TCP Performance Enhancement over Wireless Mobile Networks

MTech Project Report

By

Ajay Kumar Singh

Roll No : 00305032

Guide:

Prof. Sridhar Iyer

School of IT.

Co-Guide:

Prof. S.S.S.P. Rao

CSE Dept.

Department of Computer Science and Engineering
Indian Institute of Technology
Mumbai

Acknowledgment

I take this opportunity to express my sincere thanks to my guide **Prof. Sridhar Iyer** and co-guide **Prof. S.S.S.P.Rao** for their support, encouragement and advice. Their constant guidance was invaluable in the realization of this report.

Ajay Kumar Singh
IIT Bombay
January 2, 2002

Abstract

Transmission Control Protocol (TCP) is tuned to perform well in wired networks where packet losses are mainly due to congestion. It has been shown that TCP performance degrads while operating over **mobile wireless networks**. Mobile wireless networks have distinct characteristics of having *high bit error rate* and *intermittent disconnection period* induced by mobility of the hosts. Due to these characteristics, in these networks packet loss also occurrd by transmission errors and disconnection in addition to the congestion. As a result, *TCP assumption of packet loss means congestion does not hold always on these networks.* Therefore, TCP misinterprets packet loss on these networks as a indication of congestion and invokes congestion control mechanisms. This mechanism reduces the transmission rate drastically, resulting in lesser throughput over these networks.

Several approaches have been put forth by reaserach community for alleviating the detrimental effect of poor charactreistics of wireless channel, but only few approaches are presented for solving the mobility related issues. Even in these attempts, mostly requires support from basestation like some state per TCP connection, some changes in protocol stack etc. This requirment of these approach make them undesirable as these schemes *introduced scalability difficulty*, *hinder the inter-operability of the mobile host with different networks*, and also could not support *encrypted traffic, different acknowledgement path.*

All these observations motivated us to design of a new approach called **ATCP**. ATCP falls into the category of those approaches which mitigates the degrading effect of mobility while requiring modification only at mobile host like 3DA, Freeze TCP approach. We noticed that the authors of 3DA and Freeze TCP approach have not given any specific action to be taken when mobile host is sender. ATCP is designed for enhancing the performance of both direction data transfer i.e. MH to FH and MH to FH. ATCP is a modified version of TCP at mobile host end and requires the feedback about ongoing mobility from network layer in terms of connection and disconnection signal.

We compared ATCP with 3DA, Freeze TCP and TCP Reno over network simulator ns-2. ATCP is shown to improve performance of TCP data transfer in both direction i.e. FH to MH and MH to FH. For data transfer from FH to MH, ATCP is shown to reduce response time as 3DA approach does, but it also achieve higher TCP throughput in most cases. It has been shown that for small disconnection period ATCP and Freeze TCP performs equally well, only in case of

long disconnection period Freeze TCP performs better than ATCP. But ATCP has a significant advantage over Freeze TCP that it does not require prediction of impending disconnections. Simulations show that for FH to MH data transfer, an improvement of up to 30% is achieved over small RTT (round trip time) connections, and up to 200% over large RTT connections. For MH to FH data transfer, the improvement achieved is up to 50% over small RTT connections, and 150% over large RTT connection.

Contents

1 Introduction **4**
1.1 Transmission Control Protocol (TCP) . . . 4
1.2 Mobile Wireless networks . . . 5
1.3 Mobile Wireless networks and TCP . . . 5
1.4 Goal . . . 6
1.5 Summary and Organisation of the Report . . . 6

2 Design Spaces for adaptation to TCP **7**
2.1 Introduction . . . 7
2.2 Design Spaces . . . 7
2.3 Implications . . . 8
2.4 Summary . . . 8

3 Related Work **9**
3.1 Introduction . . . 9
3.2 3-Duplicate Acknowledgement . . . 9
3.3 Indirect-TCP (I-TCP) . . . 9
3.4 Snoop protocol . . . 10
3.5 M-TCP . . . 10
3.6 WTCP . . . 11
3.7 Freeze-TCP . . . 11
3.8 Comparison . . . 11
3.9 Motivation for ATCP . . . 12
3.10 Summary . . . 14

4 Our Approach: ATCP **15**
4.1 Introduction . . . 15
4.2 Description of ATCP . . . 17
4.2.1 Data Transfer from MH to FH . . . 17
4.2.2 Data Transfer from FH to MH . . . 21
4.2.3 Design Ideas in ATCP . . . 21

4.3 Summary . . . 22

5 Simulations **23**

5.1 Introduction . . . 23

5.2 Implementaion Details . . . 23

5.3 Simulation Parameters . . . 24

5.4 Simulation Analysis & Comparison . . . 24

5.4.1 FH to MH data transfer . . . 24

5.4.2 MH to FH data transfer . . . 30

5.5 Summary . . . 31

6 Conclusion and Future work **32**

6.1 Conclusion . . . 32

6.2 Future work . . . 33

Chapter 1

Introduction

1.1 Transmission Control Protocol (TCP)

The Transmission Control Protocol [22] is a reliable, connection-oriented, full duplex, byte stream transport control protocol. This protocol supports flow and congestion control, and is used by many end-user applications, including Web browsers and e-mail clients.

TCP is designed with a assumption that packet loss means congestion in the network. It detects the packet loss either by the timer out of retransmission timer or reception of 3 duplicate acknowledgment. After detecting packet loss, it reduces its sending rate as per congestion control mechanism.

As per congestion control mechanism, TCP maintains two variable ssthresh [25] and cwnd [25]. When TCP detects packet loss by timer-out, the congestion control mechanism reduces the ssthresh to half of present effective transmission window (effective window is minimum of advertised window by receiver and senders congestion window cwnd) and make cwnd equal to one segment and TCP enter the slow-start phase. During slow-start TCP increases its cwnd by one segment for every acknowledgment it receive. When cwnd become larger than ssthresh, TCP enter congestion avoidance phase during which it increment cwnd by one segment for one round trip time (rtt).

When TCP detects packet loss by the reception 3-duplicate acknowledgment, congestion control mechanism make ssthresh equal to half of transmission effective window and enter the fast recovery mode. During fast recovery mode, cwnd is set to ssthresh plus 3 segment. Each time another duplicate Ack, increment cwnd by the segment size and transmit a packet(if allowed by the new value of cwnd) and when the next Ack arrives that acknowledges new data, TCP set cwnd to ssthresh and enter congestion avoidance.

Thus, we have seen that a detection of packet loss by TCP causes significant reduction in sending rate. This reduction is shown to be effective in reducing congestion from the network. In following sections, we shall discuss mobile wireless networks and TCP performance over these networks.

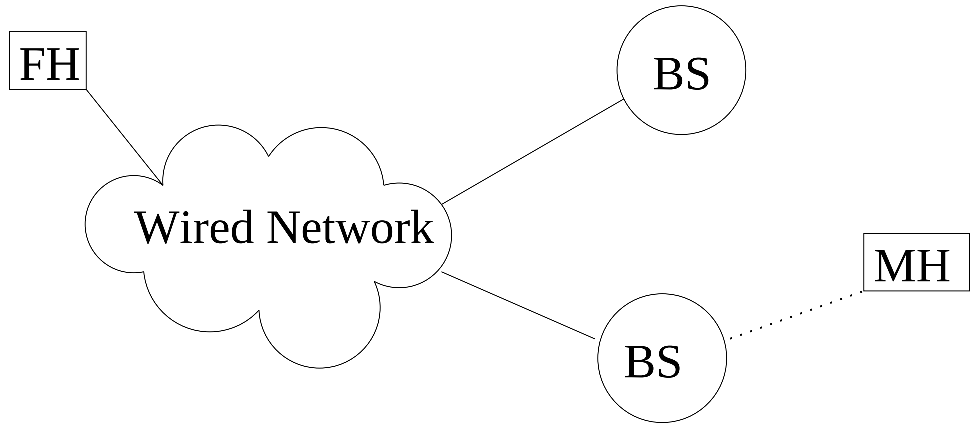


Figure 1.1: typical connection scenario

1.2 Mobile Wireless networks

A typical mobile wireless networks has mobile host and base stations. The base stations are inter-connected via wired network and Mobile hosts connect to these base stations over wireless media [33]. This configuration of base-stations and mobile hosts is also called wired-cum-wireless networks. Another typical configuration is all host are mobile and are connected through wireless media, called Ad-hoc networks. In this report, we shall consider the TCP performance over wired-cum-wireless configuration of mobile wireless networks (see figure 1.1).

The mobile wireless networks have some distinct characteristics compare to wired networks. These networks have **high bit error rate** and **intermittent disconnection periods**. The high bit error rate is incurred becouase of wireless media, where error occurs due to fading and radio signal interferences etc. To support the mobility of host, these networks do handoff when the host moves from base station region to another base staion region. This handoff may involve some period during which mobile host may be inaccessible which results in a disconnection for small period. Host may move to out of coverage region which also cause disconnection from the networks. Thus, mobility of host cause intermittent disconnections and wireless media increases bit error rate in these type of networks. In next section, the issue of TCP performance over these type of networks will be considered.

