Design of Congestion Control Protocol for Wireless Networks with Small Flow Completion Time

Atul Gosai
Department of MCA
Saurashtra University
Rajkot Gujarat, India

Bhargavi Goswami
Department of MCA
Sunshine Group of Institutions
Rajkot Gujarat, India

Udit Narayan Kar
Department of MCA
Saurashtra University
Rajkot Gujarat, India

ABSTRACT

Why the flow completion time must be faster for congestion control algorithms? How the existing and newly proposed congestion control algorithms are far away from minimizing download times? And now the question remains is how to bring solution to the tough problem of reducing flow completion time in theory and practically, especially for multimedia long flows over wireless networking equipments. When internet users are downloading web page, downloading a file, sending an attachment, send/read mail, always involve the network in almost any interaction. Every user wishes to complete the transaction in the time as small as possible. Thus, each one over the internet demands smallest possible flow completion time. Nowadays, people are less concerned about network throughput, network efficiency, optimum network utility or packet drop rate or packet delivery ratio. They behave selfish wanting always to complete their flow as fast as possible. Here in this paper we are presenting the design of a newly created protocol called TCPBooster developed and suggested by us to reduce the limitations of existing congestion control protocol over wireless networks.

General Terms

Keywords

Congestion Control, XCP, RCP, RCP-AC, TCP, TCPBooster, Comparative Study, Congestion Control Equations

1. INTRODUCTION

We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace the content with your own material.

Defining congestion, it is a situation in Communication Networks in which huge number of packets is present in one or more portion of the subnet, which at the end results to performance degradation. One of the reasons for situation of Congestion occurrence in a network is when the load on the network means, the number of packets sent to the network is greater than the capacity of the network means, the number of packets a network can handle. Congestion-controlled traffic, poses a real threat to the overall health of the Internet [RFC2914, RFC3714, RFC4336]. To our biggest surprise is that, almost all the work done on congestion control is mainly going around metrics such as throughput, delay, jitter and fairness. While they are of the interest for particular networkoperators, network managers, network researchers and network designers. But, they are not very interesting to the network users. In fact, high throughput or efficient network utilization is not necessary in user's best interest [1].

Considering the situation in India, every individual is carrying 2 mobile phones. Each mobile adds to the sharing of bandwidth and frequency which at the end results to more scanning of network due to congestion into smart phone devices. These devices are communicating through internet having more videos and audios because of smart phones. Talking about the era before 10 years where, there were no smart phones, maximum transmitting flows were text and audio based. Pipelined links are already congested and because of video streaming and video transfers there is addition of long flows to the already congested links. But the requirement is that congestion control algorithm should make long flows finish as soon as possible, so that the situation of congestion in network due to long flows is in control, while behaving fair and stable towards the network[2].

In this paper we have taken most adapted congestion control protocols that are efficient in certain extent and weak at certain parameters and that extends to the design of a new protocol called TCPBooster which have covered many points listed in the wish-list of a well performing congestion control mechanism.

Our aim was to develop a congestion control protocol to overcome the loopholes and disadvantages of existing protocols like XCP[11,18,20], RCP[19], RCP-AC and of course TCP.In fact, TCPBooster is flexible, easily implementable over wired, wireless, hybrid networks, efficient in handling diversities with very low architectural complexities and very less modification in existing TCP architecture. The design proposed has to guarantee scalability, manageability, efficiency and easy maintainability which we have tried to explain in this paper.

The rest of the paper discusses about other congestion control protocol which is previously defined and tested by other researchers and some contribution of ours. These protocols are XCP[20], RCP[18,19], RCP-AC, TCP and TCPBooster. The paper is organized as follows. In Section II we have explained brief of the protocol XCP. In Section III, we have explained RCP with its equation explanation followed with its advantages and disadvantages. In Section IV we have shown the similarities between RCP and XCP protocol look at the sensitive point we covered explaining relationship between both of them. Section V explains the extension of Section III which is improvement in RCP to overcome few of its weak points. All of them are having common disadvantage of complex implementation over wireless networks which are not in the protocol of TCP, explained in Section VI. Section VII explains the problems with existing congestion control protocols and compare it with the demands of current scenario. We proposed the solution in the form of a new protocol design called, TCPBooster which is explained in detail at Section VIII. Section IX gives the conclusion followed by the references.

