SIMULATION OF DYNAMICALLY ADAPTIVE BANDWIDTH ALLOCATION PROTOCOLS USING COLOURED PETRI NETS

J. Asmuss [(a)], V. Zagorskis [(b)], G. Lauks [(c)]

Institute of Telecommunication, Faculty of Electronics and Telecommunication,
Riga Technical University, Azenes str. 12, LV-1048, Riga, Latvia

[(a)] Julija.Asmuss@ rtu.lv, [(b)] Viktors.Zagorskis@ rtu.lv , [(c)] Gunars.Lauks@ rtu.lv

ABSTRACT

We consider the problem of resource allocation in a substrate network and suggest a simulation scheme for dynamically adaptive bandwidth allocation protocols using CPN (Coloured Petri Nets) Tools. The proposed simulation scheme based on network virtualization according to DaVinci principles allows us to describe how a single physical network can support multiple traffic classes with different performance objectives by means of multiple virtual networks constructed by subdividing each physical link into multiple virtual links.

Keywords: Bandwidth Allocation Problem, Simulation, Coloured Petri Nets, CPN Tools

1. INTRODUCTION

Network virtualization is a widely applied technique discussed in the networking research community. It is being considered as a good technique to overcome the weaknesses of the current Internet (Anderson, Peterson, Shenker and Turner, 2005). Network virtualization allows multiple virtual networks (VNs) to run parallel on the substrate network (SN). In a virtualization-enabled substrate network resources offered by SN are sharing between all VNs. An optimal allocation of resources is a fundamental problem for virtualization-enabled networking infrastructures (Haider, Potter and Nakao, 2009; Zhu and Ammar, 2006).

DaVinci approach (Dynamically Adaptive Virtual Networks for a Customized Internet) describes a technique of network virtualization, according to which all VNs are constructed over the physical SN by subdividing each physical node and each physical link into multiple virtual nodes and virtual links (He, Zhang-Shen, Li, Lee, Rexford and Chiang, 2008). We consider the problem of bandwidth (BW) resource allocation in a substrate network on the basis of DaVinci architecture of its virtualization-enabled networking infrastructure. In this context it is a maximization problem for the aggregate utility of all virtual networks (Lin and Shroff, 2006), which effective solution depends on the design of dynamically adaptive bandwidth allocation protocols.

We present the design and simulation scheme of dynamically adaptive bandwidth allocation protocols using Coloured Petri Nets.

2. COLOURED PETRI NETS BASED MODELING

The concept of Coloured Petri Nets (CPN) was introduced by Kurt Jensen in 1981. CPN is one of several mathematical modeling languages for the description of Discrete Event Systems. CPN combine 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 (Jensen, 1992–1997; Kristensen, Christensen and Jensen, 1998; Jensen, Kristensen and Wells, 2007; Gehlo and Nigro, 2010).

Definition. A Colored Petri Net is a tuple $CPN = = \langle P, T, D, Type, Pre, Post, M_0 \rangle$, *where*

- P *is a finite set of places,*
- T *is a finite set of transitions,* $P \bigcap T = \varnothing$, $P \bigcup T \neq \varnothing$,
- D *is a finite set of types,* $D \neq \varnothing$,
- $Type : P \bigcup T \to 2^D$ *is a type function assigning types to places or transactions,*
- $TRANS = \{(t,m) : t \in T, m \in Type(t)\}$ *is the set of all transition modes,*
- $\mathbb{N}^{PLACE} = \mathbb{N}^{\{(p,q) : p \in P,\, q \in Type(p)\}}$ *is the set (multiset) of all markings,*
- $Pre,\ Post : TRANS \to \mathbb{N}^{PLACE}$ *are the backward and forward incidence functions assigning marking to each transition mode,*
- $M_0 \in \mathbb{N}^{PLACE}$ *is the initial marking.*

Coloured Petri Nets have got their name because they allow 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.

