VEGAS: Better Performance Than Other TCP Congestion Control Algorithms on MANETs

M.Jehan *jehanjoy@yahoo.co.in*
Associate Prof., Department Of Computer Science,
D.J.Academy for Managerial Excellence,
Bharathiar University,
Coimbatore, 641 032, India.

Dr.G.Radhamani *radhamanig@hotmail.comProf.*
& Director, Department of Computer Science,
Dr.G.R.Damodaran College of Science,
Bharathiar University,
Coimbatore, 641 014, India.

T.Kalakumari *kalakumari_t@yahoo.co.in*
Assistant Prof., Department of Commerce & IT,
Sri Krishna Arts and Science College,
Bharathiar University,
Coimbatore, 641 008, India.

Abstract

The wireless communication TCP/IP protocol is an important role in developing communication systems and which provides better and reliable communication capabilities in almost all kinds of networking environment. The wireless networking technology and the new kind of requirements in communication systems need some extensions to the original design of TCP for on coming technology development. In this paper we have analyzed six TCP Congestion Control Algorithms and their performance on Mobile Ad-hoc Networks (MANET). More specifically, we describe the performance behavior of BIC, Cubic, TCP Compound, Vegas, Reno and Westwood congestion control algorithms. The evaluation is simulated through Network Simulator (NS2) and the performance of these congestion control algorithms is analyzed with suitable metrics.

Keywords: TCP Congestion Control Algorithms, MANET, BIC, Cubic, Compound, Vegas, Reno, Westwood.

1. INTRODUCTION

Ad-hoc networks are self-organizing wireless networks, in which all end nodes act as routers. This network improves the efficiency, range of fixed or mobile internet access and enables totally with new applications. A Mobile Ad hoc Networks (MANET) consists of a set of mobile hosts within communication range and exchange the data among themselves without using any pre-existing infrastructure. MANET nodes are typically distinguished by their limited power, processing and memory resources as well as high degree of mobility. In such networks, the wireless mobile nodes may dynamically enter the network and leave the network. Due to the limited transmission range of wireless network nodes, multiple hops are usually needed for a node to exchange information with any other node in the network.

MANET has potential use in a wide variety of disparate situations. Such situations include moving battlefield communications to disposable sensors which are dropped from high altitude and dispersed on the ground for hazardous materials detection. Civilian applications include simple scenarios such as people at a conference in a hotel with their laptops comprise a temporary

MANET to more complicated scenarios like highly mobile vehicles on the highway which form an ad-hoc network in order to provide vehicular traffic management.

In this paper, we have evaluated the Control Window (cwnd), Round Trip Delay Time (rtt) and the Throughput using the six algorithms on the performance of TCP in the wireless communication.

2. BACKGROUND WORK

2.1 Transmission Control Protocol and Congestion Control

TCP is one of the core protocols of the internet protocol family. TCP operates at a higher level, concerned only with the two end systems. In particular, TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on other computer. Among its other management tasks, TCP controls segment size, flow control, the rate at which data is exchanged, and the network traffic congestion.

For the reliable packets delivery, TCP can support the mechanisms of flow and congestion control. Due to the unconstrained movement of the mobile nodes, TCP is unable to notice network congestion or link down to activate related controls on the MANET [2]. The standard congestion control mechanism of the TCP is not able to handle the special properties of a shared wireless multi-hop channel well. In particular, the frequent changes of the network topology and the shared nature of the wireless channel pose significant challenges [4].

TCP provides reliable end-to-end delivery of data over wired networks, several recent studies have indicated that TCP performance degrades significantly in MANET [6] [7] [8]. In [13], TCP-F is proposed to overcome the TCP false reaction towards route failures in MANETs. In [14] the simulation shows that the route change results in link disconnections, which reduces TCP throughput.

Vegas 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 [9]. Westwood TCP is a new congestion control algorithm that is based on end-to-end bandwidth estimate [15].

Using TCP more computers are interconnected to increase data transaction between users rapidly. The MIMD and PIPD protocols developed and provides better throughput for the wireless networks [12], [16] and [17].

So, this study on the existing TCP congestion control algorithms and its performance on MANET will be very useful to design new algorithms exclusively for mobile wireless communication scenarios.