1.3 Mobile Wireless networks and TCP

TCP while operating over wired networks delivers good performance as the assumption made by TCP that packet loss means congestion is valid over wired network. As wired link have very low error rate which make very few packets get corrupted and dropped due to error introduce by channel. Thus packet drop over wired network usually occur due to congestion and TCP performs well over it as it is tuned for this. In case of wireless mobile networks this assumption does not hold in all cases, as described earlier that packet loss over these type network also occur due to bad wireless channel condition and intermittent disconnection introduced by mobility.

Whenever TCP operating over these type of network detects packet losses caused by either disconnection or error introduced by wireless channel, *it will unnecessarily invoke congestion control mechanism.* This mechanism reduces the sender window multiplicatively (by half), this will cause reduce throughput of channel. As when packet loss occur due to bad condition of wireless channel, very few packet get through during bad phase, but *after bad condition is over full capacity is available to use, similarly when hand-off is over MH is accessible with full capacity.* So, the reduction of congestion window, is not only unnecessary, it also decrease the throughput becuase it results in under-utilization of the channel. *This results in degradation of TCP performance over these types of network.*

1.4 Goal

We observed that the performance of TCP degrades over mobile wireless networks primarily due to **high bit error rate** and **intermittent disconnections**. We are focussing intermittent disconnection issue, as many approches including [1] [27] [8] [17] [31] [29] [19] have been proposed which are effective in mitigating the adverse effect of poor wireless channel characteristics. But, there are only few approaches [21] [30] [28] [27] [19] which tackle mobility induced problems. Still, some of these approaches require supports from base staion, which hinders the interoperability of mobile host in different networks. Also these schemes are not easily scalable. So, it is desirable to have scheme which requires modification in mobile host protocol stack only. *Therefore, our goal was to devise ascheme which alleviate the degrading effect of mobility on the TCP performance while requiring modifications at MH protocol stack only.* To this end, we proposed a scheme called **ATCP**, adapted version of TCP at mobile host. ATCP is designed for improving data transfer performance in both direction i.e. FH to MH and MH to FH. It requires feedback from network layer about the status of ongoing mobility in terms of *connection signal* and *disconnection signal.* Using these signal, it modifies the action taken when retransmission timer expires. Details will be discussed in chapter 4.

1.5 Summary and Organisation of the Report

The problem of TCP getting confused by packet losses cause by poor quality wireless channel or intermittent disconnection, with the packet losses cause by congestion, over wireless mobile networks is being exposed. In next chapter, the various design spaces and choices available to improve TCP performance will be discussed and their implications in implementation will be considered.In chapter 3, various approaches towards improving TCP performance will discussed and compared. Chapter 4 will details about future work involve in the project and conclusion will be drawn. In chapter 7 we will present the simulation result of ATCP performance and compare with 3-duplicate acknowledgement(3DA) approach [21] and FreezeTCP approach [30].

Chapter 2

Design Spaces for adaptation to TCP

2.1 Introduction

We will discuss different design spaces available for adapting TCP over the mobile wireless networks and implications of working upon these spaces. It will help us in comparing various approaches and understanding features that a desired solution should have. We will consider wired-cum-wireless configuration as shown in fig. 1.1

2.2 Design Spaces

- *Fixed Host TCP :* An approach can be designed which requires modification in TCP at Fixed Host. These modifications could be for example, making TCP aware of reason of packet loss. TCP can be notified, either by BS or MH, that due to what reason a particular packet loss has occur. Then TCP can be adapted to take appropriate action, if congestion causes packet loss than Fixed Host behave normally as before, otherwise if loss has occured due other reason than possible adaptation could be to simply retransmit the packet and do not reduce the current transmission window etc.

- *BS protocol stack :* Protocol stack at BS can be modified for implementing many schemes. It can be modified to inform modified FH TCP when MH is out of coverage region as no packet could be able reach MH. It can also be modified to isolate Normal FH TCP from hand-off. Isolation can be done by buffering packets at BS and retransmitting it when it detects that MH has not got a particular packets or by termination sender connection at BS and having a new connection with MH.

- *MH protocol stack :* MH stack can be changed by having modified TCP which can provide some mechanism to inform about mobility to FH TCP.

- *Hybrid Approaches :* These approaches may involves changing at more than one design space, typically at BSs and MHs like having specialised protocol for wireless media.

Details of various adaptation that can be applied to TCP will be given during the description of working of various approaches in chapter 3.

2.3 Implications

- *Modification at Fixed Host TCP :* As this involve modification in TCP of presently deployed large base of hosts, it is not easily feasible, but may be useful for new fixed host being installed.
- *Modification at BS protocol stack :* This may introduce the following difficulties:
  - Scalability: These approaches require some amount of state at BS for each TCP connection. Therefore, not easily scalable.
  - Interoperability: These approaches do not allow inter operability of mobile host with other types of mobile wireless networks since they use specialised protocols.
  - Encrypted Data: The encrypted data will cause these schemes to fail since an IP payload will be encrypted as in IP Security protocol [12].
  - Different Acknowledgement paths: These schemes do not work if acknowledgements from the MH take a different path from the one taken by data packets.
- *Modification at MH protocol stack :* These are the nodes which require mobility support, therefore, their protocol stack can be modified. These approaches are most desirable as they do not require any changes in fixed host TCP and any support from mobile wireless networks. This enhances the interoprability of mobile host with different mobile wireless networks.
- *Hybrid Approaches:* These approaches will inherit above mentioned merit and demarit depending on where they require modifications.

2.4 Summary

We discussed three design spaces where researchers can work out their design, namely FH TCP, BS protocol stack and MH protocol stack. Considering implications of modifications at different spaces, it is desirable that a solution should require no modification in FH TCP and no support from base station, but could adapt MH protocol stack. In next chapter, we will present many schemes attempted earlier by research community to alleviate the degrading effect of temporary disconnections caused by mobility of hosts.

Chapter 3

Related Work

3.1 Introduction

We will describe various schemes proposed for mitigating the detrimental effect of temporary disconnection induced by mobility on TCP performance. The compariosn of these approaches and motivation for our approach will be also be presented.

3.2 3-Duplicate Acknowledgement

The approach proposed in [21], 3-duplicate acknowledgements approach, requires network layer to provide information about ongoing mobility to the TCP layer at MH. At the time of reconnection, the MH sends three duplicate acknowledgements to FH. These acknowledgements cause the FH to enter the fast recovery phase and restart transmission. This scheme reduces the idle period after an MH is reconnected, otherwise TCP at the FH would have waited for an RTO event to occur before restarting transmission.

3.3 Indirect-TCP (I-TCP)

In I-TCP [19] a transport layer connection between an FH(fixed host) and MH(mobile host) is splitted in two separate connections - one between FH to BS, another between BS to MH. Normal TCP run on connection between FH to BS and specialized protocol over BS to MH connection can be used for flow and congestion control tuned for wireless condition. When a packet is received on FH to BS connection it is acknowledged independent of second connection between BS to MH and vice-versa.

When a MH moves out of old BS region and enter new BS region, the whole state consisting of two socket per connection is transferred to new BS. It may increase time involve in hand-off. This scheme does not preserve semantics of TCP that FH gets the acknowledgement only after peer TCP at MH has got the data.

3.4 Snoop protocol

In this approach [27], Base station routing code is modified by adding a module, called snoop, that monitors every packet that passes through the connection in either direction. The snoop module maintains a cache of TCP packets sent from the FH (fixed host) that haven't yet been acknowledged by MH. When a new packet arrives from the FH, snoop module add it to its cache and passes the packet on to the routing code which performs the normal routing functions. The snoop module also keeps track of all the acknowledgment sent from the mobile host. When a packet loss is detected(either by the arrival of a duplicate acknowledgment or by a local timeout), it retransmit the lost packet to the MH if it has the packet cached. Thus, the base station(snoop) hides the packet loss from the FH by not propagating duplicate acknowledgments. For data transfer from MH to FH, MH is notified for any packet loss over wireless channel by Base station. After getting notification of packet loss from BS, MH retransmit that packet and does not invoke congestion control mechanism.
The routing protocol tries to minimize handoff duration by multicasting the packets to set of base stations. Base stations nearby to the active BS through which present TCP data flow is taking place, and the active BS forms a multicast group. All base stations in multicast group buffers the packet destined for MH although only one base stations buffer is being used for retransmission purpose. When a MH moves to near by BS then no packet drop occur as already that BS has buffered the packet which could not be delivered by old BS during hand-off. No state transfer take place between old BS and new BS during hand-off.

