

An Improvement on TCP Congestion Control Protocol for Next Generation Networks

Nazbanoo Farzaneh, Reza Monsefi, Mohammad Hossein Yaghmaee, Amir Hossein Mohajerzadeh

Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Farzaneh@stu-mail.um.ac.ir, {monsefi, yaghmae}@ferdowsi.um.ac.ir, ah.mohajerzadeh@stu-mail.um.ac.ir

Abstract—Recent years have shown an increased interest towards multimedia rich applications. Multimedia content ranges from text or simple images to audio data or video data. A Next-Generation Network (NGN) is a packet-based network that handles multiple types of traffic (such as voice, data, and multimedia). Multimedia applications include large amount of data, high throughput, and multiple flows which need diverse QoS characteristics. Nowadays, many applications use TCP as their transport protocol. As the network evolves, many improvements and modifications are proposed to make TCP meet various requirements. TCP, although powerful and effective, is not sufficient to satisfy real time applications because TCP emphasizes more on reliability than timeliness. In this paper a new Improvement on TCP (ITCP) protocol is presented. Source rate is regulated based on the feedback which is received from intermediate nodes. Furthermore, in order to satisfy the requirements of different multimedia applications, a weighted random early detection (WRED) mechanism is used. Loss probability is calculated based on the packets priority in a router and the result is sent to source then it regulates its rate based on the packet loss probability values. The proposed protocol was simulated in order to evaluate its performance. Simulation results indicate that the ITCP improves overall system throughput and reduces delay and packet loss.

Keywords—Active queue management (AQM); Next Generation Network (NGN); Quality of Service (QoS); Transmission Congestion Protocol; Transport protocol.

I. INTRODUCTION

According to ITU-T recommendation the definition of Next Generation Networks (NGN) is a packet based network able to provide telecommunication services to users and able to make use of multiple broadband, QoS enabled transport technologies and in which service related functions are independent of the underlying transport related technologies [1]. NGN implies a consolidation of several transport networks each built for a different service into one core transport network (often based on IP and Ethernet) [1]. New NGN platforms must provide voice, data and multimedia communications over a converged IP infrastructure [2].

Congestion occurs when the demand for resources exceeds the capacity of the link. The constraints - unpredictable supply and demand and the desire for efficient distributed operation - necessarily lead to feedback control as the preferred approach, where traffic sources dynamically adapt their rates to congestion in their paths. On the Internet,

this is performed by the Transmission Control Protocol (TCP) in source and destination computers involved in data transfers [3]. In TCP loss of a segment indicates network congestion [4]. Existing transport protocols have limitations when they are used in new application domains and for new network technologies (such as NGNs). For example, multimedia applications need congestion control but not necessarily ordered and reliable delivery - this is not offered by TCP. Multimedia traffic requires congestion control, but it requires a new approach, which would be more efficient. The emerging approach for the multimedia traffic is adaptability TCP is not suitable for real time applications because it increases end-to-end delay and delay variations [5].

Active Queue Management (AQM) schemes perform special operations in the router to achieve better performance [6]. In multi-class queuing, packets from different upstream routers that belong to the same class are entered into the same queue at the router. Different applications have their own QoS requirements, such as delay, delay variance and packet loss ratio tolerance, and thus can be classified into different classes. Weighted Random Early Detection (WRED) algorithm that was introduced in [7] is an extension to RED that drops packets based on their weight. Different queues may have different thresholds based on their priority so the higher priority traffic enters the queue with a higher probability than the lower priority. In this paper we proposed ITCP (Improvement on TCP), which is a new extension of TCP that can be implemented using the extra 40 bytes of TCP option field. The designing goal of this protocol is to work within the existing TCP specifications, but it improves TCP protocol for a network with different traffic types that need various demands. So that it would provide good performance over networks which have heterogeneous traffics (such as NGNs).

The rest of this paper is organized as follows. In section II we present a brief review of prior related works on improving TCP. The proposed protocol is explained in section III. In section IV using computer simulation, the performance of the proposed protocol is evaluated. Finally, section V concludes the paper.

II. RELATED WORKS

In recent years, several solutions have been proposed to improve TCP performance. Two variables, congestion window (*cwnd*) and slow start threshold (*ssthresh*) are used to throttle the TCP input rate in order to match the network

available bandwidth. TCP Tahoe is the first version that includes the congestion avoidance mechanism.

