Performance Optimization of Transmission Control Protocol in Heterogeneous Wireless Network during Mobility

Muhammad Saeed Akbar, Syed Zubair Ahmed, Muhammad Abdul Qadir
Center for Distributed and Semantic Computing (CDSC),
Mohammad Ali Jinnah University, Islamabad, Pakistan

Summary
Transmission Control Protocol (TCP) has shown serious performance limitations on wireless media. The main causes of these limitations have their root in mobility and link characteristics of wireless media. Heterogeneous environment of multiple radio-access technologies generally increases handover frequency, causing adverse impact on TCP flow and congestion control operations. In this paper, we have provisioned handover state management mechanism by adding new states in TCP control operations. Performance optimization has been achieved by adding adaptive flow control, based on mobility events and by reconfiguration of TCP control parameters according to their criticality and impact on performance parameters. Proposed component also contains functionality of end-to-end Handover-state Identification and Adaptation Mechanism (HIAM), which takes care of both sender and receiver mobility events. The results show that optimization achieved through proposed scheme provides 10.26% to 53.55% improvement in throughput over other TCP variants. Similarly, proposed scheme also improves end-to-end delay for each packet from 15.23 % to 35.05%.

Index Terms: *Congestion control, mobile TCP sessions, Handoff-delay absorption, Wireless-loss absorption, Explicit Congestion Notification.*

1. Introduction

TCP is a reliable end-to-end transport protocol which has high utility over Internet. It is used by main-stream application layer protocols like *http, ftp, tftp, smtp,* etc which are source of more than 80% of Internet traffic [25]. The design goal of TCP has been centered on both user as well as network friendly service operation through its flow and congestion control, error-control, retransmission strategies and state preservation mechanisms. A number of TCP variants have been proposed such as Tahoe [21] , Reno [23], Vegas [20] and SACK [22], which is already, successfully deployed. These variants improve the performance of TCP by adding new services such as selective-acknowledgements (SACK), fast-retransmit, fast-recovery, Explicit Congestion Notification (*ECN)* and Random Early Detection (*RED*). Historical evolution of TCP flow and congestion control has optimized it into a highly reliable transport service, but has caused serious performance degradations in less-reliable, low data-rate wireless network environment. The performance of TCP degrades over wireless links due to high rate of data losses, which are falsely perceived as network congestion state. TCP performance metrics also diminish due to low data-rate, since large delays may occur in last link i.e. wireless link. Similarly in heterogeneous wireless network, packet-loss may also occur due to mobility-events that can cause burst-losses, service-disconnection. This motivates to re-evaluate TCP control operations and embed some mobility related services to optimize its performance for new generation of wireless networks.

The article optimizes TCP for mobile, wireless environment with the help of link-layer triggers which shall be standardized by IEEE 802.21 standard, through its media independent handover (MIH) services [26]. These services are used to detect link layer events in the TCP control operations, and act accordingly to adjust flow and congestion control in a way that does not seriously hamper TCP performance. The rest of paper is organized as follow. Section 2 discuses existing TCP proposals for wireless networks. Section 3 describes architecture and algorithm of the proposed scheme with its components. The Section 4 presents the simulation model and related results. Finally, in Section 5, we conclude with some future research directions.

2. Related Work

The performance attenuation of TCP metrics on wireless links has motivated researcher to find ways and means to adopt TCP in changing link-layer parameters. This has resulted in numerous TCP variants that attempt to optimize TCP performance over wireless networks. Despite high-proliferation of such schemes, none of these proposals have been widely adapted for general purpose use, to the best of our knowledge. These schemes are mostly restricted to test-bed environments. In this section, we summarize some of main proposals in this domain.

Bakre et al., proposed the Indirect-TCP (I-TCP*)* [18] 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

Manuscript received August 5, 2008.
Manuscript revised August 20, 2008.

result in serious inequalities on the two sides. In such a attempt, I-TCP eventually violates end-to-end semantics of TCP as well. The M-TCP [17] 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 the 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.

The TCP Westwood (TCP-W) and TCP Westwood + (TCP-W+) [3][4] are true extension of TCP Reno (Reno). TCP-W uses the bandwidth estimation as a parameter to control the congestion window and slow-start (SS) threshold (*ssthresh*). The protocol separates the congestion control from the error control. This mechanism helps it to improve performance over lossy wireless links. Despite, some advantages, TCP-W is unable to presents its preeminence over Reno when higher packet losses occur or the blackout time is longer. TCP-Peach (TCP-P) [6] was developed for the satellite network to overcome long propagation delays and elevated link error rate. It uses two new algorithms; namely *sudden start* and *rapid recovery* to overcome the limitations of TCP in satellite network. It also uses dummy packet to estimate the available network resources. The successful delivery of the dummy segments indicates better resource availability and transmission rate can amplify. The TCP-Peach+ (TCP-P+*)* [7] further improve the network utilization by sending *Nil* packet instead of dummy packet. It also replaces *sudden start* and *rapid recovery* algorithms by *jump star*t and *quick start. Jump start* is extension of *sudden start* and it replaces dummy segment with *Nil* segment. TCP-P+ presents improved performance as compare to TCP-P.

