Application of CPN Tools for Simulation and Analysis of Bandwidth Allocation

Julija Asmuss, Gunars Lauks, Viktors Zagorskis

***Abstract*—**We consider the problem of bandwidth allocation in a substrate network as an optimization problem for the aggregate utility of multiple applications with diverse requirements and describe a simulation scheme for dynamically adaptive bandwidth allocation protocols. The proposed simulation model based on Coloured Petri Nets (CPN) is realized using CPN Tools.

***Keywords*—**Bandwidth Allocation Problem, Coloured Petri Nets, CPN Tools, Simulation

I. Introduction

NETWORK virtualization is a widely applied technique discussed in the networking research community. It is being considered as a means to overcome the weaknesses of the current Internet [1].

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 virtual networks. An optimal allocation of resources is a fundamental problem for virtualization-enabled networking infrastructures [2].

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 the physical substrate network by subdividing each physical node and each physical link into multiple virtual nodes and virtual links [3]. 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 [4], which effective solution depends on the design of dynamically adaptive bandwidth allocation mechanisms (protocols). For simulation of such mechanisms we apply Coloured Petri Nets [5], [6] as a suitable modeling methodology and use CPN Tools [7], [8].

II. DaVinci Modeling of Substrate Network

The DaVinci architecture allows us to describe how a single substrate network 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, $k = 1, 2, ..., N$;

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

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 mechanisms.

III. Formal 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.

J. Asmuss is a PhD student in Telecommunications, Faculty of Electronics and Telecommunications, Riga Technical University, Riga, Latvia (corresponding author to provide phone: 371-28345627; e-mail: julija.asmuss@ rtu.lv).

G. Lauks is a Professor at the Institute of Telecommunications, Faculty of Electronics and Telecommunications, Riga Technical University, Riga, Latvia (e-mail: gunars.lauks@rtu.lv).

V. Zagorskis is with the Institute of Telecommunications, Faculty of Electronics and Telecommunications, Riga Technical University, Riga, Latvia (e-mail: victors.zagorskis@rtu.lv).

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 :

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

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 $g^{(k)}(\mathbf{z}^{(k)})$.

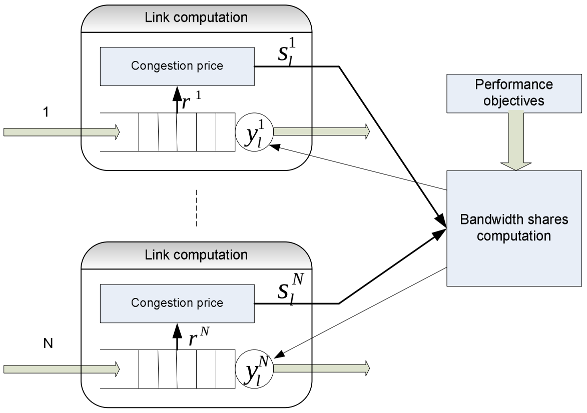

Fig. 1 Bandwidth shares computation scheme

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:

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

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:

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

An optimization scheme (Fig. 1) follows directly from DaVinci principles. First, the substrate network determines how satisfied each VN is with its allocated bandwidth. Congestion price $s_l^{(k)}$ (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.

IV. Modeling Based on Coloured Petri Nets

Our simulation scheme is based on Coloured Petri Nets. Coloured Petri Nets (CPN) is one of several mathematical modeling languages for the description of Discrete Event Systems [5]– [7].

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, $P \ne \varnothing$

T is a finite set of transitions, $T \ne \varnothing,\ P \bigcap T = \varnothing$,

D is a finite set of types, $D \ne \varnothing$,

$Type: P \cup T \to 2^D$ 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,

$N^{PLACE} = N^{\{(p,q):p\in P,\ q\in Type(p)\}}$ is the set (multiset) of all markings,

$Pre, Post : TRANS \to N^{PLACE}$ are the backward and forward incidence functions assigning marking to each transition mode,

$M_0 \in N^{PLACE}$ is the initial marking.

A CPN model of a system describes the states of the system and the events (transitions) that can cause the system to change state. By making simulations of the CPN model by using CPN Tools [7],[8] it is possible to investigate different scenarios and explore the behaviors 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.

