TCP for Wireless Environments

Mr. Ranjeet V. Bidwe, Mr. Mahesh A. Kandekar

Dept of Computer engineering, PICT, Pune. Maharashtra, India.
Dept of Computer engineering, AISSMS COE, Pune. Maharashtra, India.

Abstract— *Computer networks have experienced an explosive growth over the past few years, which has lead to some severe congestion problems. Reliable protocols like TCP works well in wired networks where loss occurs mostly because of congestion. However, in wireless networks, loss occurs because of bit rates and handoffs too. TCP responds all losses by congestion control and avoidance algorithms, which results in degradation of TCP's End-To-End performance in wireless networks. This paper discusses different issues and problems regarding use of TCP in wireless networks and provides comprehensive survey of various schemes to improve performance of TCP in Wireless Networks.*

Keywords—TCP, Mobile-IP, Wireless networks, Protocol design.

I. INTRODUCTION

Due to rapid advances in the area of wireless communications and the popularity of the Internet, the provision **of** packet data services for applications like e-mail, web browsing, mobile computing etc. ***o***ver wireless is gaining importance. The TCP/IP protocol suites have number of layers, of which transport layer is used widely for mobility. It uses protocols like TCP and UDP for transferring data.

While coming towards wireless environment, we must understand what wireless environment is first. Wireless environment can be broadly distinguished in three types: Cellular networks, Ad-hoc networks and Satellite networks. In Cellular Networks a mobile host is connected to the fixed network with the help of the Base Station. This is the most common form of Wireless Network currently in use. Mobile devices like cell phones, laptops use this network. Most of the proposed solutions to TCP use this model. All service providers are on the fixed network and hence we have to address the problem of wireless networks only at one end point. Ad – Hoc Networks are formed by mobile hosts which are connected to each other within a radio distance. This kind of a model is not well deployed and very few solutions have been proposed to this model. Satellite Networks are those where a satellite link is in between the sender and the receiver. These have very high BERs (Bit Error Rates) and high latency because the Satellite are at a great distance from the surface.

Now going towards our main concept, Transmission Control Protocol (TCP) [1] is one of the important standards in the internet world and also a very vital element in internet protocol suite. It provides a connection oriented service with reliable data transfer over the unreliable underlying protocols. It uses sequence numbering and timers to ensure reliable transfer of packets. TCP's flow control increases the data sending rate until there are signs of congestion in the network. The basis of TCP congestion control lies in the following algorithms: slow start, congestion, avoidance, fast retransmit and fast recovery [2].

In the following we first outline the different issues of TCP in wireless networks. Then the main problem regarding TCP. Then we summarize some proposed solutions with their strengths and weaknesses.

II. ISSUES IN WIRELESS ENVIRONMENTS

Wired services are relatively reliable compared to wireless networks. So if any packet get lost then it is due to congestion only, so that they can carried out a congestion control scheme to get lost packets. But in wireless networks some serious issues are found, those are:

First issue is Bit Error Rate (BER). Wireless host uses radio transmission or infrared wave transmission for communication. Experimentally found that, The BER of wireless links is typically higher than that of wired networks. Also BER also varies by a large amount when wireless environment changes quickly.

Second issue is Bandwidth. Wireless links having very less bandwidth as compared to the wired links. Wireless links offers bandwidth of 2MBPS, while wired links offers 10-100 MBPS. As wireless links offers very low bandwidth, Optimum use of available bandwidth is a major issue in heterogeneous networks that has to be taken care of.

Third issue is Mobility. As world is moving towards wireless environment, large addition of mobile devices are done. So it introduces huge amount of indeterminate mobility in rather a stationary network. This tends to introduce some amount of instability in existing network topology. When wireless host is moving in a particular network, its base station is sending data to it. But when it moves to another station during handoff, the data sent by old base station is lost as it moved out of range. Similarly data it is sending to old base station is lost.

Next issue is Round Trip Time (RTT). The wireless media exhibits longer latencies than wired media in the case of satellite networks. It is almost the same as in wired networks since Radio waves travel at the speed of light which is same as

the transmission speeds in wired media. Since the bandwidth is lower in wireless networks a packet takes longer to get transmitted in wireless networks. This affects overall throughput and increases interactive delays.

Last issue is Power consumption. Normally mobile hosts have limited power and processing speed compared to base stations, which forms inefficiency in network. Solutions that Take power consumption into account have a clear-cut advantage over the otherwise designed solutions.

