

Dataflow Features in Computer Networks

Gyula Max

Budapest University of Technology and Economics,
Department of Automation and Applied Informatics
H-1521 Budapest, Po. Box. 91. Hungary
e-mail: max@aut.bme.hu

Abstract. Dataflow in computer networks can be describe as a complicated system. To simulate behaviours of these systems are also difficult, because these systems show non-linear features. Before implementing network equipments users wants to know capability of their computer network. They do not want the servers to be overloaded during temporary traffic peaks when more requests arrive than the server is designed for. According to the non-linear character of network traffic, a system model is established to exam behaviour of the network planned. This paper presents setting up a non-linear simulation model that helps us to handle dataflow problems of the networks.

Keywords: non-linear system analysis, computer networks, data congestion, stability

1 Introduction

In a large-scale computer network data packets may choose among many routes to arrive their destination address. These selections are based on knowledge of recent nodes of the computer networks. In most cases parts of the computer networks do not know all levels of the fully network. Though almost all nodes take part in the process of forwarding packets, nodes of the computer network know more on neighbours than far ones. In this paper, we firstly summarize the contributions and limitations included in the existing works. In second session the non-linear model and its behaviours are established focused on congestion problems and stability. The third session explains the estimation process of model parameters and criteria, while in the fourth session an case study is given. These system parameters and criteria are used to analyze quantitatively behaviour and stability of system as well as to investigate the impact of system parameters, such as data density, load level, link capacity on system stability. The simulation results validate our analysis. Last session shows tendency of future works.

2 Related Works

To study network behaviour, traffic parameters are needed to measure and analyse, and to find their statistical laws, such as was done by Y. Bhole and A. Popescu [1]. After having known the statistical rules, some traffic models are built, such as the establishment of a model for Novel network traffic by J. Jiang and S. Papavassiliou in [2] or we try to understand why buffers are overflowed like S. Sidirolou et al. did in [3].

Congestion control methods. We also set up congestion models. Model of Du Haifeng et al. [4] performs an effective network congestion control method for multilayer network. If operational cost is important read I. Chuang's article [5] on Pricing Multicast Communication. If the time-scale of network traffic is considered, the network traffic behaviour will be different in different time-scale. Paxson and Floyd [6] showed that the traffic behaviour of millisecond-time scale is not self-similar by the influence of network protocol. Due to the influence of environment, the traffic behaviour whose time-scale is larger than ten minutes is not also self-similar and is a non-linear time-series. Only the traffic behaviour in second-time scale is self-similar. This problem becomes known when transactions of the computer networks have to be described. As it was presented before, the system shows non-linear features, i.e. transactions. It means no tools to describe transactions by mathematics of linear control systems. In this paper, a new non-linear traffic behaviour model is set up and by the aid of this model, the non-linear features of dataflow are described.

Traffic simulation. The most extensive and modern researches present directions of network parameter estimations [7], analyses of traffic generators [8, 9], elastic network nodes [10] etc. The non-linear analyses of network nodes [11, 12, 13, 14] is a separate important research area. Therefore, very important the optimal function of the nodes in the system. In terms of the data traffic that would either be the ideal if fewer nodes existed in the network, or in the case of huge number of nodes these all were linked to all the others. This apparently an absurd approach, already if we look at the economic sides of the solution only. A question becomes known, that necessary, that let the nodes be on the central place of the examinations. The correct answer is in this direction, that in terms of the data traffic, the whole network is necessary to put onto the examined central place.

Stability control. Many papers have theoretically analyzed computer networks in framework of feedback control theory based on the continuous time model (e.g. [15-19] etc.) or discrete-time model (such as [20] and [21]) after having made some necessary simplification and assumption, and finally provided some very revelatory and significant conclusions and judgments. In [16], a general nonlinear model was developed, then after linearisation it around equilibrium and making several simplifying assumptions, the sufficient stability condition was obtained. The main contribution in [16] was to replace the single-link identical-source model in [15] with a general model with heterogeneous sources, but components

were approximated linearly. Under this assumption, some possible factors leading to the queue oscillation have been successfully identified, however, the others will be likely hidden.