2. XCP

The fundamental algorithm of XCP has been developed and published by Dina Katabi of MIT and Mark Handley of ICIR. XCP[11,18,20] is a feedback-based congestion control system that uses direct, explicit, router feedback to avoid congestion in the network[5].Simulations indicate that XCP is powerful and scalable[5]. In collaboration with Prof. Katabi, ISI is currently developing a BSD (UNIX) implementation of XCP, funded by the NSF under the OptIPuter medium ITR, by DARPA, and by ISI. Explicit Control Protocol (XCP) represents a major advance in Internet congestion control. XCP delivers the highest possible application performance over a broad range of network infrastructure, including extremely high speed and very high delay links that are not well served by TCP[5]. By doing so, it achieves maximum link utilizations and wastes no bandwidth due to packet loss as proved by the authors of [5]. XCP is novel in separating the efficiency and fairness policies of congestion control, enabling routers to quickly make use of available bandwidth while conservatively managing the allocation of bandwidth to flows[5]. XCP is built upon a new principle: carrying per-flow congestion state in packets. XCP packets carry a congestion header through which the sender requests a desired throughput. Routers make a fair per-flow bandwidth allocation without maintaining any per-flow state. Thus, the sender learns of the bottleneck router‟s allocation in a single round trip [5,18].

The calculation of bandwidth adjustment for XCP required is done by two algorithms: 1) Efficiency Algorithm and 2) Bandwidth Calculation Algorithm [8]. Our interest is efficiency algorithm which calculates bandwidth AB which is distributed to all the flows at the interval T given as follows:

$$AB = \alpha.(C - input\_bw) - \beta.\frac{q}{T} \qquad \text{------------}(1)$$

Where, C is capacity of link, input_bw is bandwidth used during last period T calculated based on average RTT, q is minimum queue length. And α and β are performance and stability parameters [8].

XCP inherits some of the unfairness problems of TCP as it takes many RTT rounds to allocate a fair share to current traffic on network flows. Second biggest disadvantage is that it causes many router computation overheads [8]. The calculation of C, bandwidth is based on queue measured. And queue measure cannot be efficient when medium is not being fully utilized. Thus, XCP is more resource and time consuming, not very good option in very large deployments [8].

3. RCP

Nandita Dukkipatti proposed RCP (Rate Control Protocol) that enables typical large Internet Sized flows to complete in one to two orders of magnitude faster than the existing (TCP SACK) and XCP congestion control algorithms [2]. In the era of smart phones and large demand of internet by individual is increasing drastically and will be increasing for the next few years. This clearly expects from we like researchers to solve the issue by improving the congestion control mechanism.

Going to the fundamentals of congestion control, there are two types of congestion control mechanisms. One is using explicit feedback from routers to compute the rate; another is doing per packet calculations for computing the rate. In both the mechanisms, rate provided to all the flows remains approximately the same. In RCP, the function of providing same rate to all the flows based on the above stated two congestion control mechanisms is done by Router[18,19,20]. And to complete the flows faster, RCP provides with extreme quick computation the excessive bandwidth to the flows. This is the reason behind faster completion of flows by RCP.