TCP works reliably well on wired networks and fixed topologies, so it operates on assumption that packet lost is due to congestion. But this assumption is not true in case of wireless networks. There are many reasons of packet loss like disconnection, corruption by underlying physical medium, handoffs, but TCP assumes it as due to congestion in network. So it cannot find actual reason behind loss of data. But this wrong assumption degrades the TCP performance. For example, let data is lost due to temporary or short disconnection, but TCP assumes it is due to congestion and decrease the window size to minimum size, and starting the slow start mechanism [2] which means that sender unnecessarily holds back, slowly growing the transmission rate. Even though receiver recovers quickly from temporary or short disconnection. This is illustrated in following Fig.1 where it is seen that the network capacity can remain unutilized for a while even after a reconnection.

Fig.1: TCP SLOW START [10]

The fundamental problem is the underestimation of bandwidth by the network endpoints which results in reduced application layer performance, reductions in throughput and unacceptable delays. When this happens the applications don't get their fair share of the bottleneck link's bandwidth.

Another problem may happen during handoff, there are three major impacts on TCP during the handoff scenario. The packets will experience a higher delay during handoff due to packet re-routing. Secondly the packets already in transit for the old access point are generally dropped during the handoffs. Lastly TCP has to deal with massive packet re-ordering after a handoff. Same issues are discussed in various papers. [21]

III. DIFFERENT APPROACHES IN WIRELESS NETWORKS

In this section, we are discussing some approaches proposed by taking problems under consideration to improve the performance of TCP over wireless environment.

Snoop [3],[4]protocol is classified as a TCP Aware link-level protocol. In this protocol a network layer software is updated at Base station (BS) by adding module called snoop. Snoop module checks every packet travelling on the connection in both directions. It maintains cache of TCP packets which are sent by fixed host (FH) to mobile host (MH) but not acknowledged by MH. When packet is sent from FH, snoop adds it to cache and forwards it according to its routing information. It also checks acknowledgment coming from MH, if any packet gets lost or snoop got any duplicate acknowledgment about packet, and then it resends that packet if it is cached. It maintains its own timers for retransmission of buffered packets, implements selective transmission etc. by this way snoop hides loss of packets from FH, by not propagating duplicate acknowledgments, and thereby it prevents further invocations of congestion control mechanism [5].

Main disadvantage of this scheme is that, it relies on intermediaries i.e.. BS, so it does not satisfies true end to end semantic proposed by TCP. This protocol does not completely shield the sender from wireless losses as the sender may timeout due to repeated losses or bit errors caused by the wireless link.

An extension proposed to random delay detection (RED) is Explicit Congestion Notification (ECN) [5]. RED is an active queue management mechanism in routers, it detects congestion before the queue overflows and provides an indication of this congestion to the end nodes. A RED router signals incipient congestion to TCP by dropping packets before the queue runs out of buffer space. RED router operates by maintaining two levels of thresholds minimum ($\min_{th}$) and maximum ($\max_{th}$). If the average queue size lies between the $\min_{th}$ and $\max_{th}$, then It drops packets. ECN is extension to RED, which marks a packet instead of dropping in when the average queue size lies between $\min_{th}$ and $\max_{th}$. Upon receipt of congestion marked packet, the TCP receiver informs the sender (by subsequent acknowledgement) about happening congestion, which starts the congestion avoidance algorithm at the sender. ECN requires support from both the router as well as the end hosts, there is need of modification at the end host of TCP stack. If the ECN support is provided then the packets are referred as ECN capable packets. RED droops packets that are not ECN capable.

Explicit Bad State Notification (ESBN) [6] proposes a mechanism to update the TCP timer at the source to prevent source from decreasing its congestion window, if there is congestion occurring. EBSN's are sent to the source, when base station is trying to send a packet over wireless link and fails to send. EBSN would cause the previous timeouts to be

cancelled and new timeouts put in place, based on existing estimate of round trip time and variance. Thus, the new timeout value is identical to the previous one. The EBSN approach does not interfere with actual round trip time or variance estimates and at the same time prevents unnecessary timeouts from occurring. This prevents timeouts for packets that had already been put on the network before the wireless link encountered the bad state.

Explicit Loss Notification (ELN) [7] adds an ELN option to TCP acknowledgment. When a packet is dropped on the wireless networks, future cumulative acknowledgements corresponding to the lost packet are marked to identify that a non-congestion related loss has occurred. Upon receiving this information along with duplicate acknowledgements, then sender may retransmit data instead of congestion control algorithms.