2.2 The Congestion Control Algorithms Under Evaluation

In the past years, after the invention of TCP, there is numerous congestion control algorithms discovered for different purposes. Each of them has unique characteristics [3]. But in this work we have selected six congestion control algorithms only because of these are the default algorithms in most of the open source and commercial operating systems.

2.2.1 Binary Increase Congestion Control (BIC)

BIC-TCP (Binary Increase Control-TCP) incorporated binary search increase in the protocol. Binary search increase provides reliable feedback on any network congestion and lost packets, allowing BIC-TCP to aggressively increase its transmission speed toward the maximum allowed by the high-speed network.

2.2.2 CUBIC

Binary Increase congestion Control for TCP v2.0 is called as CUBIC and it is a default TCP algorithm in Linux. The protocol modifies the linear window growth function of existing TCP standards to be a CUBIC function in order to improve the scalability of TCP over fast and long distance networks. CUBIC is a less aggressive and more systematic derivative of BIC, in which the window is a CUBIC function of time since the last congestion event, with the inflection point set to the window prior to the event.

2.2.3 COMPOUND TCP (C-TCP)

Compound TCP is a Microsoft implementation of TCP which maintains two different congestion windows simultaneously, with the goal of achieving good performance on, long fat networks (LFNs) while not impairing fairness. It has been widely deployed with Microsoft Windows Vista and Windows Server 2008 and has been ported to older Microsoft Windows versions as well as Linux.

2.2.4 TCP VEGAS

Until the mid 1990s, all TCPs set timeouts and measured round-trip delays were based upon only the last transmitted packet in the transmit buffer. In TCP Vegas, timeouts were set and round-trip delays were measured for every packet in the transmit buffer. In addition, TCP Vegas uses additive increases in the congestion window.

2.2.5 TCP NEW RENO (Reno)

TCP New Reno is the most commonly implemented algorithm; SACK support is very common and is an extension to Reno/New Reno. Starting with 2.6.8 the Linux kernel switched the default implementation from Reno to BIC.

2.2.6 TCP WESTWOOD+ (westwood)

TCP Westwood+ is based on end-to-end bandwidth estimation to set congestion window and slow start threshold after a congestion episode, that is, after three duplicate acknowledgments or a timeout. The bandwidth is estimated by properly low-pass filtering the rate of returning acknowledgment packets. The rationale of this strategy is simple: in contrast with TCP Reno, which blindly halves the congestion window after three duplicate ACKs, TCP Westwood+ adaptively sets a slow start threshold and a congestion window which takes into account the bandwidth used at the time congestion is experienced. The Algorithm significantly increases throughput over wireless links and fairness compared to TCP Reno/New Reno in wired networks. A detailed description of TCP Westwood (TCPW) is shown in [15].

3. THE SIMULATION

The proposed simulations has been successfully implemented and evaluated using NS-2 simulator on a computer with Intel Core 2 Duo CPU (T6400 processor @ 2.00 GHz) 2 GB of RAM. A random wireless mobile ad hoc network topology was used for these experiments.

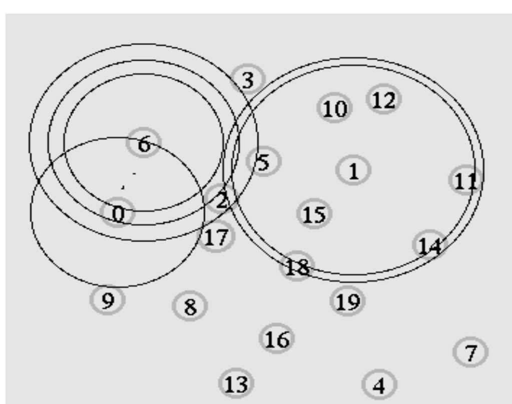


FIGURE 1: The MANET Scenario.

Some of the important parameters of the Ad-hoc Network simulation are:

Number of Nodes	20
Number of Sending Nodes	1
Topography	x=500 y=500
Mobility	0 or 20m/s
Mobility Start Time	20th Sec
Routing Protocol	AODV
Mac Type	802.11
Queue	DropTail / PriQueue
Queue Size	50
The Traffic Application	FTP
TCP Packet Size	1448
TCP Initial Window Size	30000

As far as the different parameters of congestion algorithm are concerned, all default parameters of TCP-Linux have been used in all our simulations. For simplicity and clarity of outputs, we used only one TCP flow during evaluating the algorithms.

