Adaptive Congestion Control mechanism of TCP flows for performance optimization in mobile heterogeneous wireless networks

Muhammad Saeed Akbar, Syed Zubair Ahmad, Muhammad Abdul Qadir, Ahsan Raza Sattar*
Center for Distributed and Semantic Computing (CDSC),
Mohammad Ali Jinnah University, Islamabad, Pakistan
**Department of Computer Science, University of Agriculture Faisalabad.*
*{saeedakbar, szubair, aqadir}@jinnah.edu.pk, *ahsan_raza@uaf.edu.pk*

Abstract-- The Transmission Control Protocol (TCP) provides reliable transport services between end-points and enjoys predominant share as carrier of data traffic over Internet. Last mile wireless links have challenged TCP effectiveness because of fluctuations in link quality and relatively low data rates. These fluctuations cause major performance degradation in terms of reduced throughput, increased retransmission and weak fairness attributes in TCP. The TCP performance in wireless environment has been active research issue. In this paper, we propose Adaptive TCP variant which adjust congestion window adaptively and avoid congestion on network for performance optimization. The key contribution of the proposal lies in exchanging mobility related messages between sender and receiver on the basis of lower layer triggers to handle timeout related issues in the context of mobility constraints and adjust the congestion window adaptively with respect to link characteristics. This article presents the adaptive congestion mechanism and describes the simulation results of NS 2. The results show that Adaptive TCP provides the 10.25% to 53.75% improvement in throughput over other TCP variants in the presence of varying wireless networks conditions.

Key Words: Wireless Network, Mobility, Hand off, CWND, Adaptive TCP.

I. Introduction

Transmission Control Protocol (TCP) was originally designed for wired networks in early eighties to transport packets. The TCP has been premeditated to provide in-order, connection oriented and reliable services in the ARPANET [16] initially and for internet later. TCP has been providing this service depending upon the dynamic network load conditions through its flow and congestion control mechanisms. To achieve reliability each TCP packet is associated with a sequence number and each successfully received packet acknowledged by the receiver along with sequence number of next expected packet. A number of parameters, for example checksum, duplicate acknowledgement detection, sequence number, timers and sliding window help TCP to achieve its goals.

TCP congestion control (in order to send data as per availability of network resources) is an adaptive mechanism that is based on congestion window (CWND), transmission timeouts, packet drops, and explicit backward congestion notification (EBCN). These parameters are used in three major life cycles phases, namely; Slow-Start (SS), Congestion Avoidance (CA) and Congestion Control (CC). These phases have evolved since the birth of the protocol and have helped TCP in maintaining its lion share of over 85% over the whole Internet traffic [4]. There are other significant features of TCP such as, error recovery, fast transmission, fast recovery [10], selective acknowledgement [11] random early detection (RED) [12], explicit congestion control notification (ECN) and quick-start [14], proposed to tailor TCP as per requirement.

Despite providing successful transport services since its inception, TCP has shown signs of age due to recent evolution of wireless and mobile networks. Is TCP offer the same level of reliable and efficient transport service as it has shown for wired networks, for a number of mobile devices and the diversity of wireless link technologies and modern applications. The wireless networks have some inbuilt characteristics like high bit-error-rate, channel errors, mobility, communication asymmetry, and burst loss that challenge the traffic control mechanism of TCP and result in severe performance degradations. In wireless networks the mobility event is a most common cause for data losses. A complete window of data may be lost during the handover interval. The radio signal of transmitters can interfere with each other and signal fading affects the performance of the protocol. Network throughput and good-put degrade due to number of retransmissions required when significant packets drop occur during a mobility event. Exponential growth of RTT increases stall time that can exceed the tolerable limit of TCP [15]. Specific characteristics, behavior and mobility issue of the wireless networks motivate to re-evaluate the standard transport protocol in the context of the fast growing mobile computing and prevailing mobile and wireless networks. The rest of paper is organized as follow. Section 2 discuses existing TCP proposals for wireless networks. Section 3 presents analysis of the TCP. Section 4 describes algorithm of the proposed scheme with its components. The Section 5 presents the simulation model and related results. Finally, in Section 6, we conclude with some future research directions.