Holland and Vaidya proposed a feedback based technique called TCP-ELFN [8][9]. ELFN stands for Explicit Link Failure Notification. The goal is to inform TCP sender of link and route failures so it can avoid responding to the failures as if congestion occurs. ELFN is based on DSR[10] routing protocol. To implement ELFN message, the route failure message of DSR is modify to carry payload, it is similar to "host unreachable" ICMP message. Upon receiving ELFN message, TCP sender disables congestion control mechanism and enters in stand-by mode, it sends a small packet to probe the network to see if route has been established. If new route has been established, then it leaves stand-by mode, restores its retransmission timer (RTO) and continues as normal. Though explicit route failure notification, TCP-EFLN allows sender to instantly enter in stand-by mode to avoid unnecessary transmission and congestion control, which wastes precious MH battery power and scarce bandwidth. With explicit route reestablishment notification from intermediate nodes or active route probing initiated at the sender, these two schemes enable the sender to resume fast transmission as soon as possible. But neither of these two considers the effects of congestion, out-of-order packets, or bit errors, which are quite common in wireless ad hoc networks.

Next approach I-TCP [11] splits connection between FH and MH in two parts. First part is FH to BS and from BS to MH. Firstly FH sends data to BS, BS acknowledged that data, then it is responsibility of BS to forward that data to MH. This indirection helps shield the wired network from the uncertainties of the wireless network and the TCP/IP at the fixed host side need not be changed. On the link between BS and MH, it is not necessary to use TCP. One can use any other protocol optimized for wireless links.

Using indirection in this method tends to number of benefits. It separates flow control and congestion control functionality on the wireless link from that on the fixed network. Also Indirection allows the BS to manage much of the communication overhead for a mobile host. I-TCP also optimizes the handoff by shrinking the receive window size at the MSR which forces the FH to stop sending data when the MSR buffers are full. Drawback to this protocol is again it is not following true end-to-end semantics of TCP. In case of transferring data, copying data from the incoming connection from the FH to the outgoing connection to the MH is also needed. In case of frequent handoffs, the overhead related to the connection state transfer between the base stations may be large and add delays. And also the base stations have to be complex and with large buffers in case of heavy traffic. Working of I-TCP is case of handoff to transfer connection is shown in Fig.2

Next approach is MTCP [11]. MTCP is similar to I-TCP and also splits a TCP connection into two: one from MH to BS and the other from BS to FH. The MH to BS connection passes through a

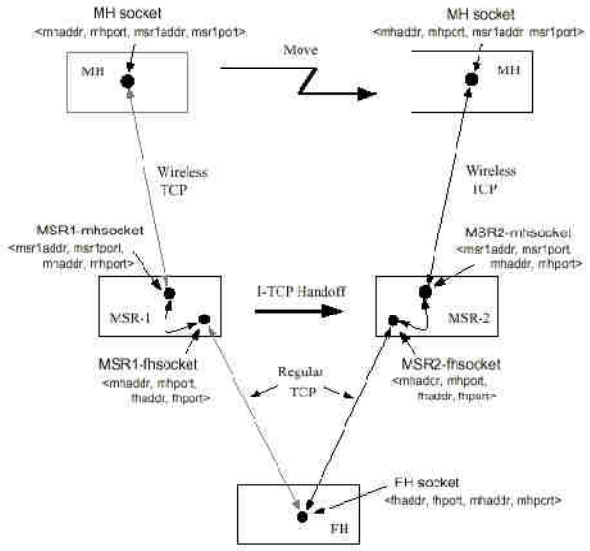

Fig.2: I-TCP Connection Transfer during Handoff [11]

Session layer protocol which can employ a selective repeat protocol (SRP) over the wireless link.Most of the schemes proposed for optimizing Transport Layer (TCP) over wireless networks needs intermediaries, due to which the end to end semantics of TCP are not maintained and problems like degradation in throughput are resulting. So next proposed protocol i.e. Freeze-TCP [13] satisfies true end to end semantics of TCP, which does not require any intermediaries; neither change in TCP code is required on the sender side or the intermediate routers. Change is limited to the mobile client side, and hence is interoperable with the existing networks.

In Freeze-TCP, receiver identifies an impending disconnection because of potential handoff, fading signal strength, or any other problem arising due to wireless media and notifies the sender of any impending disconnection by advertising a zero window size (ZWA- zero window advertisement) and prevents

the sender from entering into congestion avoidance phase. Upon getting the advertised window as zero, the sender enters the persist mode and freezes all the timers related to the session. And periodically sends the ZWP (Zero Window Probes) until the receiver's window opens up. Since the ZWPs are exponentially backed off, there is a possibility of having a long idle time after the reestablishment of connection. To avoid this, the receiver employs "TR-ACKs" (Triplicate Reconnection ACKs). As soon as the connection is reestablished, the receiver sends 3 copies of the ACK for the last data segment successfully received prior to disconnection to enable the fast transmit.