It is observed through experimentation that proposed RCP is reducing Flow Completion Time (FCT) for typical Internet Size flows by one to two orders of magnitude which in comparison of TCP and XCP are performing remarkable. Talking about the mechanism used by RCP, an improved congestion control mechanism, RCP achieves it by simply emulating processor sharing at each router. RCP assigns same rate to all the flows, i.e.R(t) passing by the link and rate is adapted to current link congestion. Each packet passing by carry a field for the smallest R(t) along the path. If R(t) is found to be smaller, optimum R(t) is calculated at each router and given to each packet passing by that router. Previous value of R(t) is always overwritten by currently computed R(t) at each router the packet pass by. The destination sends the value back to the source and so, source can compute the rate for traffic according to the fair share rate based on current available bandwidth. Here first biggest advantage of RCP is that no per-packet computation is done. This reduces much of computational overhead over the networking routers. Second biggest advantage of RCP is that unlike TCP‟s slow start, RCP can directly jump to the optimum initialization rate and remains on the optimum rate until network burst is observed.

Talking about the RCP’s disadvantages, RCP also starts with out of date Rate, i.e RTT. Likely to most conservative congestion control mechanisms, it starts faster than TCP (slow start) and XCP (AIMD- Additive Increase Multiplicative Decrease) but run the risk of Overshooting. Overshooting means when there are rapid changes in the network traffic, sudden flash crowd and busty traffic conditions where thousands of flows suddenly turn up towards to network, RCP becomes unstable. In such situations, RCP behaves unstably, fairness of protocol is vanished, RCP recovery is timely, buffers overrun, and bars of packet lost are too large to recover. RCP rate update equation designed and developed by Nandita and team is give here [1].

$$R(t) = R(t - d_o) + \frac{\underline{[\alpha(C - y(t)) - \beta\, q(t)/d_o]}}{N(t)} \qquad \text{----}(2)$$

Where, R(t) is the rate to be computed for RCP, d is the moving average of RTT, T is update interval, d is the max limit of update interval, R(t-T) is previous rate, C is the link capacity, y(t) is aggregate traffic rate, q(t) is the queue size. Here, „a‟ and „b‟ are chosen as stability and performance parameters respectively.

4. RCP &XCP SIMILARITIES

At a high level RCP behaves similar to XCP. In the sense, both, RCP and XCP assign explicit rates calculated by routers, and RCP uses a similar rate- update equation. However, XCP gives a different rate to each flow, requiring per-packet calculations [2]. Most relevant here, XCP converges slowly towards the fair-share rate in small increments so as to avoid packet loss [2]. For most flows, the flow has finished before the fair-share rate has been reached, which explains why flows take so long to complete with XCP [2,20]. On the other hand, RCP gives a high starting rate; so new flows find their fair-share rate quickly because a single rate is given to all flows (new and old)[3]. Flows achieve low completion times, but there can be sharp peaks in queue length that lead to packet loss when there is a rapid change in network conditions [3]. Both the algorithms are computing their link capacity; we also refer as Bandwidth based on queue occupancy, which is not the right measure when the network resource utilization is not to the fullest.

5. RCP-AC

Later on, RCP was extended by the same researchers stated before and they proposed RCP-AC (Rate Control Protocol – Acceleration Control) that allowed RCP configuration to become that quick that it was easily completing flows in a broad set of operating conditions. It is by behavior, both, aggressive for FCT to get completed soon and conservative for imposing more stability in abrupt networking situation. The main objective was to develop the protocol that behaves stable under the diverged set of traffic profiles. RCP-AC is the extension of RCP.

RCP-AC controls traffic in three different ways: a)Rate Control. It sets a target flow-rate, just like RCP; b)Acceleration Control. It limits the acceleration (increase in flow-rate); and c)Feedback Control. It gives priority to jump-starting new flows; only when there is congestion will RCP-AC give priority to limiting queue size. These functions are executed periodically in every router, typically about once per round-trip time[2]. And RCP-AC equation became:

$$\Phi = \alpha(C - y(t)).d + (\gamma B - \beta.q(t)) \quad \text{-------------(3)}$$

Controller computes, φ, the maximum increase in aggregate traffic volume permitted in the next time period: where B is the buffer size and γ is the user defined limit on acceleration [2]. And α and β are performance and stability parameters. The equation of RCP-AC is very near to that of XCP. But, XCP is not inter-operable with currently available protocols resulting in a poor performance of XCP flows when the resources are shared with TCP-like flows especially[4][3]. RCP-AC is much more complex (several per-packet computations) than RCP[3].

