Performance Evaluation and Comparison of TCP variants for Optimizing TCP Performance

Prof.J.VijiPriya,Dr.S.Suppiah

Abstract- Wireless networks are growing rapidly. TCP is the most widely-used protocol on Internet and so optimizing TCP performance is very important for fast efficient data transfer. The different existing TCP variants and solutions they have not been analyzed together to identify the bottlenecks in wireless networks. TCP has a major problem in its congestion control algorithm which does not allow the flow to achieve the full available bandwidth on fast long-distance links. This problem has been studied in this paper using a new high speed congestion control TCP protocol based on the Newton Raphson algorithm. In this paper we have analyzed six TCP congestion control algorithms and compared their performance behavior of Reno, Westwood, Veno, Vegas, Illinois and proposed Newton Raphson congestion control algorithms with suitable metrics. This study shows that the proposed algorithm performs better compared with the other methods of application.

Index Terms – TCP Congestion control protocol, Vegas, Reno, Westwood, Veno, Illinois, Full Bandwidth Utilization, RTT fairness, Packet Loss Rate.

1 INTRODUCTION

Wireless networks are increasingly being deployed throughout the world. Few attempts and solutions [1],[2],[3],[4],[5] have been proposed to improve TCP performance. Several high speed data applications require the stability of the Internet which is evolving from very high speed and long distance TCP network paths. One of current challenges of Internet is performance of TCP. These High speed TCP Networks are characterized by BDP (Bandwidth and Delay Product) which represents the total number of packets to be sent while keeping the bandwidth fully utilized. The stability of Internet is achieved by developing mechanisms to reduce transmission errors, to provide better bandwidth sharing of resources, to reduce the RTT and mainly to provide congestion control by TCP. TCP's end-to-end congestion control mechanism reduces the packet losses by adjusting the number of out standing unacknowledged data segments allowed in the Network [6].

TCP is not well suited for streaming real time audio and video application due to it increases end-to-end and delay variants [7]. To modified Standard TCP, the new congestion control algorithms are being developed because of more and more computers get interconnected only using TCP. The existing linear congestion algorithms generalize AIMD (Additive Increase and Multiplicative Decrease) to increase congestion window for increasing the bandwidth of the TCP connection and when the congestion occurs, the window size is multiplicatively reduced by a factor of two[8].

- *Prof.J.VijiPriya is currently pursuing Ph.D in Computer Science and Engineering .Anna University; presently he is working as Associate professor in CSE department at AMS Engineering College. E-mail: vijipriyajkv@gmail.com*
- *Dr.S.Suppiah is working as principal at P.S.Y Engineering College, Anna University.*

This paper analyses the performance of the non linear congestion control protocols for Internet transport protocols and applications and found that the rate reduction techniques cause degradation in user - perceived quality [9],[10].MIMD(Multiplicative Increase and Multiplicative Decrease), PIPD(Polynomial Increase and Polynomial Decrease) are developed and provides better throughput for wired and wireless networks [11],[12]. This analysis results into get good understand of TCP-compatible congestion control algorithms.

In a previous study [13],[14], NS-2 (Network simulator), the discrete event driven simulator is used by most of the network researchers and to analyze the fairness among high-speed TCP protocols in sharing bandwidth. The motivation for this project is to use NS-2 TCP Linux [15] instead of the standard NS-2 TCP Agent to study the performance among the protocols. NS-2 TCP Linux is a new implementation of TCP in NS-2 whose implementation is based on Linux 2.6 TCP. Future versions of NS-2 will include TCP- Linux Agent.

The proposed NRC-TCP generalizes the AIMD algorithm is analyzed in a simulated wireless TCP network. It provides additive increase and multiplicative decrease for congestion avoidance using the exponential of α and β, These are scaling factors determined by real roots of algebraic equation which have been evaluated based on the Newton Raphson method to adjust the current congestion window size. This paper also evaluates the performance of NRC-TCP using simulation of proposed model in NS-2 with TCP-Linux modification. The results of this simulation are compared with the high speed TCP variants such as TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas,TCP-Illinois. Six flows using six different high-speed TCP protocols including the proposed NRC-TCP protocol are run. The comparison shows that the proposed algorithm provides better performance in terms of performance evaluation constraints

The remainder of the paper organized as follows, Section 2 gives related work, Section 3 includes analysis and discussion related to property of NRC-TCP, Section 4 presents the results of experimental evaluation and Section 5 gives conclusion.