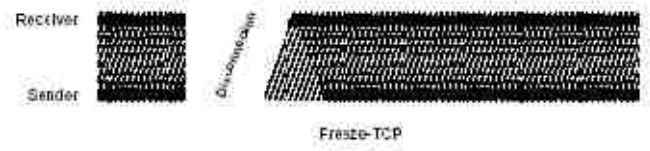

Fig.3: Increased throughput due to Freeze-TCP [13]

But main dis-advantage of Freeze-TCP is that, Freeze-TCP is only useful, if a disconnection occurs while the data is being transferred. It is not useful, in case of a disconnection when no data is being transferred between sender and receiver.

Another approach where we preserve end to end semantics is WTCP [14]. It was developed for Wireless Wide Area Networks (WWAN) where the TCP algorithms failed because it falsely assumes packet losses are due to congestion. This protocol distinguishes congestion losses and random losses. It uses packet departure time and packet arrival time for that. WTCP shapes traffic since it uses rate based transmission control, it never allows burst of packet transmission. This is useful when different connections have different Round Trip Times (RTT). The basic idea behind this protocol is that TCP should not half its transmission rate for just a packet loss which happens more frequently in wireless Networks. This is more like an algorithm where the receiver takes the responsibility of receiving all packets. The sender does not decide which packets have to be transmitted because some of the ACKs have failed but they probe the receiver to find out if a packet has to be resent. WTCP uses the ratio of the inter-packet separation at the receiver and the inter-packet separation at the sender as a metric for rate control rather than using packet losses and retransmit timeouts. WTCP reuses the standard TCP mechanism for flow control and connection management. It uses inter packet delay as a metric for congestion control, using this it performs the rate adaptation computation at receivers end. Also it provides fairly accurate measure of the available channel rate for low bandwidth channels.

However just accepting small losses as random may cause it to disregard incipient congestion. WTCP thus maintains a history of losses and reduces transmission rates more aggressively if they happened quickly. Since it does not use ACKs as a metric even Startup transmission is measured by inter-packet delay. Thus at startup WTCP sends a packet pair and uses that to adjust to the network behavior.

Next proposed approach is TCP Santa Cruz [15]. This protocol also uses same approach as WTCP. TCP Santa Cruz monitors the queue developing over a bottleneck link and this determines whether congestion is increasing in the network. Using this it identifies the type of loss, may be congestion or random and it responds it appropriately. It is able to find out direction of congestion with initial stage of congestion. Congestion is determined by calculating the relative delay that one packet experiences with respect to another as it traverses the network.

It is observed that losses due to congestion are followed by an increase in the network bottleneck queue. A wireless loss on the other hand, can be identified as a random loss that is not followed by a build-up in the bottleneck queue. TCP-SC monitors changes in the bottleneck queue over an interval equal to the amount of time it takes to transmit one window of data and receive acknowledgements corresponding to all the packets transmitted in the window. When these losses are discovered, then we expect the protocol to simply retransmit most losses without affecting the transmission window. This can be implemented as a TCP option by utilizing the extra 40 bytes available in the options field of the TCP header.

Next approach M-TCP [16] works well in frequent disconnection and low bit rate wireless links. The spurious time out as shown in the Fig.4 below proves to be very harmful to overall throughput than losses due to but errors or small congestion windows.

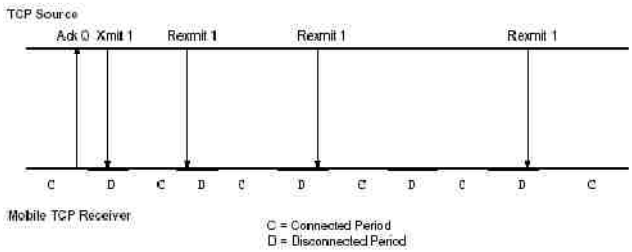

Fig.4: Serial timeouts at TCP sender [12]

In M-TCP, every TCP connection is split in two parts at the Supervisory Host (SH) . TCP connection from fixed host (FH) to the SH uses the standard, unmodified version of TCP. And connection between SH and mobile host (MH) uses the modified version of TCP. Wireless bandwidth is important resource here, and it should be keenly used. In heterogeneous systems, there is variation in available bandwidth. But SH takes care of it.