3.5 M-TCP

It assume 3-tier architecture for network [28]. Many MH can be connected to MSS(Mobile Support Station) and many MSS are connected to SH(Supervisory host) and SHs are interface to fixed networks. MSS have minimum capability just to support MH for communication and SH are responsible for bandwidth management and mobility management. It also assume that transport layer see low bit error rate over wireless link. And M-TCP approach mainly take care of frequent hand-off related problem.
It uses the split connection approach i.e. every TCP connection is split in two at SH. The TCP sender on fixed network uses unmodified TCP to send data to the SH while the SH uses modified version of TCP called M-TCP for delivering data to MH. The TCP client at the SH (SH-TCP) receives segments transmitted by the sender and it passes these segments to the M-TCP client for delivery to MH. ACKs received by M-TCP at the SH are forwarded to TCP client for delivery to the TCP sender. Thus, data is acknowledged only when MH has received it to preserve the end to end semantics. TCP client at SH advertise the zero window to FH sender when it does not received acknowledgement for unacknowledged data form MH for some estimated time or when M-TCP client at SH notify it that MH is disconnected, to prevent the retransmission timer

out event at FH sender. When SH-TCP get notification that MH is connected again by M-TCP client at SH, it advertises full window to FH sender, thus opening full window again. Thus, it ensures that the FH will not invoke congestion control due to disconnection of MH. When a MH moves from one SH region to another SH region, small state transfer takes place so that new SH can maintain the data flow between splitted connection of the TCP connection between FH and MH.

3.6 WTCP

The WTCP approach [8], is designed primarily for WWAN environments. It assumes that the MH will be connected to a dedicated proxy and that both will be mobility aware. The sender (i.e. proxy) will have a new rate based transmission control behaviour. With the help of feedback from the receiver, the sender tries to identify the reason for the losses, whether they are due to handoff, bit errors or congestion losses and thus tries to minimize the effect of non-congestion related losses on the performance.

3.7 Freeze-TCP

The main idea of this approach [30] is that when MH sense that it going to disconnect from the network it advertised its window to zero which cause FH sender to go into persist mode. And when mobile host connect back to network, it send 3-duplicate acknowledgment of last received byte. Thus forcing FH-sender to take fast retransmit and fast recovery action and start sending data immediately after MH connect back to network. This approach does not deal with high bit error rate of wireless channel. The approach requires the network layer to give an indication of impending disconnection.

3.8 Comparison

These approaches will be now compared on different factors in table 3.1, 3.2. Description of some of these factors are given below, for further clarification.

- Scalable: Can a approach is scalable if number of mobile host increases to large value.
- Interoperable: Does a approach allow mobile host to be interoperable when it moves from one network to another network.
- Encrypted traffic support: Does a approach work even if the payload of IP is encrypted.
- Different Acknowledgement path support: If acknowledgement for data takes different path other than taken by incoming data, then some technique would fail.

Table 3.1: Comparison of various approaches

Approach	FH TCP modification	BS Support required	MH Protocol stack modification	Scalable	Interoperable
3-Duplicate acknowledgement	No	No	Yes	Yes	Yes
FreezeTCP	No	No	Yes	Yes	Yes
M-TCP	No	Yes	Yes	Difficult	No
WTCP	Yes	Yes	Yes	Yes	Yes
I-TCP	No	Yes	Yes	Difficult	No
Snoop	No	Yes	Yes	Difficult	No
ATCP	No	No	Yes	Yes	Yes

- Data Transfer Direction Supported: Does a scheme try to alleviate degrading effect of mobility on FH to MH and MH to FH data transfer.
- End to end semantics: When a sender get the acknowledgment only when the intended receiver has received it, end to end semantics is said to preserved otherwise not .

3.9 Motivation for ATCP

It can be observed from table 3.1 3.2, that it is desirable for an approach to require modification only at MH compare to requiring support from BS also. It enable the approach to be applicable even when traffic is encrypted or acknowledgement pass through different path. In addition to this, the approach neither hamper interoperability nor introduce scalability difficulty. But only few approaches fall into this category, the 3DA (3-Duplicate Acknowledgement approach [21]) and Freeze TCP [30].

Though the 3DA approach increases the TCP response time by reducing idle time after mobile host get reconnected to the network. But it has the degrading side-effect on throughput. It causes TCP to enter in fast recovery mode after receiving 3 duplicate acknowledgements. This results in reduction in congestion window which reduces the transmission rate. As a result, lesser throughput is achieved over the connection.

Freeze TCP has the disadvantage of requiring lower layers to predict future disconnections. The paper suggests that by monitoring signal strength, the mobile host may predict impending

Table 3.2: Comparison of various approaches

[t]

Approach	End to end semantics preserved	Data transfer direction enhanced	Encrypted IP traffic supported	Different acknowledgement path supported	Comments
3-Duplicate acknowledgement	Yes	FH to MH	Yes	Yes	Focus on reducing Idle time after reconnection
FreezeTCP	Yes	FH to MH	Yes	Yes	Requires MH to predict disconnections
M-TCP	Yes	FH to MH, MH to FH	No	No	Split connection approach
WTCP	Yes	FH to MH, MH to FH	No	No	New algorithms at FH, MH Focus on WWANs
I-TCP	No	FH to MH, MH to FH	No	No	End to end semantics not preserved Split connection approach
Snoop	Yes	FH to MH	No	No	Improves routing protocol to reduce the handoff time and packet loss
ATCP	Yes	FH to MH, MH to FH	Yes	Yes	Focus on both way data transfer performance (delay and throughput)

disconnection. But, it is difficult for the MH *to detect the future disconnections with sufficient accuracy*. Also the issues arise of how early this prediction should be available. If it is earlier than the RTT of the connection, FreezeTCP's action of freezing the sender may lead to degraded performance. It can be noticed that accurate predictions depend on round trip time and that different TCP connections may have different RTT values.

Additionally, these approaches have not mention what actions should be taken when data is transferred from MH to FH, after getting the information about ongoing mobility from lower layers.

All these factors lead us to design of a new approach called ATCP, which require modification at mobile host only, does not require prediction of future disconnectins, does not reduce the congestion window of the sender after mobile host get reconnected while it reduces the idle time successfully. ATCP also focuses on MH to FH data transfer. It utilises network layer feedback about the status of connectivity to networks. Lower layers give connectivity status in terms of connection signal and disconnection signal, corresponding to connection or disconnection with the network. Details of ATCP will be given in next chapter and results of comparison through simulation in ns-2 will be presnted in chapter 6.

3.10 Summary

In this chapter, we reviewed many existing approaches which attempt to alleviate the degrading effect of mobility on TCP performance. Some approaches require modifications at base-station and mobile host while some other approaches requie modifications only at mobile host. Later approaches have many advantages over the former one, like support for encrypted traffic, different acknowledgement path, interoperable and scalable. Motivation for new approach ATCP was also discussed. Details of ATCP will be described in next chapter.

Chapter 4

Our Approach: ATCP

4.1 Introduction

ATCP is an approach to improve data transfer performance in mobile wireless networks in the presence of temporary disconnections caused by mobility. It is designed to improve performance from FH to MH as well as MH to FH, but involves modifications to the network stack only at the MH. ATCP requires network layer feedback regarding the status of connectivity with the network in the form of *connection* and *disconnection* signals. The network layer sends a *connection* signal when the MH gets connected to the network and a *disconnection* signal when the MH gets disconnected from the network. We believe that this is a reasonable requirement since in mobile wireless networks the network layer typically has this information (for instance, Mobile IP [16]).

For MH to FH data transfer, ATCP uses these signals to freeze or unfreeze data transmission and changes the action taken at RTO (retransmission timer out) event. ATCP assumes that if a disconnection has occurred, acknowledgements may be lost during the disconnection. With this assumption, ATCP modifies the action taken by TCP at occurrance of RTO event. At RTO event it checks whether a has disconnection occurred, if so then instead of reducing ssthresh, ATCP increases the ssthresh to the value of cwnd at the time of disconnection. This action helps ATCP in regaining earlier achieved window in slowstart phase itself, thus reducing under-utilization of the available link capacity.

For FH to MH data transfer, ATCP uses connection signal to first freezing and then unfreezing the FH sender. It freezes the sender with zero window advertisement (ZWA) and unfreeze it with full window advertisement (FWA). Freezing causes FH to freeze it retransmission timer, without reducing its congestion window. Unfreezing results in FH sending all unacknowledged packet with no reduction in congestion window. Thus, ATCP reduces idle time after reconnection without reducing congestion window unlike 3DA approach.