2 RELATED WORKS

Networks are mainly divided into wired and wireless groups. In wired network, router is one of the network elements to route the packets. Wireless networks [16] are classified into infrastructure based networks and infrastructure less networks. The infrastructure network is Ad-hoc network [17] where there is no fixed infrastructure and separate network element called router. Thus, mobile nodes are act as the routers. The Internet can be thought of as a network of links, where packets share the link along certain routes.

Congestion control is an important component of a transport protocol in a packet-switched shared network. The congestion control algorithm of the widely used transport protocol TCP is responsible for detecting and reacting to overloads in the Internet and has been the key to the Internet's operational success. However, as the link capacity grows and new Internet applications with high-bandwidth demand emerge, the performance of the TCP is unsatisfactory, especially on high speed and long distance networks. The main reason is the conservative behavior of TCP in adjusting its congestion window that governs the senders' transmission rates. A number of solutions have been proposed to overcome the aforementioned problem of TCP by changing the way in which TCP adapts its congestion window: BIC-TCP, CUBIC, FAST [18], HS-TCP, H-TCP, STCP, TCP-Westwood [19], and TCP-Africa [20]. These new protocols promise to improve TCP performance on high-speed networks significantly and are hence usually called TCPs for high-speed networks.

Floyd [21] proposed a framework for evaluating congestion control algorithms. The framework includes a number of metrics to be considered such as throughput, packet loss rates, delays, and fairness as well as a range of network environments. Along the same line, Wei et al. [22] proposed that the networking community establishes a TCP benchmark suite to leverage comparative performance evaluations of TCP variants. The benchmark includes various scenarios for realistic performance evaluations such as heavy-tail file size distributions and ranges of propagation delays. The frameworks proposed by Floyd and by Wei et al. illustrate the need for realistic performance evaluations of new congestion control algorithms and accentuate the motivation for our work and existing evaluation work that we briefly review below.

Bullot et al. [23] compared the performance of TCP New Reno with HSTCP, FAST, STCP, HSTCP-LP, H-TCP, and BIC TCP on high-speed production networks. They reported that TCP Reno gave low and unstable performance and most TCPs for high-speed networks delivered significant improvement over TCP Reno. Bullot et al.'s results are very encouraging. Nevertheless, as their experiments are performed over a real production network path, they don't have any control over the background traffic on the network. They only included UDP background traffic and did not consider the impact of network environments created by various mixes of background traffic on protocol behaviors. Li et al. [24] performed experiments for STCP, HSTCP, BIC TCP, FAST, and H-TCP in a lab network. They noted that most protocols, especially FAST, STCP, HSTCP and BIC, exhibit substantial unfairness in their experiments and highlighted the good performance of HTCP. Since Li et al. did not have any background traffic in their experiments and the results may be subject to the deficiencies such as unfairness, degradation in throughput, unfriendliness and packet loss.

TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded End-to-End performance in wireless environment. The TCP congestion control should be modified to utilize the available bandwidth efficiently in wireless environments. The proxy enables the TCP sender to confirm a packet loss due to wireless error when it receives an acknowledgement with the RNF flag set [25]. Snoop outperforms split TCP schemes [26], even when TCP with the SACK option is used over the wireless link, without violating TCP semantics, since TCP itself remains unmodified. It also avoids conflicting local and TCP retransmissions [27] by suppressing duplicate TCP acknowledgments whenever it performs local error recovery.