6. TCP

TCP is the protocol widely accepted and implemented in the network world in all the diversified network situations, compatible with each networking device and each network mechanism. Researchers prefer experimenting with TCP as it is the only protocol in real world with all the simulators support and easy equipments configuration for model based or real time experimentation testing. But, few problems are observed in TCP protocol.While the global network access speeds increased dramatically on average in the past decade, the standard value of TCP‟s initial congestion window has remained unchanged [7]. As per a 2010 study [4], the average connection bandwidth globally is 2Mbps with more than 50% of clients having bandwidth above 2Mbps, while the usage of narrowband (< 256Kbps) has shrunk to about 5% of clients. In spite of avoiding congestion in the early stages on the Internet, TCP is finding it increasingly difficult to cope with the growth of communication network capacities and applications as shown in the literature [1,6,13,14,21].TCP's inability to properly utilize network links is one of the problems of TCP [6]. Any delay or packet loss can cause TCP to dramatically reduce its congestion window and hence under-utilize the link [6]. This is mainly because TCP doesn't have proper method to find the available capacity in the path of the flows and set the rate of each flow. So it takes many rounds for TCP to find the correct rate at which flows need to send data. This means that flows are taking unnecessarily long duration to get completed. Besides, TCP even takes long time to achieve fairness between flows [6].There have be many improvements proposed for TCP to change its behaviour for slow start and AIMD behaviour like High Speed TCP, Scalable TCP, Cubic TCP, etc. Main aim here is to develop an algorithm that should be able to achieve large equilibrium window size considering realistic drop ratio. Window size remains small for many RTT with additional drawback of window size reduced to half on each packet drop. Because of AIMD, drastic reduction in rate on each packet drop demands more RTT to reach to desired expected rate for each particular flow. Thus, to finish long flows by achieving rate near to desired rate, we need a more adaptive and robust solution for diverse network like the one proposed in the next section.

7. PROBLEM WITH EXISTING ALGORITHMS

The biggest issue with the entire list of above stated algorithm is that, they are not adaptive to the wireless scenarios. Internet was less mobile before and more infrastructures dependent. But now, in the world of smart phones, tablets, PDAs and GPS, internet is demanded more by wireless equipments in comparison of fixed equipments. While the global adaptation of broadband is growing with the speed of light, TCP‟s init_cwnd has remained unchanged since 2002[4]. Average TCP‟s bandwidth for any connection is 1.7Mbps, but more than 70% average client connections are having 2G networks.

The above described protocols are also proved heavy for implementation into light devices with the reference of OS, Processing, Memory Requirement and Power consumption. They perform complex computation which is consuming power, requiring more memory for storing per packet states, heavy for Hot Spot Tethering and might get hang with heavy processing. Simulation results prove that XCP and RCP are not easily adaptive to wireless implementation in comparison of TCP. TCP is adaptive to wireless implementation, but is having issues for multimedia flow inclusion and justice are not given to multimedia flows co existing with short and non multimedia flows. There is a clear requirement of the congestion control mechanism that is adaptive to wireless scenarios, less demanding for computational and storage resources, compatible to current internet demand of multimedia and short flows and with high networking parameter performances.

As a solution, we propose, a new scheme in the next section that would solve the above stated issues with current congestion control mechanisms.