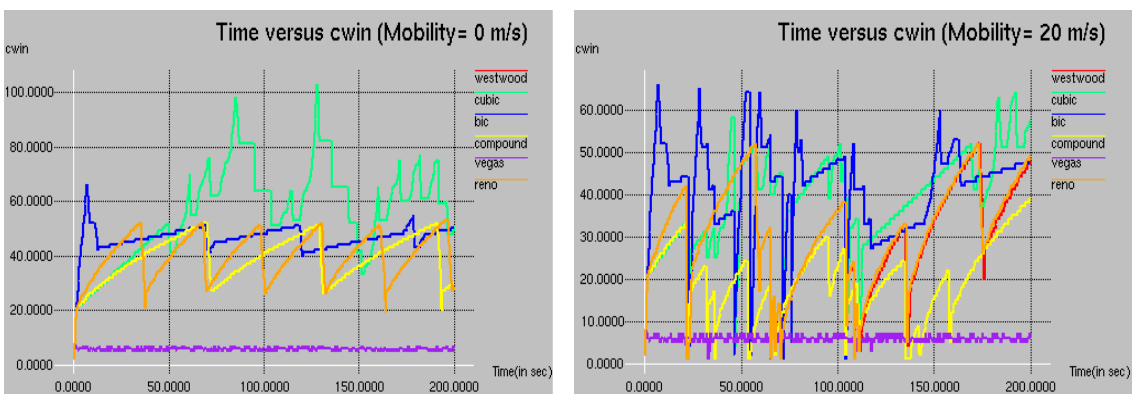
4. RESULT AND DISCUSSION

4.1 Simulation Results in MANETs

In this section, we carried out the simulation results of congestion control window, Round Trip Delay Time and Throughput in the Wireless Ad hoc Network. This simulation has been run for 200 seconds.

4.1.1 Control Window in MANETs

In the experimental network, we have used to perform evaluation of congestion control window comparison between above mentioned six algorithms as shown in the simulation result.



(a) (b)

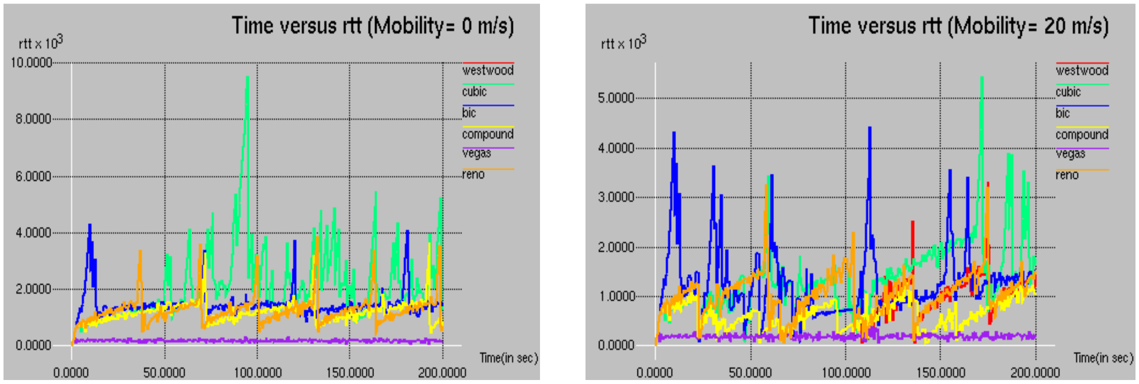
FIGURE 2: The cwnd Dynamics on Ad hoc Network – With and Without Mobility.

Figure 2 shows the, congestion control window increasing and decreasing in the all algorithms without any sequence except TCP Vegas.

In the both Figure 2 (a) and 2 (b), the exponential window size increase, linear increase and drop-off occurs irregularly during the simulation. In this Mobile Ad-hoc Networks scenario the TCP Vegas giving good result than other algorithms from starting to end of the evaluation.

4.1.2 Round Trip Delay Time in MANETs

The Round Trip Delay Time estimation of congestion control algorithms on wireless ad hoc network as shown in the Figure 3 (a) and 3 (b). This simulation result shows the TCP Vegas performance is better than other algorithms from the group of algorithms. As per the simulation setup, all nodes started to move randomly after 20 m/s.