A survey on congestion control for MANETs [28] argues that the majority of work is concerned with improving the performance of TCP in MANET scenarios [29] indicates that TCP-like congestion control mechanisms suffer fundamental problems in the presence of wireless interference. Recent developments in social-based opportunistic forwarding [30],[31] have identified that load is unfairly distributed towards nodes which are better connected. By making informed forwarding decisions based on a heuristic that favors connectivity, delivery probabilities increase, but load distribution becomes more unbalanced. Congestion control mechanisms also have to contend with sudden changes in the bandwidth-delay product due to mobility. Such bandwidth-delay product changes are expected to become more frequent and to have greater impact than path changes today. As a result of both mobility and of the heterogeneity of wireless access types (802.11b,a,g, WIMAX, HS-WCDMA Bluetooth, etc.), both the bandwidth and the round-trip delay can change suddenly, sometimes by several orders of magnitude. Evaluating the response to sudden or transient changes can be of particular concern for slowly responding congestion control mechanisms such as equation-based congestion control [RFC3448] and AIMD (Additive Increase Multiplicative Decrease) or for related mechanisms using parameters that make them more slowly-responding than TCP [32].

High-speed TCP protocols can be broadly categorized into two categories based on how they sense congestion in the network:

1) Loss -Based Protocols

2) Delay-Based Protocols

Loss-based protocols use packet loss in the network to detect congestion where as delay-based protocols use queuing delays at the routers, in addition to loss, to detect congestion. FAST TCP is the only high-speed TCP protocol which is delay based, but its implementation is not present in Linux, so all the protocols we have considered in our experiments are loss-based protocols.

TCP-Hybla [33] scales the window increment rule to ensure fairness among the flows with different RTTs. TCP-Hybla behaves as TCP-NewReno when the RTT of a flow is less than a certain reference RTT (e.g., 20ms). Otherwise, TCP-Hybla increases the congestion window size more aggressively to compensate throughput drop due to RTT increase.

TCP-Veno [34] determines the congestion window size very similar to TCP-NewReno, but it uses the delay information of TCP-Vegas to differentiate non-congestion losses. When packet loss happens, if the queue size inferred by the delay increase is within a certain threshold, which is the strong indication of random loss, TCP-Veno reduces the congestion window by 20%, not by 50%.

TCP-Illinois [35] uses a queuing delay to determine an increase factor α and multiplicative decrease factor β instantaneously during the window increment phase. Precisely, TCP-Illinois sets a large α and small β when the average delay d is small, which is the indication that congestion is not imminent, and sets a small α and large β when d is large because of imminent congestion.

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.

3 NRC ALGORITHM

Newton Raphson Congestion Control TCP is similar to High Speed TCP. It uses the value of the previous congestion window to compute its new congestion window value. It behaves like standard TCP when the congestion window is below a threshold value. Above the threshold, High Speed TCP acts more aggressively in attaining bandwidth by increasing its congestion window size aggressively. It suggests a modified slow start, congestion avoidance, modified fast retransmission and Fast recovery mechanism. It generalizes AIMD. If sending rate is too fast between two communication hosts, result in congestion. The router will start to discard packets to avoid congestion. As the sender detects to packet loss, it infers that congestion happens in the network. NRC-sender will start a succession of congestion control at the moment and reduces sending rate. In this case, action is taken following a recovery procedure.

3.1 Modified Slow Start

NRC-TCP differs from other algorithms during its slow start phase. The reason for this modification is that when a connection first starts it has no idea of the available bandwidth and it is possible that during exponential increase it over shoots the bandwidth by a big amount and thus introduces congestion. To end this, it increases exponentially only every other RTT and calculates the actual sending throughput to the expected. When congestion window exceeds or equals slow start threshold, it exits slow start and enters the congestion avoidance phase.

3.2 Congestion Avoidance

When congestion window exceeds or equals slow start threshold, the state enters congestion avoidance. The congestion window is increased by eα where α is window scaling factor determined by real roots of algebraic equation which are found by Newton Raphson method for every arrival of a new acknowledgement until congestion occurs.

3.3 Modified Retransmission

As packet loss happens, the TCP sender receives three duplicate ACK and triggers modified fast retransmission and fast recovery immediately. Then the sender does not wait for retransmission timeout to send lost packet back. Besides slow start threshold will be set as eβ or double MTU (Maximum Transmission Unit) and set up congestion window as slow start threshold plus 3 MTU.

