

# TCP Extensions for Wireless Networks

Nachiket Deshpande, [deshpand@cis.ohio-state.edu](mailto:deshpand@cis.ohio-state.edu)

<http://www.cis.ohio-state.edu/~deshpand>

---

## Abstract

Traditional transport protocols (like TCP) are tuned to perform well in wired networks where the packet losses are mainly due to congestion. Wireless networks are characterized by losses due to transmission errors and handoffs. TCP interprets these losses as congestion and invokes congestion control mechanisms resulting in degradation of performance. In this paper, I discuss proposed schemes to improve performance of TCP in wireless networks. Proposed schemes for satellite networks are also discussed.

---

See Also: [Wireless Networking and Mobile IP References](#) | [Routing Protocols for Ad Hoc Mobile Wireless Networks](#) | [Evolution Toward Third Generation Wireless Networks](#) | [In-building Wireless LAN](#)  
[Other Reports on Recent Advances in Networking](#)  
[Back to Raj Jain's Home Page](#)

---

**Raj Jain is now at**  
**Washington University in Saint Louis**  
**Jain@cse.wustl.edu**  
<http://www.cse.wustl.edu/~jain/>

## Table of Contents:

1. [Introduction](#)
2. [Issues with wireless networks](#)
  - 2.1 [Link Error Rate](#)
  - 2.2 [Bandwidth](#)
  - 2.3 [Mobility](#)
3. [TCP extensions for wireless networks](#)
  - 3.1 [Wireless unaware TCP at the fixed host](#)
    - 3.1.1 [I-TCP \(Indirect TCP\)](#)
    - 3.1.2 [Berkeley Snoop Module](#)
    - 3.1.3 [M-TCP](#)
    - 3.1.4 [Delayed duplicate acknowledgements](#)
  - 3.2 [Wireless aware TCP at the fixed host](#)
    - 3.2.1 [Fast Retransmit](#)
    - 3.2.2 [Mobile TCP](#)
    - 3.2.3 [Multiple Acknowledgements](#)
    - 3.2.4 [Discriminating congestion losses from wireless losses](#)
    - 3.2.5 [Distinguish losses by making two connections](#)

4. [TCP extensions for satellite networks](#)
  - 4.1 [Issues](#)
  - 4.2 [Proposed Solutions](#)
    - 4.2.1 [Slow Start and congestion avoidance](#)
    - 4.2.2 [Fast Retransmit](#)
    - 4.2.3 [Large Window Sizes](#)
    - 4.2.4 [Delayed Acknowledgements](#)
    - 4.2.5 [Selective Acknowledgements](#)
    - 4.2.6 [TCP for transaction](#)
    - 4.2.7 [Explicit Congestion Notification](#)
5. [Conclusion](#)
6. [References](#)

[List Of Acronyms](#)

---

# 1. Introduction

Wireless networks are an emerging technology of this decade and have become very popular in last few years. Recent developments in this field demonstrate that mobile computers with wireless communication links will form an integral part of the future networks. This technology provides the user with a wireless equipment unlimited access to information and services like video telephony, e-mail, news, stock quotes etc. Future networks will be a combination of high speed wired networks and wireless links. It is required that this wireless link does not become a bottleneck in the network. The transport layer protocols (TCP) are designed to perform well for wired networks. Even a single wireless link reduces the throughput considerably. This can be avoided by buffering the packets at the network layer and by intelligent scheduling at the link layer [[Mate98](#)]. In theory we can use existing transport protocols like TCP on wireless host to communicate with the fixed network. This keeps the transport layer transparent to mobility, disconnection and other features of wireless and mobile hosts but all this at the cost of performance. In this paper we discuss some solutions to the problem of degraded performance of standard TCP on wireless networks and on satellite networks.

[\[Back to Table of Contents\]](#)

---

## 2. Issues with wireless networks

Hosts remain at a fixed position in a wired network. They are connected by cables and hence there is least transmission error due to interference from the environment. For instance, the bit error rate (BER) for a fiber optic link is usually about  $10^{-12}$  or better. In contrast, the hosts on wireless networks frequently move while communicating and as they share the media for communication they experience a lot of interference from the environment [\[Liu99\]](#).

### 2.1 Link Error Rate

Wireless hosts use radio transmission or infrared wave transmission for communication. This mode of communication is vulnerable to interference from the environment. Bit error rate on wireless links is found to be about  $10^{-1}$  or worse. This high error rate poses problems for the wired networks with a wireless link [\[Liu99\]](#).