C. P Fu et al., presented their work with the name; TCP-Veno (Veno) [1] showing improvements by introducing sender side solution, by combining the feature of the Reno and TCP Vegas (Vegas). The Reno and Vegas use reactive and proactive congestion control strategies respectively. Veno integrates the congestion detection mechanism of Vegas with Reno to distinguish between the congestive and non-congestive loss. Veno efficiently deals with single packet loss but it suffers form performance degradation when multiple packet losses occur in the network. The reason behind this degradation is the continuous reduction of window size on successive loss. *Kai Xu et al.* proposed a sender side solution named; TCP-Jersey (Jersey) to improve the TCP performance in heterogonous wireless networks [2]. Jersey adopts the similar idea of TCP-W to estimate the bandwidth at sender side by monitoring arrival rate of *Ack*. Jersey uses the *Available Bandwidth Estimator (ABE*) at sender side to estimate the bandwidth and re-computes the *cwnd* on every *rtt*. It uses *ECN* for identifying congestion in the network. For further tuning of congestion control, Jersey also uses two parameters, namely; *intervals of jitter (IJ)* and *jitter ratio (JR)* to determine the ratio of congestion on the network. Freeze TCP (F-TCP) [8] proposed a recipient side solution which improves the performance of the transport protocol TCP in networks where handoffs and disconnections are more common and significant. If MN detects the low quality signal it sends *zero-window-probes (zwp)* with *Ack* to the sender. On the receipt of *zwp* sender freezes the timers and waits for positive *zwp*.

M. chinta et al., presented an inter-layer collaboration protocol for TCP (ILC-TCP*)* [9] by introducing a sender side solution for mobile and wireless network. ILC-TCP introduces 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. TCP Delayed Congestion Response (TCP-DCR) [12] also improves congestion handling mechanism for wireless networks. It identifies the congestion by sensing retransmission timeout or receipt of DUPACKs. It also suggests modification in calculation of retransmission timer. On the receipt of duplicate acknowledgement, it introduces the bounded delay period τ and sets it equal to one *rtt*. This helps the sender to recover the link-layer loses otherwise the retransmission algorithm is activated. Adaptive Delayed Acknowledgement (TCP-ADA) [11] is also a sender side solution for mobile ad-hoc networks. It uses Delayed Acknowledgement (DelAck) to block acknowledgement for a specific time period. The actual proposal for DelAck [13], blocks acknowledgement until specific number of data packets has been received.

Chandran et al., proposed a router assisted solution named as; TCP-Feedback (TCP-F*)* [10][9] for Ad hoc wireless networks. TCP-F detects route failures in case an intermediate node move or fails the next node detects route fail. When route failures detected the next node transmits *route failure notification (RFN)* packet to destination. On the receipt of *RFN* packet each node invalidates the previous route and stops incoming packets. If the alternate path does not exist each node relays the *RFN* packet to source node.

David X. Wei et al., [16] presents a TCP congestion control variant for high speed and high latencies suffering networks. It highlights the four difficulties which TCP faces in high speed networks. It devised a mathematical and analytical model of FAST-TCP. The results show that FAST-TCP shows some improvements over other TCP variants with respect to the throughput and fairness. *M. Todorovic et al.* presented a study of performance comparison of four protocols [15], namely; TCP-W, Jersey, Veno and Jitter TCP (J-TCP) [5]. The simulation study focuses on the three parameters, namely; throughput,

Average congestion window and Time to complete a file transfer. The study is performed in presence of random losses and author highlights that J-TCP performance better in the presence of random losses. The performance of the TCP-W is better in the LAN and WAN environment. The performance of J-TCP is better during random and disconnection packets losses but throughput degrades while competing with other flows. *J. Liu et al.* presents ATCP [14] for mobile ad hoc networks. ATCP uses the ECN and ICMP to put the sender in one of four states such as normal congested, loss, and disconnected. Beside some improvement over ad hoc networks it unable to prove strength in situation where blackout and disconnection are frequent.