V. Simulation Scheme

The aim of our simulation experiment is to analyze virtual network switching due to demand for extra link bandwidth. To simplify the problem we experiment with bandwidth allocation protocols for two virtual networks.

Let us consider simulations on link level. The following simulation scheme for two nodes topology (Fig. 2) includes: G1, G2 – packet generators; D1, D2 – destination nodes.

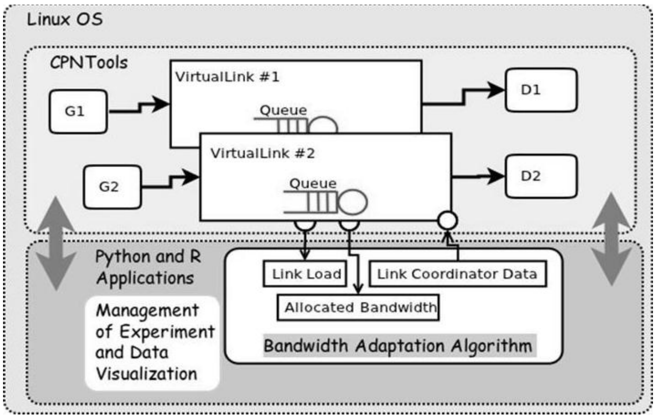

Fig. 2 Simulation scheme using CPN Tools

According to DaVinci architecture all data packets which are generated by both generators are handled and transmitted over virtual networks. Both virtual links (VL) reside on the substrate link (SL) and share the same link bandwidth. Bandwidth of SL in our experiment is determined 100 Mbps or exactly 10 ns one bit. In order to present a real network problem, in CPN models we use simulation timing determined by Model Time Units (MTU). That gives opportunity to simulate concurrent systems like two or more simultaneously running networks or links. First two columns of Table I show network BW corresponding MTU values. Columns 3–6 show some combinations how the capacity 100 Mbps can be shared with two different virtual links ($y^{(1)}$, $y^{(2)}$) in many different ways.

TABLE I
Network BW Corresponding MTU Values

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

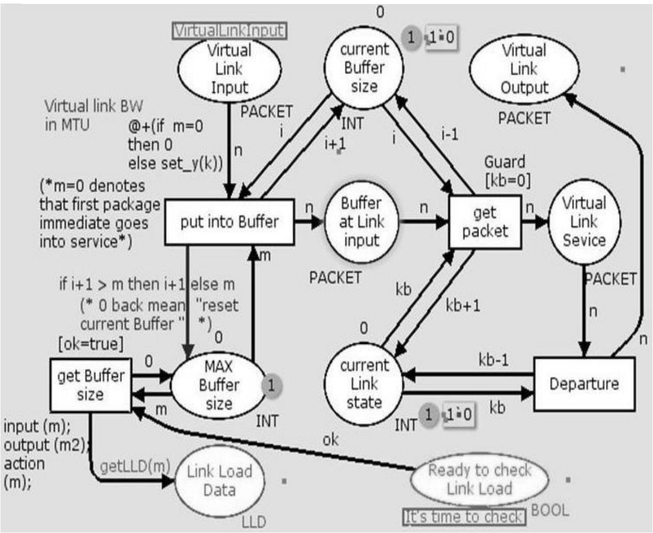

Fig. 3 Modeling of a virtual link using CPN Tools

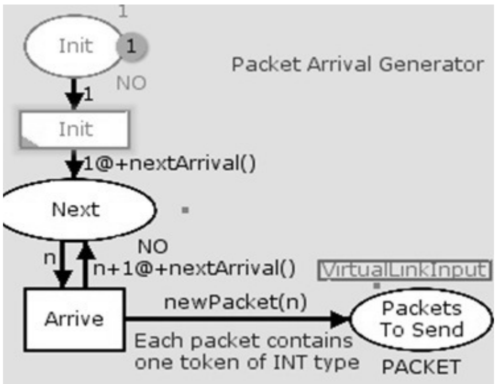

Fig. 4 Traffic simulation using CPN Tools

In our CPN network model packets are generated (Fig. 4) with exponentially distributed inter arrival time. There is time scheduled monitoring of load of each virtual link (Fig. 5). For instance, if 10 MTU correspond to 10 ns, then for 0.1 s time delay in simulator model there should be delay in 10 MTUs.