Tahoe refers to the TCP congestion control algorithm which was suggested by Van Jacobson [8]. Tahoe uses fast retransmit algorithm, which uses the retransmission strategy without waiting for retransmission timeout. Upon receiving three duplicate acknowledgements, Tahoe immediately retransmits the unacknowledged segment.

TCP Reno [9] evolves from Tahoe and includes the additional algorithm, fast recovery, to reach the available bandwidth more quickly than Tahoe after recovering packet losses by fast retransmit. TCP Reno implements the TCP's AIMD mechanism of increasing the congestion window W by one segment per round-trip time for each received ACK and halving the congestion window for each loss event per round-trip time. When the link bandwidth does not change, TCP Reno periodically repeats the window increase and decreases TCP Reno's congestion window in terms of packet loss. TCP NewReno[10] is a modification of TCP Reno. TCP NewReno adds the bandwidth-delay product estimation algorithm to Reno to avoid packet losses during the slow-start phase. TCP NewReno can detect multiple packet losses. NewReno changes the fast recovery algorithm in Reno. It does not exit the fast recovery phase until all the unacknowledged segments at the time of fast recovery are acknowledged. The remaining three phases (slow start, congestion avoidance, and fast retransmit) are similar to TCP Reno. Vegas [11] TCP was the first attempt to depart from the loss-driven paradigm of the TCP by introducing a mechanism of congestion detection before packet losses. In particular, Vegas TCP computes the difference between the *actual input rate* ($cwnd/RTT$) and the *expected rate* ($cwnd/RTT_{min}$), where RTT is the Round Trip Time and RTT_{min} is the minimum measured round trip time, to infer the network congestion. In particular, if the difference is smaller than a threshold α then the $cwnd$ is additively increased, whereas if the difference is greater than another threshold β then the $cwnd$ is Additively Decreased; finally, if the difference is smaller than β and greater than α , then the $cwnd$ is kept constant.

III. THE PROPOSED ALGORITHM

In this paper, a method for congestion control is presented. Next generation networks, encounter with different types of traffic flows with different QoS requirements. As a result, the network behavior should be different for each class of traffic flow. In this paper, three types of traffic classes are considered, 1) Assured Forwarding (AF) class is related to delay and loss intolerant traffic, 2) Expedited Forwarding (EF) class is loss intolerant data traffic and 3) the Best Effort (BE) class is belonging to loss and delay tolerant traffic.

In WRED, loss rate in each group varies with other groups. EF traffic class is delay and loss intolerant; therefore it must experience less packet loss than other traffic classes. AF class packets have the middle priority. This packet group is loss intolerant so it experiences a little loss. AF class packets are not delay intolerant, hence in case of loss, the retransmission mechanism is used to send lost packets by the

source again. BE class has low priority, so in congestion condition it is more prone of being dropped. Higher level of congestion caused other kinds of traffic to gradually be dropped. In sever congestion AF class traffic packets are also dropped. Finally, in conditions of intense congestion, loss includes EF class packet as well. It is obvious that the percentage of packet loss is different for each class for various congestion levels. The proposed congestion control protocol consists of various units. Fig. 1 illustrates the operation model of congestion control protocol.

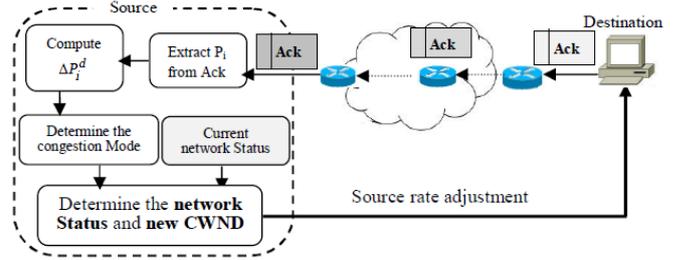


Figure 1. The Operation model of congestion control protocol

In ITCP, source uses loss probability values to determine the congestion rate. As soon as a packet enters a node, loss probability is calculated, depending on packet type (class, EF or AF or BE). As mentioned before since loss probability values are calculated using *packet type*, it can be a good representative to show network congestion. Higher packet loss probability shows higher network congestion level.