### 2.2 Bandwidth

Bandwidth is a scarce resource in case of wireless networks. Compare the bandwidth of a typical Ethernet which is around 10 Mbps (100 Mbps for fast Ethernet) to that of Lucent Wavelan which is just 2 Mbps. Bandwidth also varies highly on wireless networks. The higher layers may have to take this into consideration and use different methods (e.g. compression) to take care of this problem [\[Liu99\]](#).

### 2.3 Mobility

Wireless hosts may move frequently while communicating. During this movement the data sent to the wireless host is lost. TCP at the destination interprets this loss as congestion and invokes congestion control mechanisms, which is unnecessary, as when the move is complete the wireless host will start receiving data again. This causes the performance of TCP to degrade. There also might be degradation of performance due to frequent recalculation of routes to the moving wireless host.

[\[Back to Table of Contents\]](#)

---

## 3. TCP extensions for wireless networks

Several schemes have been proposed to improve performance of TCP over wireless links. These can be classified into two classes. In first approach the TCP sender is unaware of the losses due to wireless link so the TCP at the sender need not be changed. In second approach the sender is aware of the existence of the wireless link in the network and attempts to distinguish the losses due to wireless link from that due to congestion. So the sender does not invoke congestion control algorithms when the data loss is due to the wireless link [\[Bala97\]](#). We describe the proposed solutions in this section.

## 3.1 Wireless unaware TCP at the fixed host

In this approach, the non-congestion related losses are hidden from the TCP at the fixed host (sender) and hence the TCP at the fixed host remains unmodified. This approach is based on the intuition that as the problem is local it should be solved locally and the TCP should be independent of the behavior of the individual links. We now present some solutions based on this approach.

### 3.1.1 I-TCP

This was implemented at Rutgers University as a part of the Dataman Project. The scheme works by breaking the connection between the machine on the fixed wired network and the wireless mobile host in two connections. One connection is between the fixed host and the base station and the other connection is between the base station and the wireless host. Data sent to the wireless host is first received by the base station, it then sends an acknowledgement to the fixed host and then the data is forwarded to the wireless host. The base station and the wireless host need not use TCP for communication instead a specialized protocol that is optimized for mobile applications and for low speed and unreliable wireless medium can be used. 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. Because of this indirection the wireless host can be very simple and the base station can handle much of the complexity about communication overhead. If the wireless host moves to a different cell while communicating with a fixed host, the whole connection information is transferred from the current base station to the new base station and the new base station takes over from here. The fixed host is unaware of this indirection and is not affected when this switch occurs.