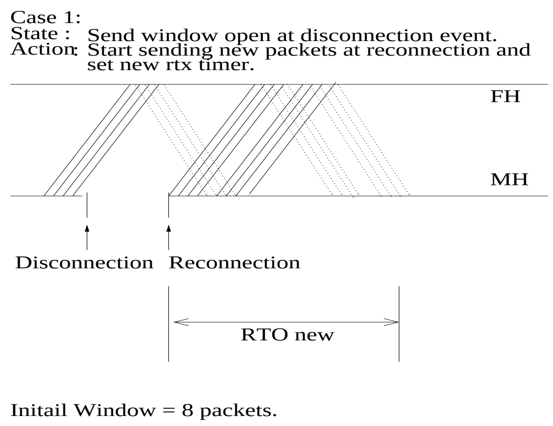


Figure 4.1:

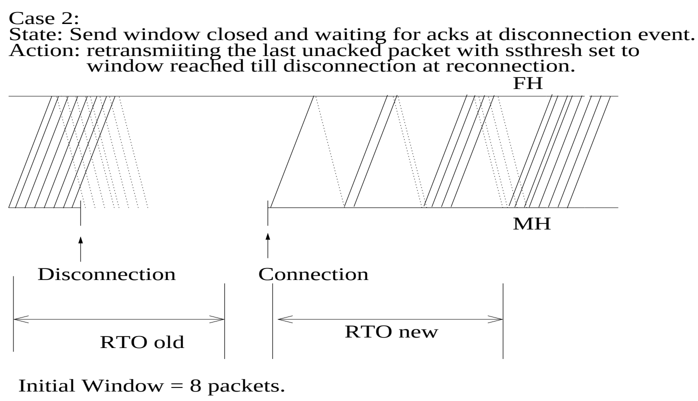


Figure 4.2:

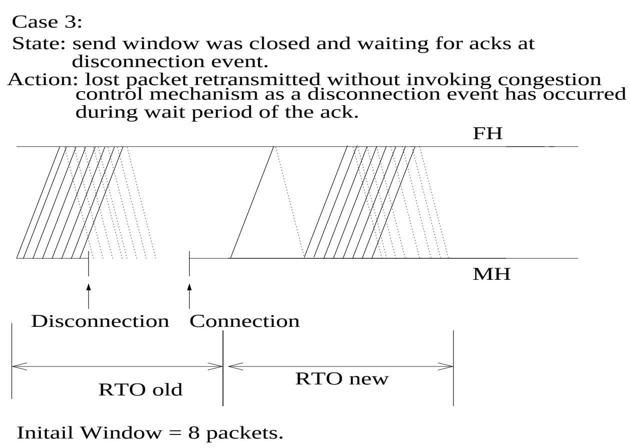


Figure 4.3:

4.2 Description of ATCP

The algorithm is presented in *Algorithm part 1* for MH to FH data transfer and in *Algorithm part 2* for FH to MH data transfer.

4.2.1 Data Transfer from MH to FH

The actions taken by ATCP in various scenarios are described.

- Disconnection event:
  - If the MH's transmission was interrupted and its sending window was open (case 1, see fig. 4.1), the MH does not wait for acknowledgements of packets sent before disconnection, therfore cancels the retransmission timer (RTX).
  - If MH's sending window is closed and it was waiting for acknowledgements to arrive, it does not cancel the retransmission timer. ATCP assumes that the acknowledgements might have been lost during the disconnection event and behaves accordingly on the occurence of an RTO event.
- RTO event:
  - if MH is disconnected from the network(case 2, see fig. 4.2), ATCP increases the ssthresh to the window size reached before disconnection and sets the cwnd to one.
  - if MH is connected to the network (case 3, see fig. 4.3), retransmits the lost packet without modifying the ssthresh and cwnd parameters. This is done when a disconnection has occurred while waiting for acknowledgement to come. Otherwise normal TCP actions are taken.
- Connection event:
  - if send window is open, MH sends new data as permitted by the open window and sets a new retransmission timer. As acknowledgements are cumulative, the new acknowledgement will acknowledge the data sent before disconnection.
  - if send window is closed
    - If RTO event has happened while MH was discnnected, ATCP starts sending new data and quickly reaches the previous window size since ssthresh is set to the earlier full window value. The low value of cwnd (i.e. 1) at reconnection will reduce the probability of congestion in the network.
    - else ATCP waits for RTO event.

ATCP

Events

SendCall: Application layer deliveres data to TCP for transmission.

AckRx: TCP has received acknowledgement from peer TCP layer at other end.

RTO: Retransmission Timer at MH has expired.

Disconnection: Lower layer has given disconnection signal to TCP layer.

Connection: Lower layer has given connection signal to TCP layer.

DataRcvd: Data packet received from peer TCP layer of FH.

AckSend: time to send pending acknowledgement.

Variables

NetworkStatus: It maintains status of connectivity with the network.

StatusRecorded: It records the state of MH TCP at the time of last disconnection event. It will be *sending* if MH window is open and it has more data to send or *waiting* if MH has window closed and waiting for acknowledgement or *idle*.

DisconnectionOccurred: It is use to record that a disconnection event has occurred when waiting for acknowledgement after sending packets from present window. It used at the time of RTO event to check whether MH got disconnected while acknowledgements were in transit.

RTOoccurred: This is used to record the occurrence of a RTO event when MH was disconnected.

DuplicateAllowed: This is used to check that if duplicate acknowledgments can be send or not.

d: The delay given to last acknowledgement when data transfer occurs from FH to MH.

AckPending: This is TRUE if any acknowledgement is pending for transmission.

LastSentAck: record the sequence no of last sent acknowledgement.

PendingAckSeq: record the sequence no to be acknowledge.

srtt: This is the smooth round trip time at mobile host.

Initialisation

NetworkStatus = connected

StatusRecorded = idle

DisconnectionOccurred = FALSE

RTOoccurred = FALSE

DuplicateAllowed = TRUE

d = 200ms

AckPending = FALSE

status()

if packets in send buffer & send window is open **then**
  return sending ;
end if
if send window is closed & waiting for acknowledgements **then**
  return waiting;
end if
return idle ; /* connection is idle */

SendCall()

if $NetworkStatus == connected$ **then**
  normal TCP Behaviour ;
else
  Buffer the data ;
end if

```
Disconnection()
  NetworkStatus = Disconnected ;
  StatusRecorded = Status() ;
  if StatusRecorded == Idle then
    no action ;
  end if
  if StatusRecorded == Sending then
    Cancel the Retransmission Timer (RTX) ;
    Stop Sending ;
    Record sequence. no. of last send packet ;
  end if
  if StatusRecorded == Waiting then
    RTOoccurred = FALSE ;
    DisconnectionOccurred = TRUE ;
  end if
RTO()
  if NetworkStatus == Connected then

    if DisconnectionOccurred == TRUE then
      DisconnectionOccurred = FALSE ;
      retransmit last sent packet;
    else
      normal TCP behaviour;
    end if
  else
    ssthresh = max( ssthresh, cwnd, receiver. advertised window) ;
    cwnd = 1 ;
    no change in RTO value ;
    RTOoccurred = TRUE ; /* record this event to be checked at reconnection */
  end if
Connection()
  NetworkStatus = Connected ;
  if StatusRecorded == idle then
    normal TCP Behaviour;
  end if
  if StatusRecorded == sending then
    Resume sending the packet from last sent packet onwards ;
    Set RTX timer ;
  end if
  if StatusRecorded == waiting then
```



```
    if RTOoccurred == TRUE then
      Retransmit last sent packet;
      RTOoccurred = FALSE ;
    else
      no action; /* wait for RTO to occur, acknowledgements may be in transit */ ;
    end if
  end if
AckRx()
  if (is this acknowledging new data) then
    /* Last disconnection has not cause loss of packet, should be ignored */
    DisconnectionOccurred = FALSE;
  end if
```

Algorithm 1: For Data Transfer from MH to FH

```
DataRcvd()
  /* let sequence no. to acknowledge as per TCP Reno algorithm is S */
  if NoAckPending then

    if S > LastSentAck then
      /* keeping last 2 bytes unacknowledged */
      send acknowledgement with sequence no S-2 ;
      LastSentAck = S-2 ;
      PendingAckSeq = S ;
      set timer to invoke AckSend() call after d ms ;
      AckPending = TRUE ;
    else

      if DuplicatAllowed then
        send acknowledgement with S sequence no.;
      else
        return ; /* suppressing Dup-Acks after sending ZWA & FWA */
      end if
    end if
  else

    if PendingAckSeq < S then
      send acknowledgement with sequence no S-2 ;
      LastSentAck = S-2 ;
      PendingAckSeq = S ;
      reset timer to invoke AckSend() call after d ms ;
      AckPending = TRUE ;
    else
      /* S is duplicate acknowledgement: so sending 2 Dup-Acks as would be send by TCP-
      Reno */
      send two acknowledgement with seq. no. S;
      AckPending = FALSE ;
      cancel timer used for invocation of AckSend() call;
    end if
  end if
AckSend()
  if NetworkStatus == connected then
    Send acknowledgement with seq.no. = PendingAckSeq ;
    LastSentAck = PendingAckSeq ;
    AckPending = FALSE ;
  else
    return ;
  end if
Disconnection()
  NetworkStatus = disconnected ;
Connection()
  NetworkStatus = connected ;
  if AckPending then
    /* Let the pending acknowledgement has advertised window = W, S = PendingAckSeq */
    send acknowledgement with sequence no = S-1 and advertised window = 0; (ZWA)
    send acknowledgement with sequence no = S and advertised window = W ; (FWA)
    DuplicateAllowed = FALSE ;
    set timer for DupAlwd() call for duration of 2 * srtt.
  end if
DupAlwd()
  DuplicateAllowed = TRUE
```

Algorithm 2: For Data Transfer from FH to MH

4.2.2 Data Transfer from FH to MH

- DataRcvd: ATCP acknowledges all the bytes received so far except the last two bytes during active data transfer from FH to MH. The acknowledgement for these two bytes will be delayed at most by d msec, so that FH does not have to wait until it times out. This action of delaying the acknowledgement for the last two bytes increases the probability that in case of a disconnection event, ATCP has the last two bytes unacknowledged.

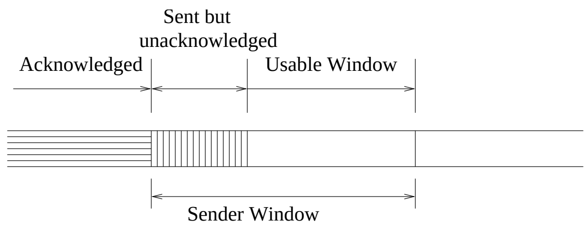
- Disconnection; Update network connectivity status.