3. TCP Review

TCP has been keenly studied by the researchers that have helped TCP attaining many fine-tuned operations such as flow control, congestion control, fast retransmission, fast recovery, and selective acknowledgement etc. In this section we briefly study some control parameters that help TCP in controlling it operation under varying network conditions. 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

4. Proposed Technique

TCP is the choice of majority of legacy applications requiring elastic service model, due to its highly reliable transport facility. These applications shall continue in the mobile computing era as well due to their wide scale deployment and utility. The discussion given in the last section highlights the need of adaptive provisions in TCP to compensate for the link-layer data-rate fluctuations and disconnections. In this section we propose a scheme that can robustly adapt to above mentioned link conditions and minimizes impact of mobility and wireless events. The main theme of the scheme is to minimize timeout events of TCP sessions resulting in reduced probability of segment loss that helps in maintaining *cwnd* value reasonably less perturbed, during mobility. The architectural view of the proposed scheme is presented in figure 1. We introduce a sub-module Mobile TCP (Mo-TCP) in core TCP module. This module consists of Handoff-state Identification and Adaptation Mechanism (HIAM). It is responsible for the receipt of Media Independent Handover Function (MIHF) triggers and generates appropriate message for the Mo-TCP module. On Mobile Node (MN) side, the HIAM sub-module can intercept MIHF triggers like link_going_down, link_detected, handover_imminent, handover_command and link_up. These triggers are transformed to the sender as Handover_Initiated (HoI) or Handover_Completed (HoC) messages. On the receipt of HoI message, the sender starts a handover timer (nHoT) and adds previously calculated (HoT) into current timeout value. Sender preserves the present parameter of connection and on receipt of HoC message stops presently running nHoT timer, estimate HoT value by adding weighted contribution of nHoT time. This helps in maintaining a realistic estimate of a probable next handover event on the basis of historical data.

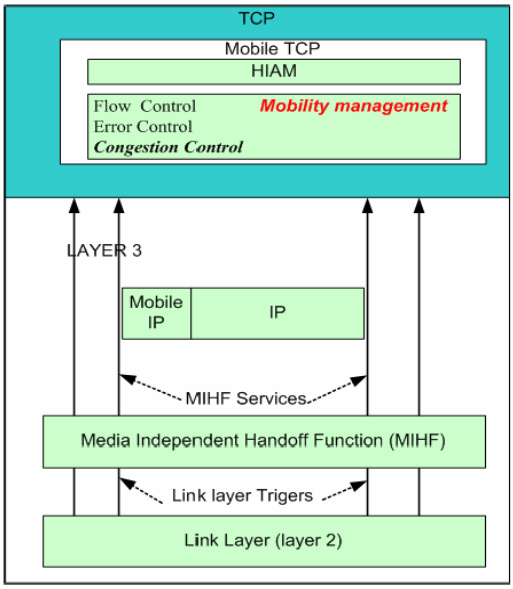

Figure 1: Architecture of the proposed scheme

4.1 State Model

The proposed scheme is not based on a specific side of the two communication ends, rather it can cope mobility at both ends. The functionality at two ends is different. These differences of functionality at two ends are highlighted in the state-transition diagrams as shown in figure 2 and 3. The sender-side state model consists of five states namely; TCP, Mo-TCP, Handover Wait (HoW), Probe HoW (P-HoW) and Segment Lost (SL), as shown in figure 2. A brief description of these states is given next. TCP is the most common state in which system remains under normal, non-mobile conditions. It's particularly of interest that the handover events are the core mobility events and some wireless signal condition events don't lie in scope of this paper. The MIHF events if generated are intercepted and analyzed for necessary action. The main interest in this state, as a sender lies in handover initiated (HoI) message received from receiver-side which takes system into Mo-TCP state. This message further takes system into Handover Wait (HoW) state which is locked with a handover timer (HoT). At this point the normal timeout value of TCP is also extended to a value of HoT. In case a HoC message reaches the sender within HoT, system comes back to the Mo-TCP state where normal TCP state is restored and full rate transmission is possible. In case HoC does not arrive with in HoT, a probe message with timeout value of HoT is sent to the receiver for getting the connectivity status. This further extends the normal timeout of TCP for another HoT value. In case HoC message arrives within the response time HoT of P-HoW message, the system is restored for normal TCP operation, otherwise a segment drop event of TCP is raised at the completion of response time of P-HoW.

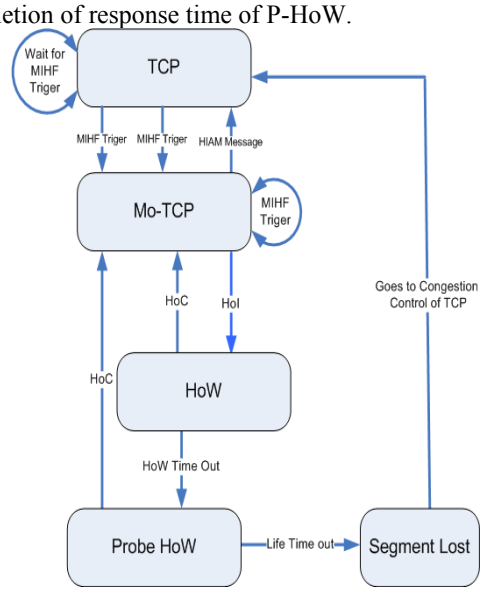

Figure 2: Handoff states transition of Mo-TCP at the sender side

Figure 3 describes the receiver-side state model of Mo-TCP, which comprises of two states; namely, TCP and Mo-TCP. TCP state represents normal TCP operation where as, a mobility event takes this state to the Mo-TCP state. In this state Mo-TCP, generates HIAM messages which are presently two namely; HoI and HoC. There is no wait timer in this state model due to the event-driven architecture of system.