3 Data Flow Model

The computer network models known by the literature trace back the data traffic to the description of a communication happening between edges and nodes of a communication graph. The models handle the nodes as an important element in this description method. The outcome of model is the communication graph that faithfully imitates the physical arrangement of the computer network, where the nodes, the active elements of the system are the peaks of the graph, which are connected to each other by the transfer mediums called edges. This statement is important, because in the network actually the nodes, the active elements of the network communicate with each other, and the nodes form the peaks of the network graph in the graph theory model of the computer network. Figure 1 shows a plain example how nodes communicate:

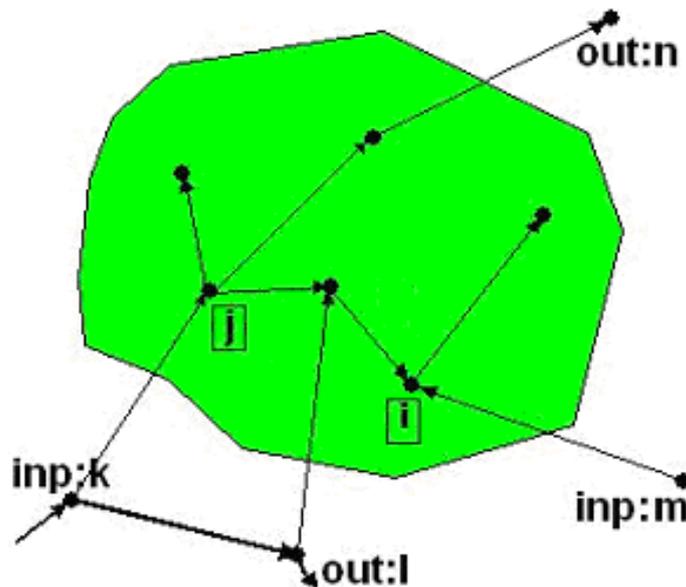


Figure 1

Internal and external elements of a network

This description gives back that view on a natural manner. In this model the central place are occupied by nodes as the peaks of the network graph and the edges of the network graph show transactions of the traffic in which the peaks

communicate with each other along the data lines connecting them. This process separates our internal nodes, where switches or bridges are located with a closed curve from their input and output workstation nodes, as it is shown in figure 1. In the following number of internal, input and output nodes are marked by n , l and m respectively.

Consider the network is known in a time t and numbers of bits stored in the nodes are marked with $N(t)$. Now let us examine status of the network in time moment $(t+\Delta t)$. During time Δt if transfer speeds between nodes (v_{ij}) are known, the data $(N_i(t+\Delta t))$ stored in node i change as (1):

$$N_i(t+\Delta t) = N_i(t) + \Delta N_i^{\text{inter}} + \Delta N_i^{\text{input}} - \Delta N_i^{\text{output}}, \quad (1)$$

$$\text{where } \Delta N_i^{\text{inter}} = \sum_{\substack{j=1; \\ i \neq j;}}^n C_{ij}(v_{ij}) \cdot \Delta t, \quad \Delta N_i^{\text{input}} = \sum_{\substack{k=1; \\ i \neq k;}}^l C_{ik}(v_{ik}) \cdot \Delta t, \quad \Delta N_i^{\text{output}} = \sum_{i=1;}^m C_{ii}(v_{ii}) \cdot \Delta t.$$