- Connection: At reconnection, ATCP uses these two unacknowledged bytes to send ZWA (zero window advertisement) and FWA (full window advertisement) acknowledgements. *The FH will process these acknowledgements as they have a higher sequence number* [23] than all acknowledgements seen so far. The first acknowledgement is with a Zero Window Advertisement (ZWA) and the second with a Full Window Advertisement (FWA) to open the window. The ZWA causes the FH to assume that all the unacknowledged packets are dropped by the MH as it will not have space in its buffer. Therefore, FH freezes sending the data and also freezes the retransmission timer [23].When a FH receives a ZWA, its sending window will shrink from the right as shown in Fig. 4.4. Although a receiver is discouraged from such an action since the usable window [25] becomes negative, the sender must recover from this [23]. The FWA causes FH to resume sending the unacknowledged data as FWA opens up the sender's window. This action ensures that FH will not take congestion control action when packets are lost while MH was disconnected. Thus ATCP causes FH to retransmit the packets lost during disconnection without any invokation of congestion control mechanisms.

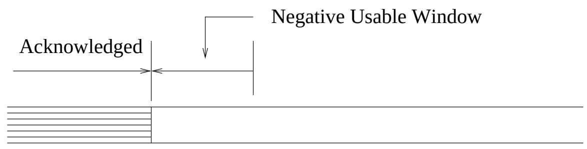
- DupAlwd: This event allow ATCP to send duplicate acknowledgement. Transmission of duplicate acknowldgement is disabled after reconnection for a time period of one RTT. This is done with considering the case when FH might have sent data packets before receiving the ZWA and FWA acknowledgements. These packets may cause the MH to generate duplicate acknowledgements instead of dropping these packets as these packet will be retransmitted by FH after recieing FWA. The duplicate acknowledgements subsequently may cause the FH to unnecessarily enter the fast recovery phase, reducing its congestion window. Therefore, after sending the ZWA and FWA acknowledgements, ATCP suppresses duplicate acknowledgements for one round trip time.

4.2.3 Design Ideas in ATCP

For MH to FH data transfer, ATCP is designed to behave like new connection after long disconnection (i.e. one RTO), as the characteristics of the connection may have changed due to the



FH's sending window space before ZWA received.



FH's sending window space after ZWA received.

Figure 4.4:

Figure 4.5: TCP Window Management

MH entering a new network or due to different load conditions prevailing in the same network. Therefore, the cwnd is set to one and ssthresh to window value reached when disconnection event occurred. This design ensures that ATCP will quickly achieved full window while TCP may have waited for a long time (upto 60 seconds) for restarting transmission itself, after reconnection.

In FH to Mh data transfer, The value of d in our simulations is chosen as 200ms since it is the value by which most practical implementations of TCP delay the acknowledgement [22]. The maximum allowed value for delaying the acknowledgement is 500ms according to RFC1122 [23].

4.3 Summary

In this chapter, we presented our approach, ATCP, an adapted version of TCP at mobile host. It has been designed for alleviating the degrading effect of temporary disconnections induced by mobility of the hosts. It requires modification only at mobile host and requires feedback about ongoing mobility from the lower layers. This approach falls in to the category of 3DA and Freeze TCP approach. These approaches also mitigate detrimental effect of mobility while requiring modification only at mobile host. In next chapter, we will present various simulation results and compare these approaches.

Chapter 5

Simulations

5.1 Introduction

To compare various approaches, we have implemented these in network simulator ns-2 [26]. This simulator is widely used in research community for simulations of protocols used in internet like TCP. Therfore, we are using this for comparison of various modifications to TCP. This results in a meaningful comparison of various schemes. Therefore, to compare 3DA, Freeze TCP and ATCP, we have implemented these in ns-2. Next section will provide the various implimentation details.

5.2 Implementaion Details

We have implemented 3DA, Freeze TCP and ATCP, in the network simulator **ns-2.1b8a**. We also modified ns for simulating the actions taken by TCP at the FH when zero window advertisement packet is received. Mobility of the MH is simulated by maintaining a variable, *NetworkStatus*, in TCP Agent whose value changes from *Connected or Disconnected* as determined by a *handoff timer handler*. The handoff timer can be configured to alternately connect and disconnect to simulate handoff. We have also modified **MobileIP** [16] in ns for providing mobility information to the TCP Agent, but in this case granularity of disconnection period was in order of seconds. Therfore, we chose to simulate mobility internally that allows granularity level in the order of milli seconds. Following is the listing of file names which are modified and the purpose of modifications.

- tcp.h, tcp.cc: These files are modified for implementing zero window advertisement processing (for the case of FH to MH data transfer) and to process connection & disconnection signals given by lower layers (for the case of MH to FH data transfer).
- tcp-sink.cc: These is modified to simulate temporary disconnection of mobile host and action taken at connection & disconnection events as per 3DA, Freeze TCP and ATCP algorithms.

- mip.cc. mip-reg.cc: This is modified to give feedback about network connectivity to upper layer i.e. TCP.

Various other files are also modified for small changes which are required to fulfill above purposes.

5.3 Simulation Parameters

This section will gives the values of various parametrs involved in the simulations and justifications for those chosen values.
The capacity of both links i.e. FH to BS and BS to MH, were maintained to be equal so that no packet losses due to buffer overflow in router could occur. Throughput of TCP connections were measured for different durations of handoff periods with a constant frequency of one handoff every 10 seconds. These simulations were conducted for different values of round trip time from the FH to MH.
The simulation scenario is shown in figure 5.1. The simulation parameters and reasons for particular choices of values are stated below.
Application type: An FTP application was chosen to simulate a large data transfer.
Handoff duration: Values ranging from 50ms to 4s were chosen to cover a wide range hand-off durations. Typically small values occur in wireless LANs and large values occurs in wireless WANs [29].
Round Trip Time: The values chosen were 4ms and 600ms. The 4ms value was chosen to represent a wireless LAN (WLAN) environment where RTT values are in the order of milliseconds. The large values of 600ms were taken to represent a wireless WAN (WWAN) environment [29][5][2].
Handoff frequency: was chosen as 10 seconds, indicating moderate to high mobility.
Link capacity: Link capacity for the small round trip time (RTT) of 4ms was selected as 10Mbps to represent a WLAN [2] Link capacity for a large RTT of 600ms was chosen as 100Kbps, to represent a WWAN [2][5].

5.4 Simulation Analysis & Comparison

5.4.1 FH to MH data transfer

We will analyze and compare the performance of 3DA, Freeze TCP, ATCP and TCP-Reno, when FH is sender. Following observations could be made from the graphs plotted (see figure 5.5, **??**).

- **TCP-Reno throughput has step-wise behaviour against disconnection period:**
  It can be observed clearly in graph shown in figure **??** for **wireless LAN environments**, that TCP-Reno throughput exhibits stepwise variation as the disconnection period is lineraly increased. This is because of it does the retransmission at discrete time after packet

