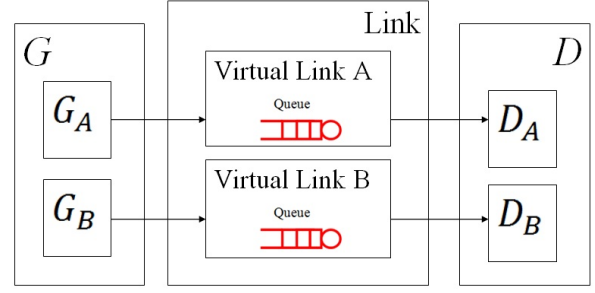


Fig. 5: Simulation scheme for two nodes topology.

The CPN Tools based model of the system is given in Fig. 6. Packets are generated by the traffic generators (Arrivals) and stored in the FIFO (first in, first out) queues (Buffer). The goal is to route packets to the output port (Transmitted). Packets cross the FIFO queue and are transmitted under the condition that the transmission link is free. The transmission time depends on the size of a packet and the bandwidth of the corresponding virtual link and is calculated for each packet transmission. Module Link Load Data assigns a bandwidth value for each virtual link. Two traffic FIFO queues and two virtual links are separated due to colours A and B and two flows of packets are parallel controlled and analyzed.

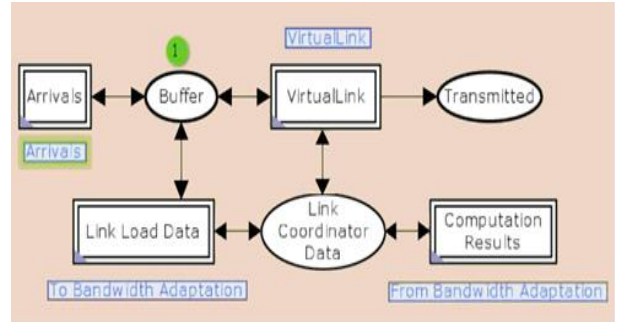


Fig. 6: Bandwidth adaptation simulation scheme using Coloured Petri Net Tools.

This model involves the bandwidth shares adaptation modules (see Fig. 6). The adaptation scheme is realized accordingly to the proposed fuzzy logic based algorithm described in Section 4.

The bandwidth allocation adaptation modules are designed to update the virtual link resources allocation for dynamically changing traffics. Special predicate and observation functions and data collection monitors (Buffer Delay, Buffer Length, Virtual Link Utilization) are included for monitoring the performance of the system. The monitoring mechanism is used not only to control, but also to modify a simulation of the net. It is done by reassigning bandwidth shares between virtual links A and B . The decision making system is based on data collection monitors that allow to calculate the system performance measures such as the delay in each queue, the length of each queue, the utilization of each link. The criteria for decision making described in Sec-

tion 4 depend on parameters d_1 and d_2 . Fig. 7 illustrates the results of simulation experiments for $d_1 = 0.4$ and $d_2 = 0.6$. By changing the values of d_1 and d_2 it is possible to adjust the traffic management policy.

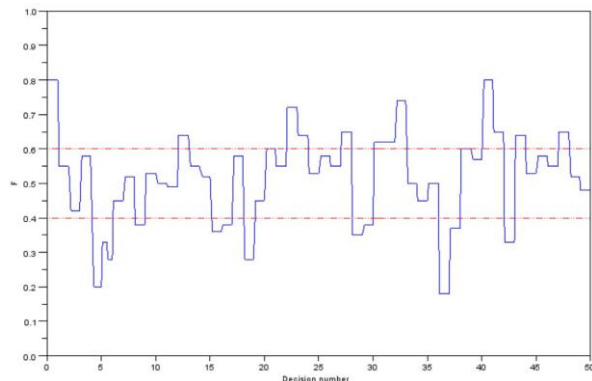


Fig. 7: Simulation result output values after defuzzification.

In our experiments the initial resource allocation to the virtual links is uniform, we set the capacity of the physical network as 100 Mbps. Packets A and B are generated with exponentially distributed arrival time and uniform distributed size. An important issue is the frequency of adaptation. Bandwidth resources are reassigned every 10000 MTU. By changing traffic parameters we observe the adaptation process.

