

Attractor Selection Mechanism Simulink Model

Dmitrijs Finaskins

Riga Technical University
st. Azenes 16, LV-1048, Riga, Latvia
Email: dmitrijs.finaskins@rtu.lv

Gunars Lauks

Riga Technical University
st. Azenes 16, LV-1048, Riga, Latvia
Email: gunars.lauks@rtu.lv

Abstract—Telecommunication network topologies are changing nowadays. The core telecommunication networks are based on fiber optics infrastructure. WDM technology is used to transmit multiple flows of traffic over a single fiber. Virtual network topology (VNT) is used to route IP traffic over WDM networks. VNT control system must be adaptive to traffic changes and dynamically reconfigure VNT. Langevin equation is used in statistical physics to describe the time evolution of a subset of the degrees of freedom. These degrees of freedom typically are collective (macroscopic) variables changing only slowly in comparison to the other (microscopic) variables of the system. The fast variables are responsible for the stochastic nature of the Langevin equation. Langevin equation could be used to model attractor selection mechanism, which is used by biological systems to adapt to unknown changes in environment. In this paper we describe attractor selection mechanism Simulink model and analyze simulation results.

I. INTRODUCTION

CORE telecommunication networks nowadays are based on Wavelength Division Multiplexing (WDM) technology. In such division multiplexing technology transmitting/receiving channels are divided by wave length. It allows transmitting multiple traffic channels over a single fiber. Traffic by fiber optics can be transmitted over long distances without any additional equipment. Because of that WDM networks are commonly used to carry Internet traffic at backbone level. The major protocol of Internet is IP protocol. One of possibilities to carry IP traffic over WDM network is to construct virtual network topology, which includes transmitting/receiving channels (lightpaths) and IP routers. There are many VNT control methods, which configure/reconfigure VNT according to traffic demand matrices. Traffic demand matrices show how traffic flows are distributed via lightpaths.

One of the most exciting Internet opportunities for end-users is to share their pictures, videos and so on with other users; to communicate with each other using online services such as Skype. These cause constant and rapid changes in traffic flows between IP routers in VNT. There is a need to reconfigure VNT over a period of time in order to provide high-level service with minimal delays. VNT needs to be adaptable to changes in traffic demand.

There are two modes of constructing traffic demand matrices—offline and online or dynamical mode. In offline approach traffic demand matrices are constructed using previous known information about changes in traffic demand in [1], [2], [6]. The major weakness of this approach—offline methods will not work correctly if traffic flow changes will be

different from ones expected. Online approach allows reconfiguring VNT dynamically. In this case periodical measurement results are used. To evaluate VNT status, information about average or maximum link utilization, packet delays can be used. Based on this information new lightpaths are added to source-destination pair of nodes if, for example, link utilization between these nodes is more than threshold and deleted if lightpath is underutilized.

The majority of online VNT control methods are used when traffic demand is changing periodically and gradually [5]. This approach will not work if changes of traffic demand are not predictable. There is a need to develop such VNT control method, which is adaptable to unknown changes in network environment.

One of the approaches is to use attractor selection, which represents mechanism of adaptation to unknown changes of biological systems [4]. The main idea of attractor selection—the system is driven by two components—deterministic and stochastic. Attractors are a part of the equilibrium points in the solution space. Conditions of such system are controlled by very simple feedback. When conditions of a system are suitable (close to one of the attractors), it is driven almost only by deterministic behaviour, stochastic influence is very limited. When conditions of the systems are poor, deterministic behaviour influence is close to zero and in this case system is driven by stochastic behaviour. It randomly fluctuates searching for a new attractor. When this attractor is found, deterministic behaviour again dominates over stochastic.

There are some publications, which show how to use attractor selection mechanism to manage network resources (bandwidth), but in most cases authors do not write which tool was used to create a model. In this paper we propose a Simulink model, which describes approach proposed by [4]. The model is relatively large and it could be reduced by assigning multiple tasks to one block, but in this case the model will not be so easy to understand.