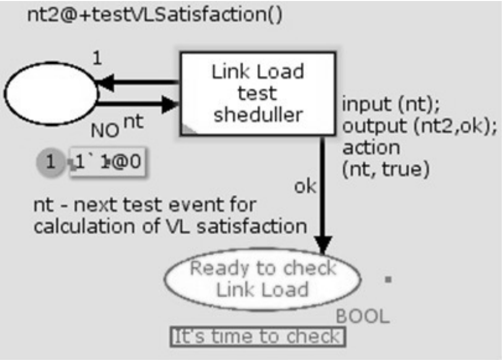

Fig. 5 Load monitoring using CPN Tools

VI. Some Numerical Results

Up to now there where some experiments with BW adaptation for two virtual links. After 10000 MTU since the start of simulation there was a scheduled event to check link load and calculate satisfaction for both links.

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).

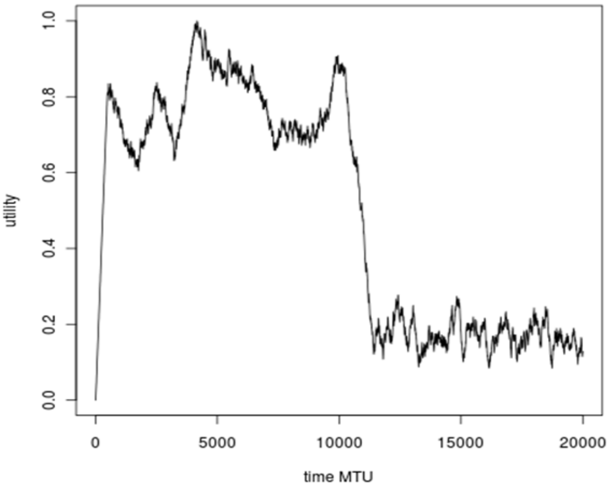

Fig. 6 Utility of the first virtual link

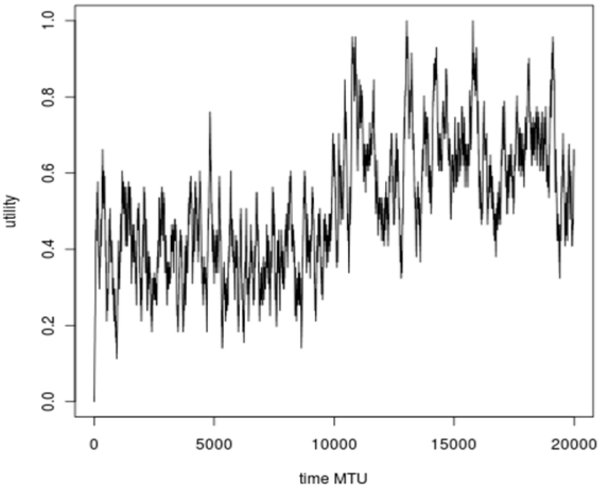

Fig. 7 Utility of the second virtual link

As a result (Fig. 6, Fig. 7), 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. But there still is a subject for optimization after next scheduled step because allocated bandwidth for the first link is more than necessary. But the parameters of arriving packets generators are not changed.

References

[1] T. Anderson, L. Peterson, S. Shenker and J. Turner, “Overcoming the Internet impasse through virtualization”, *IEEE Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[2] A. Haider, R. Potter, A. Nakao. “Challenges in resource allocation in network virtualization”, *20th ITC Specialist Seminar*, vol. 20, Hoi An, Vietnam, May 2009.

[3] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, M. Chiang, “DaVinci: dynamically adaptive virtual networks for a customized Internet”, *ACM CoNEXT Conference*, ACM, New York, Dec. 2008.

[4] X. Lin, N. Shroff. “Utility maximization for communication networks with multi-path routing”, *IEEE Trans. Automatic Control*, vol. 51, pp. 766–781, May 2006.

[5] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. Volumes 1–3, Monographs in Theoretical Computer Science, Springer-Verlag, 1992–1997.

[6] V. Gehlot and C. Nigro, “An introduction to system modeling and simulation with Coloured Petri Nets”, *Proc. of Winter Simulation Conference*, Baltimore, Maryland, USA, pp. 104–118, Dec. 2010.

[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*, no. 9, pp. 213–254, 2007.

[8] Home Page of the CPN Tools, http://cpntools.org. Accessed on May 10th, 2012.