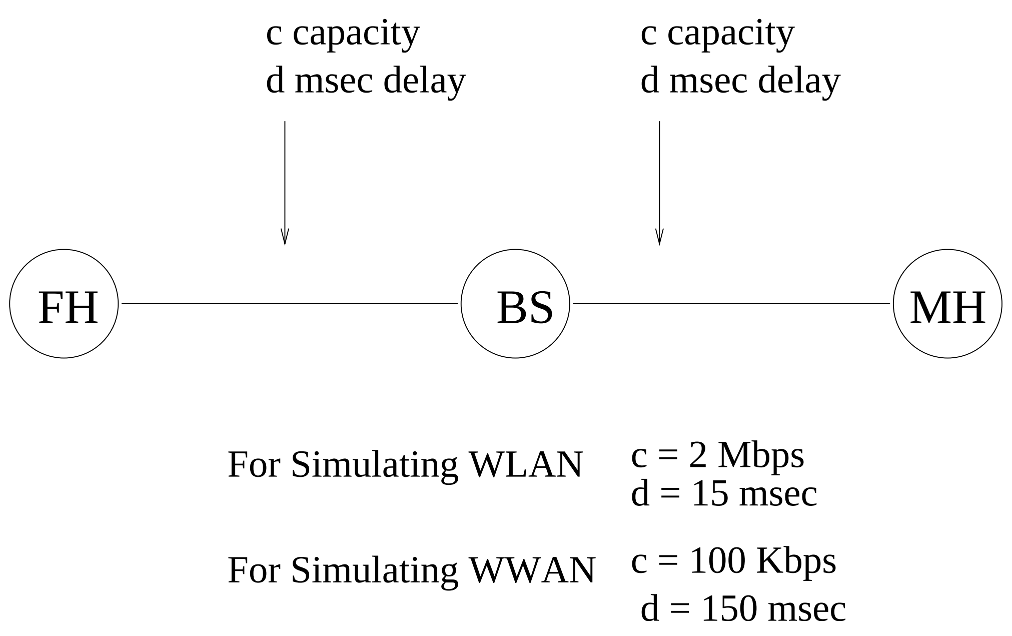


Figure 5.1: Simulation setup

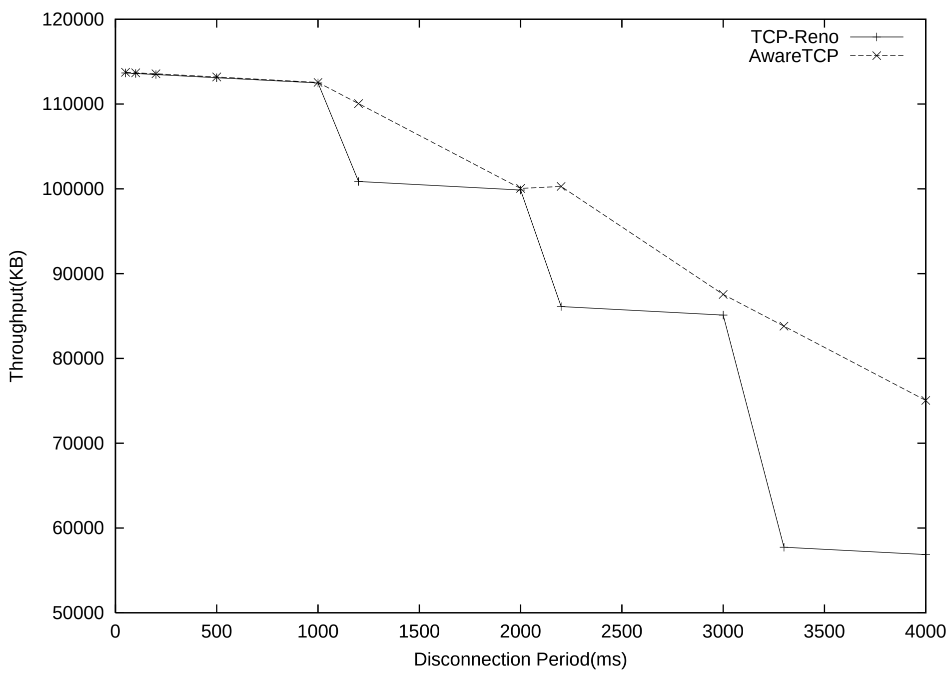


Figure 5.2: MH to FH data Transfer: RTT = 4ms

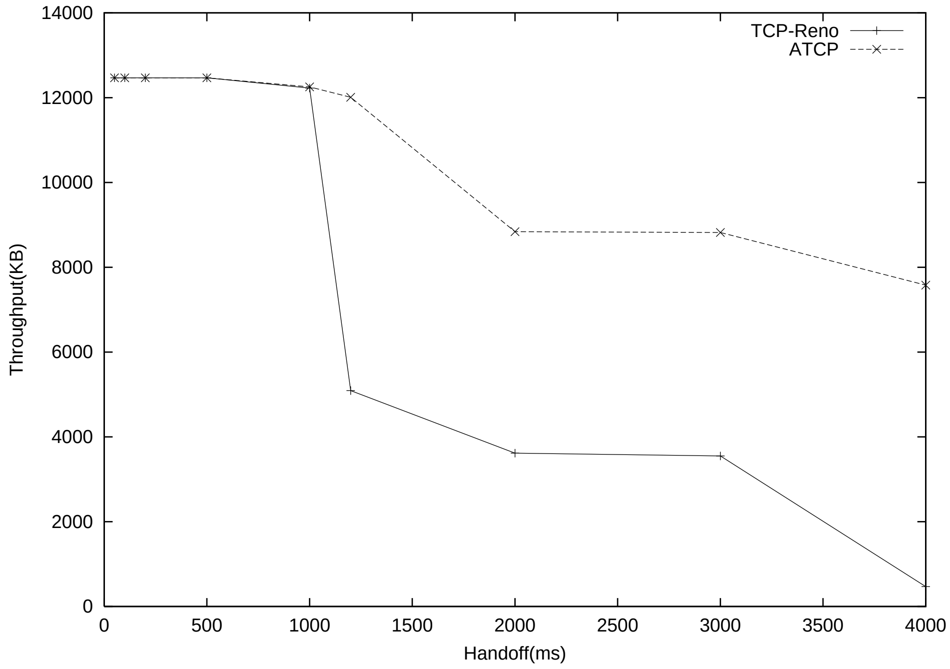


Figure 5.3: MH to FH data Transfer: RTT = 600ms

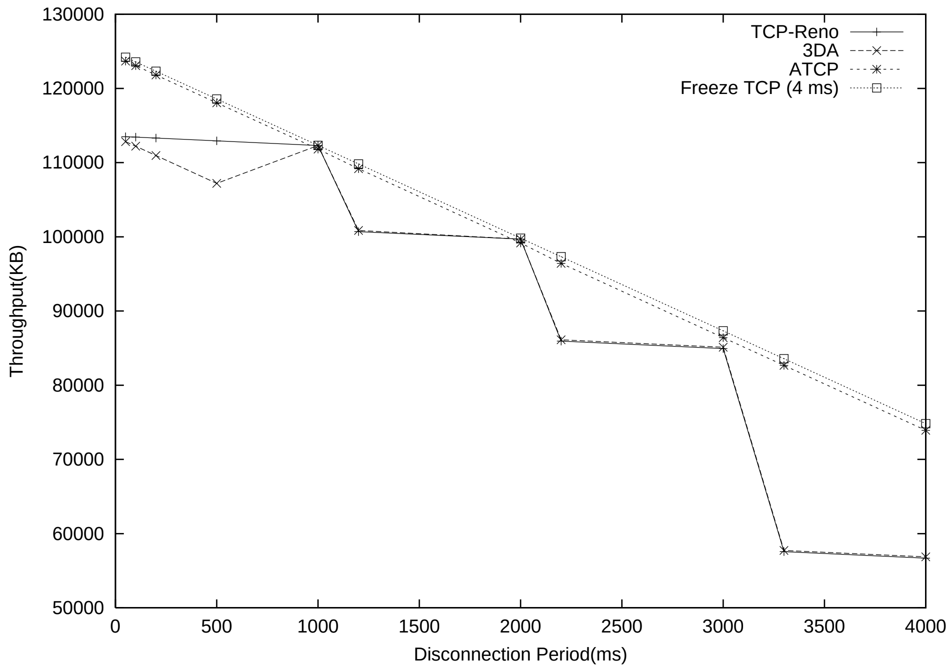


Figure 5.4: FH to MH data Transfer: RTT = 4ms

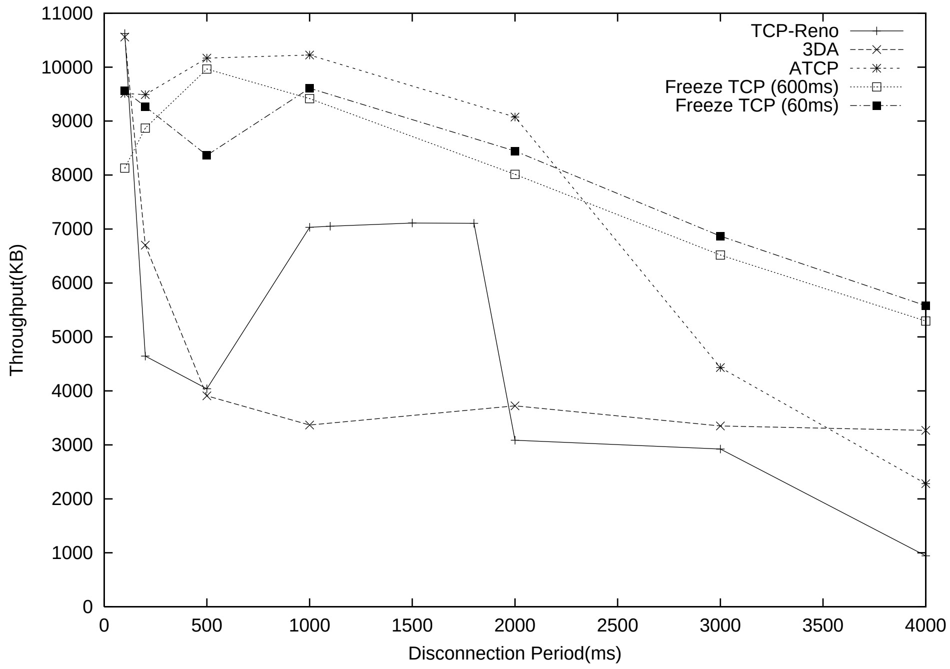


Figure 5.5: FH to MH data transfer: RTT = 600ms

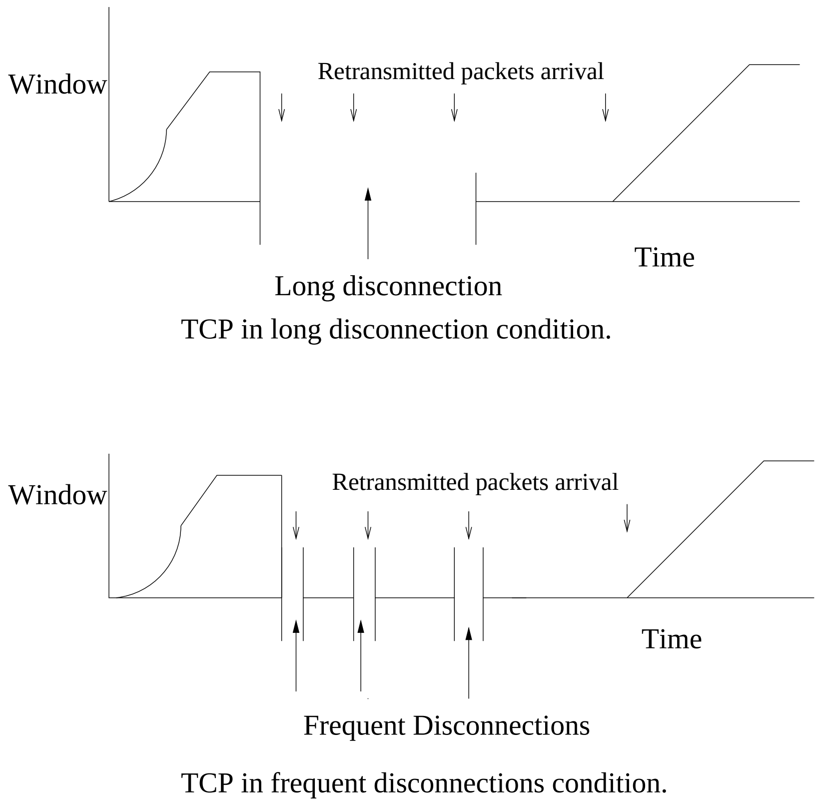


Figure 5.6:

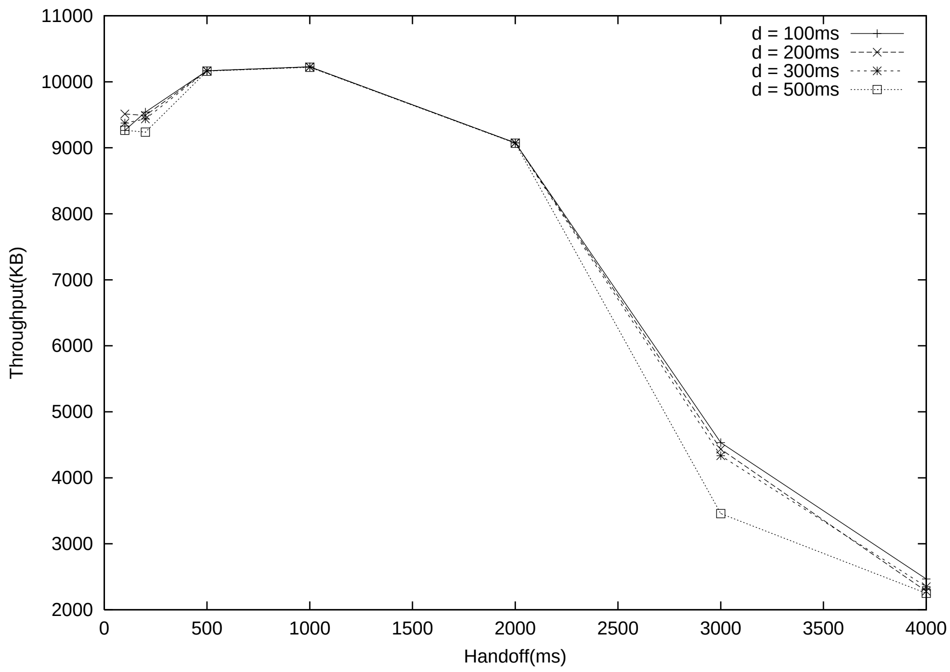


Figure 5.7: FH to MH: ATCP throughput, RTT = 600ms

losses are detected. TCP-Reno retransmitt packet after retransmission timer expire. So, when packets are lost either in small disconnection period or longer, TCP is going to retransmitt at RTO event only. Thus, there is same idle time over the connection for a range of disconnection period. For disconnection period, when retransmiited packet is also lost, a major change in throughput occurs copare to disconnection period when retransmitted packet is not lost. Then, again for a range of disconnection period, throughput remains almost same.

In case of **wireless WAN environments**, refer figure 5.5, we observed that throughput of TCP-Reno is less for 500ms disconnection period compare to 1000ms disconnection period and after that it follows step-wise behaviour as described earlier. For 500ms disconnection period, TCP-Reno enters into fast recovery mode and due to multiple packets losses, it could recovers slowly. Slow recovery is caused by deecting further packet losses by retransmission timer out event. For 1000ms disconnection period, it does not enter in fast recovery mode as it detects all packet loss by RTO event and able to quickly recover from it.

- **3DA approach does not always improves response time:**

  We observed during simulation that 3DA is not able to reduce idle time after MH is reconnected even though it sends 3 duplicate acknowledgement. This happens when there are multiple packet losses in a window. These losses make TCP-Reno first to take fast recovery action and then detect further packet losses by retransmission timer out. 3DA approach could not help in this case because TCP sending window is already has small value.

- **3DA, ATCP, Freeze TCP approach some times degrades the throughput:**

  3DA approach causes TCP-Reno to reduce it window, which is some time not required as TCP-Reno just has retransmitted the lost packet after RTO event. So, these extra 3 duplicate packets becomes useless. This is evident from the graph in figure **??**, here 3DA has almost same throughput as TCP-Reno.

  Additionally, some of disconnection period is not experience by TCP-Reno, as its packets were in transit, but 3DA, ATCP & Freeze TCP takes corrective action even for these disconnections which is undesired. Thus, causing some degradation in throughput, refer figure 5.5 for 50ms disconnection period.

- **In WLAN environment, ATCP and Freeze TCP performs almost equally well:**

  In wireless LAN environment,round trip time (RTT) is small which helps in achieving maximum window value again after a congestion control action. Therefore, in these environment, idle period (the period after MH got reconnected and data transfer starts) is primary factor in degrading the throughput rather than reduced congestion window value. As ATCP and Freeze TCP, both reduces the idle period, they almost achieved same throughput, see figure **??**.

- **In WWAN environment, ATCP performs almost equal to Freeze TCP for small disconnection period:**