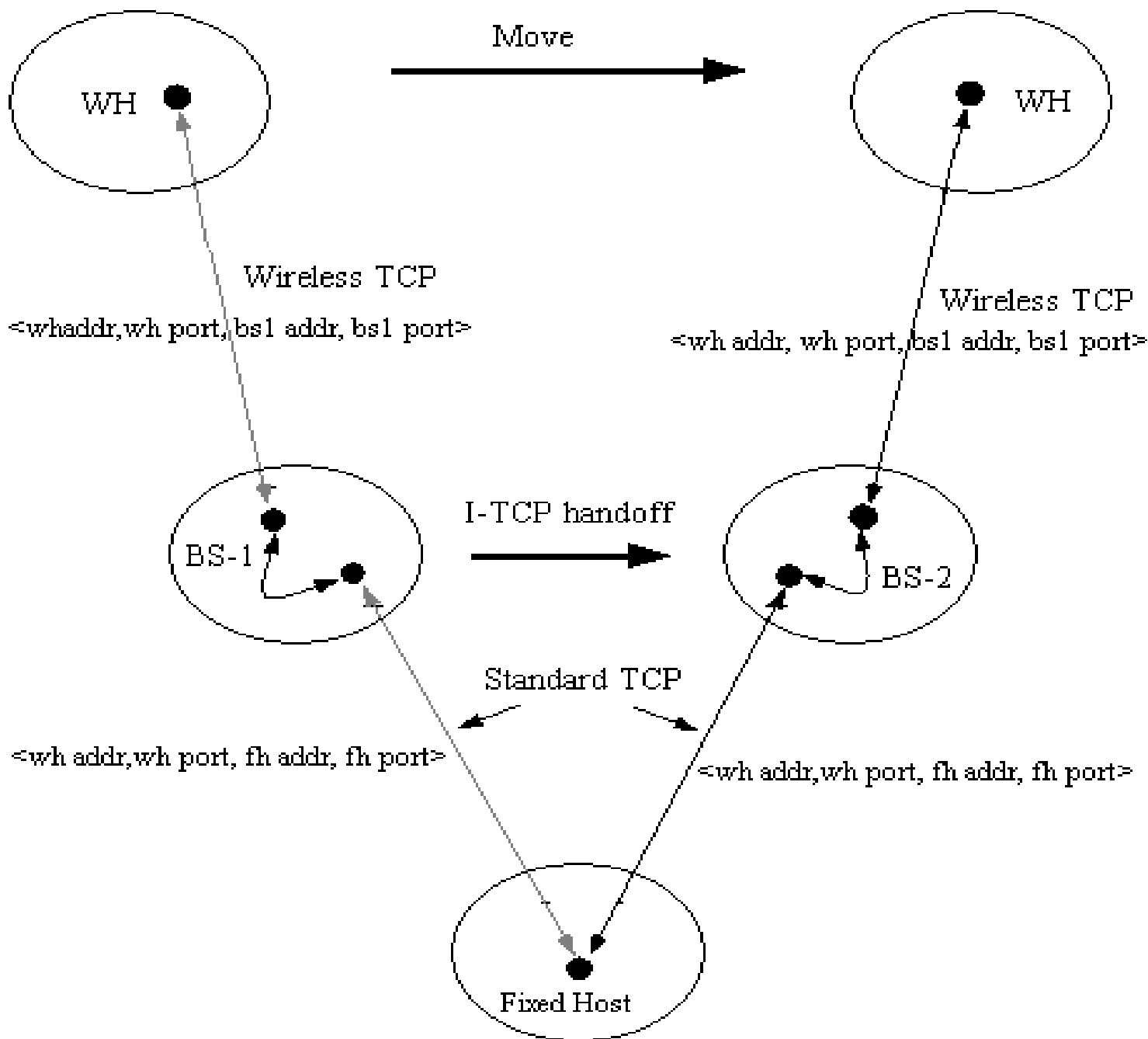


Figure 1. I-TCP Connection transfer

Figure 1 explains the operation of this protocol. The wireless host in the figure is using the I-TCP. When it wants to open a connection with a host on a wired network it sends a request to its base station (BS1). This base station now opens a TCP connection with the wired host on behalf of the wireless host. The wireless host and the base station communicate through a separate connection. Now the wireless host moves to a different cell with base station BS2. The state associated with the connection is transferred from the BS1 to BS2. As the endpoints for the connection don't change for both the wireless host and the wired host new connection need not be established and hence this indirection is transparent to the fixed host [Badrinath95].

The drawbacks of this method are:

1) If there are frequent handoffs then the overhead related to the connection state transfer between the

base stations may be large and add delays.

2) The base stations have to be complex and with large buffers in case of heavy traffic.

### 3.1.2 Berkeley Snoop Module

In this approach the network layer software is not modified anywhere in the wired network except at the base station. The router code at the base station is modified to cache data meant for the wireless host. A layer called, snoop layer is added to the base station [Bala95][Katz95]. This layer looks at every packet on the connection in either direction. The module caches the packets that are sent by the fixed host to the wireless host but have not yet been acknowledged by the wireless host. Actions at the base station when it receives a data from the fixed host are shown in fig.2. On receiving a packet from the fixed host the module stores the packet in its cache and then passes it to the wireless host. If the packets are lost on the wireless link then the base station gets repeated acknowledgements for the lost segment from the wireless host. This scenario has been explained in fig.3. The snoop module on detecting this loss checks if it has the packet in the cache if it has the packet retransmits the packet and suppresses the ACK to the fixed host, otherwise forwards the ACK to the fixed host and lets the sender recover from the loss.