Firstly FH sends segment, then it is taken by SH. Then SH forwards that segment to MH. Then MH gives acknowledgment for that segment.SH, upon getting acks,

acknowledges back to FH. Unlike other split connection techniques, it saves the ack of the last byte, in order to prevent loss of outstanding packets. Now in case, if the MH is disconnected from nowhere, then the SH stops getting the acks and assumes that MH has been temporarily disconnected and sends the ack of the last byte that it saved previously. This ack will contain the advertised window of the MH as "zero", then sender enters the persist mode and freezes all the timers related to the session, and starts sending the exponentially backed off persist packets to the SH. The SH responds with the zero window size at the receiver, to each persist packet, until it receives some nonzero window size indication from the receiver. When it receives, then SH immediately replies to the persist packet as the appropriate window size and resumes all its freezed timers. Thus the sender can resume transmitting at full-speed. The FH again starts transmitting from the next byte that is unacknowledged.

The state transition diagram for ATCP at the sender is shown in Fig. 6. Upon receiving a "Destination Unreachable" message, the sender enters the persist state. The TCP at the sender is frozen and no packets are sent until a new route is found, so the sender does not invoke congestion control. Upon receipt of an ECN, congestion control is invoked without waiting for a timeout event. If a packet loss happens and the ECN flag is not set, ATCP assumes the loss is due to bit errors and simply retransmits the lost packet. In case of Multi-path routing, upon receipt of duplicate ACKs,

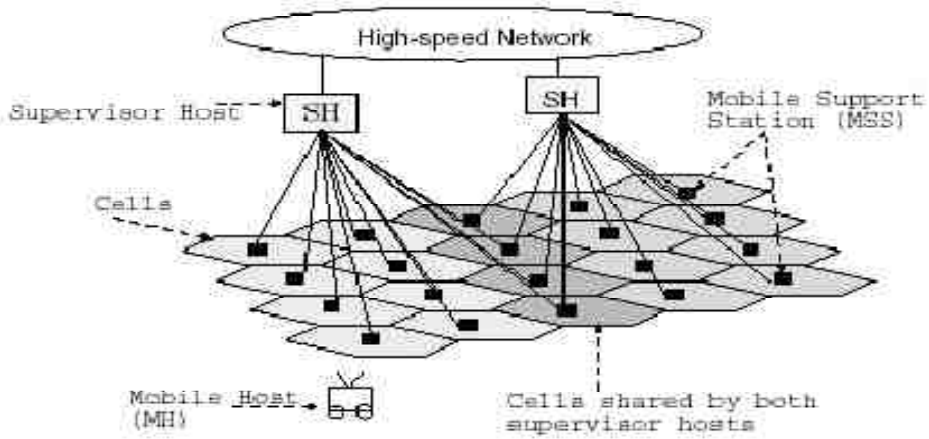

Fig.5 Architecture for M-TCP[12]

TCP sender does not invoke congestion control, because multi-path routing shuffles the order in which segments are received. So ATCP works well when the multi-path routing is applied.

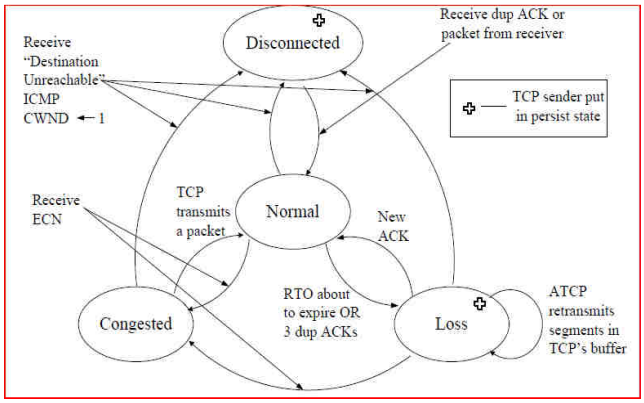

Fig.6: State transition diagram of ATCP at sender[17]

Next approach is TCP Westwood[18]. It is a sender-side modification of the TCP congestion window algorithm that improves upon the performance of TCP Reno in wired as well as wireless networks. General idea used here is to use bandwidth estimate (BWE) to set the congestion window (cwin) and slow start threshold (ssthresh) after congestion episode. The main difference between TCP Reno and TCP Westwood is, TCP Reno halves the congestion window after

three acknowledgments where as TCP Westwood attempts to select a slow start threshold and a congestion window which are consistent with the effective bandwidth used at the time congestion is experience.