A CPN model can be effectively applied to describe and analyze multiple virtual networks. CPN model of substrate network describes the states of each virtual network of the system and the events (transitions) that can cause the system to change state.

Our simulation scheme is based on Coloured Petri Nets Tools (Jensen, Kristensen and Wells, 2007). CPN Tools is a discrete event modeling language combining Coloured Petri Nets and the functional programming language CPN ML which is based on Standard ML. Standard ML provides the primitives for the definition of data types, describing data manipulation, and for creating compact and parameterisable models. By making simulations of the CPN model with CPN Tools it is possible to investigate different scenarios and explore the behaviours of the system, to verify properties of the model by means of state space methods and model checking, and to conduct simulation-based performance analysis.

3. DAVINCI PRINCIPLES FOR MODELING OF SUBSTRATE NETWORK

The DaVinci architecture allows us to describe how a single 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 the DaVinci approach, each traffic class is carried on its own 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 $k = 1, 2, ..., N$. Let the key notations be the following:

$\mathbf{y}^{(k)}$ bandwidth of virtual network k, $k = 1, 2, ..., N$;

$\mathbf{z}^{(k)}$ path rates for virtual network k, $k = 1, 2, ..., N$;

$\lambda^{(k)}$ satisfaction level degree of virtual network k, $k = 1, 2, ..., N$;

$U^{(k)}$ performance objective for virtual network k, $k = 1, 2, ..., N$.

Bandwidth values $\mathbf{y}^{(k)} = (y_l^{(k)})_{l \in E_S}$ 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 $\mathbf{y}^{(k)}$ for each substrate link between its virtual links. Thus, values $\lambda^{(k)} = (\lambda_l^{(k)})_{l \in E_S}$ and $U^{(k)}$ are periodically updated by the substrate network and used to compute virtual link capacity $\mathbf{y}^{(k)}$.

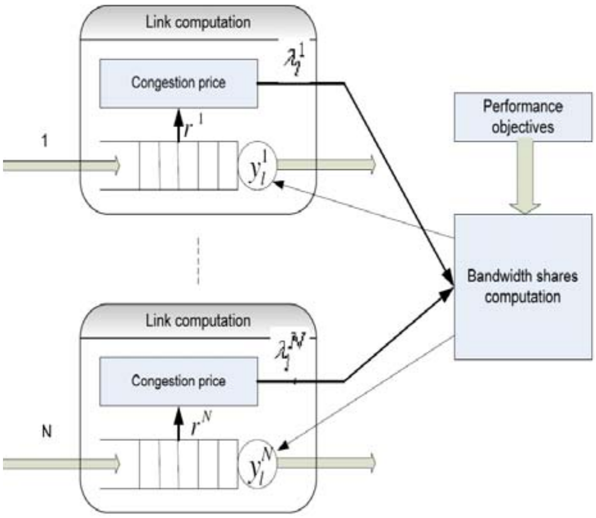

Figure 1: Bandwidth Shares Computation Scheme

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.

4. DESCRIPTION OF BANDWIDTH ALLOCATION PROBLEM

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

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

where $w^{(k)}$ is the weight the substrate assigns to represent the importance of VN virtual network k. If the substrate wants to give virtual network k strict priority, then $w^{(k)}$ 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:

$$\begin{array}{rl} \text{maximize} & U^{(k)}(\mathbf{z}^{(k)},\mathbf{y}^{(k)}) \\ \text{subject to} & \mathbf{H}^{(k)}\mathbf{z}^{(k)} \le \mathbf{y}^{(k)}, \\ & g^{(k)}(\mathbf{z}^{(k)}) \le 0, \\ & \mathbf{z}^{(k)} \ge \mathbf{0}, \\ \text{variables} & \mathbf{z}^{(k)}. \end{array}$$