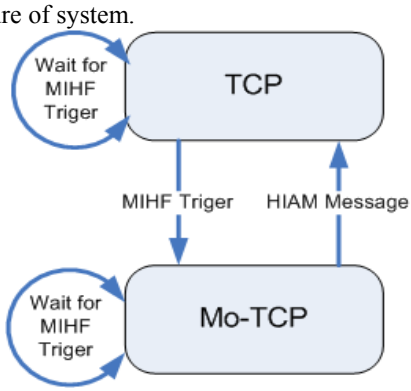

Figure 3: Handoff states transition of Mo-TCP at the receiver side

Figure 4, 5 describe the relationship between the timeout and HoT with effect of these on connection parameters. When the HIAM on MN side receives the MIHF trigger handover_imminent the HIAM module generate HoI message. As Sender receive HoI message, it preserves the present connection parameters and starts a timer HoT. The cwnd remains same because sender extends the timeout for transmitted packets. In response of HoI message the sender also sets mobility status true to distinguish between static

and mobile state of session. It also helps in avoiding unnecessary communication during handover. The MN side of Mo-TCP generates HoC on receiving a link_up event. Sender side, on receiving HoC message, may face following possible cases:

Case 1: HoC arrives before HoT expires

This is the case when the HoT value shall be reduced due to a faster handover completion. The normal TCP session shall be restored. Mobility flag is cleared for full rate TCP operation.

Case 2: HoC does not arrive within HoT expires

In this case a probe message shall be sent to the mobile side regarding the HoC with a response time constraint of pTimeout. There are two purposes for this message mainly; to check connectivity status of other side, and second to extend segment timeout to another timer slot. The obvious reason for further wait is the adverse impact that session may face on a timeout event. Mobility flag is not cleared.

Case 3: HoC arrives in response to probe message

This possibility arises due to partial link connectivity at the MN side or segment lost event might have occurred at the handover completion. Congestion may also be a cause of this case. In this situation pTimeout is also included in handover completion time and HoT is estimated accordingly. Mobility flag is also cleared.

Case 4: HoC does not arrive even in response to probe message

In this case a timeout event is allowed to occur to readjust *cwnd* accordingly. Mobility flag is not cleared.

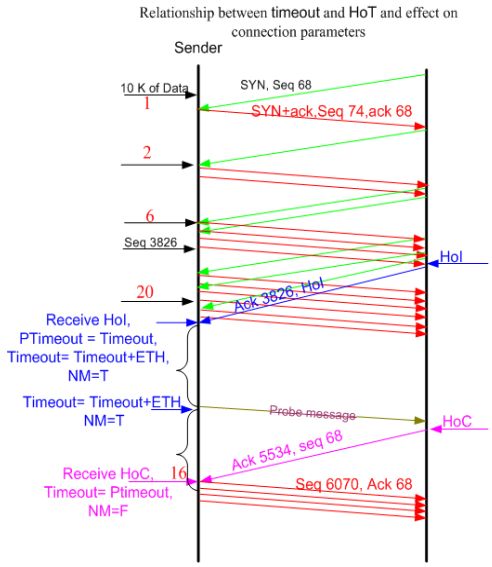

Figure 4: Timing Diagrams for Connection Time Out and Handoff

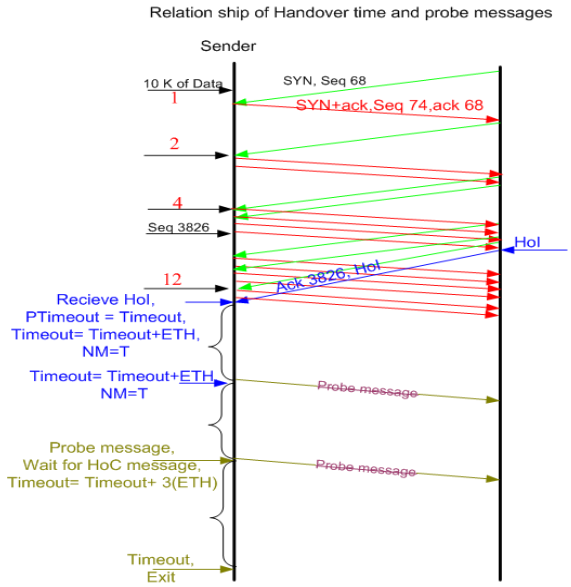

Figure 5: Timing Diagram for Hand off Time Calculation, expiry of HoT and behavior of sender