II. ATTRACTOR SELECTION MECHANISM

Every pair of nodes, between which connection can be established, is represented by control unit u_{ij} , where i and j are indexes of nodes. Every control unit has its control value $x_{u_{ij}}$. Indexes s and d refers to source and destination nodes. How control values change over time can be expressed by

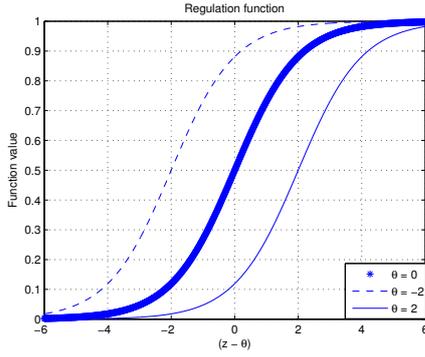


Fig. 1. Regulation function

differential equation, also known as Langevin equation:

$$\frac{dx_{u_{ij}}}{dt} = v_g \cdot f \left\{ \sum_{u_{sd}} W(u_{ij}, u_{sd}) \cdot x_{u_{ij}} - \theta_{u_{ij}} \right\} - v_g \cdot x_{u_{ij}} + \eta, \quad (1)$$

where v_g indicates conditions of VNT, $W(u_{ij}, u_{sd})$ —regulatory matrix of VNT, $\theta_{u_{ij}}$ is coefficient, which depends on minimum and maximum loads on links in VNT and η —Gaussian noise. We use Gaussian noise with 0 mean value and 0.1 variance. The first term in Equation (1) represents regulation function $f(z)$:

$$f(z) = \frac{1}{1 + e^{-z}}; z = \sum_{u_{sd}} W(u_{ij}, u_{sd}) \cdot x_{u_{ij}} - \theta_{u_{ij}} \quad (2)$$

Regulatory matrix $W(u_{ij}, u_{sd})$ is the most important parameter of Equation (1), since it shows the relationships between node pairs. Every element of this matrix can be -1, 0, or 1; it represents the influence of node pair u_{ij} on node pair u_{sd} . According to model, proposed in [3], -1 corresponds to inhibition of the node u_{ij} by u_{sd} , 0 corresponds to no relation and 1 to activation. These values are multiplied by corresponding control values of u_{ij} . In such way node pairs affect each other—if u_{sd} is activated by u_{ij} , increasing in $x_{u_{ij}}$ will increase $x_{u_{sd}}$ and, as it was mentioned above, increase the amount of lightpaths between nodes s and d . In [3] has been shown that if node pairs can activate or inhibit each other with probability 0.03 and no relation with probability 0.94, such network will be extremely adaptable to changes in traffic demand.

So for every node pair we construct regulation function, which is defined by Equation (2). Adjusting $\theta_{u_{ij}}$, we can move regulation function in the negative or positive direction, as it is shown in Figure 1.

The amount of assigned lightpaths is a function of $x_{u_{ij}}$ normalized by all control unit values of all node pairs, which use transmitters and receivers from node i to node j . The number of lightpaths $G_{u_{ij}}$ between nodes i and j is defined as follows [4]. The total number of lightpaths between nodes i and j is K_r :

$$G_{u_{ij}} = \min \left\{ K_r \cdot \frac{x_{u_{ij}}}{\sum_s x_{u_{sj}}} \right\} \quad (3)$$

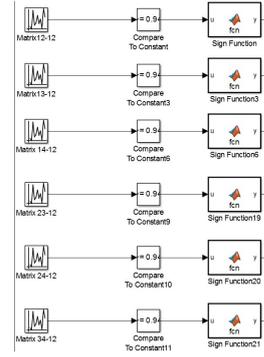


Fig. 2. Regulation matrix generation

III. MODEL LIMITATIONS, ASSUMPTIONS, OUT OF SCOPE ELEMENTS

In our Simulink model, described in this paper, we simulate 4-node network. We chose 4 nodes because creating wider network does not add additional value to the model, but only makes it more difficult. But it is not a model limitation, using our model a network with unlimited number of nodes could be simulated.