  In these environment, both the idle period and reduced congestion window play significant role in degrading the throughput. ATCP achieves almost the same throughput as Freeze TCP when no reduction in retransmission timer out event has occurred, as result congestion window value remains unchanged. For long disconnection period, ATCP is able to reduce idle period, therefore performs better than TCP-Reno, but could not prevent reduction in congestion window value, so have lesser thoughput than Freeze TCP.

- **Freeze TCP throughput is sensitive to prediction period variation:**

  Freeze TCP requires lower layer to predict disconnection of mobile host in advance, so that it can take proper actions. We observed that throughput is function of this advance period, if it is too early, then throughput may decreases compared to TCP-Reno. The variation in throughput for different prediction period are plotted in graph, figure 5.5, prediction periods are 600ms (one RTT) and 60ms.

- **The value of d for ATCP:**

  Different values of the variable d were tried to find an optimum value, see figure **??**, 5.7. For small RTT connection, we vary d from 20ms to 500ms and for large RTT connection, we vary it from 100ms to 500ms. The graphs show that the value of d does not make much difference in throughput achieved, except at the large value of 500ms. In this case , the throughput decreases sometimes due to large idle period over the connection. Since acknowledgements are generally delayed by 200ms [22], the value of d can be selected to 200ms.

- **Thoughput Comparison of 3DA, Freeze TCP and ATCP:**

  In wireless LAN environment, TCP-Reno and 3DA approach have almost same throughput, and ATCP and Freeze TCP are able to improve throughput up to 30 to 40% for long disconnection period compare to TCP-Reno throughput.

  In wireless WAN environment, significant improvements are observed by ATCP and Freeze TCP in compared to TCP-Reno and 3DA. For disconnection period range to 1000ms, ATCP and Freeze TCP achieved up to 150% improvement and for longer disconnection ATCP achieved up to 50% and Freeze TCP more than 150%.

5.4.2 MH to FH data transfer

In ATCP, we have considered this scenario and specified the action to take upon reception of connection & disconnection signal from network layer. We have noticed that author of 3DA and Freeze TCP approach does not mention any specific action to take upon receiving signal about mobility from network layer, while mobile host is sender. Therefore, we have done the comparison of ATCP with TCP-Reno only. Following observations are made from the graphs (See figure **??**, 5.2).

- **TCP-Reno throughput has step-wise behaviour against disconnection period:**

  This occurs due to the same reason , for which it occurs in case of FH to MH data transfer.

- **The enhancement in throughput increases as the duration of a single disconnection increases**. This increase in performance is due to the following reasons:

  - TCP Reno backs off exponentially when an RTO event occurs during the disconnection. When the MH is reconnected, TCP-Reno waits for its retransmission timer to expire. ATCP does not have this idle period which results in enhanced performance. Also, as the disconnection period increases, the number of RTOs increases which in turn increases the waiting time for TCP-Reno to attempt retransmission (see fig.5.6).
  - At each RTO event, TCP-Reno decreases the slow start threshold (ssthresh) by half. This is undesirable if it is occured due to MH is disconnected. ATCP sets the ssthresh to window value reached upto disconnection instead of decreasing it. This results in ATCP achieving full capacity quickly when compared to TCP-Reno.