The source performs end-to-end estimate of the bandwidth available along a TCP connection by measuring and averaging the rate of returning ACKs. Whenever a sender perceives a packet loss (i.e. a timeout occurs or 3 duplicate ACKs are received), the sender uses the bandwidth estimate to properly set the congestion window (cwin) and the low start threshold (ssthresh). This way TCP Westwood avoids overly conservative reduction of cwin and ssthresh; and thus it ensures faster recovery.

TCP Westwood satisfies true end to end semantic of TCP. Also it works well in mixed wired and wireless networks. Better throughput, goodput and delay performance, fairness as well as friendliness when coexisting with TCP Reno were observed in experimental studies. TCP Westwood does not require inspection and/or interception of TCP packets at intermediate (proxy) nodes and complies with the end-to-end TCP design principles. Only disadvantage is that it performs poorly when random packet loss rate exceeds a few percent.

TCP-F[19] is specially designed for Ad-hoc networks. All previously proposed schemes depend on the base station and so cannot be applied to the multihop wireless networks, since there are no base stations in such a network. At a time of large data transfer from one MH to another MH through number of MH's, if an intermediate MH detects a route failure, due to which it cannot send the data any further, then it sends a route failure notification (RFN) to the source. Each intermediate router that receives the RFN, invalidates all packets traveling through that failed route and prevents more incoming packets. The intermediate node than tries to find an alternate route for the destination. If any alternate path exists, then packets are routed through that path, otherwise RFN is forwarded towards the source. Upon receiving RFN, source goes into the snooze state and remains until it is notified of any updates. That time source stops all packet sending, Marks all existing timers as invalid, Freezes the send window of packets, Freezes value of other state variables such as retransmission timer value and window size and Starts a route failure timer which corresponds to a worst case route reestablishment time. If any intermediate router knows about a new route to the destination, then it sends a route reestablishment notification (RRN) packet to the source, whose identity was previously stored. As soon as the source receives the RRN, it comes to an active state from the snooze state. Since almost all packets in transit would have been affected by the failure, the source flushes out all unacked packets in its current window. Communication would then resume at the same rate prior to the route failure.

The TCP-Bus algorithm [20] is very similar to the TCP-F algorithm. The basic idea is to use buffering capacity of mobile nodes. It uses a source-initiated on-demand routing protocol for the underlying layer. It uses two control messages (ERDN and ERSN) related to route maintenance to notify the source of route failures and route re-establishments. These indicators are used to distinguish between network congestion and route failure as a result of node-movement.

ERDN (Explicit Route Disconnection Notification) message is generated at an intermediate node upon detection of a route disconnection. When the sender receives the ERDN message it stops transmitting. Similarly after discovering a new path from the node that initiated the ERDN message the sender is informed by using a ERSN message (Explicit Route Successful Notification). On receiving the ERSN message the source starts retransmission. However the retransmission of lost packets due to congestion relies on timeout mechanism. Since it increases the timeout to avoid retransmission during a disconnection it must also request the lost packets as they will be retransmitted only late. The packets from the node that initiated the ERDN message, to the point where the node previously existed are flushed after receiving the ERSN message.

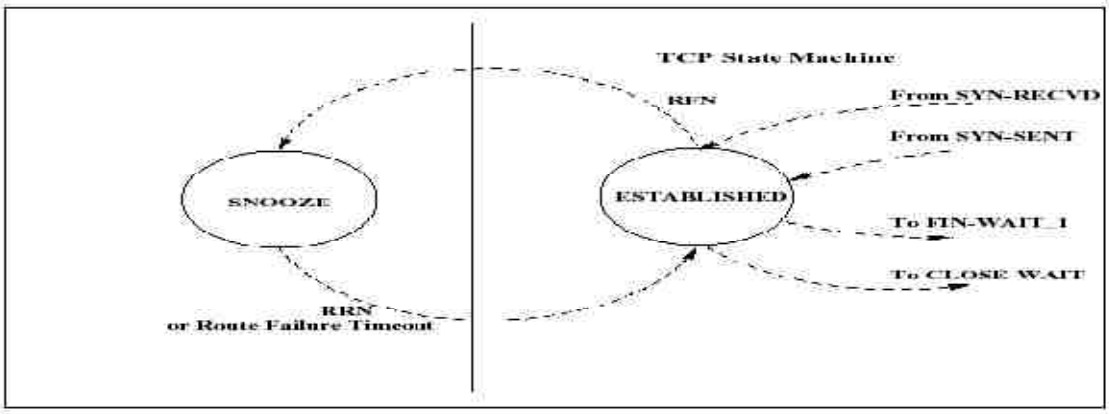

Fig.7: The TCP-F State Machine [19]

Hence to avoid further packet flows to the mobile node all nodes that received the ERDN message for a particular destination must stop forwarding those packets. Also to ensure that an ERSN message is successfully delivered to the source, all intermediate nodes must time out and retransmit the ERSN message if they do not hear the upstream nodes forwarding the ERSN message.