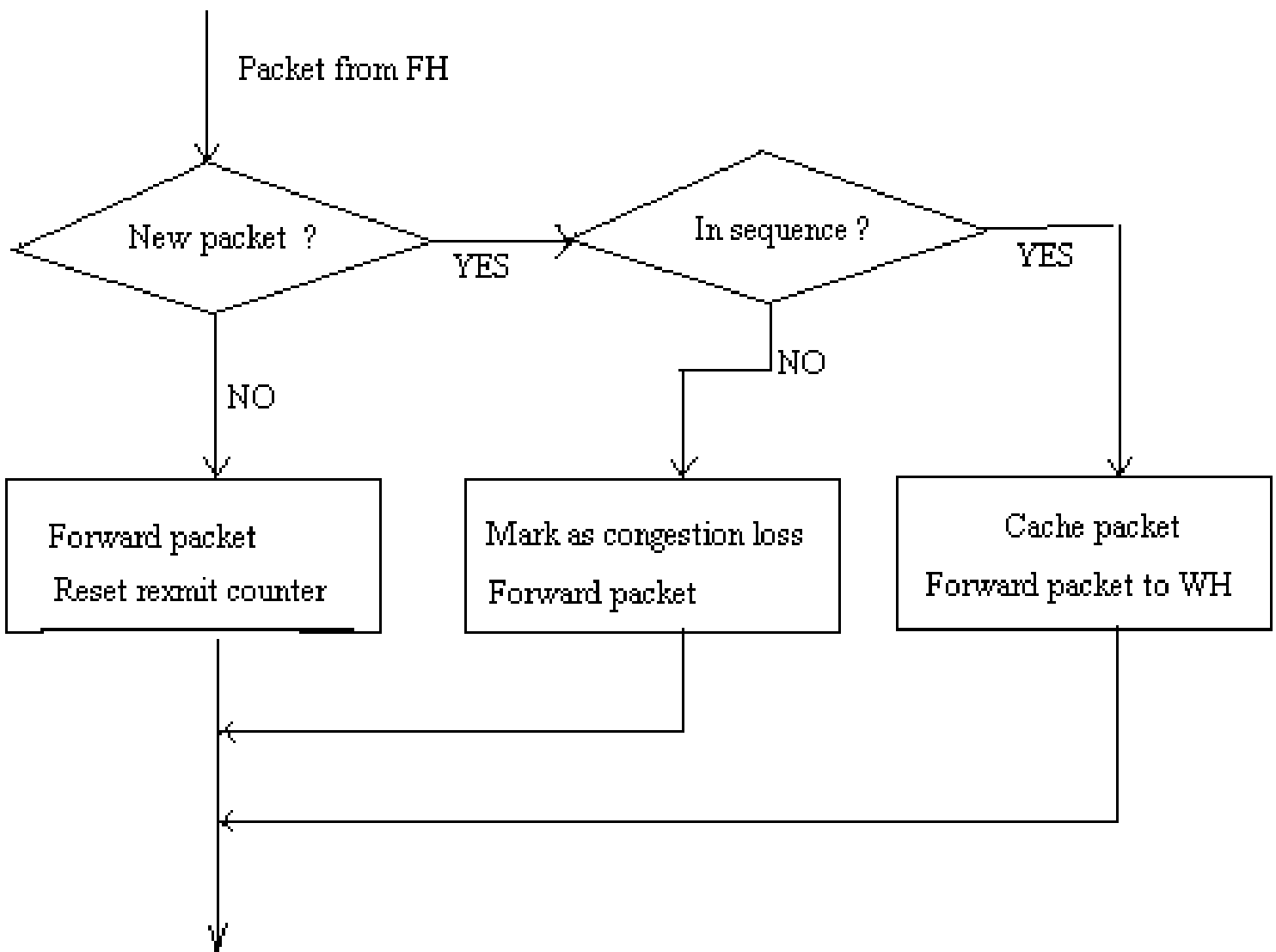


Figure 2. Actions at base station on receiving data from fixed host

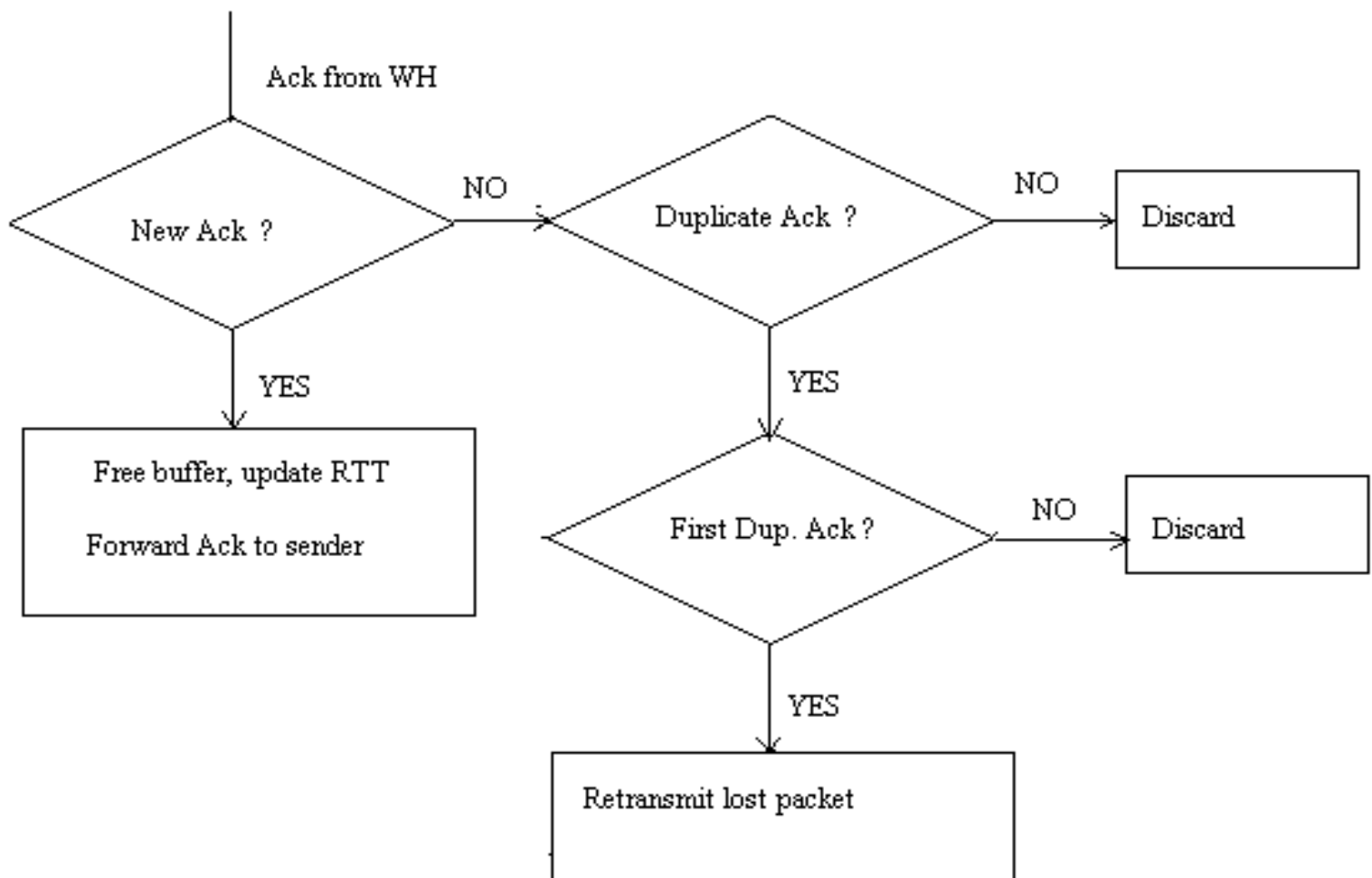


Figure 3. Actions at base station on receiving an ACK from wireless host

It also identifies a group of base stations where the wireless host can move, and broadcasts the packets meant for the wireless station to this group. This avoids expensive state transfer during the handoff. When the wireless host moves to the new base station in the group it already has the cache of packets so the transport state gets recovered quickly. However if the wireless host moves to a base station out of the group it takes some time to built up the cache.