(a) (b)

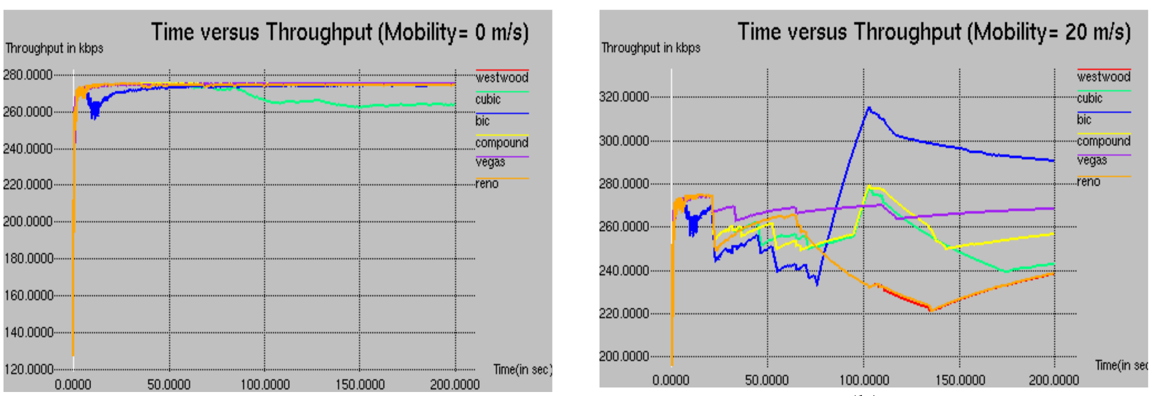
FIGURE 3: The rtt Dynamics on Ad hoc Network – With and Without Mobility.

As per the simulation result, the Figure 3 (a) and 3 (b) shows the TCP Vegas giving better result than other algorithms in the node mobility and non mobility environment.

4.1.3 Throughput Over Time in MANETs

The amount of data transferred from sender to receiver is processed in specified time duration. Data transfer rates for nodes and networks are measured in terms of throughput.

In this simulation, the throughput is the number of packets arriving at the sink per ms/second. Here we have find out the instant throughput over time during nodes movement and non movement.



(a) (b)

FIGURE 4: Throughput – without and with Mobility.

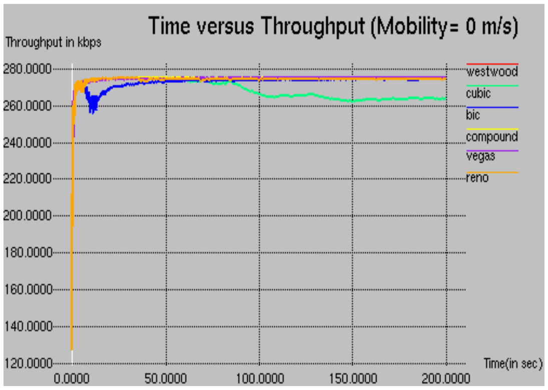
As shown in the Figure 4 (a), algorithms in the scenario without mobility, there was not much difference in provided throughput. But in the case of mobile scenario, as shown in the Figure 4 (b), the algorithm Vegas provided better throughput over time. The algorithm BIC provided best throughput after 75 seconds only but it provided poor results during initial phase of the communication.

4.2 Discussion of the Closer Analysis of Throughput Over Time

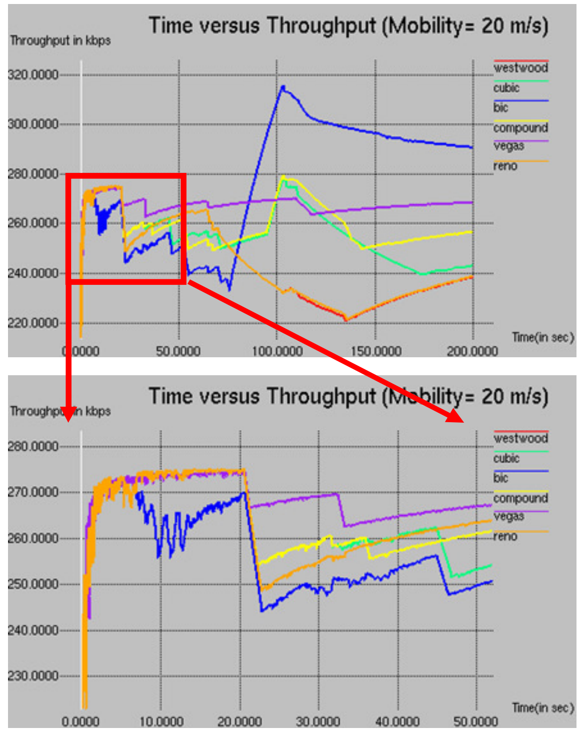
The following graphs illustrate the closer analysis of the congestion control algorithms in terms of throughput in the whole time duration.