Each node puts three values in the returned ACK packets from destination to the source. These three values are related to loss probability for three classes (EF, AF and BE classes). To prevent ACK packets enlargement, only maximum loss values among nodes are placed in ACK packet. Each node puts its calculated loss probability when it is larger than current value in ACK packet. Thus by receiving each ACK packet, the source node can be informed of the maximum amount of loss packet during path. ΔP_{EF}^d , ΔP_{AF}^d , ΔP_{BE}^d are maximum loss probability of EF, AF and BE class respectively. By receiving each ACK packet, source calculates amount of changes on loss packet based on the difference function as shown in equation (1).

$$\Delta P_i^d = Cur(P_i^d) - Old(P_i^d) \quad (1)$$

$Old(P_i^d)$ is the maximum loss probability in the previous ACK packet and $Cur(P_i^d)$ is the maximum loss probability in the current ACK packet. Based on ΔP_i^d values, network congestion level can be found. Table I describes the meaning of various ΔP_i^d values.

TABLE I. DESCRIBE OF DIFFERENT VALUE OF ΔP_i^d

$\Delta P_i^d = 0$	Loss rate for packet type i has not been changed.
$\Delta P_i^d < 0$	Loss rate for packet type i has been decreased in

	network. ΔP_i^d values, shows the decrement rate.
$\Delta P_i^d > 0$	Loss rate for packet type i has been increased in network. ΔP_i^d values, shows the incremented rate

Based on changes that emerged in ΔP_i^d values, the following *modes* –which are shown in table II- can be defined. These modes help to identify the network current level of congestion.

TABLE II. POSSIBLE NETWORK MODE OF CHANGES ON THE PROBABILITY OF PACKET LOSS

Mode	Description
I	$\Delta P_{EF}^d \leq 0$ and $\Delta P_{AF}^d \leq 0$ and $\Delta P_{BE}^d \leq 0$
II	$\Delta P_{EF}^d \leq 0$ and $\Delta P_{AF}^d \leq 0$ and $\Delta P_{BE}^d > 0$ OR
	$\Delta P_{EF}^d > 0$ and $\Delta P_{AF}^d \leq 0$ and $\Delta P_{BE}^d \leq 0$ OR
III	$\Delta P_{EF}^d \leq 0$ and $\Delta P_{AF}^d > 0$ and $\Delta P_{BE}^d > 0$ OR
	$\Delta P_{EF}^d > 0$ and $\Delta P_{AF}^d > 0$ and $\Delta P_{BE}^d \leq 0$ OR
IV	$\Delta P_{EF}^d > 0$ and $\Delta P_{AF}^d > 0$ and $\Delta P_{BE}^d > 0$

In mode I, packet loss rate is unchanged or reduced in all types of traffic. Mode II shows increased packet loss in one class. In mode III, loss is constant or it is low in one class and packet loss has been increased in the other two classes. Intense congestion occurs in IV mode, this means, loss increase is observed in all three types of traffic packets. In proposed protocol three *statuses* are defined for the network as table III.

TABLE III. NETWORK CONGESTION STATUS

Status	Description
Normal	In this case there is no congestion in the network.
Low congestion	There is the risk of emerge congestion or a little congestion has been emerged. Increased loss rate in the network, shows congestion has been started.
High congestion	The amount of loss probability has been increased in all three types of traffic classes. Thus, sever congestion has been occurred.

Based on three defined statuses in table III and discussed modes in table II, a Graph (*D*), that is shown in fig. 2, can be drawn.

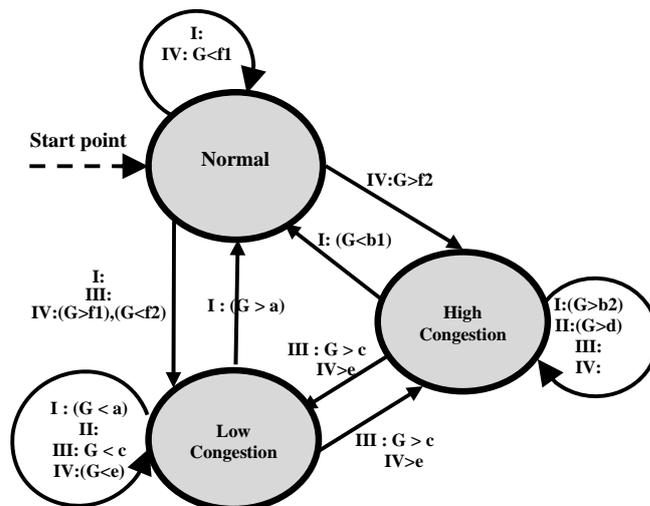


Figure 2. Network status graph

The State diagram is used to give an abstract description of the behavior of a system. In graph *D*, nodes represent status and directed edges represent transitions. “Normal status” being the start status. G is a convex combination of ΔP_i^d as follows:-