4.2 Algorithm

The proposed algorithm suggests some modification in the header of the TCP. It uses two reserved bits of the TCP segment to incorporate the mobility messages. The bits no 6 and 7 are reserved for future in TCP packet header, as given in [28]. We use the bit no 6 for indication of the HoI and bit 7 for HoC. If sender receives *Ack* with bit no 6 is enabled the sender determine that the MN is in mobility state and performing the handoff. If sender receive *Ack* packet with bit no 7 is enabled it is indication that MN completes the handover and now can receive packets. In case if no *Ack* is pending on MN side for sender then a separate HoI or HoC message shall be generated that comprises of only 20 bytes of header. On the receipt of HoI or HoC message sender proceeds according to the mobility algorithm. The mobility algorithm consists on five condition of the Mo-TCP. If the MN with in the range of HA and sender received the *Ack* it precedes as follows. If timeout occurs or dupack received and the *ECN* bit is enabled the sender executes normal TCP operations. In case, timeout occurs and *ECN* not enabled in previous *Ack* sender checks mobility bit. If mobility bit enabled then sender executes mobility algorithm. If timeout occurs with both *ECN* and mobility bits are not enabled then sender should retransmit the packet.

One important component in proposed model is MIHF, which sends mobility events to facilitate HIAM module of Mo-TCP to take mobility related control decisions. Mo-TCP module transforms MIHF triggers into mobility messages and optimizes protocol for mobility events which

is presented in figure 6. On receipt of two consecutive layer 2 triggers MIHF generate trigger link_Going_Down which is indication that handoff is about to occur in near future and it is precautionary information [26]. When MIHF generates a trigger handover_imminent HIAM module generates HoI message for sender. On receipt of HoI message sender preserve value of present Timeout and increase Timeout for transmitted packets. It also changes MN status from static to mobile.

```
If (ACK) then Proceed Normal
If (Timeout !! dupack && ECN = 1 ) then Proceed Normal
If (Timeout && ECN ! = 1 && Mobility=1)
        Execute Mobility Algorithm
Else
        Proceed Normal
End if
```

Figure 6: Mobility Algorithm

On the receipt of HoI message on sender side it change state from normal to HoW. Our previous work [24] describes the major steps taken by the sender side on the receipt of mobility event (HoI) shown in figure 7.

```
Mobility handler
{
StartTimer (HoT)
pTimeout = Timeout
Timeout ←Timeout + HoT
}
```

Figure 7: Proposed Timeout extension

In case, sender receives HoC message before the expiry of HoT, it changes state form HoW to normal and restores connection parameters to pre-HoI values. Handoff initiated procedure is presented in figure 8.

```
HandoffInitiated
{
Start timer pHoT
pTimeOut = TimeOut
Timeout = Timeout + HoT
Mobility = True
}
```

Figure 8: Handoff Initiated

In case HoC message is received within HoT, normal TCP session is restored. If sender receives HoC message from receiver side, the HIAM module of sender stores elapsed HoT value and stops timer. On the completion of handover the HoT value updated and used as reference value for upcoming mobility events. Sender also changes the mobility status of mobile node equal to false. It updates the value of timeout which is equal to value of pTimeout. The figure 9 describes the procedure of HoC event and presents its functionality.

```
HandoffCompleted
{
HoT = ElapsedTimerValue
Stop_Timer
TimeOut = PTimeOut
}
```

Figure 9: Handoff Completed

A situation in which HoT expires and sender does not receive the HoC message can be handled by algorithm shown in figure 8. As discussed above, on receipt of HoI message sender extends Timeout for transmitted packets to avoid unnecessary retransmissions during the mobility events. If Timeout expires and sender does not receive HoC message, the following algorithms show in figure 10 performs the remedial action.

```
HoITimeoutExpire
{
Start Timer
Timeout = TimeOut + HoT
SendProbe message
}
```

Figure 10: HoI Timeout Expire

If sender side receive HoC message before expiry of extended HoT, normal TCP session is restored. On receipt this message, sender stops the timer gets the value of HoT and updates value of HoT. It also sets timeout equal to ptimeout. The HoT is used as reference value for future mobility events. The procedure for ProbeHoC is shown in figure 11.

```
ProbeHoC
{
EsHoT = Get Timer
Stop Timer
EsHoT = TimeOut
TimeOut = pTimeOut
}
```

Figure 11: Extended Timeout Handoff Completed

If sender does not receive HoC message even in response to P-HoC, algorithm shown in figure 12 is activated. In this scenario Mo-TCP change its state from Probe HoW to segment lost and control handed over to normal TCP which wait for activating congestion control operation.

```
ProbeHoCTimeoutExpire:
{
EsHoT = Get Timer
Stop Timer
Exit
}
```

Figure 12: Extended Time out Expire

5. Results and discussion

The simulation study has been performed on wired-cum-wireless networks through NS2 discrete-event simulator. A system level diagram of simulation scenario is shown in figure 13. 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 Mo-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 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.