In our model we assume that connections between all node pairs have equal parameters (bandwidth, delay, etc.). This was done to simplify the model, but, in general, it is not a mandatory requirement and links with not equal parameters could be also used. As [4] we used only link utilization for network activity control. But more complex metrics could be used as well, e.g. link utilization and total delay.

Loads on links are randomly generated at certain time moments. Initial loads on links are randomly generated as well. We use uniform distribution function for this purpose. We chose uniform distribution to make generation results unpredictable and therefore to simulate such network in with traffic flow can change rapidly and unpredictably. We also assume that between two loads on links generation moments total traffic does not change.

IV. ATTRACTOR SELECTION MECHANISM SIMULINK MODEL

A. Regulatory matrix generation

As it was mentioned above, each node pair in the network affects all other nodes, including itself. We simulate 4-node network, it means that we have 6 node pairs— $x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, x_{34}$. Regulatory matrix's $W(u_{ij}, u_{sd})$ dimensions will be 6x6. In Figure 2 we show how the first row of this matrix is generated. We use "Uniform random number" block, "Compare to constant" block and "User defined function" block for generating 1 element of matrix with the given parameters (0.94 probability of no relation, 0.03 of inhibition and 0.03 for activation). First of all we generate a uniformly distributed random number $\epsilon \in [0; 1]$. Next, we compare this generated number with constant—0.94. In 94 cases "Compare to constant" block will output 0, because in 94 cases out of

Algorithm 1 Sign function block

```

function y = fcn(u)
if u==0;
    u=0;
else m=sign(0.5 - rand);
    if m>=0
        u=1;
    else u=-1;
    end
end
y = u;

```

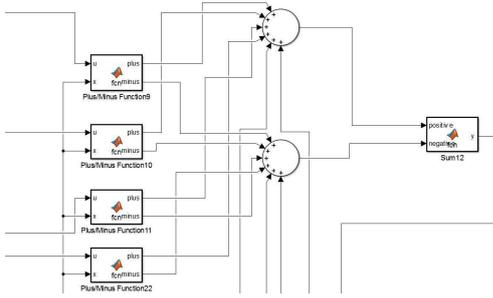


Fig. 3. Calculation of total inhibition/activation value

100 generated number will be less or equal to 0.94. In 6 cases out of 100, this block will output 1. This signal (0 or 1) is an input variable for "Sign function". In this block it is decided -1, 0, or 1 will be put in the regulatory matrix. The code of this block is listed in Algorithm 1. If "Sign function" block receives 0 as an input signal, it does nothing, but if 1, it is assumed that -1 or 1 decision should be made. Both inhibition and activation probabilities are equal, so result of rand function is subtracted from 0.5. By this we ensure that in 50 cases out of 100 we have -1 and in other 50 cases 1. In Figure 2 it could be seen how the signal is processed further. We need to calculate the total inhibition and activation value—this is done by multiplying -1 or 1 with $x_{u_{ij}}$. We divide this flow into two parts—positive and negative, corresponding to activation and inhibition. The total value of $W(u_{ij}, u_{sd})$ for $x_{u_{ij}}$ is calculated in block "Sum 12". The code for this block could be found in Algorithm 2. This value is used for regulatory function calculation.