3.4 Binomial Congestion Control

In [36] the authors proposed a class of non-linear TCP compatible congestion control schemes called Binomial Congestion Control schemes, which are well suited for real time streaming applications. AIMD can be considered as one of congestion control schemes in the subset of binomial algorithm. Formally, every TCP-like congestion control equation can be generalized by the binomial algorithm as the following two equations:

$$W' = W + \alpha / w^k ; \alpha > 0 \text{ and if no loss} \quad (1)$$

$$W' = W - \beta\, w^l ; 0 < \beta < 1 \text{ and if loss} \quad (2)$$

Where W' is the congestion window after adjusting, k and l are window scaling factors for increasing and decreasing respectively, α and β are proportionality constants. For any given value of α and β and k + l = 1 and l ≤ 1, this class of congestion control will be TCP-Friendly. Furthermore, all the binomial control protocols converge to fairness as long as k ≥ 0, l ≥ 0 and k + l > 0.

3.5 NRC – TCP Window Growth Function

The window adjustment policy is only one component of the congestion control protocol derived from Newton Raphson congestion control algorithms. The proposed algorithms mainly aim in increasing the window size faster and to gain the bandwidth quicker.

ISSN 2229-5518

NRC-TCP is similar to High Speed TCP. It uses the value of the previous congestion window to compute its new congestion window value. NRC- TCP behaves like standard TCP when the congestion window (cwnd) is below a threshold. Above threshold NRC-TCP acts more aggressively in attaining bandwidth by increasing its congestion window size aggressively.

On each arrival of a new acknowledgement, NRC-TCP increases its congestion window by the following:

$$W' = W + e^{\alpha} \;; \quad if \; no \; loss \qquad (3)$$

When congestion is detected through packet loss, the congestion window is decremented as follows:

$$W' = W - e^{\beta}; \quad if \; loss \qquad (4)$$

where α and β are real root of rational integral equation determined by Newton Raphson Method.

4 EXPERIMENTAL EVALUATION

4.1 Experimental Setup

A lot of experiments have been performed with any congestion in the network to study the performance of TCP by changing bandwidth and RTT of bottleneck link [37]. It consists of six wired nodes, one router, one base station and six mobile nodes. First six wired nodes are connected with a router through 100MB (Mega Bytes) bandwidth and 10ms delay duplex wired connection. The router is connected with a base station through 10Mb and 100ms (milli seconds) delay duplex wired connection. Then seven mobile nodes and the base station are connected with a wireless duplex link .The sources (N0…N5) emits a number of flows towards corresponding destination (N8…N13) which traverses the bottleneck Router R and Base-station BS in between. Here the base-station which uses its adhoc routing protocol (DSDV) to route the packets to its correct destination. The traffic used File Transfer Protocol (FTP).

A lot of experiments on the above topology are carried to test the performance of High speed TCP variants with the new NRC congestion control mechanism and the results are compared with the previous high speed TCP variants such as TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas and TCP-Illinois using NS-2 with TCP-Linux modification.

4.2. TCP-Full Bandwidth Utilization

The experiment has been performed with flows of NRC-TCP, TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas and TCP-Illinois.The experiments are run for 300ms and the congestion window for each flow is measured [38].

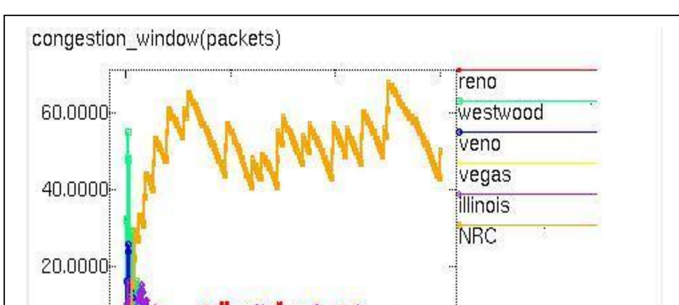

Figure 1 shows the congestion window of NRC-TCP and High Speed TCP-Variants flow, run separately. The different color lines represent NRC-TCP,TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas and TCP-Illinois. The graph shows that the NRC-TCP is able to increment its congestion window very quickly so that it can attain the whole of available bandwidth by increasing the congestion window accordingly.