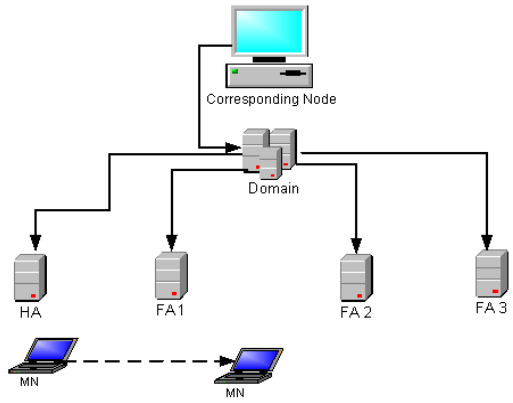

Figure 13: Simulation Scenario

Figure 14, 15, and 16 presents end-to-end delay performance comparison of Mo-TCP with well known TCP variants. In figure 14 it is noticeable that Mo-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 Mo-TCP. Figure 15 shows a plot of end-to-end delay of Mo-TCP and Vegas with results slightly in favor of Mo-TCP. Vegas end-to-end delay is in the range of 0.08 to 0.3 which is much better than Reno. Figure 14 shows a plot of end-to-end delay of Mo-TCP and Tahoe showing superior performance of Mo-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 Mo-TCP, only possible due to the continuous operation of Mo-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 *Ack* 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 Mo-TCP end-to-end delay plot.

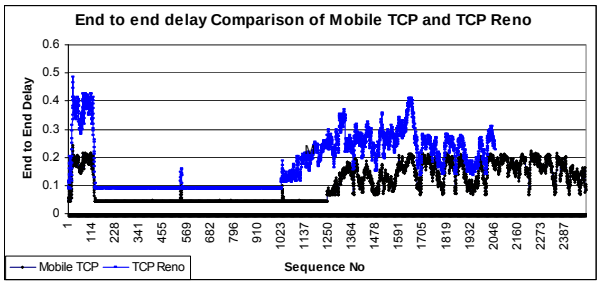

Figure 14 : Comparison of Reno and Mo-TCP End to End Delay

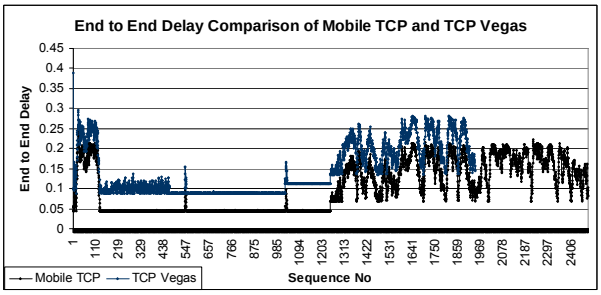

Figure 15: Comparison of Vegas and Mo-TCP End to End Delay

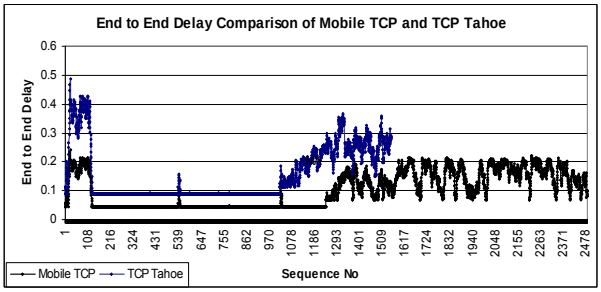

Figure 16: End to End Delay of Mo-TCP and Tahoe

Figure 17, 18 and 19 presents the congestion windows behavior of Mo-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 Mo-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, Mo-TCP *cwnd* does not drop at the handover events. It can be seen that at time around 15.8 seconds of simulation run Mo-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, Mo-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 Mo-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.

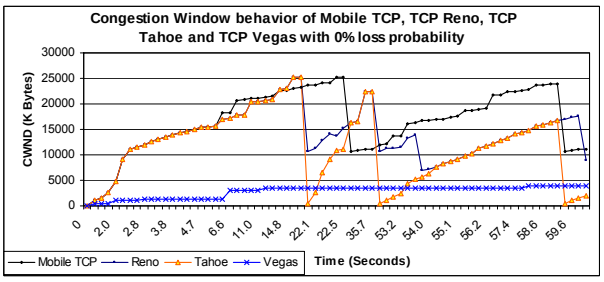

Figure 17: Congestion Windows behavior of Mo-TCP, Reno, Tahoe and Vegas with 0% loss probability

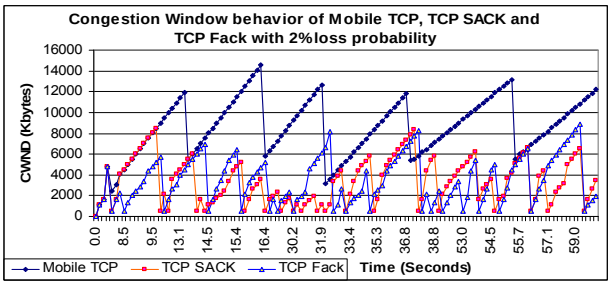

Figure 18: Congestion window behavior of Mo-TCP, SACK and FACK with 2% loss probability