TABLE I: Comparison for all the congestion control Algorithm

Points of Difference	**Comparison of Protocols**			
	XCP	**RCP**	**TCP**	**TCP Booster**
Per Packet Calculations	Yes	Yes	No	No
Problem of large data transfer in high bandwidth delay networks	Problem partially solved	Near to complete solution	Problem unsolved	Proposed solution would solve this problem
Router‟s Involvement	Involves Routers in congestion control	Also involves Routers	No direct information passed from routers to end users.	No direct information passing required.
Flows	Faster flow completion time but less than RCP	Short flow completion time	Not specific over flows (short/large)	As per the demand and size of flow, bandwidth is allocated.
Bandwidth Utilization	Average or more utilization of Bandwidth	Complete Utilization of bandwidth	Low utilization of high bandwidth delay product network	Complete Utilization of bandwidth is expected and under tests.
Load	High	High	High	Undefined
Overhead	High	High	High	Expected to be low
Congestion Control	Average	High	Very low	Expected to be High
Mechanism	Inform sender to get max rate of transmission by information header	By stamping each packet with max rate and by processor sharing	Using AIMD, Slow Start and Fast Retransmission, Fast Recovery	By utilizing max congestion windows and keeping the queues smallest

8. TCP BOOSTER

We recently described a simple congestion control algorithm called TCPBooster which reduces flow completion time for diverse flow types to large extend for broad range of traffic conditions and network situations.

In TCPBooster, unlike XCP, RCP and RCP-AC, we are not following feedback mechanism but we are adapting the ancient congestion window based congestion control mechanism. The reason behind sticking to the congestion window for congestion control is that, while each second new flow are entering and moving out of network, it is tough to obtain exact number of flows at particular RTT. This inspired us to stay with congestion window based mechanism instead of feedback based mechanism. We set the „cwnd‟ value completely based on the rate computation, but that doesn‟t impose any overhead over the router. In addition, we are setting the value of „maxcwnd‟ to obtain the better start for our optimum data rate calculation at the initial stage.

The basic TCP Booster algorithm operates as follows:

1) It sets maxcwnd value before the transmission starts.
2) It sets rate based on „cwnd‟ value computed.
3) Every router maintains and updates rate R(t) approximately once per RTT.
4) The Source rate is observed by Destination and it adapts to the available flow rate.
5) The Source rate is observed by Destination and it adapts to the available flow rate

The steps expected to be followed for the implementation of TCPBooster over simulator of NS2 [9,12,15,17] gives us following briefing:

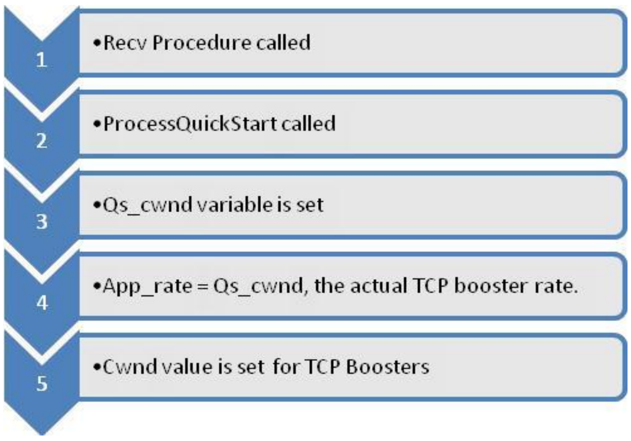
1. Sendmsg (nbytes) function is invoked.
2. It creates Packet and forwards it using the instance of target_ ->recv(p,h) where, p is packet and h is header called as argument when called procedure of recv.
3. Once TCP receiver receives packet, we call
4. “TCPSink”.
5. TCP receiver creates ACK packet and returns to TCP sender via de-multiplexer.

TCP protocol in simulation implementation is following congestion window based mechanism to calculate optimum rate for the transmitting flows. This equation can be given by:

$$R(t) = [\{R(f)qsh * \alpha(T - RTT)\} - q(s)/\sum(p)size + \sum(p)header]/C \qquad \text{----------}(4)$$

Where, R stands for Rate, f stands for flow, α is stability parameter and p stand for packet. RTT stands for Round Trip Time and the T is current time, term becomes interval. We are considering queue by the term q(s). C is the link capacity. The rate calculated is given to the procedure of TcpAgent class‟s “ProcessQuickStart”, calling for parameter of Packet class. The obtained rate is given to “qx_cwnd”. The procedure is shown in the Fig.1.

