Coloured Petri Nets based simulation scheme for adaptive bandwidth management

Julija Asmuss, Gunars Lauks

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

Abstract: *We consider the problem of resource allocation for multiple classes of traffic in a substrate network with DaVinci architecture. According to DaVinci (Dynamically Adaptive Virtual Networks for a Customized Internet) approach a single physical network can support multiple traffic classes with different performance objectives by means of multiple virtual networks constructed over the physical one. In this context the problem of bandwidth allocation is a conditional maximization problem for the aggregate performance of all virtual networks. We describe Coloured Petri Nets (CPN) based models and present CPN based schemes for simulation of dynamically adaptive bandwidth allocation mechanisms proposed to support multiple traffics in a substrate network. The simulation study focuses on two traffic types: delay sensitive and throughput sensitive. Obtained simulation results demonstrate the achievement of dynamically adaptive bandwidth management strategies.*

Keywords: Bandwidth Allocation Problem, Coloured Petri Nets, Simulation.

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 (Anderson et al., 2005) can be used for building experimental platforms that run multiple virtual networks. 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 a substrate network are shared between all virtual networks. Finding the proper bandwidth allocation to virtual networks is one of the key problems of network virtualization (Szeto et al., 2003; Zhu et al., 2006; Haider et al., 2009; Dramitinos, 2009; Zhang et al., 2009; Zhou et al., 2010).

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 (He et al., 2008). 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 (Lin et al., 2006), which effective solution depends on the design of dynamically adaptive bandwidth allocation protocols. When different types of traffic coexist over the same substrate network, 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.

Bandwidth allocation problem

The DaVinci architecture (He et al., 2008) allows us to describe how a single substrate network can support multiple traffic classes, each with a different performance objective. We describe the topology of a substrate network by a graph $G_S = \{V_S, E_S\}$, given by a set V_s of physical nodes (or vertices) and a set E_S of physical 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 virtual networks, indexed by k, where $k = 1, 2, ..., N$. According to the DaVinci approach, each traffic class is carried on its own virtual network 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 key notations for virtual network $k, \quad k = 1, 2, ..., N,$ be the following:

- $\mathbf{y}^{(k)}$ – bandwidth of virtual network k,
- $\mathbf{z}^{(k)}$ – path rates for virtual network k,
- $\lambda^{(k)}$ – satisfaction level degree of virtual network k,
- $U^{(k)}$ – performance objective for virtual network k.

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 $\mathbf{y}^{(k)}$.

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 $g^{(k)}(\mathbf{z}^{(k)})$. The goal of the substrate network is to optimize the aggregate utility of all virtual networks

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

under constraints

$$\sum_{k=1}^{N} \mathbf{y}^{(k)} \le \mathbf{C}, \quad \mathbf{H}^{(k)}\mathbf{z}^{(k)} \le \mathbf{y}^{(k)}, \quad g^{(k)}(\mathbf{z}^{(k)}) \le 0, \quad \mathbf{z}^{(k)} \ge \mathbf{0}, \quad k = 1, 2, ..., N.$$

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 the link load 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.

An optimization scheme follows directly from DaVinci principles. First, the substrate network determines how satisfied each virtual network is with its allocated bandwidth. Satisfaction level degree $\lambda_l^{(k)}$ (for link l of virtual network k) is an indicator that a virtual network may want more resources. 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.

Coloured Petri Nets based models

The concept of Coloured Petri Nets (Jensen, 1992–1997) 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.

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

- P is a finite set of places;
- T is a finite set of transitions, $P \cap T = \varnothing$, $P \cup T \ne \varnothing$;
- F is a set of directed arcs, $F \subset (P \times T) \cup (T \times P)$;
- Σ is a finite set of types (colour sets), $\Sigma \ne \varnothing$;
- W is a finite set of typed variables, $Type(w) \in \Sigma$ for all $w \in W$, where $Type: W \to \Sigma$ is a type function assigning types (colour sets) to variables;
- $C: P \to 2^{\Sigma}$ is a colour function assigning colour sets to each place, $C(p) \subset \Sigma$ for all $p \in P$;
- $G: T \to EXPR(W)$ 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 $Bool = \{true, false\}$;
- $H: F \to EXPR(W)$ is an arc expression function assigning an expression $H(f) \in F$ to each arc $f \in F$, $Type(H(p,t)) \in C(p)$ for all arcs $(p,t) \in F$ and $Type(H(t,p)) \in C(p)$ for all arcs $(t,p) \in F$;

- 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,F) 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 modelling 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 behaviour of systems (Jensen, 1992–1997; Kristensen et al., 1998; Jensen et al., 2007; Gehlo et al., 2010).

Coloured Petri Nets, proposed by Kurt 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 Kurt 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 Coloured Petri Nets Tools (Ratzer et al., 2003; Jensen et al., 2007). CPN Tools is a discrete event modeling computer tool for CPN models supporting interactive and automatic simulations, state spaces and performance analysis, and combining Coloured Petri Nets and the functional programming language CPN ML, which is based on Standard ML.