4.2.1 Throughput Without Mobility

Figure 5 (a) shows the throughput over time in the case of mobility=0ms or without mobility. After 50 seconds, mostly all algorithms provided equal performance except CUBIC. During initial stage, even TCP BIC also very low throughput; but over time, it started to perform equal to other algorithms except CUBIC.



(a)



(b)

FIGURE 5: Throughput -without Mobility and with mobility closer analysis up to 50 seconds.

4.2.2 Throughput With Mobility

The Figure 5 (b) shows the throughput over time in the case of mobility=20ms. In this wireless network, the nodes started to move after 20 seconds. So up to 20 seconds, the graph is almost similar to after 50 seconds; all the algorithms provided equal performance except CUBIC. During the initial stage, TCP BIC also provided very low throughput; but all over the time TCP BIC performance is excellent than other algorithms from the group.

If we carefully observe the two sections (up to 50 seconds and 100 to 200 seconds) of Figure 5 (b), we can say that Vegas is the only algorithm which tried to provide almost constant and better throughput from starting to end of the simulation time. TCP BIC has started to provide better throughput after 75 seconds only. But it provided very poor throughput in the initial stage.

So as a final verdict, we have selected Vegas as the best performer in mobile ad hoc network scenario. As per the arrived results, we can say the algorithm Vegas can be used for short time TCP communication applications as well as long time TCP communication applications.

5. CONCLUSION & FUTURE WORK

We have successfully evaluated six congestion control algorithms using NS2 simulation tool in the Mobile Ad hoc Networks. The results are more significant and comparable. We have appraised the performance of these congestion control algorithms in very ideal condition without any cross traffic and any additional flows. In this small MANET scenario, the algorithm BIC provided good throughput after 75 seconds but algorithm Vegas provided stable and excellent throughput almost all over on the whole run time. So we move towards to the wrapping up that the algorithm Vegas will be the suitable algorithm for small and dynamic mobile ad hoc network scenario. Except Vegas, all other assessed algorithms provided very poor throughput during initial stage of the communication (less than 50 Seconds). So we conclude TCP Vegas will be the best algorithm from the list.

In this work, we have preferred six algorithms for evaluation, because they are default algorithms in several standard operating systems. But we have planned to do another evaluation based on the types of algorithms namely, Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery. In our future work, we will select few algorithms from each of these four categories and will evaluate their performance in MANET scenario. There are other varying network parameters and metrics that the authors are working on the same. Based on the results, we can extend the further enhancement towards specific application on MANETs.

6. REFERENCES

[1] D. X. Wei and P. Cao, "*NS-2 TCP- Linux: An NS-2 TCP Implementation with Congestion Control Algorithms from Linux*", proceedings of ValueTool'06 -- Workshop of NS-2, Oct, 2006.

[2] Shin-Jer Yang , Yung-Chieh Lin, Soochow University, Taipei, Taiwan , "*Tuning Rules in TCP Congestion Control on the Mobile Ad Hoc Networks*", Proceedings of the 20th International Conference on Advanced Information Networking and Applications, Volume 01 , 2006.

[3] Jehan.M, G.Radhamani, T.Kalakumari, "*A survey on congestion control algorithms in wired and wireless networks*", Proceedings of the International conference on mathematical computing and management (ICMCM 2010), Kerala, India, June 2010.

[4] Christian Lochert, Björn Scheuermann, Martin Mauve, "*A survey on congestion control for mobile ad hoc networks*", Wireless Communications & Mobile Computing, Volume 7 , Issue 5, pp. 655 – 676, June 2007, ISSN:1530-8669.

[5] M.Allman, V.Paxson, and W.Stevens. *TCP Congestion Control. RFC2581 (Proposed Standard), Apr.1999. Updated by RFC3390*.

[6] L. A. Grieco, S. Mascolo, "*Performance evaluation and comparison of Westwood+, Vegas and New Reno TCP congestion control*", ACM Computer Communication Review, April 2004.