Fig. I: TCP Booster rate setting procedure


1
•Recv Procedure called
2
•ProcessQuickStart called
3
•Qs_cwnd variable is set
4
•App_rate = Qs_cwnd, the actual TCP booster rate.
5
•Cwnd value is set for TCP Boosters

9. CONCLUSION

Our As more and more wireless transmission has evolved, more of the congestion control has to be adaptive to diverse flow and hybrid network situations.The emergence of next generation wireless communication networks must support wide range of multimedia applications with heterogeneous network situation support and faster congestion clearance capability. The proposed TCPBooster is adaptive framework design which generalizes the previously done work on congestion control and is easily supporting heterogeneous network scenario and traffic. The reason of selecting among feedback and cwnd based approach for rate calculation is given which would improve the performance of proposed design. We have also shown equations of each protocol talked about in our paper. TCPBooster is to be implemented on simulator first and there after upon real world‟s routing networks to check for the performance parameters testing. Channel and queue aware cross layer scheduling and packet header + size (i.e 40 + 1500 = 1540) and interval conscious resource allocation strategy would fairly allocate channel resources among heterogeneous traffic situations with minimum queue occupancy and minimum packet loss. Proposed solution to the listed congestion control protocol issues would allow research community to look into the fair performance comparison with its clear specification of characteristics among widely talked congestion control protocols of XCP, RCP, RCP-AC[18,19,20]. Our paper also gives reason for selecting TCP the base for further modification as it is the only one protocol with maximum flexibility and adaptive to most variable traffic situations and widely accepted among end users, network engineers and network researchers. But, TCP is not a good congestion control protocol for current network demand due to weak signal strength and its slow start and AIMD strategies. XCP and RCP have a problem of wrong bandwidth availability measurement in wireless networks[9,10,12,16]. Consideration performance of TCPBooster with consideration of NAV (Network Allocation Vector) and RTS/CTS (Request / Clear to Send)is of prime interest for us as we are being targeting wireless communication.

Future research may check for trade off between efficiency, delay, fairness and coverage. To widen its application scope, current work need to focus upon extension of this approach by applying it to sensor scenarios because they are also light weight devices which may include more sophisticated congestion control mechanism with less overhead. We are currently studying and working upon all these parameters to implement the performance check for the proposed solution by the means of TCPBooster to provide accurate mechanism that optimize available bandwidth utilization in dynamic networks with light weight wireless and sensor networks.

6. REFERENCES

[1] NanditaDukkipati and Nick McKeown, Why Flow-Completion Time is the Right metric for Congestion Control and why this means we need new algorithms, ACM SIGCOMM Computer Communication Review, New York, NY, USA Volume 36 Issue 1, January 2006 Pages 59 – 62

[2] N. Dukkipati , Nick Mckeown, RCP-AC: Congestion Control to make flows complete quickly in any environment. in Proc. High-Speed Networking Workshop: The Terabits Challenge, IEEE INFOCOM, 2006.

[3] NanditaDukkipati, YasharGanjali and Rui Zhang-Shen, Typical versus Worst Case Design in Networking, In Fourth Workshop on Hot Topics in Networks, College Park, Maryland, November 2005.

[4] HamsaBalakrishnan, NanditaDukkipati, Nick McKeown and Claire J. Tomlin, Stability Analysis of Explicit Congestion Control Protocols, IEEE COMMUNICATIONS magazine, vol. X, no. X, month 2007.

[5] Dino Lopez, LaurantLefevre, Congduc Pham, “Lightweight Fairness Solutions for XCP and TCP Cohabitation” in IFIP International Federation for Information Proc. Networking, 2008, LNCS – 4982, pp715-726, 2008.