- **Large Improvements in throughput for large RTT connections:**

This is due to the large value of retransmission timer. At every RTO event, this value doubles. Due to this TCP-Reno achieved less throughput compared to ATCP as shown in Fig. 5.3.

- **Percentage improvement in throughput:**

  TCP-Reno over a small RTT connection is able to quickly reach its full window size. Therefore, percentage improvement of upto 40% is observed over TCP-Reno for long disconnection period. This is achieved primarily by eliminating the idle period after reconnection. Large RTT connections suffer due to the large idle period after disconnection and the large time taken to reach the full window size. ATCP achieves an increase of upto 150% in throughput for disconnection period of 3000ms.

5.5 Summary

Network simulator ns-2 is used for comparing various approaches i.e. 3DA, Freeze TCP, ATCP with TCP-Reno. For Fh to MH data transfer, it has been shown that in the case of multiple packet losses in a single window, sometimes 3DA is not able to reduce idle time and some times it also decreases the throughput in comparison to TCP-Reno. ATCP is shown to perform comparable to Freeze TCP in wireless LAN environment. ATCP is also able to give equal throughput as Freeze TCP for short disconnection period in wireless WAN environment. ATCP does not require disconnection prediction as by Freeze TCP, which is a major drawback of Freeze TCP. Throughput of Freeze TCP is also shown to be function of prediction period and this is undesirable.

For MH to FH data transfer, ATCP is compared with TCP-Reno only, as 3DA and Freeze TCP have not consider this direction of data communication. ATCP performs better than TCP-Reno and the improvement is shown to be larger over wireless WAN environment. In next chapter, we will draw the conclusion and mention the scope for future work.

Chapter 6

Conclusion and Future work

6.1 Conclusion

We analyzed the issue of poor performance of TCP over mobile wireless networks. The high error rate and intermittent disconnections induced by mobility is found to be the primary factors which degrads the performance of TCP. We observed that while several approaches have been put forth by reaserach community for alleviating the detrimental effect of poor charactreistics of wireless channel, only few approaches are presented for solving the mobility related issues. Even in these few attempts, mostly requires support from base-station like some state per TCP connection, some changes in protocol stack etc. This requirment of these approach make them undesirable as these schemes could not support encrypted traffic, different acknowledgement path, also they introduced scalability problem and hinder the inter-operability of the mobile host with different networks. All these observations motivated us to design of a new approach called ATCP. ATCP falls into the category of those approaches which mitigates the degrading effect of mobility while requiring modification only at mobile host like 3DA, Freeze TCP approach. The ATCP is a modified version of TCP at mobile host end. It requires the feedback about ongoing mobility from network layer. We compared ATCP with 3DA, Freeze TCP and TCP Reno. This is also noticed that the authors of 3DA and Freeze TCP approach have not given any specific action to be taken with regrad to MH to FH data transfer. ATCP is shown to improve performance of TCP data transfer in both direction i.e. FH to MH and MH to FH. For data transfer from FH to MH, ATCP is shown to reduce response time as 3DA approach does, but it also achieve higher TCP throughput in most cases. It has been shown that for small disconnection period ATCP and Freeze TCP performs equally well, only in case of long disconnection period Freeze TCP performs better than ATCP. But ATCP has a significant advantage over Freeze TCP that it does not require prediction of impending disconnections.

Thus, we can conclude that ATCP is an approach which has reasonable requirement of feedback from network layer about network connectivity status and performs comparable to the 3DA and Freeze TCP approach for FH to MH data transfer and better than TCP-Reno for MH to FH data transfer.

6.2 Future work

We have simulated our approach on ns-2 network simulator. Though these simulations assisted us in testing our apprach and comparing it with others, yet exact real life scenario could not be created as it is the limitation of the simulator. So, we would like to implement our approach in real protocol stack and test its inter working with several real life TCP implementations in different operating systems.

By testing ATCP with real life TCP Fixed Host implementation, we could also find out how various implementations behave when their sending window goes negative. Ideally, they should be able to recover without any failure as per RFC1122 requirements.

We also want to explore about the various possible way by which lower layers can give feedback about network connectivity to TCP layers.

References

[1] S. Mascolo and Claudio Casetti, *TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links*, ACM SIGMOBILE 7/01 Rome italy, ACM ISBN 1-58113-422-3/01/07, July 2001.

[2] George Xylomenos, G.C. Polyzos, Petri Mahonen and Mika Saaranen, *TCP Performance Issues over Wireless Links*, IEEE Communications Magazine, April 2001.

[3] Cooperative Association for Internet Data Analysis, *www.caida.org*, April 2001.

[4] K.Pahkwan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J. Makela, R. Pichma, and J. Vallstrom, *Handoff in Hybrid Mobile Data Networks*, IEEE Personal Communications Magazine, vol.7, no.2, April 2000.

[5] Roger Kalden, Ingo Meirick and Michal Meyer, *Wireless Internet Access Based on GPRS*, IEEE Personal Communications, April 2000.

[6] Jochen Shiller, *Mobile Communications*, Addison-Wesley, 2000

[7] V. Paxson, M. Allman, *RFC 2988, Computing TCP's Retransmission Timer*, November 2000.

[8] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan, *WTCP: a reliable transport protocol for wireless wide-area networks*, Proceedings of ACM MOBICOM 99, Seattle, Washington, August 1999.

[9] K. Ratnam, I. Matta *WTCP: An efficient Mechanism for improving TCP Performance over Wireless Links*, 1999, http://www.citeseer.com.

[10] T. Goff , J. Moronski. *Freeze-TCP: A True end-to-end TCP enhancement mechanism for mobile environment*, 1999, http://www.citeseer.com.

[11] Nitin Vaidya and Miten Mehta, *Delayed Duplicate Acknowledgement : A TCP-Unaware approach to improve performance of TCP over wireless*, 1999, http://www.citeseer.com.

[12] S. Kent, R. Atkinson *RFC 2401: Security Architecture for the Internet Protocol*, November 1998.

[13] K. Thompson, G.J. Miller and R. Widler *Wide-area Internet traffic patterns and characteristics*, IEEE Networks, Vol 11, no. 6, November 1997.

[14] H. Balakrishnan, V. N. Padmanabhan, S. Sechan and R.H. Katz, *A Comparison of Mechanisms for Improving TCP Performance over Wireless Links*, IEEE/ACM Transactions on Networking, December 1997.

[15] K. Brown and S. Singh *M-TCP: TCP for Mobile Cellular Networks*, ACM Computer Communications Review, vol27, no.5, 1997.

[16] C. Perkins, *RFC 2002: IP Mobility Support*, October 1996.

[17] B. Bakshi, P. Krishna, N.H.Vaidya *Improving Performance of TCP over wireless Networks*, 1996, http://www.citeseer.com.

[18] Pertro Manzoni, Dipak Ghosal and Giuseppe-Serazzi, *Impact of Mobility on TCP/IP: An Integrated Performance Study*, IEEE JSAC, VOL. 13, NO. 5, June 1995.

[19] Ajay Bakre, B.R. Badrinath *I-TCP: Indirect TCP for Mobile Hosts*, Tech Rep., Reuters university, May 1995, http://www.cs.rutgers.edu/ badri/journal/contents11.html.

[20] H. Balakrishnan, V.N.Padmanbham and R.Katz *Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks*, Wireless Networks, vol.1. no.4., Dec 1995.

[21] Ramon Caceres and Liviu Iftode, *Improving the performance of reliable transport protocol in mobile computing environments*, IEEE JSAC Special Issue on Mobile Computing Network, vol. 13, no. 5, June 1995.

[22] W. Richard Stevens *TCP/IP Illustrated, Volume 1, The Protocols*, AWL, 1994.

[23] R. Braden, *RFC 1122 Requirements for Internet Hosts - Communication Layers*, October 1989.

[24] Van Jacobson, Michael J. Karels, *Congestion Avoidance and Control*, ACM Computer Communication Review , Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988.

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

[26] The network simulator ns-2.1b8a/,*http://www.isi.edu/nsnam/ns/*.

[27] H. Balakrishnan, V.N.Padmanbham and R.Katz *Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks*, Wireless Networks, vol.1. no.4., Dec 1995.

[28] K. Brown and S. Singh *M-TCP: TCP for Mobile Cellular Networks*, ACM Computer Communications Review, vol27, no.5, 1997.

[29] K. Ratnam, I. Matta *WTCP: An efficient Mechanism for improving TCP Performance over Wireless Links*, 1999, http://www.citeseer.com.

[30] T. Goff , J. Moronski. *Freeze-TCP: A True end-to-end TCP enhancement mechanism for mobile environment*, 1999, http://www.citeseer.com.

[31] *Delayed Duplicate Acknowledgement : A TCP-Unaware approach to improve performance of TCP over wireless*, 1999, http://www.citeseer.com.

[32] B. Bakshi, P. Krishna, N.H.Vaidya *Improving Performance of TCP over wireless Networks*, 1996, http://www.citeseer.com.

[33] J. Schiller *Mobile Communications*, AWL, 2000.