[7] I. Chlamtac, M. Conti, and J. Liu, "*Mobile ad hoc networking: imperatives and challenges*", Ad Hoc Networks Journal, vol.1, no. 1, pp. 13-64, Jul, 2003.

[8] A. Al Hanbali, E. Altman, P. Nain, "*A Survey of TCP over Mobile Ad Hoc Networks*", Research Report no. 5182, INRIA Sophia Antipolis research unit, May 2004.

[9] Brakmo, L. S., O'Malley, S.W., and Peterson, L.,"*TCP Vegas: End-to end congestion avoidance on a global Internet*", IEEE Journal on Selected Areas in Communications (JSAC), 13(8), pp. 1465-1480 (1995).

[10] Mascolo, S. "*Congestion control in high-speed communication networks*", Automatica, Special Issue on Control Methods for Communication Networks, Vol. 35, no. 12, pp. 1921-1935 Dec, 1999.

[11] Sally Floyd and Kevin Fall, "*Promoting the use of end-to end Congestion control in the Internet*," IEEE/ACM Transactions on Networking, vol. 7(4), pp. 458–472, Aug, 1999.

[12] Chandrasekaran M. and Wahida Banu R.S.D., "*Interaction Between Polynomial Congestion Control algorithms Queue Management schemes in wired TCP Networks*," International Journal of Soft Computing, Vol. 1, No. 2, pp.83-90, 2006.

[13] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash,, "*A feedback based scheme for improving TCP performance in Ad-Hoc wireless networks*", Proceedings of the International Conference on Distributed Computing Systems (ICDCS 98), Amsterdam, Netherlands, May 1998.

[14] Foez ahmed, Sateesh Kumar Pradhan, Nayeema Islam , and Sumon Kumar Debnath, "*Performance Evaluation of TCP over Mobile Ad-hoc Networks*" in (IJCSIS) International Journal of Computer Science and Information Security ,Vol. 7, No. 1, 2010.

[15] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M., Wang. R. "*TCP Westwood: End-to-End Bandwidth Estimation for Efficient Transport over Wired and Wireless Networks*", In the Proceedings of ACM Mobicom 2001, (Rome, Italy, July 2001).

[16] Chandrasekaran M, Kalpana M and Wahida Banu R.S.D., "*Congestion Control using Polynomial Window size adjustment Algorithms for wired and wireless networks*", In the Proceedings of International Conference on Network –ICN06 conducted at Mauritius, Apr, 2006.

[17] Chandrasekaran M, , Kalpana M and Wahida Banu R.S.D., "*Interaction between MIMD-Poly & PIPD-Poly Algorithms and other TCP Variants in Multiple Bottleneck TCP Networks*", In the Proceedings of IEEE Conference WOCN2006, Bangalore, Apr, 2006.

[18] Tomoya Hatano, Hiroshi Shigeno, Ken-ichi Okada, "*TCP-friendly Congestion Control for High Speed Network*", International Symposium on Applications and the Internet-SAINT'07, pp.10, 2007.

[19] David X. Wei, Cheng Jin, Steven H. Low, and Sanjay Hedge., "*Fast TCP: Motivation, Architecture, Algorithms, Performance*", IEEE/ACM transactions on networking, 2006.

[20] K. Satyanarayan Reddy and Lokanatha C. Reddy., "*A survey on congestion control mechanisms in high speed networks*", IJCSNS-International Journal of Computer Science and Network Security, vol. 8, no. 1, pp. 187 – 195, 2008.

[21] Colin Perkins and Ladan Gharal., "*Rtp and the datagram congestion control protocol*", In Proceedings of IEEE International Conference on Multimedia and Expo, Toronto, Canada, July 2006.

[22] Van Jacobson., "*Congestion Avoidance and Control*", Computer Communications Review, Volume 18 number 4, pp. 314-329, Aug, 1988.

[23] Van Jacobson., "*Modified TCP Congestion Control Avoidance Algorithm*", end-2-end-interest mailing list, pp.1-14 April 30, 1990.

[24] A Linux TCP implementation for NS-2. URL: http://www.cs.caltech.edu/weixl/ns2.html

[25] A mini – tutorial for NS-2 TCP-Linux. URL: http://www.cs.caltech.edu/weixl/ns2.html