II. Related Work

Bakre et al., proposed the Indirect-TCP (I-TCP) [9] that splits the connection on the wireless edge to protect TCP session from wireless media inconsistencies and losses. This results in two different flow and congestion controls working in wired and wireless sections separately and may result in serious inequalities on the two sides. In such attempt, I-TCP eventually violates end-to-end semantics of TCP. *K. Brown and S. Sing* M-TCP [8] is similar to I-TCP and splits the connection at super host. It differs only in a way, that super host does not generate acknowledgment (Ack) of last received segment until it receives the Ack from mobile node. In this way it too breaks end-to-end semantics of the TCP and disparities of two sides flow and congestion control mechanism still exist. Another problem that may arise is that source can take inaccurate decision about receipt of all data on the basis of last byte acknowledgement. If the acknowledgment of last byte is not received, the sender resends the all data which MN already received.

M. Chinta, et al. ILC-TCP [1] presents an interlayer collaboration protocol for TCP by introducing a sender side solution for mobile and wireless network where sender node is mobile host. The basic idea of the proposal is introduction of a new layer parallel to the network protocol stack named as State Manger (SM). The SM communicates with core layers and notifies the status to network. It can handle the temporary disconnection. The ILC TCP is similar to the Freeze TCP except that it is sender side solution while Freeze TCP is a receiver side solution. So both protocol shares the same merits and demerits. SM collects information's from TCP, IP and Link/Physical and presents information to other layers.

Qi Wang, et al. Cross Layer [2] optimize the wireless network design and presents solution which breaks layering principles and presents a joint design of protocol which consists on different layers. It takes the advantages of layered architecture and presents solution for next generation networks. It breaks the basic idea of layering ordering constraints while keeping layering structures. *Raisinghani. V.T* et al. presents solution to Improving TCP performance over mobile wireless environments using cross layer feedback [3]. This paper present a cross layer approach called Link Layer ARQ Exploitation TCP (LLE-TCP). Proposed technique use feedback to exchange among the network, transport layer and between user/ application. This approach takes the advantage of network information connection \disconnection to improve the performance of TCP. In this way it tries to make the TCP network aware and presents some improvements. An agent called snoop is introduced which helps to create collaboration between link and transport layers. Therefore, the main performance advantages are achieved through the optimization of interlayer Automatic Repeat Request (ARQ) scheme functionality.

Dzmitry Kliazovich and Fabrizio Granelli, Cross-layer approach LLE-TCP [4] aims to improve performance of TCP with respect to through-put by using ARQ snoop agent. The ARQ snoop agent actually a software module which used the link layer ARQ messages to facilitates the TCP in wireless LAN environment. When packet delivered at link layer the acknowledgement not packet sent through channel ARQ snoop automatically generates the acknowledgment. Major advantage of this scheme is that it reduces the medium busy time. It presents some improvements over the channel errors which can achieve better through put. LLE-TCP present solution for WLAN and it does not testes over wireless WAN and other network. Performance of proposed scheme is compared with TCP Tahoe which is proposed for wired network environment. Comparison with I-TCP and ILC-TCP present only with respect to performance enhancement and no other parameter has been considered. LLE-TCP functionality look like split connection which violates the end to end semantics of TCP. ARQ is involved at three levels which add additional overhead. It is also required additional resources at sender side to analyze the traffics

III. Analysis of TCP

TCP has been comprehensively studied by the researchers that have helped TCP attaining many fine-tuned operations over wired networks such as flow-control, congestion control, fast retransmission, fast recovery, and selective acknowledgement etc. In this section we briefly discuss some important control parameters that help TCP in controlling it operation under varying network conditions and presents reliable in order data delivery. TCP's flow control is managed through a sliding window protocol using a window, which is controlled by receiver-side according to its receiving capacity, based on available buffers and link capacity. A receiver advertises its received window (*wnd*) on messages that move towards the sender. Sender side, on receiving the advertised window readjusts the send window, whose sending behavior is now controlled by congestion window size. The congestion window (*cwnd*) grows conservatively through a slow-start phase which grows at the rate of two segments on every acknowledged segment. A control variable, *ssthresh* is the point where the slow-start phase completes and enters into a congestion avoidance phase where *cwnd* grows linearly. In case a timeout occurs for any transmitted packet during slow-start or congestion avoidance phase, the *cwnd* drastically reduces (different approach for handling this situation for each of TCP variants), resulting in poor throughput. This is the point where TCP performance metrics are seriously reduced. In the wired links, such situation happens with less frequency and most probable cause of timeout is network congestion. In wireless network, however, such situation arises quite frequently, and is convoluted with many of the wireless specific event which has been discussed in introduction section. The segment loss event in wireless networks is much more probable as compared to its wired counterpart, and causes repeated reduction in the *cwnd* value. This also causes repeated session/connection timeouts and waste of link resources. To achieve optimum performance of TCP during mobility, especially in heterogeneous wireless environment, we believe TCP control parameter needs to re-examine and re-configure in order to meet the demands of future networks. Re-evaluation of parameters helps TCP to move in the next generation communication networks as a modern transport service for new and legacy applications. In this regard, following points are considered desirables to be achieved:

1. TCP should have the ability to take advantage of lower-layers variations and cross-layer information in order to yield high interoperability.
2. It should distinguish between congestive losses and losses due to mobility.
3. It should avoid injecting new segments during mobility events such as handover.
4. Reduce the algorithmic complexity to increase performance metrics

IV. Proposed Scheme

Proposed scheme extends our previous work Mobility Aware Transmission Control Protocol (MA-TCP) [7]. The MA-TCP consist off the handoff state identification and adaptation mechanism (HIAM) which is responsible for the receipt of the triggers from media independent handoff (MIHF) [17] and generate appropriate message. On the Mobile Node (MN) the HIAM sub module configure to intercept the two types of the MIHF events. These events include Link Handover Imminent (LHI) and Link Handover Completed (LHC). MIHF is supposed to generate these events through a continuous monitoring mechanism and uses well defined thresholds. When MIHF received LHI event at HIAM it sets mobility flag on to indicate possible mobility decision in near future. There is a possibility that this event is only generated because of a possible fading channel or change in direction of movement. MIH Link layer Handover Imminent event received at the HIAM provides a solid confirmation of handoff and on its arrival, HIAM generates a message handoff initiated (HoI) for the sender to adjust its control parameters such as Timeout, Flow Control and congestion control which shall be discussed later. Other important factor related to the transmission of such message relates to the possibility of an extra message or piggybacking the same on an Ack message. In case no Ack is pending to be sent then a separate HoI message shall be generated that comprises of only 20 bytes of header. The second type of trigger is MIH Link layer Handoff Completed. This event is indication that link layer handoff is complete with new point-of-attachment so HIAM transmits a Handoff Completed (HoC) message for the sender. This message has been synchronized with the Mobile IP (MIP). The updation of its registration process during this process all traffic is tunneled through MIP.

Figure 1 presents functionality of protocol on receipt of network parameters and it executes relevant procedure. On receipt of mobility event it executes the MobilityHandler. The functionality of MobilityHandler describes in the figure 2. One added function of MobilityHandler is to reduce the congestion window size during the mobility event.

```
IF ACK && ECN != 1 && mobility bit != 1 then proceed as normal
IF ACK && ECN = 1 && mobility bit != 1 then Execute congestion algorithm
IF HoI then Execute MobilityHandler
```

Figure 1: Abstracted model of MA TCP Algorithm

```
MobilityHandler {
StartTimer (HoT)
PTimeout = Timeout
Timeout ←Timeout + HoT}
cwnd= cwnd*3/4
```

Figure 2 Mobility Handler Routine

Figure 3 presents the adaptive congestion window mechanism for heterogeneous wireless networks. It adjusts the value congestion window by monitoring the value of RTT on receipt of each acknowledgement. On the receipt of each acknowledgment new RTT calculated based on the previous history of the RTT values. It adds value of previously received RTT and adds the last three RTT values. If there is any increasing trends in the value of RTT then value of congestion window is reduced by one on receipt of each RTT. If there is a decreasing trend in the value of RTT on the receipt of each acknowledgment then congestion window increase by 2. The value of RTT is calculated based on the average value of RTT which is calculated based on average of present and previous three RTT values. If the average value of RTT remains same as the previous average value of RTT then congestion window will remain as the previous value of the congestion window.

Current = $'$

Previous = without $'$

Let P1, P2…………, Pn, Pn+1 be the transmitted packets where 1, 2,………, n, n+1, n-1 are sequence numbers

When RTT Pn+1 is received and the there is decreasing, increasing trend or RTT remain smooth the congest window calculated as follow.