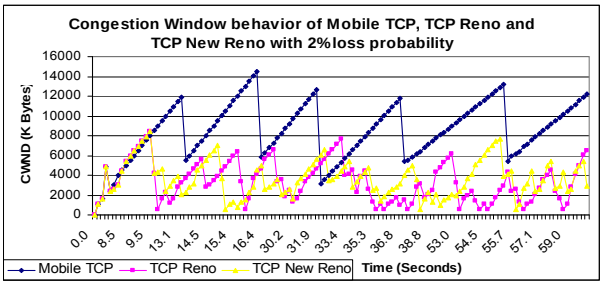

Figure 19: Congestion Window behavior of Mo-TCP, Reno and New Reno with 2% loss probability

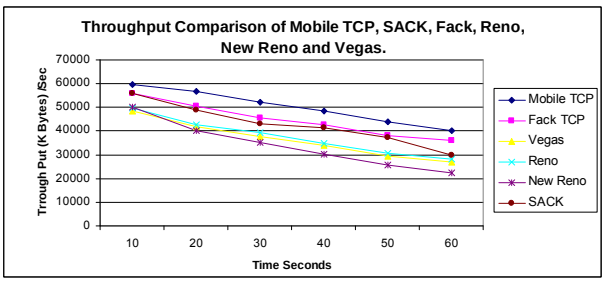

Figure 20: Through put comparison of Mo-TCP

The throughput comparison of schemes under study is presented in figure 20. 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 Mo-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 Mo-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 Mo-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 Mo-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 14, 15, and 16. Similarly attaining a better average *cwnd* is another important reason for throughput improvement. Further it is noticed that Mo-TCP handles mobility events more efficiently and prove its strength over its counterparts.

6. Conclusion

In this paper, we proposed a new TCP variant called Mo-TCP for heterogeneous wireless network. Mo-TCP has an embedded support for mobility handling and adjusting its operations according to underlying link and network conditions. In this proposal, we have added few states in the sender and receiver side operation to reduce repeated timeouts and retransmissions, which are major causes of poor 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)*. The Mo-TCP also reduces disconnections during long handover durations. 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 Mo-TCP is a viable solution which improves the performance in wireless networks. The results show that the Mo-TCP performs better with other TCP variants and achieves 10.26% to 53.55% improvement in throughput 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 Mo-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] C. P. Fu and S. C. Liew, TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks, IEEE JSAC, vol. 21, no. 2, pp. 216–28, Feb. 2003.

[2] K. Xu, Y. Tian, and N. Ansari, TCP-Jersey for Wireless IP Communications, IEEE JSAC, vol. 22, no., pp. 747–756, 4 May 2004.

[3] Casetti Claudio Casetti, Mario Gerla, Saverio Mascolo, M.Y. Sanadidi and Ren Wang, TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks, Wireless Networks, vol. 8, no. 5, pp. 467–479, 2002.

[4] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, G. Piscitelli, Performance Evaluation of Westwood+ TCP Congestion Control, Performance Evaluation, vol. 55, no. 1–2, pp. 93–111, Jan. 2004.

[5] E. H.-K. Wu and M.-Z. Chen, JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks, IEEE JSAC, vol. 22, no. 4, pp. 757–766, May 2004.

[6] I. F. Akyildiz, G. Morabito, and S. Palazzo, TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks, IEEE/ACM Trans. Net., vol. 9, no. 3, pp. 307–321, June 2001.

[7] I. F. Akyildiz, X. Zhang, and J. Fang, TCP-Peach+: Enhancement of TCP-Peach for Satellite IP Networks, IEEE Commun. Letters, vol. 6, no. 7, pp. 303–305, July 2002.

[8] Tom Goff, James Moronski, D. S. Phatak and Vipul Gupta, Freeze-TCP: A True End-to-End TCP Enhancement Mechanism for Mobile Environments, Proc. IEEE INFOCOM 2000, vol. 3, Tel Aviv, Israel, pp. 1537–1545, Mar. 2000.

[9] 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, pp. 1004–1010, Mar. 2003.

[10] Kartik Chandran, Sudarshan Raghunathan, S. Venkatesan and Ravi Prakash, A Feedback-Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks, Published in IEEE Pers. Commun., vol. 8, no. 1, pp. 34–39, Feb. 2001.

[11] A. K. Singh and K. Kankipati, TCP-ADA: TCP with Adaptive Delayed Acknowledgement for Mobile Ad Hoc Networks, Proc. IEEE WCNC 2004, vol. 3, Atlanta, GA, pp. 1685–90, USA, 21–25 Mar. 2004.