TCP-P [22] stands for TCP-Performance. Talking about basic TCP functionalities, then yes, TCP-P satisfies true End to-End semantics of TCP since no intermediaries involved. It provides reliable, connection oriented service for mobile nodes. TCP-P uses the standard TCP mechanisms for flow control and connection management. Mainly TCP-P tries to solve three important issues of TCP that are Congestion. So, TCP-P Disconnection and Random Packet Losses mainly having three functionalities. Working with these three functionalities TCP-P is able to detect packet losses due to congestion in network, disconnection in network links and random lost packets. TCP-P is more successful than other TCP versions since it is having more Packet Delivery Ratio as well as able to solve more issues [23].

First function of TCP-P deals with issues like mobility and handoff, Disconnections. It is a receiver modifying stage, here receiver senses wireless medium continuously for detecting fading signals which in turn detects happening disconnection. In certain cases, it might even be able to predict a temporary disconnection (signal strength is fading for instance). In such a case, it advertises a zero window size, then it forces sender into the ZWP mode and prevent it from dropping its congestion window. When the receiver senses an impending disconnection, first it advertises its window size as zero and a zero window acknowledgement (called as ZWA) to sender prior to disconnection to inform sender about disconnection. This period is called as "warning period" (provided that warning period should be long enough than time required for one ZWA to get across sender). If warning period is any longer, then sender is forced into ZWP mode. If warning period is small then receiver will not have enough time to inform sender, and sender have to drop its congestion window. When connection is established again then receiver sends three ACKs for last received packet and sender starts sending data again. To check connection is established or not sender sends zero window probes to receiver after an interval of time.

When data sending is going on at the same time, sender is continuously computing the connection Bandwidth Estimate (BWE) which is equal to the rate at which data is delivered to the TCP receiver. The BWE value is computed by performing end-to-end estimate of the bandwidth available along with the TCP connection by measuring and averaging the rate of returning ACKs. This estimated BWE value is used to set congestion window (cwin) and slow start threshold (ssthresh) before congestion episode. Whenever sender perceives a packet loss (i.e. a timeout occurs or 3 duplicate ACKs are received), the sender uses the BWE to properly set the congestion window (cwin) and the slow start threshold (ssthresh) and sends data accordingly. This mechanism very little bit different from slow start mechanism. In slow start mechanism, packet sending rate is increased exponentially i.e. if 2 packets delivered successfully, then it will try for 4 packets, then try for 8 and so on. But in this approach packet sending rate is not increased exponentially i.e. if 2 packets delivered successfully, but it is not able to delivered 4 packets, can only delivered 3 packets then it will try for sending 3 packets only, not 4 packets. It will prevent system from loss of packets.

Initially TCP was designed with the notion in mind that wired networks are generally reliable and any segment loss in a transmission is due to congestion in the network rather than an unreliable medium (The assumptions is that the packet loss caused by damage is much less than 1%) . This notion doesn't hold in wireless parts of the network. Wireless links are highly unreliable and they lose segments all the time due to a number of factors. According to [24], noise in network is main reason behind randomly lost packets. Up to 30% of messages can be lost because of noise. For randomly lost packets TCP-P also provides the solution. TCP-P just modifies the header part of the packet. When packet is lost, i.e. its lifetime exceeds TTL value of packet, and then lost packet itself sends loss notification message to sender. To gain this functionality we can modify TTL field in TCP header to send a ICMP message to sender. This message can use value from sender IP address from header part of TCP to send ICMP message. By this way sender can detect lost packet and resends same packet again to receiver.

IV. DISSCUSION

By studying above approaches, we observe that an ideal solution should have following characteristics.

It should maintain true end to end approach without involving any intermediaries. When we are coming towards network security, encryption is adopted widely. whereby the whole IP payload is encrypted, and the intermediate may not know about the transport layer protocol used. In briefly, it should be able to handle encryption.

Any scheme proposed must be interoperable with the existing network infrastructure. There should be no change required in the sender or the intermediate routers.