In (1) $C_{ij}(v_{ij}) = c_{ij} \cdot v_{ij}$ is an elements of the communication matrix depending on v_{ij} and c_{ij} describes features of communication between node i and node j . The communication matrix is described detailed in session 5. After summarizing data changes in all nodes and supposed that node i has only one output, we get total data change (2) in our network:

$$\begin{bmatrix} N_1(t + \Delta t) \\ N_2(t + \Delta t) \\ \dots \\ N_n(t + \Delta t) \end{bmatrix} = \begin{bmatrix} N_1(t) \\ N_2(t) \\ \dots \\ N_n(t) \end{bmatrix} + \begin{bmatrix} 0 & c_{12} \cdot v_{12} \cdot \Delta t & \dots & c_{1n} \cdot v_{1n} \cdot \Delta t \\ c_{21} \cdot v_{21} \cdot \Delta t & 0 & \dots & c_{2n} \cdot v_{2n} \cdot \Delta t \\ \dots & \dots & 0 & \dots \\ c_{n1} \cdot v_{n1} \cdot \Delta t & c_{n2} \cdot v_{n2} \cdot \Delta t & \dots & 0 \end{bmatrix} \quad (2)$$

$$+ \begin{bmatrix} 0 & c_{12} \cdot v_{12} \cdot \Delta t & \dots & c_{1n} \cdot v_{1n} \cdot \Delta t \\ c_{21} \cdot v_{21} \cdot \Delta t & 0 & \dots & c_{2n} \cdot v_{2n} \cdot \Delta t \\ \dots & \dots & 0 & \dots \\ c_{k1} \cdot v_{k1} \cdot \Delta t & c_{k2} \cdot v_{k2} \cdot \Delta t & \dots & 0 \end{bmatrix}$$

$$- \begin{bmatrix} c_{11} \cdot v_{11} \cdot \Delta t & 0 & \dots & 0 \\ 0 & c_{22} \cdot v_{22} \cdot \Delta t & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & c_{nn} \cdot v_{nn} \cdot \Delta t \end{bmatrix}$$

or (2) in matrix form (3)

$$N(t+\Delta t) = N(t) + C_{\text{int}}(v_{\text{int}}) \cdot \Delta t + C_{\text{inp}}(v_{\text{inp}}) \cdot \Delta t - C_{\text{out}}(v_{\text{out}}) \cdot \Delta t \quad (3)$$

Forming (3) and do $\Delta t \rightarrow 0$, the result is shown by (4):

$$\lim_{\Delta t \rightarrow 0} \frac{N(t+\Delta t) - N(t)}{\Delta t} = N'(t) = C_{\text{int}}(v_{\text{inz}}) + C_{\text{inp}}(v_{\text{inp}}) - C_{\text{out}}(v_{\text{out}}) \quad (4)$$

(4) becomes more simply (5) if $v_{\text{int}} = v_{\text{inp}} = v_{\text{out}} = v$

$$N'(t) = v * (c_{\text{int}} + c_{\text{inp}} - c_{\text{out}}) \quad (5)$$

Complete calculation of (2) by Taylor series (6)

$$N(t+\Delta t) = N(t) + N'(t) * \Delta t + \frac{N''(t)}{2} * \Delta t^2 + \dots + \frac{N^{(n)}(t)}{n!} * \Delta t^n \quad (6)$$

Comparing (2) to (6) we say that

$$N''(t) * \Delta t^2 + \dots + N^{(n)}(t) * \Delta t^n = 0 \quad (7)$$

4 State Variables

In the network the transmission speed means how many bits can be transported in one seconds between two nodes. Transmission speed may not be equal in every part of the network, but from node j to node i the same value is supposed and marked with v_{ij} . Each node has some buffer for their messages. This capacity is measured by data density (8) that shows rate of recent data quantity in time and the maximal data quantity in node i .

$$x_i(t) = \frac{\text{number of recent data bits in buffer of node } i}{\text{maximal number of data bits in buffer of node } i} = \frac{N_i(t)}{N_{\text{imax}}} \quad (8)$$

In our model the data density of node i is a number without dimension between $0 \leq x \leq 1$. Traditionally, the numeration of data density shows the difference of outgoing and incoming bits in time t or in other words it shows the maximum number of data bits that can be transferred in the next time unit. Now data density is introduced as a rate of the length of messages stored in node i and the maximal message length. This property shows non-linearity, because no chance to receive more bits than the maximal size of the data buffer or node i can not receive any bit, if node j has no any one to send. In our network data density of node i is marked with x_i and data density of input node j that is located outside of the network is marked with s_j .