4.3 TCP-Throughput

The average throughput for each flow is measured as mentioned in [39],[40].

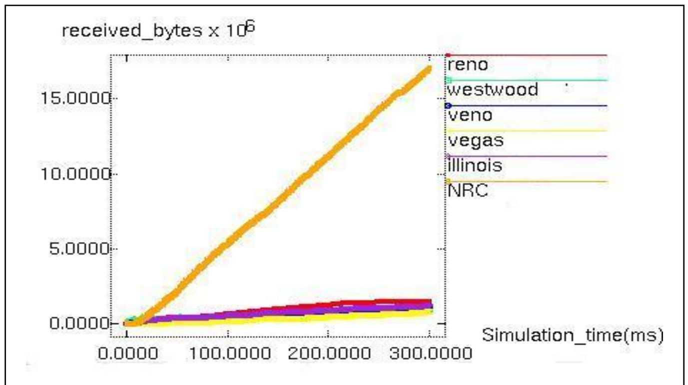

Fig. 2. Accumulated Throughputs

Figure 2 shows the comparison of the accumulated throughput. It can be seen that the mean throughput of the existing algorithms is quantity low. A better performance with the NRC -TCP Agent is observed. NRC-TCP gives an improvement in the value of mean throughout.

4.4 TCP-Packet Loss Rate

The Packet Loss Rate and Number of packets are sent for each flow is measured.

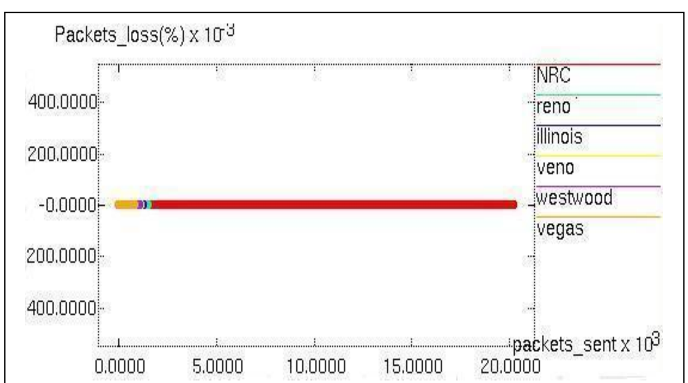

Fig. 3. TCP-Packet Loss Rate

Figure 3 shows the performance of six algorithms with congestion inserted in the network. From this graph it is found that the performance with NRC-TCP algorithm is better and its PLR is zero, even if the number of packets sent is very high compared with TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas and TCP-Illinois.

4.5 TCP- Fairness

This fairness issue has been discussed in more detail and [41] also describing the ways that packet size can affect the packet drop rate experienced by a flow. This experiment is run with flows of NRC-TCP, TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas and TCP-Illinois. The experiments are run for 500ms and the average throughput for each flow is measured between the interval [250, 500] ms. The results obtained are compared to the earlier results.

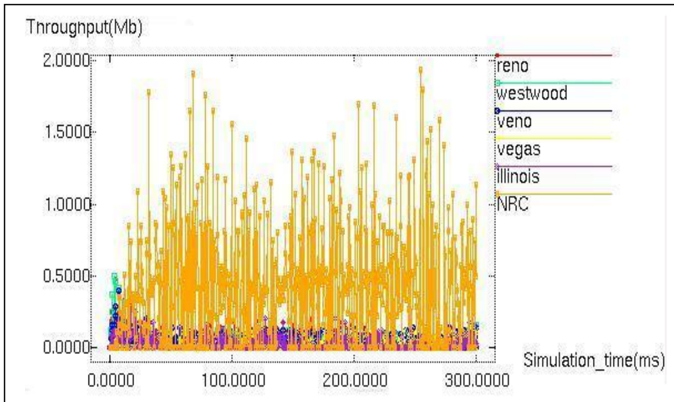

Fig. 4. TCP- Fairness

In Figure 4 shows that the experiment in which a new High Speed NRC-TCP is much fair in sharing the bandwidth compared to previous High Speed TCP Variants.