If at all there is any intermediate node involved in any scheme, care should be taken of its 100% efficiency, since the processing overheads involved with those nodes may add to the original problem. The processing overheads may include extra buffer space or transfer of complete state information of a mobile node from one base station to another.

In ideal solution code at the sender should be affected. Means

we should have a static code at sender.

Also the solution should be robust against high BER. It should be robust against frequent disconnections. And its performance should not degrade with long disconnections.

V. CONCLUSION

In this paper, we present a comprehensive survey of the various schemes proposed in the literature that try to solve this problem, classified them according to their characteristics and mentioned their limitations.

We conclude that different schemes have their own advantages and disadvantages. But it seems that a combination of pure link level and end to end scheme is a good combination to alleviate the problem. Further research is needed to investigate other approaches to help TCP discriminate between host mobility and network congestion. Although most schemes would yield improvement in throughput, the key factor will be the ease with which the modification can be incorporated in the existing infrastructure.

REFERENCES

[1] W. Richards Stevens, “TCP/IP illustrated, Volume I; The protocols.” AWL 1994.

[2] V, Jacobson, “congestion avoidance & control”, ACM proceedings. SIGCOMM 88.

[3] H. Balakrishan, V. N. Padmanabhan and R. Katz “Improving reliable transport and handoff performance in cellular wireless networks.”, wireless networks, vol.1 , no.4 Dec 1995.

[4] H.Balakrishan, V. N. Padmanabhan and R. Katz,”A comparison of mechanisms for improving TCP performance over wireless links”, in proceedings of ACM SIGCOMM’96, palo alto, CA, Aug 1996,pp 256-269.

[5] Sarama Vangala, Miguel A. Labrodor,” Performance of TCP over wireless Network with snoop protocol”

[6] Bikram Bakshi and N.H. Vaidya,”Improving performance of TCP/IP over wireless networks”.

[7] R. Ramani, A. Karandikar, “Explicit congestion notification in TC over wireless networks”.

[8] G. Holland and N. H. Vaidya, “Analysis of TCP Performance over Mobile Ad Hoc Networks,”.

[9] J. P. Monks, P. Sinha and V. Bharghavan, "Limitations of TCP-ELFN for Ad Hoc Networks,".

[10] D. B. Johnson, D A. Maltz, Y. Hu, “The dynamic souce routing protocol for mobile ad hoc networks,”.

[11] Ajay bakre & B. R. Badrinath, “I-TCP: indirect TCP for mobile hosts”.

[12] Raj Yavatkar & Namrata Bhagvat,”Improving end-to-end performance of TCP over mobile envioronment”.

[13] Tom goff, James Moronski, D. S. Phatak, Vipul Gupta, “Freeze-TCP: A true end to end TCP enhancement mechanisms for mobile environment”

[14] P. Sinha,N. Venkitaraman,R Sivakumaran & V. Bharghavan,”WTCP: realible transport protocol for wide area network”.

[15] C. Parsa & J.J. Garcia-Luna-Aceves, “Improving TCP congestion control over internets with heterogeneous transmission media”.

[16] K. Brown & S. Singh, “M-TCP: TCP for mobile cellular networks”.

[17] J. Liu, S. Singh, “ATCP: TCP for Mobile Ad Hoc Networks”.

[18] Severio Moscola, Claudio casetti, Mario Gerla,M.Y.Sanadidi & Ren Wang, “TCP WESTWOOD:Bandwidth estimation for enhanced transport over wireless links”.

[19] K. Chandran, R. Prakash, “Feedback based scheme for improving TCP performance in Ad-hoc networks”.

[20] Dongkyun Kim, C-K Toh & Yanghee Choi “TCP-Bus-Improving TCP performance in wireless Ad-Hoc Networks”.

[21] Ranjeet V. Bidwe, “Different Issues and survey of proposed solutions in TCP over Wireless Environment.”, International Journal of Future Computer and Communication, 2012.

[22] Ranjeet V. Bidwe, Amar R. Buchade, “TCP-P: A new approach in wireless environment to solve issues currently TCP facing” Second International Conference on Computational Intelligence and information Technology – CIIT 2012, Elsevier, 2012.

[23] Ranjeet V. Bidwe, Amar R. Buchade, “Improving Performance of TCP in Wireless Environment using TCP-P”, International Journal on Communication, Vol. 4, No. 1, 2013.

[24] Shamimul Qamar, Kumar Manoj, “Impact of Random Loss on TCP Performance in Mobile Ad-hoc Networks (IEEE 802.11): A Simulation-Based Analysis”, Arxiv preprint arXiv9:1002.2403,arxiv.org, 2010.