$$G = \mu_1(\Delta P_{EF}^d) + \mu_2(\Delta P_{AF}^d) + \mu_3(\Delta P_{BE}^d) \quad (2)$$

$$\mu_1 + \mu_2 + \mu_3 = 1, \quad \mu_1 > \mu_2 > \mu_3$$

The value of G is in the $[-1, 1]$ range and represent the overall loss rate in packets. Each edge is labeled by the phrase that cause the transition represented by the edge. The phrase **S: $G < L$** means that network state has been changed whenever the congestion mode is $S = \{I, II, III, IV\}$ and G values is less than $L = \{a, b1, b2, c, d, e, f1, f2\}$ coefficient.

Based on the current *mode* and the current *status* of the network, next network state is determined.

a) *Determination of the congestion window size*

Source adjusts its congestion window size in order to set its own sending rate. Congestion window size will be configured according to network mode.

- **In Normal state:** congestion window size (*cwnd*) has been added *linearly*. Cwnd is changed depends on the G value.
- **In Low congestion state:**
 - ✓ If the current mode is I or II, then according to slow start mode in TCP, Congestion window size is doubled in every RTT.
 - ✓ If the current mode is III or IV, then congestion window size is decreased linearly. The amount of decrement is depended on G value.
- **In High congestion state:**
 - ✓ If the current mode is III, then congestion window size is decreased in non-linear form. The decrement amount depends on the G value.

- ✓ If the current mode is II, congestion window size will be halved, similar to the time when receiving three duplicated ACK in TCP.

We have modified the *cwnd* update function to implement the proposed protocol. The modifications are shown in fig. 3.

Algorithm : *cwnd* update Function

```

// snd_mss = Maximum Segment Size for outgoing segments
// a = -1 ; lower bound of G
// b = 1 ; upper bound of G
// c = 1 ;
if (tcp_flavor == TcpC_Flavor_Multi_ClassTCP) )
{
    if (Current_state == 'Normal')
    {
        newG = ((-G)-((a+b)/2))*((alpha - c) / (b-a))+ ((c+alpha)/2);
        cwnd = cwnd + (newG * snd_mss);
    }
    else if (Current_state == 'Low Congestion')
    {
        if (mode == I || mode == II)
        {
            cwnd = cwnd + (1 * snd_mss);
        }
        else if (mode == III || mode == IV)
        {
            newG = (G-((a+b)/2))*((beta-c) / (b-a))+ ((c+beta)/2);
            temp = cwnd - (newG * snd_mss);
            if (temp < snd_mss)
                cwnd=snd_mss ;
            else
                cwnd = cwnd - (newG * snd_mss);
        }
    }
    else if (Current_state == 'High Congestion')
    {
        if (mode == III)
        {
            newG = (G-((a+b)/2))*((gama-c) / (b-a))+ ((c+gama)/2);
            temp = cwnd / (newG * snd_mss);
            if (temp < snd_mss)
                cwnd=snd_mss ;
            else
                cwnd = cwnd / (newG * snd_mss);
        }
        else if (mode == IV)
        {
            temp = cwnd / 2;
            if (temp < snd_mss)
                cwnd=snd_mss ;
            else
                cwnd = cwnd / 2;
        }
    }
}
    
```

Figure 3. Pseudo-code of proposed protocol

IV. SIMULATION RESULTS

In this section, we present simulation results using Opnet Modeler simulator [11]. The Opnet simulator is a discrete-event network simulation development and used for the analysis of the computer networks. The following parameters are evaluated to investigate end-to-end congestion control algorithm performance.

- Total packet loss
- Throughput: the number of received packets per simulation time
- Average end to end delay

Fig. 4 shows the topology that was used in our simulation and the simulation parameters values were given in table IV.