B. Load on links generation

As it was mentioned above, we use uniform distribution function to generate load on links. We randomly change load on links at time moments $t \in [0,2,5,8]$. At all other time moments we use load on links calculated by attractor selection mechanism. Please refer to Algorithm 3 to see how it happens. We use variable t to control the current simulation time. ry_{12} variable is used for random load on link generation, y_{12} corresponds to load on link calculated by algorithm and yo_{12} is load on link, which is processed to the next blocks—"Min" and "Max" blocks. We multiply yo_{12} by 100 to reflect the

Algorithm 2 Calculation of the total value of $W(u_{ij}, u_{sd})$ for $x_{u_{ij}}$

```

function y = fcn(positive, negative)
m=0;
n=0;
if positive == 0
    m=0;
else m = (1.08*6)/positive
end

if negative == 0
    n=0;
else n = (1.08*6)/negative
end
y = m + n;

```

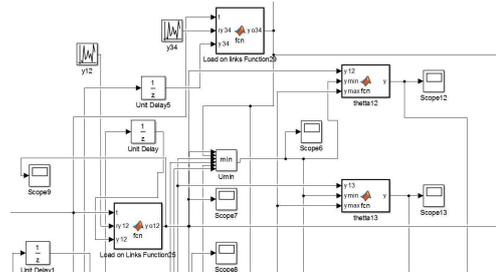


Fig. 4. Load on links generation

Algorithm 3 Load on links management

```

function yo12 = fcn(t, ry12, y12)
if t==0
    yo12=100*ry12;
elseif t==2
    yo12=100*ry12;
elseif t==5
    yo12=100*ry12;
elseif t==8
    yo12=100*ry12;
else
    yo12=y12;
end
end

```

Algorithm 4 $\theta_{u_{ij}}$ value calculation

```

function y = fcn(y12, ymin, ymax)
tc = 5;
if ymin == ymax
    m = 0;
    y = m;
else
m = -((y12 - ymin) / (ymax - ymin)) * 2 * tc - tc
y = m;
end

```

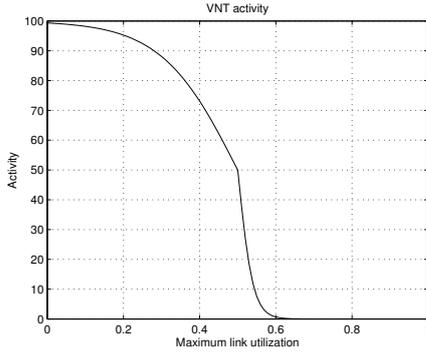


Fig. 5. VNT Activity

fact that 100 Gbps links are used, but this coefficient is not mandatory and could be excluded.

We need to know minimal and maximal load on link in our network to calculate regulation function. "Min" and "Max" blocks allow to do this very easily. These values are transferred to block "Theta", which calculates $\theta_{u_{ij}}$ value. Please refer to Algorithm 4 for code details. Condition $y_{min} = y_{max}$ is mandatory for catching exceptions and is highly recommended to be used.

C. Network activity and regulation function

We have discussed already how initial control values x_{sd} and load on links y_{sd} are generated, to integrate Equation (1) we need to calculate activity function. Activity function is defined [4] as follows:

$$v_g = \frac{100}{1 + \exp(\delta \cdot (u_{max} - \xi))}, u_{max} \geq \xi \quad (4)$$

$$v_g = \frac{100}{1 + \exp(\frac{\delta}{5} \cdot (u_{max} - \xi))}, u_{max} < \xi \quad (5)$$

Figure 5 shows how activity function looks like. From Equations (4) and (5) it can be seen that ξ is a threshold for VNT activity v_g . If maximum link utilization is more than ξ , activity of VNT dramatically degraded and stochastic behaviour dominates over deterministic and the system is searching for a new attractor. Algorithm 5 shows Simulink code for activity calculation and Figure 6 shows network activity v_g for 4-node network. For regulation function calculation we

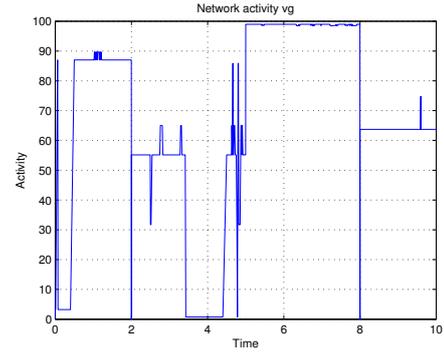


Fig. 6. Activity function

Algorithm 5 v_g calculation

```

function y = fcn(umax)
igrek = 100;
beta = 50;
ksi = 0.5;
if umax >= ksi
vg = (igrek / (1 + exp(beta * (umax - ksi))));
else
vg = (igrek / ((1 + exp((beta / 5) * (umax - ksi))));
end
y = vg;

```

use Equation (2). All needed parameters/variables have been already calculated in the previous steps.