Adaptive Congestion Control Mechanism

If

$$RTT' = (rtt_{n+1} < rtt_n < rtt_{n-1})$$

$$cwcd' = cwnd + 2$$

Elseif

$$RTT = (rtt_n > rtt_{n-1} > rtt_{n-2})$$

$$cwnd' = cwnd - 1$$

Else

$$cwnd' = cwnd$$

Figure 3: Adaptive Congestion Mechanism Procedure

V. Results and Discussion

The simulation study has been performed on wired-cum-wireless networks through NS2 which is a discrete-event simulator. The simulated network consists of basic mobility support components such as Home Agent (HA) and three Foreign Agents (FA's). Both sender and receiver may either be mobile, which means that either sender or receiver may operate on mobile node. In principle, both sides may also be mobile but such scenario has not been evaluated in this work. The mobile node roams between HA and different FA's. It performs multiple handovers during simulation time of 60 seconds. We perform the simulation study of Reno, New Reno, SACK, FACK, Vegas and performed comparison with proposed Adaptive TCP. The simulation parameters for network are as follow. Link capacity is 5 Mb and processing delay is 10 milli-seconds. Traffic pattern used is constant-bit rate (CBR) which generates packets after every 20 milli-seconds for all variants. The MN moves at speed of 5 m/s during the entire simulation time. Random mobility model has been used for fair distribution of node density across the domain. It's important to note here that we only focused on the isolation of congestion events from mobility events, hence other related attributes such as fairness and other inter-variant operational issues were not studies and lie outside the scope of this study. The simulation scenario configures in a way that there may be negligible congestion and available bandwidth is sufficient to maximize the performance of each protocol in mobile environment. The study focuses on parameters like behavior of congestion window, throughput response, end-to-end delay, and impact of mobility-events. Congestion window behavior and end-to-end delay can describe internal behavior of TCP under varying network conditions and mobility-events, where as throughput and end-to-end delay provides external behavior related to performance metrics.

Figure 4, 5, and 6 presents end-to-end delay performance comparison of Adaptive TCP with are well known TCP variants. In figure 4 it is noticeable that Adaptive TCP has end-to-end delay, fluctuating in a narrow range of 0.05 to 0.25 seconds where as Reno has a much wider range of 0.08 to 0.49 seconds. Similarly other variants also show higher end-to-end delay as compared to the Adaptive TCP. Figure 5 shows a plot of end-to-end delay of Adaptive TCP and Vegas with results slightly in favor of Adaptive TCP. Vegas end-to-end delay is in the range of 0.08 to 0.3 which is much better than Reno. Figure 4 shows a plot of end-to-end delay of Adaptive TCP and Tahoe showing superior performance of Adaptive TCP. The most important aspect of these graphs relates to the extended delay line of three variants (Reno, Vegas, & Tahoe) that indicate slow rate of data packets flow and higher flow rate of Adaptive TCP, only possible due to the continuous operation of Adaptive TCP which could not be possible for the Reno, Tahoe, and Vegas due to handover discontinuities. These plots can be divided into three regions namely; initial start-up region, then a partially-mobility region and finally full mobility region. In the startup region, all the variants are trying to achieve optimum state and various parameters are in adjustment stage to attain stable values. For example the timeout is adjusted every time a new Acknowledgement has arrived. The second region reflects a stable region where TCP has attained stability and the session does not face high rate of mobility events like handovers though terminals may be mobile. The third region represents major impact of mobility events and correspondingly a highly volatile end-to-end delay pattern. Despite high volatility of end-to-end delay, the variations remain range bound within 0.25 seconds. This aspect is specifically visible in the Adaptive TCP end-to-end delay plot.