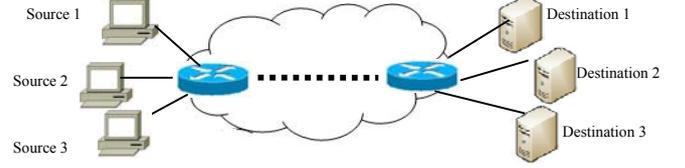


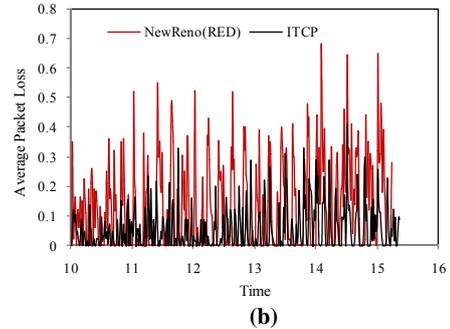
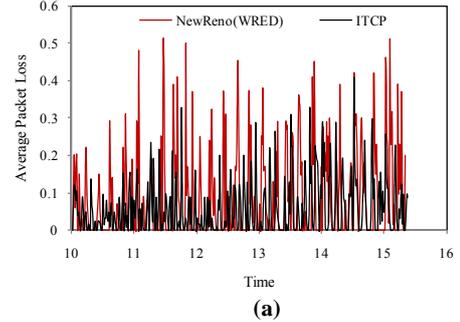
Figure 4. The topology that is used in simulation

TABLE IV. SIMULATION PARAMETERS VALUES

Parameter	Value	Parameter	Value
a	-0.6	c	0.2
b1	-0.8	b2	-0.2
d	0	e	0.3
f1	0.2	f2	0.4
alpha	3	betha	5
gama	1.5		

In many scenarios in order to compare transport section of ITCP and NewReno we have used WRED as AQM mechanism for both of them. But in other scenarios we use RED and FIFO as AQM besides NewReno.

In fig. 5.a, fig. 5.b and fig. 5.c shows the number of the lost packets versus time against different AQM algorithm used for NewReno. In comparison packet loss rate in all three figures 5.a, 5.b and 5.c it's seen that the amount of packet loss in ITCP protocol is much less than NewReno protocol because AQM based congestion control protocol, set source sending rate better and prevent congestion and loss using feedback from the network. Although the NewReno protocol increases and decreases source sending rate base on some parameters, but as it's shown in fig. 5.a, fig. 5.b and fig. 5.c, ITCP protocol has been more successful to choose the source optimum sending rate.



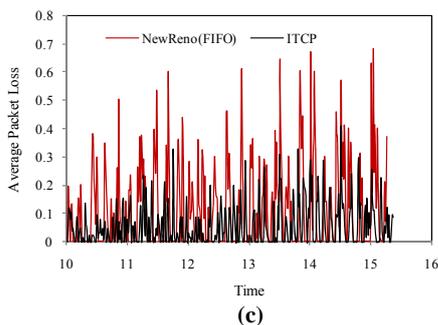


Figure 5. loss percentage average in routers of the proposed protocol versus a) NewReno with WRED b) NewReno with RED c) NewReno with FIFO

Fig. 6.a, fig. 6.b and fig. 6.c, present the end-to-end delay average versus time against different AQM algorithm which are used for NewReno. Using effective AQM mechanism which calculates the loss probability based on packet priority besides on AQM-based congestion control protocol that is responsible to determine the source rate, leads to stable queue length not to add suddenly and queue length always sit in appropriate range. So average delay decreases after a while because the more queue length is, the more delay in packets service and end-to-end delay.

Fig. 7 shows throughput versus time. Throughput is the number of received packets at destination per time unit. Two main goals of congestion control protocols are to reduce the packet loss and to increase the network efficiency. To reduce packet loss when congestion occurs, source should reduce its rate. Sending rate reducing should not reduce the network efficiency. As shown in fig. 5 and fig. 7, ITCP protocol has less packet loss rate than NewReno protocol and has increased the network throughput significantly as well.