For Gaussian noise generation random number generation block is used. As it was mentioned above, mean value of this signal is 0 and variance 0.1. It was shown in [3], using Gaussian noise with these parameters makes attractor selection mechanism extremely adaptable for unknown and unpredictable changes in environment.

D. Numerical integration of system of differential equations

Equation (1) has 3 terms—regulation function multiplied by activity function, control value $x_{u_{ij}}$ multiplied by activity function and Gaussian noise. As it can be seen in Figure 7, we use sum block with 3 inputs to sum these three terms. Regulation and activity functions calculations are represented

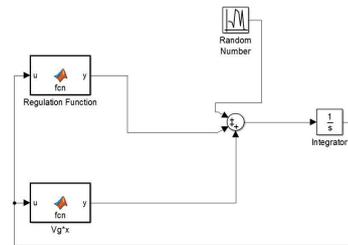


Fig. 7. Integration concept

as "User defined function" blocks. Both blocks use during the previous timeslot integrated $x_{u_{ij}}$ value. We have to solve such differential equation for each node pair in the network. As it was mentioned above, each two nodes in the network form a node pair, regardless where these nodes are located and with no respect to the fact if they are directly connected. For example, in our simulated network we have 4 nodes. It means that we have 6 nodes pairs. So at each time moment we need to solve a system of 6 differential equations. In general, if we have N nodes in the network, we will have $\frac{N \cdot (N-1)}{2}$ differential equations.

E. Lightpaths assignment and load on links recalculation

As soon as we have integrated Equation (1), that is, have $x_{u_{ij}}$ values, we need to assign/reassign lightpaths to node pairs according to Equation (3). We use the following model for load on links recalculation—at time moment $t = 0$ we randomly generate load on links. When we have $x_{u_{ij}}$, we assign additional lightpaths for node pairs (At time moment $t = 0$ each node pair have only 1 lightpath assigned). When additional lightpaths are assigned, we recalculate load on links by dividing current load on link with assigned number of lightpaths. For example, if at time moment $t = 0$ we have $y_{12} = 80$ Gbps and there are 3 additional lightpaths assigned to this node pair, then the new load on this link would be $\frac{80}{4} = 20$ Gbps.

After one integration period (sample period) passed and we got new $x_{u_{ij}}$ values, we need to recalculate load on links. First of all, we calculate the total load on link for y_{ij} by multiplying current load on link distributed by all lightpaths with the number of lightpaths. For example, if we take y_{12} mentioned in the previous paragraph, we multiply 20 Gbps with number of lightpaths (4) and get initial value 80 Gbps. Then we recalculate number of lightpaths which will be assigned to this node pair according to Equation (3) and divide load on link with this new amount of lightpaths. For example, at step 2, 5 lightpaths were assigned to y_{12} . It means that new load on link for this node pair will be $\frac{80}{5} = 16$ Gbps.

To make this algorithm work, we need to somehow save the previous value of lightpaths assigned to node pair y_{ij} at time moment T_{t-1} as well as current number of lightpaths for time moment T_t . This could be easily done in Simulink by using one of two blocks—"Memory block" or "Unit delay block". Both these blocks delay an input signal for one integration period (sample time), we used "Unit delay block" in our simulation. You can see a fragment of our Simulink model which recalculate load on links in Figure 8 and Algorithm 6 shows source code. If we refer to Figure 8, t represents simulation time, y_{ij} current load on links, G_{ij} new amount of lightpaths assigned to y_{ij} , dG_{ij} —number of lightpaths for y_{ij} from the previous integration step and ey_{ij} is a new load on link. As it can be seen from Algorithm 6, at time moments $t \in [0, 2, 5, 8]$ load on links are randomly generated and in that cases we do not use previous number of lightpaths to restore the original load on link.