In our model $N(t)$ and $x(t)$ are vectors of n elements. From (8) $N_i(t) = N_{\text{imax}} * x_i(t)$, consequently N_{max} is a matrix of $n \times n$ elements. Elements of N_{max} represent the buffer size of nodes. Since no buffer between node i and node j , N_{max} is a diagonal matrix.

5 Communication Matrix

In (1) – (5) the communication matrix plays big role in our model. There are some functions implemented in $C_{ij}(v_{ij})$. Let us examine parts of this matrix. $C_{ij}(v_{ij})$ must contain at least the following:

- internal connections of the network
- environmental connections of the network
- size of internal buffers in nodes
- capabilities of nodes to send and receive data

The capability of node i shows when no data to send, no more room in its buffer as well as the probability of transmission in case of more output channels.

An element of the Communication matrix grants the connection when node j communicates with node i . Creating Communication matrix is done column by column. We go through each element of column j , and if a connection exists between nodes j and node i , that is node j works to node i , the communication function marked C_{ij} - is created, where ($i \neq j, 1 \leq i, j \leq n$). All features are necessary to be taken into consideration at the time of the forming of the communicational matrix by the aid of connection functions C_{ij} . The most important function is the model parameter of c_{ij} showing properties of the connection. Inside traffic regulations are also necessary to be taken into consideration at the time of the forming of the communicational matrix (i.e. data link mechanisms depending on the density of the traffic). In our model, the inner traffic regulations depend on density three functions of $c_{ij}(t)$, $S_j(t)$, $R_i(t)$ and $x_i(t)$, where $S_j(t)$ defines properties of sender node, $R_i(t)$ declares features of the receiver, while C_{ij} are defined by product of these four factors (9):

$$C_{ij} = c_{ij}(t) \cdot S_j(t) \cdot R_i(t) \cdot x_i(t) \quad (9)$$

Structure of $C_{ij}(t)$:

If there is the opportunity of the permanent communication between two nodes, and the node j works for node i , then $c_{ij} = e_{ij} \cdot p_{ij}$, where e_{ij} is the coefficient of effectiveness that shows the time rate of real data transfers and total working time, while p_{ij} is the probability of data transmission from node j to node i . If there is not a physical connection between these nodes, then $c_{ij} = 0$. The probability of data transmission, $p_{ij}(t)$ presents the distribution proportion of given routes belonging to a node with relative weighting, where $0 \leq p_{ij}(t) \leq 1$, if node j works for more than one node. The relative weightings of node j are shown in column j of the communication matrix and $\sum p_{ij}(t) = 1$ for each column.

$S_j(t)$ is an automatic internal self-regulation function (10) of transmitter node with values of 1 or 0. It shows whether node j has message to send or not. Connection is disable if data density of node j ($x_j(t)$) equals to 0, anyway 1.

$$S_j(t) = \begin{cases} 1 & , & x_j(t) > 0 \\ 0 & , & x_j(t) = 0 \end{cases} \quad (10)$$

$R_i(t)$ is another automatic internal self-regulation function (11) of the receiver node with values of 1 or 0. Connection is enable if data density of node i ($x_i(t)$) smaller, than 1, anyway 0. Value of 1 means that buffer of node i has been overloaded, so node i closes its communication port to direction of node j , therefore node i does not receive any message from node j .

$$R_i(t) = \begin{cases} 1 & , & x_i(t) < 1 \\ 0 & , & x_i(t) = 1 \end{cases} \quad (11)$$