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

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

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

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

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:

$$\begin{array}{rl} \text{maximize} & \sum_{k=1}^{N} w^{(k)} U^{(k)}(\mathbf{z}^{(k)},\mathbf{y}^{(k)}) \\ \text{subject to} & \mathbf{H}^{(k)}\mathbf{z}^{(k)} \le \mathbf{y}^{(k)},\ k=1,2,\ldots,N, \\ & \sum_{k=1}^{N} \mathbf{y}^{(k)} \le \mathbf{C}, \\ & g^{(k)}(\mathbf{z}^{(k)}) \le 0,\ k=1,2,\ldots,N, \\ & \mathbf{z}^{(k)} \ge \mathbf{0},\ k=1,2,\ldots,N, \\ \text{variables} & \mathbf{z}^{(k)},\ \mathbf{y}^{(k)},\ k=1,2,\ldots,N. \end{array}$$

An 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 congestion 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 value $y_l^{(k)}$ proportional to the satisfaction level $\lambda_l^{(k)}$ on link l and proportional to the relative importance $w^{(k)}$ of virtual network k, taking into account the capacity constraint.

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.

5. SIMULATION SCHEME

The aim of our simulation experiment is to get 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.

We experiment with two nodes topology (Fig. 2) and use the following notations: G1, G2 – packet generators; D1, D2 – destination nodes.

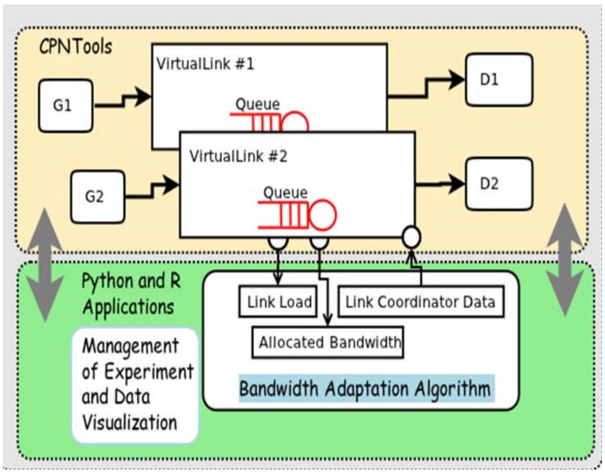

Figure 2: Simulation Scheme Using CPN Tools

According to DaVinci architecture all data packets which are generated by both generators are handled and transmitted over virtual networks. The queueing model of each virtual link (Fig.2) is described independently.

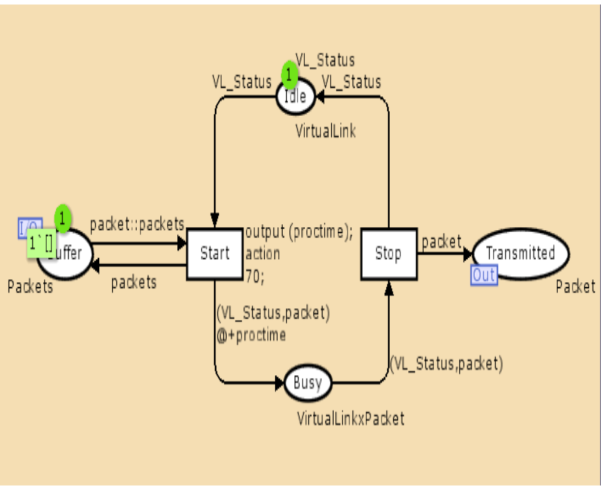

Figure 3: Queueing Model of Virtual Link Using CPN Tools

In our CPN network model packets are generated (Fig. 4) with exponentially distributed inter arrival time.