5 CONCLUSIONS

A detailed evaluation of a new friendly NRC-TCP with previous high speed TCP variants has been developed using NS2 simulator, because they are default algorithms in several standard operating systems. The outcome of the study shows that

1) NRC-TCP is able to increment its congestion window rapidly when compared to previous high speed TCP variants.
2) The mean throughput of NRC-TCP has improved
3) When compared to PLR, the present algorithm (NRC-TCP) performs better.
4) The fairness has been improved when NRC-TCP algorithm is being improved.

The above comparison shows that the present study based on NRC-TCP performs better than existing TCP variants such as TCP-Reno,TCP-Westwood,TCP-Veno,TCP-Vegas and TCP-Illinois.

REFERENCES

[1] S. Floyd ,"High-Speed TCP for Large Congestion Windows", *IETF,* RFC 3649, December 2003

[2] Tom Kelly, "Scalable TCP: Improving Performance in High-Speed Wide Area Networks", *Computer Communication Review*, April 2003

[3] D.Leith , R.Shorten , "H-TCP: TCP for high-speed and long-distance networks", *In Proceedings of PFLDnet*, Feb 2004

[4] Lisong Xu, Khaled Harfoush, and Injong Rhee1, "Binary Increase Congestion Control for Fast, Long Distance Networks", *In the proceedings of INFOCOM*, 2004

[5] Injong Rhee and Lisong Xu. "CUBIC: A New TCP-Friendly High-Speed TCP Variant", *In Proceedings of PFLDnet,* 2005

[6] J.Widmer,R.Denda and M.Mauve," A Survey on TCP-Friendly Congestion Control", *IEEE Network Magazine*,2001

[7] Shudong Jin, Liang Guo, Ibrahim Matta, and Azer Bestavros,"A spectrum of TCP-friendly window-based congestion control algorithms," Tech. Rep. BU-CS-2001-015, Computer Science Department, Boston University, July2001

[8] Y. Richard Yang and S.Simon , Lam, "General AIMD congestion control," *in Proceedings of ICNP*, November 2000

[9] M. Allman and V. Paxson, "TCP Congestion Control", Internet Engineering Task Force, April 1999, RFC 2581

[10] 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, August 1999

[11] M.Chandrasekaran andR.S.D Wahida Banu , "Performance Evaluation of Polynomial Congestion control Algorithms MIMD-Poly and PIPD-Poly in TCP networks", *Asian Journal of Information Technology*, Vol.5, No.3, pp.346-357,2006

[12] M.Chandrasekaran and R.S.D.Wahida Banu, "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] Michele C.Weigle, Pankaj Sharma, and Jeese R. Freeman IV, "Performance of Competing High-Speed TCP Flows" *In the Proceedings of NETWORKING*, May 2006

[14] Pankaj Sharma, Michele C.Weigle, "Performance Analysis of High-Speed Transport Control Protocols", A Thesis Presented to the Graduate School of Clemson University, August 2006

[15] David X. Wei, Pei Cao. "NS-2 TCP-Linux: An NS-2 TCP Implementation with Congestion Control Algorithms from Linux". In the proceedings of the workshop on NS-2, Oct, 2006

[16] Subir Kumar Sarkar, T G Basavaraju & C Puttamadappa. "Adhoc Mobile Wireless Networks -Principles, Protocols and Applications" Auerbach Publications

[17] Larry L., Peterson and Bruce S. Davie,"Computer Networks - A Systems Approach. Edition-3", Morgan Kaufmann Publishers

[18] Cheng Jin, David X Wei and Steven H Low, "FAST TCP: motivation, architecture, algorithms, performance", *INFOCOM,* 2004

[19] Ren Wang, Kenshin Yamada, M. Yahya Sanadidi, and Mario Gerla, "TCP with sender-side intelligence to handle dynamic, large, leaky pipes", *IEEE Journal on Selected Areas in Communications*, 23(2):235-248, 2005