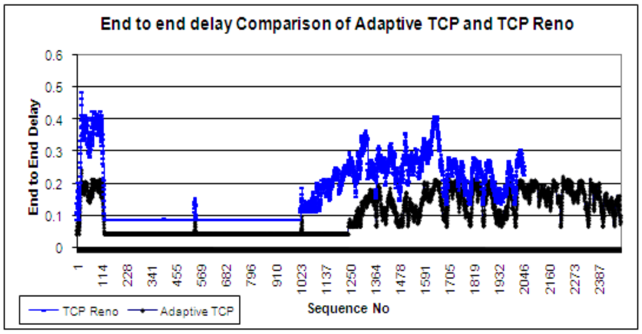

Figure 4 : Comparison of Reno and Adaptive TCP End to End Delay

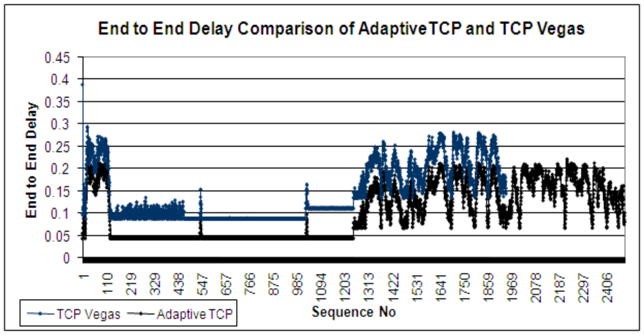

Figure 5: Comparison of Vegas and Adaptive TCP End to End Delay

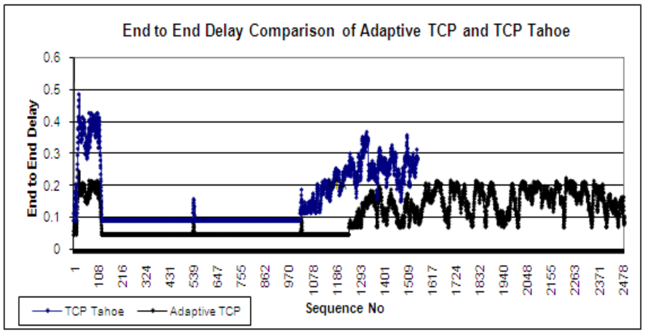

Figure 6: End to End Delay of Adaptive TCP and Tahoe

Figure 7, 8 and 9 presents the congestion windows behavior of Adaptive TCP in comparison to each of TCP variants under study with 0 and 2 percent loss rate. The most important aspect of these plots is that Adaptive TCP cwnd is decoupled from handover events. It can be noticed that all variants Reno, New Reno, Vegas, SACK, FACK and Tahoe cwnd values drop at the same time indicating packet drops at handover events. In contrast to this, Adaptive TCP cwnd does not drop at the handover events. It can be seen that at time around 15.8 seconds of simulation run Adaptive TCP cwnd value still extends to a value of 2500. Since in our proposed scheme, Timeout has been extended for an approximated handover time, congestion window does not grow rapidly but sustains an already achieved value. The random drop events do cause decline on the cwnd size as per defined model, but this is not directly related to handover event. There is a possibility that the approximated handover delay added to TCP timeout may still not be enough to cover the actual handover time and result in a packet drop. The rest of peaks in graph further highlight this specific aspect of the proposed scheme. After the initial settling time, Adaptive TCP shows a consistent growth in cwnd proving the strength of the scheme in the mobility scenario. More significant impact of proposed scheme is visible in the plot of Adaptive TCP with Vegas where Vegas remains subdued with respect to achieving a reasonable cwnd size. This is primarily due to the defensive readjustment of cwnd in Vegas within a tight-bound limit. The bounding value of α and β are calculated in an adaptive manner that does not encourage a larger value for both, specifically in mobility scenario. This also indicates that performance of Vegas may be seriously degraded in any hostile environment with respect to either mobility of multiple sessions or high density of TCP variants executing mobile nodes.

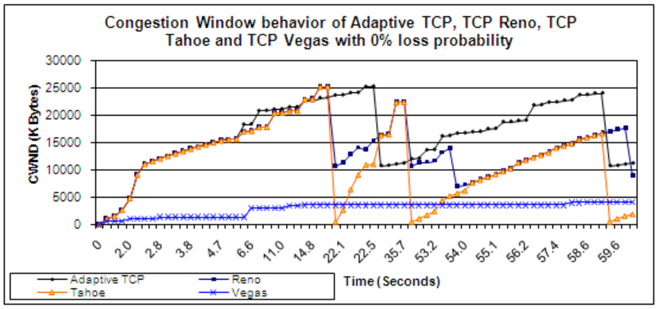

Figure 7: Congestion Windows behavior of Adaptive TCP, Reno, Tahoe and Vegas with 0% loss probability