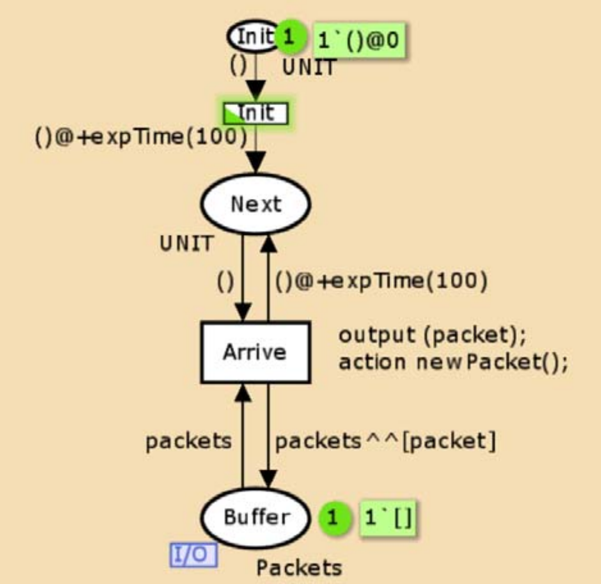

Figure 4: Traffic Simulation Using CPN Tools

Both virtual links (VL) reside on the substrate link (SL) and share the same link bandwidth. Packets are transmitted over corresponding virtual link (Fig. 5) with determined BW, each virtual link is acting independently.

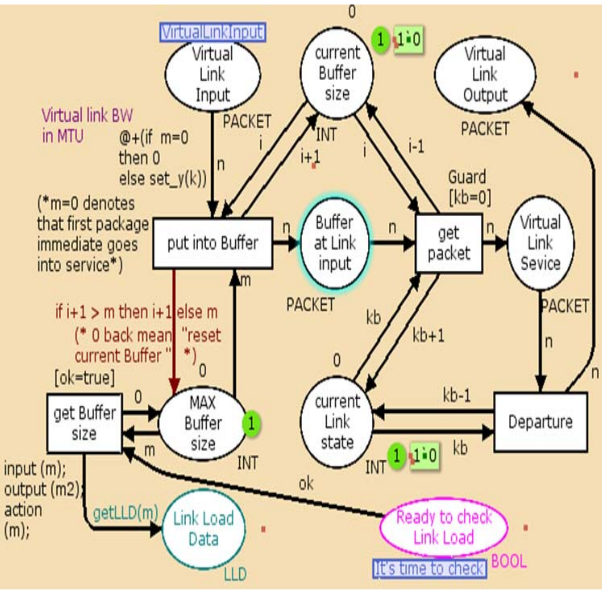

Figure 5: Modelling of a Virtual Link Using CPN Tools

The substrate periodically reassigns bandwidth shares between virtual links based on local information. In our model there is time scheduled monitoring of load of each virtual link (Fig. 6). For instance, if 10 MTU correspond to 10 nS, then for 0.1 sec. time delay in simulator model there should be delay in 10*1,000 MTUs.

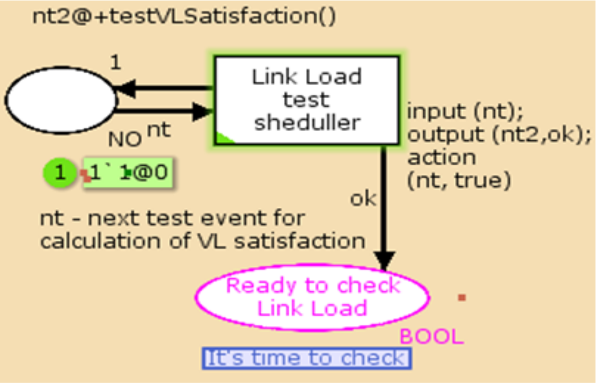
nt2@+testVLSatisfaction()
1
Link Load test shedduller
NO
nt
input (nt);
output (nt2,ok);
action
(nt, true)
1
1`1@0
ok
nt - next test event for calculation of VL satisfaction
Ready to check Link Load
BOOL
It's time to check

Figure 6: Load Monitoring using CPN Tools

The model involves bandwidth shares adaptation module (Fig. 7). An adaptive scheme can lead to a solution maximizing the aggregate utility of all SL. At the same time small-timescale isolation between VLs ensures the stability of the overall system. Longer-timescale adaptation of resource shares ensures efficiency.