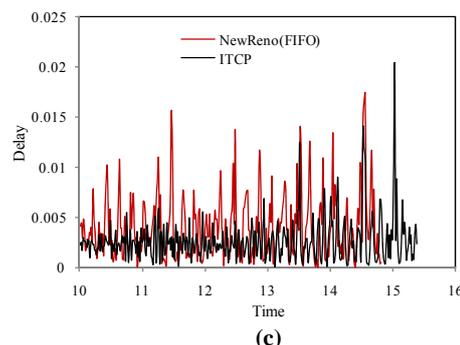
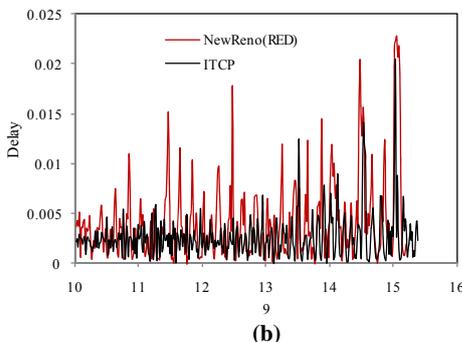
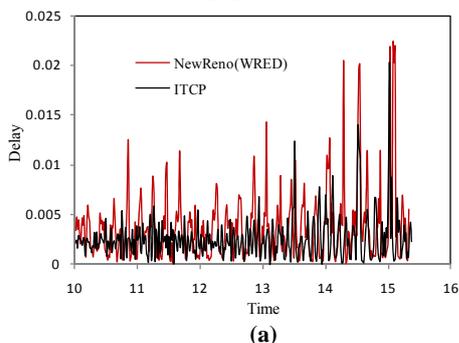


Figure 6. end to end delay of the proposed protocol versus a) NewReno with WRED b) NewReno with RED c) NewReno with FIFO

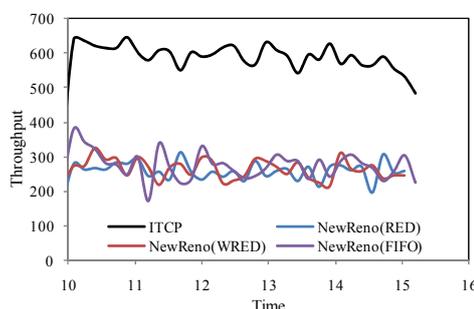


Figure 7. Network Throughput

V. CONCLUSION

In this paper we have presented a new extension on TCP that is better suited to NGNs than TCP New Reno or Tahoe. In the proposed protocol which is called ITCP (Improvement on TCP), Intermediate nodes inform source nodes about congestion occurrence and its level, by determining loss probability values. Source regulates its congestion window (*cwnd*) according to loss probability values. ITCP protocol can distinct between different kinds of data flows. ITCP can support three kind of traffic source. We used a weighted AQM which accept the packets according to their priority. In ITCP due to feedback received from the intermediate nodes, packet loss reason is detectable well. Congestion prediction possibility decreases loss packet very much and increase network throughput. Number of dropped packets in ITCP method is much less than TCP due to early detection of congestion.

ACKNOWLEDGEMENT

The research team wishes to express their gratitude to Iran Telecom Research Center (ITRC) for their valuable help support. This research has been sponsored by Iran Telecom Research Center (ITRC) under contract T/500/9966.

REFERENCES

- [1] ITU-T Recommendation Y.2011, "General principle and general reference model for Next Generation Networks", 2004.
- [2] W. Wang, M. Palaniswami, S.H. Low, "Application-Oriented Flow Control: Fundamentals, Algorithms and Fairness", IEEE/ACM

TRANSACTIONS ON NETWORKING, VOL. 14, NO. 6,
DECEMBER 2006.

- [3] D. X. Wei, C. Jin, S. H. Low, S. Hegde, "Fast TCP", IEEE/ACM Transactions on Networking, 14(6):1246-1259, Dec 2006.
- [4] M. Allman, V. Paxson, et al., "TCP Congestion Control", RFC 2581, April 1999.
- [5] V. Firoiu, and M. Borden, "A Study of Active Queue Management for Congestion Control," IEEE INFOCOM, 2000.
- [6] William Stallings, "Data & Computer Communication", Sixth edition, Prentice Hall International
- [7] W. Stevens. "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery", RFC 2001, January 1997.
- [8] V. Jacobson "Modified TCP Congestion Control and Avoidance Algorithms". Technical Report 30, Apr 1990.
- [9] S. Floyd, T. Henderson "The New-Reno Modification to TCP's Fast Recovery Algorithm" RFC 2582, Apr 1999.
- [10] L.S. Brakmo, L.L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communication, vol. 13[1995],(1465- 1490).
- [11] <http://www.opnet.com>