Equation (9) transforms (3) into (12)

$$N(t+\Delta t) = N(t) + c_{ij}(t) \cdot S_j(t) \cdot R_i(t) \cdot x_j(t) \cdot v_{ij} \cdot \Delta t + c_{i,inp}(t) \cdot R_i(t) \cdot x_i(t) \cdot v_{i,inp} \cdot \Delta t - c_{outm,i}(t) \cdot S_i(t) \cdot v_{outm,i} \cdot \Delta t \quad (12)$$

In (12) we supposed that input node always can send data to and output nodes can receive all data from our network. In longer time period $x_j(t) \cdot v_{ij}$ shows real transmission speed. While $S_j(t) = 1$, i.e. $v_{ij} \cdot \Delta t \leq x_j(t) \cdot N_{imax}$, transmission works. Transmission stops if no more data to transfer. In other words, the average transfer speed for the total time period is $x_j(t) \cdot v_{ij}$.

6 Simulation

That takes too much simulation time if dataflow is simulated by bit by bit. We wanted to make a unit for our simulation. So in first step we measured frame lengths. We made data transfers in different time period (early morning, afternoon, midnight), with different data lengths (from some hundred bytes to hundred megabytes) and different transmission speed (from 24 Kbits/s up to 100Mbits/s). Measurements were made by the Wireshark Network Protocol Analyzer. After having collected more hundred thousands messages we analyzed incoming and outgoing frames. The result showed two significant figures among the values. One of them was characteristic of confirmation messages, while the other one was typical data messages. Distribution function of data measured is presented in Table 1. It shows the most frequented values.

Considering figures table 1 we chose 55 bytes as unit of frame length. Using this frame length an average data length is 26 units.

Table 1
 Number of pieces of the most frequented frame length

Length	Number of Pieces	Total Length
54	165 535	8 938 890
66	11 773	777 018
1 082	10 730	11 609 860
1 434	271 383	389 163 222
1 514	43 127	65 294 278

An existing network was simulated that can be seen in Figure xxx. Two files were transmitted through the real network. The first file of 89 MB was transported from Inp1 and the second one of 91 MB started from Inp2. First channel used TCP protocol, while second file was transported by FTP protocol. In our 100Mb/sec channels the calculated efficiency rate was about 72%, because the transportations took appr. 20 sec. Our model used the same files during the simulation.

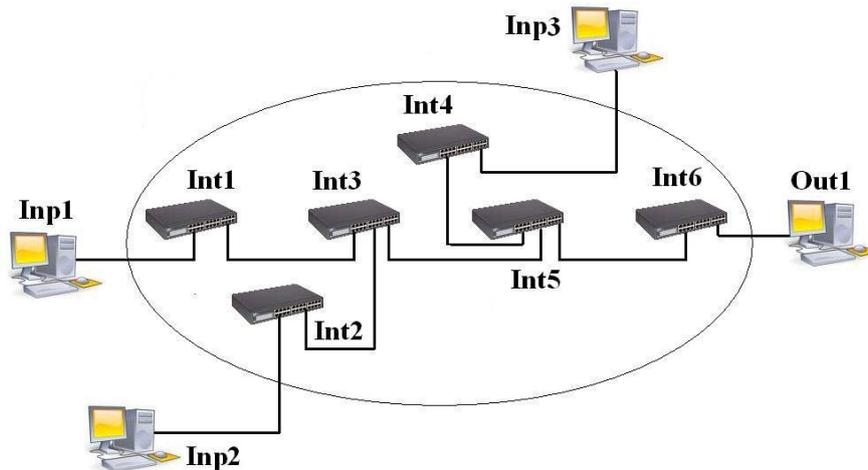


Figure 2
 Structure of the network measured and simulated

The second file was started in 8 seconds after the first one. The third input channel named Inp3 did not work. The throughput can be seen in Figure 3. Process time of the simulation is a bit less than the real communication. In the figure 3 we can see that the real communication used different channel speeds, while the transfer speed was the same in our simulation. During our simulation we supposed that speeds of transmission equal between nodes. This supposition was false as we could see it in Figure 3.