Colours can be effectively used for modelling virtual networks accordingly to the DaVinci architecture. A CPN model of a substrate network describes the states of each virtual network of the system and the events (transitions) that can cause the system to change state. By making simulations of the network CPN model with CPN Tools it is possible to investigate different scenarios and explore the behaviours of the system, to use simulation-based performance analysis for decision making and adaptation processes.

Simulation scheme

The simulation study focuses on two traffic types: delay sensitive and throughput sensitive. Accordingly to DaVinci principles 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. 1) and use the following notations: G_A, G_B – packet generators; D_A, D_B – destination nodes.

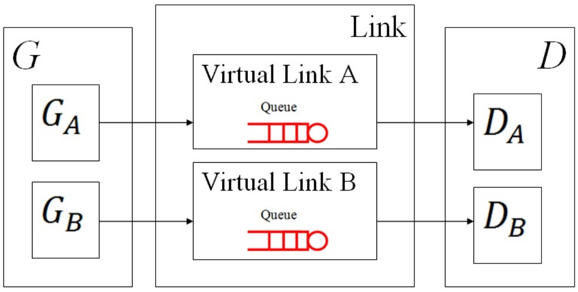

Fig. 1. **Simulation scheme for two nodes topology.**

Colours *A* and *B* are effectively used for modelling and simulating such system. The colours and variables associated with the tokens which represent packets are defined as follows:

```
colset PacketType = with A | B;
colset PacketSize =int;
colset Packet = record packetType : PacketType * packetSize : PacketSize * AT:int;
colset Packets = list Packet;
var packet: Packet;
var packets: Packets.
```

We also use colours *A* and *B* as identifiers for the corresponding virtual links. It is the number of tokens and the token colours which represent the state of a virtual link:

```
colset VirtualLinkType = with A | B;
colset VirtualLinkStatus = with VL_Status timed;
colset VirtualLinkBandwidth = int;
colset VirtualLink = record VLtype : VirtualLinkType * VLstatus : VirtualLinkStatus * VLbandwidth :
VirtualLinkBandwidth;
colset VirtualLinkxPacket = product VirtualLink * Packet timed.
```

In addition to token colours we use time stamps. As usual in timed CPN models, we use Model Time Units (MTU) and define additional time variables:

```
colset INT = int;
var proctime : INT.
```

The CPN Tools model (Fig. 2) involves transmission modules and bandwidth shares adaptation modules:

- Arrivals;
- Virtual Link;
- To Bandwidth Adaptation;
- From Bandwidth Adaptation.

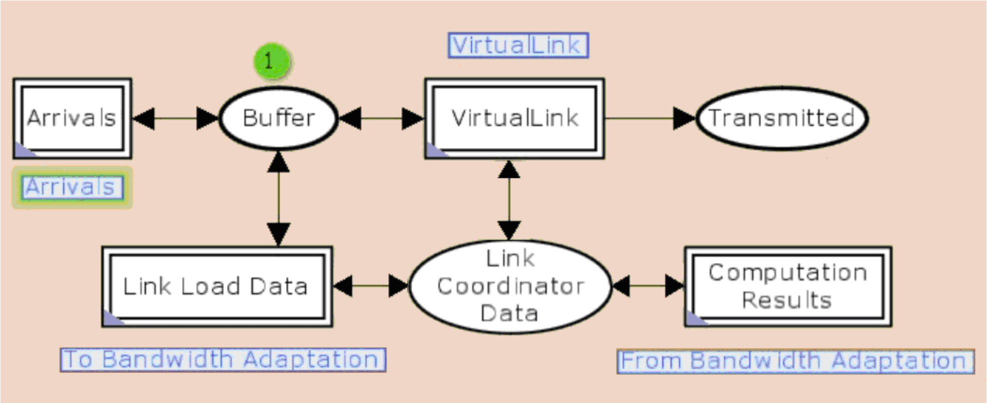

Fig. 2. **Bandwidth adaptation simulation scheme using CPN Tools.**

Packets are generated by the traffic generators (Arrivals) and stored in the FIFO 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. This condition fulfilled if the place Idle (Fig. 3) is marked. If this place is empty, then the transmission link is busy. Technically token VL_Status is used to organize the queue and packets transmission as it is shown in Fig. 3. 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 (Fig. 2) 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 analysed.