[20] Ryan King, Rudolf Riedi, Richard Baraniuk, "Evaluating and Improving TCP-Africa: an Adaptive and Fair Rapid Increase Rule for Scalable TCP", PFLDnet 2005

[21] Sally Floyd, Metrics for the Evaluation of Congestion Control Mechanisms, August 2005, Internet draft, draft-irtf-tmrgmetrics- 00.txt

[22] David X. Wei, Pei Cao, and Steven H. Low, "Time for a TCP Benchmark Suite?", Technical report, 08/2005

[23] Hadrien Bullot, R. Les Cottrell, and Richard Hughes-Jones, "Evaluation of Advanced TCP Stacks on Fast Long-Distance Production Networks", PFLDnet 2004

[24] Yee-Ting Li, Douglas Leith, and Robert Shorten, "Experimental Evaluation of TCP Protocols for High-Speed Networks", Technial report, Hamilton Institute, 2005

[25] Okorafor N. Godfrey, Opara K. Felix, Ikpeazu S. Chukwuka "Radio Network Feedback (RNF) mechanisms for improving TCP performance over Wireless Networks" *International Journal of Academic Research*, Vol. 3. No. 2. March, 2011

[26] Balakrishnan, H; Padmanabhan, V.N; Seshan, S. & Katz, R.H. (1996) ,"A comparison of mechanisms for improving TCP performance over wireless links", *In Proceedings of the ACM SIGCOMM*, pages 256–269,1993

[27] DeSimone, A; Chuah, M. C. and Yue, O.C. "Throughput performance of transport-layer protocols over WLANs". *In Proceedings of the IEEE GLOBECOM,* pages 542–549, (1993)

[28] V. Raghunathan and P. Kumar, "A counterexample in congestion control of wireless networks," *Performance evaluation*, vol. 64, no. 5, pp. 399– 418, 2007

[29] C. Lochert, B. Scheuermann, and M. Mauve, "A survey on congestion control for mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 655–676, 2007

[30] J. Pujol, A. Toledo, and P. Rodriguez, "Fair Routing in Delay Tolerant Networks," *in Proceedings of IEEE INFOCOM*, 2009

[31] Grundy and M. Radenkovic, "Decongesting opportunistic socialbased forwarding," in Seventh Annual Conference on Wireless On demand Network Systems and Services (WONS), Kranjska Gora, Slovenia, 2010

[32] D. Bansal and H. Balakrishnan," Binomial Congestion Control Algorithms", *IEEE Infocom*, April 2001

[33] Caini, C., and Firrincieli, R. "TCP hybla: a TCP enhancement for heterogeneous networks", *International Journal of Satellite Communication and Networking*, 22, 5, 547–566 (September 2004)

[34] Fu, C. P., and Liew, S. C," TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks", *IEEE Journal of Selected Areas in Communications,* (Feb 2003).

[35] S.Liu, T. Basar, and R.Srikant, " TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks", *In Proceedings of VALUETOOLS* (Pisa, Italy, October 2006)

[36] D. Bansal and H. Balakrishnan, "TCP-friendly congestion control for real-time streaming applications," Technical Report MIT-LCS-TR-806, Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A., May 2000

[37] Ijaz Haider Naqvi and Tanguy Pérennou,"A DCCP Congestion Control Mechanism for Wired-cum-Wireless Environments", *in the WCNC* proceedings , 2007

[38] Luigi A.Grieco and Saverio Mascolo ,"Performance Evaluation and Comparison of Westwood+,New Reno and Vegas TCP Congestion control",*ACM CCR*,Vol.34 No.2,April 2004

[39] Xiaomeng Huang, Chuang Lint, Fengyuan Rent, Guangwen Yang, Peter D. Ungsunant, and Yuanzhuo Wangt. "Improving the Convergence and Stability of Congestion Control Algorithm",*IEEE*,2008

[40] Dzmitry Kliazovich and Fabrizio Granelli,"Cross-Layer Congestion Control in Multi-hop Wireless Local Area Networks" ,*Proceedings of the First International Conference on Wireless Internet*,2005

[41] Floyd, S. and E. Kohler, "TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant", RFC 4828, April 2007