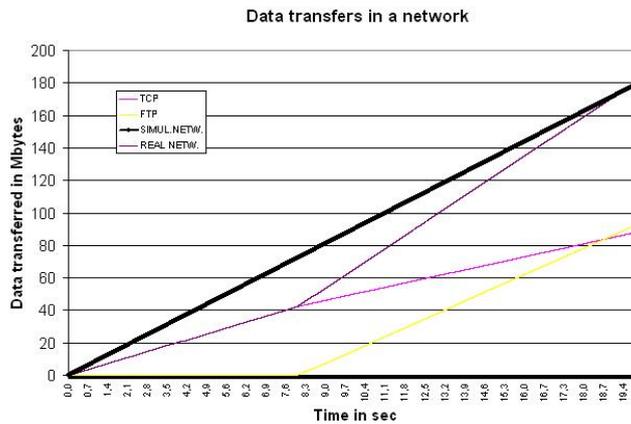


Figure 3

Internal and external elements of a network

In the simulation the data density of the internal nodes are also observable. Using different internal buffers in the network observed bottleneck effects can be tested. Figure 4 shows two simulation with different buffer sizes. On the left 5 MByte internal buffer was used, while in second case only 1 MByte buffer was allocated. On the left side inputs work linearly, while on the right side shows non-linear faces.

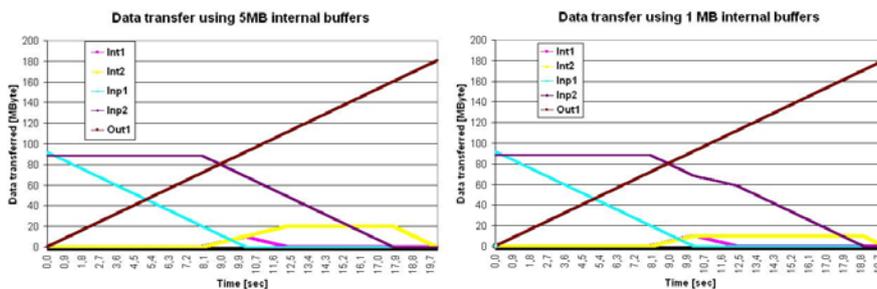


Figure 4

Data transfers using different internal buffers in the simulated network

Figure 4 helps us to find bottlenecks of the network. In the right picture Inp2 was blocked by node Int1, because there were no enough room for incoming frames.

Conclusion and future works

This paper looks for possibilities to describe dataflow model of large-scale computer networks. A model was presented that was applicable to simulation, planning and regulation of computer network’s traffic. At the time of the model's establishment, the partial differential equations were avoided in the mathematical model because of the specially chosen state variables. The nodes have honoured

roles in this non-linear model because storage capacities of the transmission medium are practically zero. Nodes either communicate to each other or not. In our model, the mean of the data density is the proportion of the size of data stored in the single node and the data quantity, which can be stored maximally. Our model examines change of data density occurred by data flow among the nodes in a region demarcated by close curve. Input and output data densities are regarded as known. At first sight, these processes are the inputs and outputs of the model. Effectively, these processes together form the actual inputs of the mathematical model. State variables present data densities arising in the internal nodes of the system. Our system applies a data traffic model involving n internal and m external nodes. To create the mathematical model the communication matrices defining the network has fundamental importance. Our model applies four communication matrices. During simulation we supposed that speed of transmission equals between nodes. It must be changed in the future. We have to work out a quick test to calculate possible congestion points and gives some criteria to stability of the system. Finally, a simply example was presented to demonstrate usage of the model and how this model is applicable to the simulation, planning or regulation of computer networks.