The simulations show that the proposed fuzzy logic based adaptation scheme has good results. We experiment with delay sensitive traffic and throughput sensitive traffic as A and B correspondingly, as well as with two delay sensitive traffic classes and with two throughput sensitive traffic classes. Our simulation results clearly demonstrate that the adaptive bandwidth allocation mechanism can dynamically and efficiently react to traffic changes in both cases: when traffic classes are with different performance objectives or with the same one.

7. Conclusions and future research

In this paper we present the simulation scheme of the adaptive bandwidth allocation mechanism which is realized for two nodes topology. Future work will mainly focus on extension of this simulation scheme in two directions: from the local link level to a global network level and from two traffic classes to multiple traffic classes. At the same time our goal is to modify and improve monitoring and decision making systems to optimize adaptive bandwidth management, to develop the experimental network system and simulation environment as well as to accomplish the practical realization of the classic and fuzzy logic based decision making systems under identical conditions to make a further comparison of produced results.

We hope that in comparison with the classical decision making system, based on the successive evaluation of multiple parameters and large scale bases of if-then rules, the fuzzy solution can provide fast and effective real-time modification of traffic management policy within the MPLS (as well as GMPLS in future) routers, adopting it to an unsteady network environment.

References

- [1] T. Anderson, L. Peterson, S. Shenker and J. Turner, Overcoming the Internet impasse through virtualization, *IEEE Computer*, 38(4):34–41, 2005.
- [2] J. Asmuss, V. Zagorskis and G. Lauks, Simulation of dynamically adaptive bandwidth allocation protocols using Coloured Petri Nets, *Proceedings of the 24th European Modeling and Simulation Symposium EMSS2012*, pages 408–413, 2012.
- [3] M.L. Dramitinos, A bandwidth allocation mechanism for 4G. *Proceedings of European Wireless Technology Conference*, Rome, pages 96–99, 2009.
- [4] J. He, R Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford and M. Chiang, DaVinci: dynamically adaptive virtual networks for a customized Internet, *ACM CoNEXT Conference*, ACM, New York, 2008.
- [5] K. Jensen. Coloured Petri Nets: basic concepts, analysis methods and practical use. Volumes 1–3, Springer-Verlag, 1992–1997.
- [6] K. Jensen and L. Kristensen. *Coloured Petri Nets. Modelling and validation of concurrent systems*. Springer-Verlag, 2009.
- [7] K. Jensen, L. Kristensen and L. Wells, Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9:213–254, 2007.
- [8] L. Kristensen, S. Christensen and K. Jensen, The practitioner's guide to Coloured Petri Nets. *International Journal on Software Tools for Technology Transfer*, 2:98–132, 1998.
- [9] X. Lin and N. Shroff, Utility maximization for communication networks with multi-path routing. *IEEE Trans. Automatic Control*, 51:766–781, 2006.
- [10] E.H. Mamdani, Applications of fuzzy algorithms for control of simple dynamic plant, *Proceedings IEEE*, 121(12):1585–1588, 1974.
- [11] E. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies*, 7(1): 1–13, 1975.
- [12] A. Piegat. *Fuzzy modeling and control*. Physica-Verlag, Heidelberg, 2001.
- [13] A.V. Ratzer, L. Wells, H.M. Larsen, M. Laursen, J.F. Qvortrup, M.S. Stissing, M. Westergaard, S. Christensen and K. Jensen, CPN Tools for editing, simulating and analysing Coloured Petri Net. *Lecture Notes in Computer Science*, 2679, pages 450–462, 2003.
- [14] T.E. Ross, *Fuzzy logic with engineering applications*. McGraw-Hill, New York, 1995.
- [15] W. Szeto, Y. Iraqi and R. Boutaba, A multi-commodity flow based approach to virtual network resource allocation. *Proc. of the Global Communications Conference IEEE GlobeCom*, San Francisco, CA, USA, pages 3004–3008, 2003.
- [16] Y. Zhu and M. Ammar, Algorithms for assigning substrate network resources to virtual network components. *INFOCOM 2006. Proceedings of the 25th IEEE International Conference on Computer Communications*, pages 1–12, 2006.