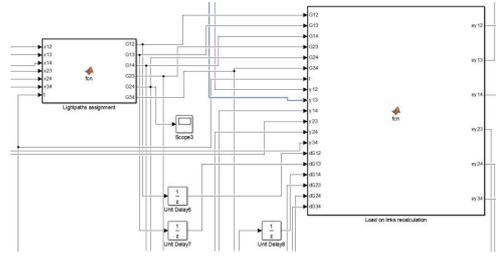


Fig. 8. Load on links recalculation

Algorithm 6 Load on links recalculation

```

if t==0
    ey12=y12/G12;
ey13=y13/G13;
ey14=y14/G14;
ey23=y23/G23;
ey24=y24/G24;
ey34=y34/G34;
elseif t==2
    ey12=y12/G12;
ey13=y13/G13;
ey14=y14/G14;
ey23=y23/G23;
ey24=y24/G24;
ey34=y34/G34;
elseif t==5
    ey12=y12/G12;
ey13=y13/G13;
ey14=y14/G14;
ey23=y23/G23;
ey24=y24/G24;
ey34=y34/G34;
elseif t==8
    ey12=y12/G12;
ey13=y13/G13;
ey14=y14/G14;
ey23=y23/G23;
ey24=y24/G24;
ey34=y34/G34;
else
    ey12=(y12*dG12)/G12;
    ey13=(y13*dG13)/G13;
    ey14=(y14*dG14)/G14;
    ey23=(y23*dG23)/G23;
    ey24=(y24*dG24)/G24;
    ey34=(y34*dG34)/G34;
end
end

```

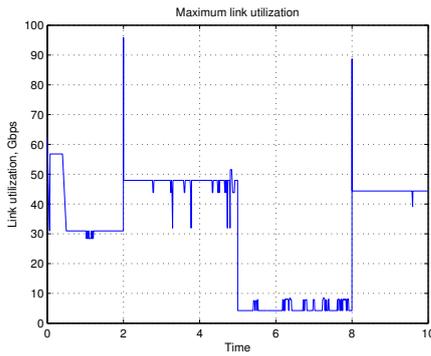
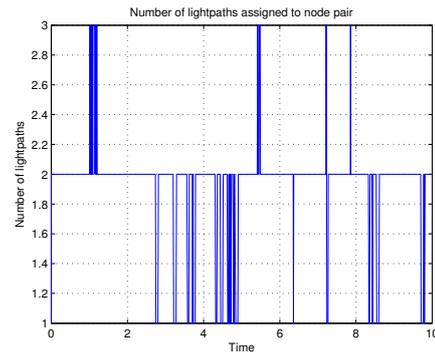


Fig. 9. Maximum link utilization in the network

Fig. 10. Number of lightpaths assigned to node pair y_{ij}

V. SIMULATION RESULTS

In this section we present results of simulating 4-node network using proposed attractor selection mechanism Simulink model. Please refer to Figure 9 to see how maximum link utilization in the network changes over time. The maximum link capacity is 100 Gbps, at time moments $t \in [0, 2, 5, 8]$ traffic was randomly generated. As it can be seen from Figure 9, attractor selection mechanism works well and when load on links changes, network adapts to this changes and maximum link utilization decreases.