References

- [1] Y. Bhole, A. Popescu: Measurement and Analysis of HTTP Traffic. *Journal of Network and Systems Management*, Vol. 13, No. 4 (2005) pp. 357-371
- [2] J. Jiang, S. Papavassiliou: Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction. *Journal of Network and Systems Management*, Vol. 12, No. 1, (2004) pp. 51-72
- [3] S. Sidiroglou, G. Giovanidis, A. D. Keromytis: A Dynamic Mechanism for Recovering from Buffer Overflow Attacks, J. Zhou et al. (Eds.): *ISC 2005*, LNCS 3650, pp. 1-15, 2005
- [4] D. Haifeng, X. Yang, L. Lingyun: An Effective Network Congestion Control Method for Multilayer Network, *Journal Of Electronics (China)*, Vol.25, No.4, July 2008
- [5] John C.-I. Change, Marvin A. Sirbu: Pricing Multicast Communication: A Cost-Based Approach, *Telecommunication Systems* 17:3, 281-297, 2001
- [6] Paxson, V., Floyd, S.: Wide Area Traffic: The Failure of Poisson Modelling, *IEEE/ACM Transactions on Networking* 3(3), pp. 226-244 (1995)
- [7] A. M. Galkin, O. A. Simonina, G. G. Yanovsky: Multiservice IP Network QoS Parameters Estimation in Presence of Self-similar Traffic, *NEW2AN 2006*, LNCS 4003, pp. 235-245, 2006

- [8] J. Sommers, H. Kim, P. Barford: Harpoon: A Flow Level Traffic Generator for Router and Network Tests, SIGMETRICS/Performance '04, 2004, pp. 392-393
- [9] S. Avallone, A. Pescape, Giorgio Ventre: Analysis and Experimentation of Internet Traffic Generator, NEW2AN 2004, pp. 1-6
- [10] M. A. Mansour, L. Scriali: Hölder Continuity of Solutions to Elastic Traffic Network Models, J. Glob. Optim. (2008) 40:175-184
- [11] H. F. Du, Y. Xiao, D. Y. Xu: Nonlinear Stability Analysis for Active Queue Management, Proceedings of the IEEE International Symposium on Communications and Information Technology (ISCIT'2005), 2005, Vol. 2, 1314-1317
- [12] T. Peter, Analysis of Large Scale Transportation Networks, MMA Conference, Symposium, Budapest (2007)
- [13] Kihl, M., Robertsson, A., Wittenmark, B.: Performance Modelling and Control of Server Systems Using Non-Linear Control Theory, Proceedings of the 18th International Teletraffic Congress (2003)
- [14] Kihl, M., Robertsson, A., Wittenmark, B.: Control Theoretic Modelling and Design of Admission Control Mechanisms for Server Systems, Proceedings of IFIP Networking (2004)
- [15] Holot, C., Misra, V., Towsley, D., Gong, W. B.: A Control Theoretic Analysis of RED, Proceedings of IEEE INFOCOM 01, Anchorage, Alaska, 2001
- [16] Low, S. H., Paganini, F., Wang, J., Adlakha, S., Doyle, J. C.: Dynamics of TCP/RED and a Scalable Control, Proceedings of IEEE INFOCOM 02, New York, USA, 2002
- [17] Holot, C. V., Chait, Y.: Nonlinear Stability Analysis for a Class of TCP/AQM Networks, IEEE Conference on Decision and Control, 2001, Orlando, pp. 2309-2314
- [18] Aweya, J., Ouellette, M., Montuon, D. Y.: A Linear System Analysis of RED, Computer Communications 25 (2002) 1736-1750
- [19] Ohsaki, H., Murata, M.: Steady State Analysis of RED Gateway: Stability, Transient Behavior and Parameters Setting, IEICE Transactions on Communications E85-B (1) (2002) 107-117
- [20] Firoiu, V., Borden, M.: A Study of Active Queue Management for Congestion Control, Proceedings of IEEE INFOCOM 00, Tel Aviv, Israel, March 2000
- [21] Ranjan, P., Abed, E. H., La, R. J.: Nonlinear Instabilities in TCP-RED, Proceedings of IEEE INFOCOM 02, New York, USA, 2002