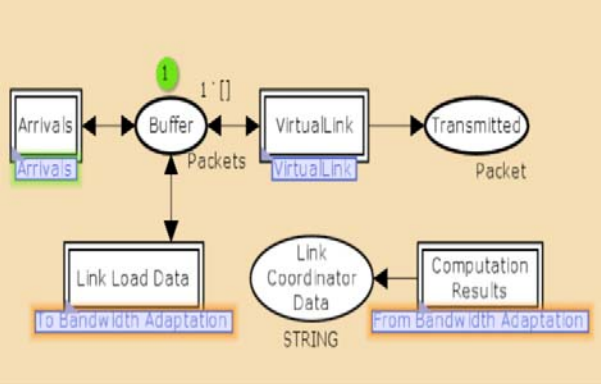

Figure 7: Virtual Network Switching Scheme Using CPN Tools

6. SOME NUMERICAL RESULTS

Up to now there where some experiments with BW adaptation for two virtual links. Bandwidth of SL in our experiment is determined 100 Mbps. In order to present a real network problem, CPN models are using simulation timing determined by Model Time Units (MTU). That gives opportunity to simulate concurrent systems like two or more simultaneously running networks or links. Two columns (Table 1) show network BW corresponding MTU values.

Table 1: Network BW Corresponding MTU Values

Network BW [Mbps]	Network BW [MTU]
100	10
90	11
80	13
70	14
60	17

50	20
40	25
30	33
20	50
10	100

Columns of Table 2 show some combinations how the capacity 100 Mbps can be shared with two different virtual links ($y^{(1)}, y^{(2)}$) in many different ways by means of MTU.

Table 2: Subdividing 100 Mbps of the Virtual Link Capacity in MTU

$y^{(1)}$ [MTU]	$y^{(1)}$ [Mbps]	$y^{(2)}$ [MTU]	$y^{(2)}$ [Mbps]
11	90	100	10
13	80	50	20
14	70	33	30
17	60	25	40
20	50	20	50
25	40	17	60
33	30	14	70
50	20	13	80
100	10	11	90
0	0	10	100

After 10000 MTU since the start of simulation there was a scheduled event to check link load and calculate satisfaction for both links. As a result (Fig. 8, Fig. 9), the first link allocated bandwidth was increased from 10 Mbps to 50 Mbps in order to reduce utilization. For the second link shared bandwidth was reallocated from 90 Mbps to 50 Mbps.

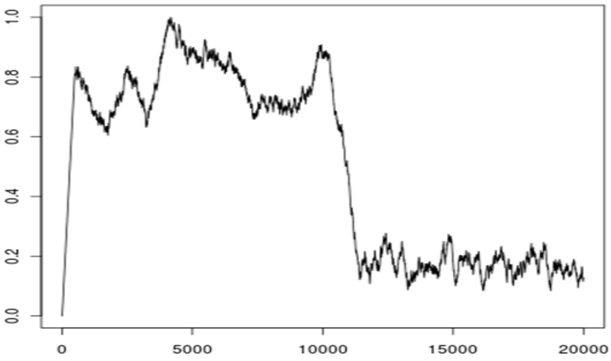

Figure 8: Utility of the First Virtual Link

For computation of the satisfaction level for each virtual link we use the following formula:

$$\lambda^{(k)}(t+T) = \frac{bsa^{(k)}}{y^{(k)}(t+T)} + \frac{y^{(k)}(t+T)}{y^{(k)}(t)},$$

where $bsa^{(k)}$ is a buffer size averaged for the link of virtual network k and T is the time period between bandwidth assignments (the time between iterations).