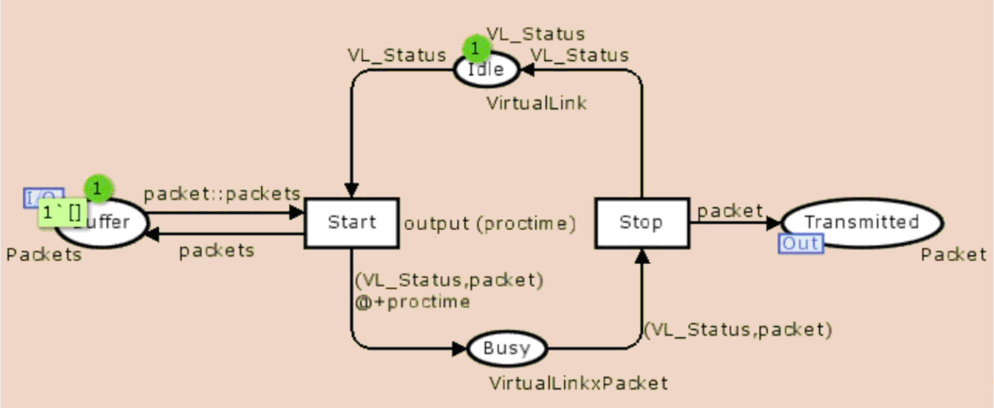

Fig. 3. **Virtual link simulation scheme using CPN Tools.**

Bandwidth allocation adaptation modules are designed to update the virtual link resources allocation for dynamically changing traffics. Special predicate and observation functions (fun pred, fun obs) and data collection monitors (Buffer Delay, Buffer Length, Virtual Link Utilization) are included for monitoring the performance of the system. A 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*. A 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 depend on the aggregate objective function for both virtual links.

In our experiments the initial resource allocation to 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 every 50000 MTU we observe the adaptation process and obtain good adaptation results after 2-3 adaptation iterations.

We experiment with delay sensitive traffic (the objective is to minimize the average delay) and throughput sensitive traffic (the objective is to maximize the average link rate) 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 show 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.

Conclusion

In this paper we present the design and experimental evaluation of a simulation scheme for an adaptive bandwidth allocation mechanism, which is realized for two nodes topology. This simulation scheme is based on Coloured Petri Nets and realized by using CPN Tools. The effectiveness of our scheme is evaluated within simulation experiments with two types of traffic: delay sensitive and throughput sensitive.

We see several possible extensions of this work. Firstly, our future work will focus on extension of the proposed simulation scheme in order to generalize the network topology. Instead of two nodes topology we plan to consider Access-Core topology, Abilene topology, etc. We should extend the model from local link level to global network level by involving a global coordinating mechanism. It is clear that the virtual link bandwidth assignment on one physical link would be dependent on the virtual links bandwidth assignments on another physical links. Of course, the model mentioned above requires more complicated bandwidth allocation scheme. Another future extension will generalize the proposed simulation scheme from two traffic classes to multiple traffic classes by involving additional colours in the CPN model. At the same time we will enhance our model with more functions in order to modify and improve monitoring and decision making systems and, as a result, to optimize adaptive bandwidth management.

References

Anderson, T., Peterson, L., Shenker, S. and Turner, J., 2005. Overcoming the Internet Impasse through Virtualization. *IEEE Computer*, 38 (4), pp.34-41.

Dramitinos, M. L., 2009. A Bandwidth Allocation Mechanism for 4G. *Proceedings of European Wireless Technology Conference*, Rome, pp.96-99.

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, pp.104-118.

Haider, A., Potter, R. and Nakao, A., 2009. Challenges in Resource Allocation in Network Virtualization. *Proceedings of the 20th ITC Specialist Seminar*, Hoi An, Vietnam. Available at: www.itcspecialistseminar.com/paper/itcss09_ Haider.pdf, 20.12.2012.

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. *Proceedings of ACM CoNEXT Conference*, Article No. 15, ACM, New York, USA. Available at: www.cs.princeton.edu/~jrex/papers/davinci.pdf, 20.12.2012.

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, pp.213-254.

Jensen, K. and Kristensen, L., 2009. *Coloured Petri Nets. Modelling and Validation of Concurrent Systems*. Springer-Verlag.

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, pp.98-132.

Lin, X. and Shroff, N., 2006. Utility Maximization for Communication Networks with Multi-Path Routing. *IEEE Trans. Automatic Control*, 51, pp.766-781.

Ratzer, A.V., Wells, L., Larsen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S. and Jensen, K., 2003. CPN Tools for Editing, Simulating and Analysing Coloured Petri Net. *Lecture Notes in Computer Science*, 2679, pp.450-462.

Szeto, W., Iraqi, Y. and Boutaba, R., 2003. A Multi-Commodity Flow Based Approach to Virtual Network Resource Allocation. *Proceedings of the Global Communications Conference IEEE GlobeCom*, San Francisco, CA, USA, pp.3004-3008.

Zhang, Y., Wang, C. and Gao, Y., 2009. A Virtualized Network Architecture for Improving QoS. *International Journal on Information Engineering and Electronic Business*, 1, pp.17-24.

Zhou, Y., Li, Y., Sun, G., Jin, D., Su, L. and Zeng, L., 2010. Game Theory Based Bandwidth Allocation Scheme for Network Virtualization. *Proceedings of the Global Communications Conference IEEE GlobeCom2010*, Miami, Florida, USA. Available at: http://ieeexplore.ieee.org, 20.12.2012.

Zhu, Y. and Ammar, M., 2006. Algorithms for Assigning Substrate Network Resources to Virtual Network Components. *Proceedings of the 25th IEEE International Conference on Computer Communications InfoCom2006*, Barcelona, Spain, pp.1-12.