The throughput comparison of schemes under study is presented in figure 10. The throughput can best describe the performance of a specific scheme under given set of operating conditions. It is anticipated that a good transport protocol is one that can optimally utilize available data-rate provide applications, their desired data-rates and network resources with other competing transport sessions. Importance of reliable data transport is even more meaningful in mobile environment; since in the absence of such provision, service provision may become unpredictable and even in some cases applications may face starvation during mobility events. Initially when MN remain within the range of the HA the Adaptive TCP present 10.26, 15.4 18.23, 22.45 and 28.9 percent improvement over the FACK, SACK, New Reno, Vegas and Reno respectively. During mobility and handover events overall throughput of each variant degrades with exception of Adaptive TCP which shows a robust performance by showing 15.23%, 17.21% 20.38%, 25.34% and 35.05% improvements over FACK, SACK, Vegas, New Reno and Reno respectively. The Adaptive TCP proves its dominance over all competitors even when MN performs the multiple handovers and changes its point-of-attachment. The throughput improvements over FACK, SACK, New Reno, Vegas and Reno are 22.71, 21.23, 32.71, 38.56 and 53.55 percent respectively. It is also observed that Adaptive TCP throughput also degrades up to some extent due to the mobility events. The major cause of throughput increase is primarily contributed due to lower end-to-end delay values as we saw in the figure 4, 5, and 6. Similarly attaining a better average cwnd is another important reason for throughput improvement. Further it is noticed that Adaptive TCP handles mobility events more efficiently and prove its strength over its counterparts.

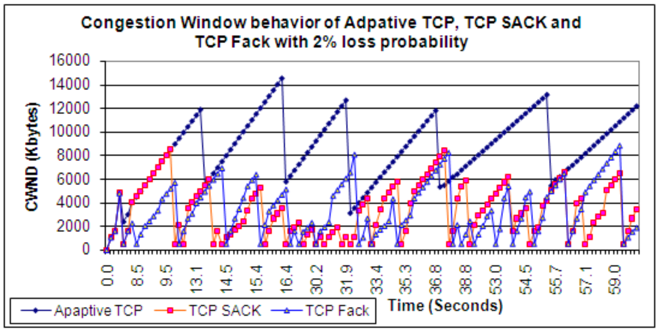

Figure 8: Congestion window behavior of Adaptive TCP, SACK and FACK with 2% loss probability

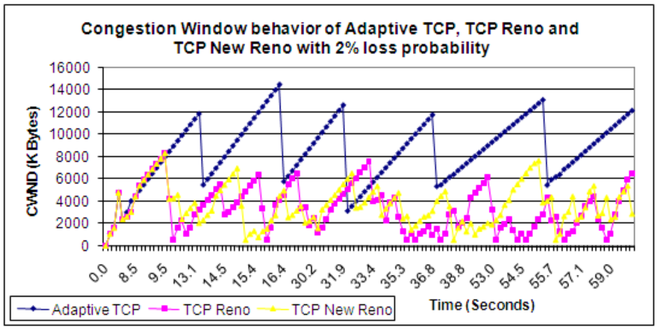

Figure 9: Congestion Window behavior of Adaptive TCP, Reno and New Reno with 2% loss probability

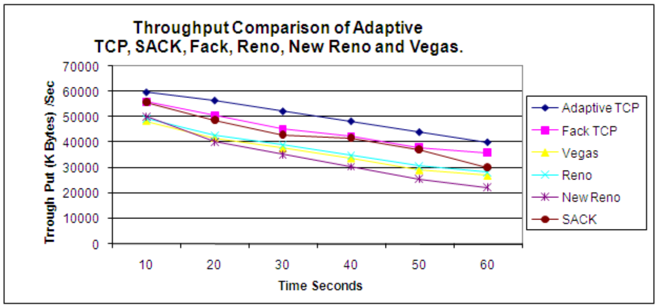

Figure 10: Through put comparison of Adaptive TCP

VI. Conclusion

In this paper, we proposed a new congestion control algorithm of TCP for heterogeneous wireless network. Adaptive TCP has an embedded support for mobility handling and adjusting its congestion window according to underlying link and network conditions. In this proposal, we have added congestion control algorithm adjust congestion window adaptively because congestion over network is major setback for TCP performance and network resource utilization. It has potential of taking advantage of any lower layer improvements of link utilization, bandwidth aggregation; which can increase traffic reliability of traditionally unstable radio access networks (RAN). It presents a model that stands on standard technology components rather than individual models for different type of RANs, making it more interoperable in complex heterogeneous wireless environment.