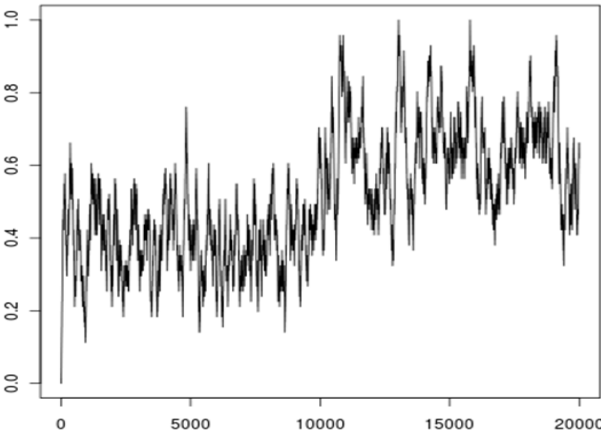

Figure 9: Utility of the Second Virtual Link

At the beginning of the simulation process the utility of the substrate link (Fig. 10) still is a subject for optimization because allocated bandwidth for the first link is more than necessary.

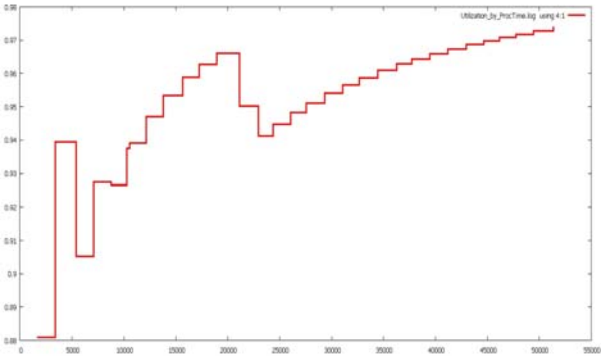

Figure 10: Small-Timescale Utility of the Substrate Link

If the parameters of arriving packets generators are not changed, then adaptive scheme leads to a solution maximizing the aggregate utility of all SL (Fig. 11).

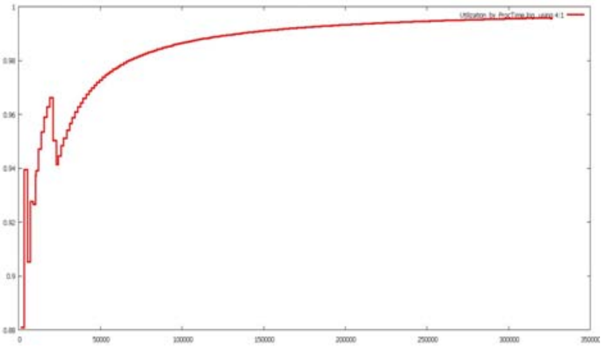

Figure 11: Long-Timescale Utility of the Substrate Link

REFERENCES

Anderson, T., Peterson, L., Shenker, S. and Turner, J., 2005. Overcoming the Internet impasse through virtualization. *IEEE Computer*, 38 (4), 34–41.

Gehlo, V. and Nigro, C., 2010. An introduction to system modeling and simulation with Coloured Petri Nets. *Proceedings of Winter Simulation Conference, Baltimore, Maryland, USA*, December, 2010, 104–118.

Haider, A., Potter, R. and Nakao, A., 2009. Challenges in resource allocation in network virtualization. *20th ITC Specialist Seminar*. May, 2009: Hoi An (Vietnam).

He, J., Zhang-Shen, R., Li, Y., Lee, C.-Y., Rexford, J. and Chiang, M., 2008. DaVinci: dynamically adaptive virtual networks for a customized Internet. *ACM CoNEXT Conference,* December, 2008: ACM, New York (USA).

Jensen, K., 1992–1997. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use.* Volumes 1–3. Springer Verlag.

Jensen, K., Kristensen, L. and Wells, L., 2007. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9, 213–254.

Kristensen, L., Christensen, S. and Jensen, K., 1998. The practitioner's guide to coloured Petri nets. *International Journal on Software Tools for Technology Transfer*, 2, 98–132.

Lin, X., Shroff, N., 2006. Utility maximization for communication networks with multi-path routing. *IEEE Trans. Automatic Control,* 51, 766–781.

Zhu, Y. and Ammar, M., 2006. Algorithms for Assigning Substrate Network Resources to Virtual Network Components. *INFOCOM 2006. Proceedings of the 25th IEEE International Conference on Computer Communications.* April, 2006, 1–12.