The one could see an interesting effect at time moment $t \in [2; 5]$ and $t \in [5; 8]$. Although traffic amount does not change over time during these time intervals (according to our simulation traffic changes are done only at time moments $t \in [0, 2, 5, 8]$, at all other time moments traffic is constant), we can see fluctuations in maximum link utilization. It is a side effect of using Gaussian noise in Equation (1). This effect was described in [4]. The system conditions are good, but the maximum link utilization increases unexpectedly (see Figure 9), although there are no changes in traffic demand at this time moment. It happens because stochastic behaviour always influences system conditions, even if VNT conditions are good. It causes VNT activity degradation. At this moment noise drives the system and a new attractor is being searched. After some time a new attractor has been found and system activity recovers. Because of noise random nature it cannot be predicted when it will happen.

But on the other hand the same effect helps to reduce maximum link utilization at time moments $t \in [0, 2, 5, 8]$. At these time moments, due to changes in traffic demand and as a result changes in maximum link utilization, VNT activity degrades. In these moments stochastic behaviour dominates over deterministic as the first two terms in Equation (1) are zero or close to zero. Control unit values randomly fluctuate searching for a new attractor. As soon as a new attractor is found, VNT activity recovers and deterministic behaviour again drives VNT.

Figure 10 shows how amount of lightpaths assigned to node pair y_{ij} changes over time. As it can be seen from this figure, there are some fluctuations in number of lightpaths assigned

to y_{ij} . There are two main reasons for that. The first one, as it was mentioned above, is stochastic fluctuations which affect network activity and network activity affects all other network parameters. The second reason is that we do not use any mechanisms for such fluctuation filtering. If filtering is used, we could prevent number of lightpaths from flapping.

Overall, simulation results show that our proposed attractor selection mechanism Simulink model works well. This proves that Simulink provides great opportunity for researches and scientists to use this program to get practical results for their theories/algorithms.

VI. DISCUSSION AND FURTHER WORK

Uniformly distributed function for traffic demand matrix generation has been used. This distribution has been selected as it allows to study how control method will adapt to unknown changes if traffic demand will vary from 0 to 1 (with respect to link capacity) with equal probability. On the other hand, it is vital to know how attractor selection mechanism deals with traffic in case when traffic demand matrix is generated using normal distribution (Gaussian distribution). Some other distribution functions such as logarithmic and Poisson are also needed to be studied—if attractor selection Simulink model is able to successfully deal with such distributed traffic.

Another improvement is to use more complicated network for simulation. In this case, a model should be modified, as it does not support transit lightpaths from some nodes to others, because in real networks each node is not directly connected any other. Because of that we need to modify block which assigns lightpaths to support some lightpaths reservation for transit purposes.

We also want to make metrics for network activity measurements more complex. Now the decision about network conditions is made based only on maximum link utilization. Some others metrics could be added, for example, total delay between x_s and x_d .

REFERENCES

- [1] G. Agrawal and D. Medhi, "Lightpath topology configuration for wavelength-routed IP/MPLS networks for time-dependent traffic," Proceedings of GLOBECOM, 2006, pp. 1-5.

- [2] B. Chen, G. N. Rouskas, and R. Dutta, "On hierarchical traffic grooming in WDM networks," *IEEE/ACM Transactions on Networking*, vol. 16, 2008, pp. 1226-1238.
- [3] C. Furasawa and K. Kaneko, "A generic mechanism for adaptive growth rate regulation," *PLoS Computational Biology*, vol. 4, 2008, p. e3.
- [4] Y. Koizumi, T. Miyamuta, S. Arakawa, E. Oki, K. Shiimoto, M. Murata, "Application of attractor selection to adaptive virtual network topology control," *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems*, 2008, Article No. 9.
- [5] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E.D. Kolaczyk and N. Taft, "Structural analysis of network traffic flows," *Proceedings of ACM Sigmetrics*, 2004, pp. 61-72.
- [6] F. Ricciato, S. Salsano, A. Belmonte and M. Listanti, "Off-line configuration of a MPLS over WDM network under time-varying offered traffic," *Proceedings of IEEE INFOCOM*, vol. 1, 2002, pp. 57-65.