[6] Fesehaye, Debessay, and KlaraNahrstedt, Implementation of the Network Control Protocol using ECN Bits (eNCP), Technical report, University of Illinois at Urbana-Champaign (UIUC), 05 2011.

[7] Dukkipati, N., Refice, T., Cheng, Y., Chu, J., Herbert, T., Agarwal, A, Jain, A., And Sutin, N, An argument for increasing TCP‟s initial congestion window, SIGCOMM Comput. Commun. Rev. 40 (June 2010), 26–33.M.

[8] LufsBarreto, Susana Sargento,"TCP, XCP and RCP in Wireless Mesh Networks: An Evaluation Study", Computers and Communications (ISCC), 2010 IEEE Symposium, on 22-25 June 2010, ISSN :1530-1346, pg. 351 - 357.

[9] Dr. AtulGonsai, BhargaviGoswami, "Network Simulator forEfficient Performance Parameter Testing & Evaluation" in National Journal of System and Information Technology - 2012, volume 5 Issue 1, pg. 89-105

[10] Dr. Atul Gonsai, Bhargavi Goswami, "Experimental PerformanceTesting of TCP and UDP Protocol over WLAN 802.11b and 802.11g" Karpagam Journal of Computer Science, Volume 07, Issue 03, March April2013, pg. no. 168 to 183

[11] Dr. Atul Gonsai, Bhargavi Goswami, Uditnarayan Kar "Evolution of Congestion Control Mechanisms for TCP and Non TCP Protocols"Matrix Academic International Journal of Engineering and Technology, MAIOJET, Volume 01, Issue 02, October2013

[12] Dr. Atul Gonsai, Bhargavi Goswami, Uditnarayan Kar "Experimental based performance Analysis of IEEE 802.11/g Hybrid Network" in Journal of Network and Information Security(JNIS), Volume 1 Issue 1 Pg. No. 61-66, October 2013

[13] Martin Bateman, SaleemBhatti, Greg Bigwood, DevanRehunathan, ColinAllison, TristanHenderson, DimitriosMiras, "A comparison of TCP behaviour at high speeds using ns-2 and Linux", Proceedings of the 11th communications and networking simulation symposium CNS '08, SpringSim, New York, Feb 2008, Pg. No. 30-37

[14] W.Stevens, "RFC 2001, TCP Slow Start, Congestion Avoidance, Fast Retransmission and Fast Recovery Algorithms, http://www.ietf.org/rfc.rfc2883.txt, July 2000

[15] Kevin Fall, Sally Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP", Newsletter ACM SIGCOMM Computer Communication Review, Volume 26 Issue 3, July 1996 Pg. No. 5-21

[16] Mohit P. Tahiliani, K. C. Shet and T. G. Basavaraju, "Comparative Study of High-Speed TCP Variants in Multi-Hop Wireless Networks "International Journal of Computer Theory and Engineering, IJCTE 2013 Vol.5 Issue.5 Pg. 802-806.

[17] Marc Greis Tutorial for NS2, maintained and being expanded by the VINT group, http://www.isi.edu/nsnam/ns/tutorial/ :as on date 29/10/2013.

[18] Luıs Barreto, Susana Sargento,"XCP-Winf and RCP-Winf: Congestion Control Techniques For Wireless Mesh Networks", iscc, The IEEE symposium on Computers and Communications, IEEE ICC 2011 proceedings.

[19] N. Dukkipati, G. Gibb, N. McKeown, and J. Zhu, "Building a RCP (rate control protocol) test network", In Proceedings of Hot Interconnects, August 2007.

[20] H. Balakrishnan, N. Dukkipati, N. McKeown and C. Tomlin, "Stability Analysis of Switched Hybrid Time-Delay Systems – Analysis of the Rate Control Protocol", http://yuba.stanford.edu/rcp/, Stanford University Department of Aero/Astro Technical Report.

[21] RFC: 793, Transmission Control Protocol, under Darpa Internet Program, http://www.ietf.org/rfc/rfc793.txt