[12] Sumitha Bhandarkar Nauzad Sadry A. L. N. Reddy and Nitin Vaidya, TCP-DCR: A Novel Protocol for Tolerating Wireless Channel Errors, IEEE Trans. Mobile Computing, vol. 4, no. 5, pp. 517–529, Sept./Oct. 2005.

[13] E. Altman and T. Jiménez, Novel Delayed ACK Techniques for Improving TCP Performance in Multihop Wireless Networks, Lecture Notes in Computer Science, vol. 2775, pp. 237–50, Sept. 2003.

[14] J. Liu and S. Singh, ATCP: TCP for Mobile Ad Hoc Networks, IEEE JSAC, vol. 19, no. 7, pp. 1300–1315, July 2001,.

[15] Milan Todorovic, Noe Lopez-benitz, Efficiency Study of TCP Protocols in Infrastructured Wireless Networks, Proceeding of 0-7695-2622-5/06/ (c) 2006 IEEE.

[16] David X. Wei, Cheng Jin, Steven H. Low, and Sanjay Hegde, FAST TCP: Motivation, Architecture, Algorithms, Performance, IEEE/ACM Transaction on Networking, VOL 14, NO6, PP. 1246-1259, 6 December 2006.

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

[18] Bakre, B.R. Badrinath, I-TCP: Indirect TCP for Mobile Host, Proceedings of the 15th international conference on distributed computing system (ICDCS '95), PP.136 – 143, 30 May-2 Jun 1995,

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

[20] L. S. Brakmo and L. L. Peterson, TCP Vegas: End to end congestion avoidance on a global Internet, IEEE J. Select. Areas Commun., vol. 13, pp. 1465-1480, Oct. 1995.

[21] J. Postel. Transmission Control Protocol. RFC 793, September 1981.

[22] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, TCP Selective Acknowledgment Options, Request for Comments, RFC 2018, Network Working Group, Internet Engineering Task Force, October 1996.

[23] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP, Computer Communication Review, 26(3): 5–21, July 1996.

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

[25] M. Saeed Akbar, Syed Zubair Ahmed and M. Abdul Qadir, A Study of Dynamics of Transport Protocols over Wireless LAN during Mobility, 4th IEEE International Symposium on High Capacity Optical Networks and Enabling Technologies' HONET 2007, PP. Dubai, UAE, 18-20 November 2007.

[26] 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.

[27] Syed Zubair Ahmed, M. Saeed Akbar and M. Abdul Qadir, A Cross-Layer Vertical Handover Decision Model for Heterogeneous Wireless Networks, 4th IEEE, International Conference on Innovations in Information Technology, Innovations' 2007, Dubai UAE, 18-20 November 2007

[28] K. Ramakrishnan, S Floyd and D. Black, "The Addition of Congestion Notification (ECN) to IP", RFC 3168 Sep. 2001.

Muhammad Saeed Akbar is a PhD fellow at Mohammad Ali Jinnah University, Islamabad, Pakistan. He completed his graduation in Computer Science in 1999 and received his MS degree in Computer Science with specialization in Computer Networks, from University of Agriculture, Faisalabad, Pakistan in 2004. He is lecturer in Department of Computer Science, University of Agriculture Faisalabad, Pakistan. His area of interest is mobility management and optimization of transport services in heterogeneous mobile wireless networks. He has produced more than 10 research papers in international refereed conferences.

Syed Zubair Ahmad is a PhD fellow at Mohammad Ali Jinnah University, Islamabad campus, Islamabad, Pakistan. He graduated in 1988 from Quaid-i-Azam University, Islamabad with an MS Electronics degree. Later on, he got an MS Software Engineering degree from National University of Science and Technology, Rawalpindi in 2003. He has experience of working in both industrial projects and academia. His main areas of interest include mobile wireless networks, mobility management in heterogeneous environment and future transport services in the high-speed data networks.

Muhammad Abdul Qadir is a professor of computer science and Dean of the Faculty of Engineering & Applied Sciences at Mohammad Ali Jinnah University, Islamabad, Pakistan. He is the head of the Center for Distributed & Semantic Computing (CDSC), a research group in the department of computer science. He received the MSc degree in Electronics from Quaid-i-Azam University Islamabad, Pakistan, and the PhD degree from University of Surrey, Guildford, UK in parallel processing / distributed computing. He has more than 20 years of work experience in industry, and academia.

His present research interests are in the areas of distributed and semantic computing. In distributed systems the focus is on the optimization of transport services in mobile and wireless networks, media-independent handovers in heterogeneous mobile networks, and fault monitoring in fault-tolerant distributed systems. In semantic computing, the focus is on the development of semantic-based search engine, ontology evaluation, mapping and merging, extraction of semantics from multi-media contents, context-based query refinement for the web, and semantic cache for relational and XML databases. About ten graduate students are working under his supervision and the group has produced more than 40 research papers in international refereed conferences and journals. He is on the technical and organizing committee of some conferences, and a member of several technical organizations like IEEE.