The simulation results show that the Adaptive TCP is a viable solution which improves the performance in wireless networks. The results show that the Adaptive TCP performs better with other TCP variants and achieves 10.26% to 53.55% improvement in throughput in congested network and with frequent host mobility. It also shows some improvement over other variants by providing low end-to-end delay. The future research directions are to improve the Adaptive TCP with respect to the wireless network conditions which cause for performance degradation. These wireless network conditions are random losses, burst losses, signal quality and channel fading.

References

[1] M. Chinta, A. Helal, and C. Lee, "ILC-TCP: An Interlayer Collaboration Protocol for TCP Performance Improvement in Mobile and Wireless Environments," Proc. IEEE WCNC 2003, vol. 2, New Orleans, LA, Mar. 2003, pp. 1004–10.

[2] Qi Wang, Abu-Rgheff, M.A., "Cross-layer signaling for next-generation wireless systems," IEEE Wireless Communications and Networking Conference (WCNC 2003), Vol. 2 , pp. 1084-1089, 16-20 March 2003.

[3] Raisinghani, V.T.; Singh, A.K.; Iyer, S, "Improving TCP performance over mobile wireless environments using cross layer feedback," IEEE International Conference on Personal Wireless Communications, pp. 81–85, 15-17 Dec. 2002.

[4] Dzmitry Kliazovich and Fabrizio Granelli, "A Cross-layer Scheme for TCP Performance Improvement in Wireless LANs," IEEE Communications Society Globecom 2004.

[5] M. Saeed Akbar and M. Abdul Qadir, "Optimization of Transmission Control Protocol (TCP) Parameters for Highly Dynamic Link Characteristics of Mobile Wireless Networks", Doctoral Symposium on Research in Computer Science, August 9-10 2008 University of Central Punjab Lahore Pakistan.

[6] M. Saeed Akbar, Syed Zubair Ahmed and M. Abdul Qadir, "Performance Optimization of Transmission Control Protocol in Heterogeneous Wireless Network during Mobility", IJCSNS VOL.8, No.8, August 2008 (In press)

[7] M. Saeed Akbar, Syed Zubair Ahmed and M. Abdul Qadir, "Mobility Aware Transmission Control Protocol (MA-TCP)", the Fourth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2008) June 11-13, 2008 Miraikan, Tokyo, Japan. PP 146-151.

[8] K. Brown and S. Sing, "M-TCP: TCP for Mobile Cellular Networks," ACM SIGCOMM Computer Communication Review Volume 27, Issue 5 (October 1997) Pages: 19 – 43, 1997.

[9] Bakre, B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Host" Proceedings of the 15th international conference on distributed computing system (ICDCS '95).

[10] W. Stevens, "TCP slow start, Congestion avoidance, Fast retransmit and Fast recovery algorithms," RFC 2001 January 1997.

[11] M. Mathis et al., "TCP selective acknowledgment options" Request for Comments, RFC 2018, Network Working Group, Internet Engineering Task Force, October 1996

[12] S. Flyod, V. Jacobson, "Random early detection gateways for congestion avoidance" IEEE/ACM Transaction on Networking, vol. 1, no 4, Aug., 1993.

[13] S. Floyd and T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm," Request for Comments, RFC2582, Network Working Group, Internet Engineering Task Force, April 1999.

[14] S. Floyd, M. Allman, A. Jain, and P. Sarolahti. Quick-Start for TCP and IP. RFC 4782, January 2007.

[15] Y. Tian, K. Xu, and N. Ansari. TCP in Wireless Environments: Problems and Solutions. IEEE Radio Communications Magazine, 43(3): S27–S32, March 2005.

[16] R. S. Prasad, M. Jain, and C. Dovrolis. On the effectiveness of delay-based congestion avoidance. In Proceedings of Second International Workshop on Protocols for Fast Long-Distance Networks, 2004.

[17] S. Schuetz, N. Koutsianas, L. Eggert, W. Eddy Verizon, Y. Swami and K. Le, Response to Lower-Layer Connectivity-Change Indications draft-schuetz-tcpm-tcp-rlci-02, November 19, 2007, Expires: May 22, 2008.