### 3.1.3 M-TCP

A new architecture is proposed for modern cellular networks to support high bandwidth multimedia services and real time audio and video [Brown97]. It has three layers.

As shown in fig.4, Mobile Hosts are at the lowest level and communicate with MSS (base station) nodes in each cell. A supervisory host controls several base stations (SH). SH is connected to a wired network and handles most of the routing and other protocol details. It also maintains connections and handles flow control. When a wireless host moves from one cell into another, two base stations need not perform state transfer if the same SH controls them.

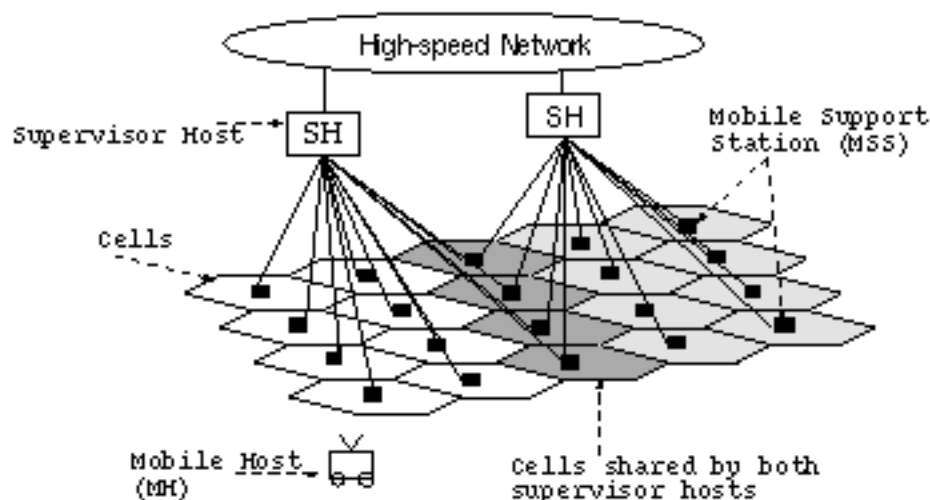


Figure 4. Architecture for M-TCP (taken from [\[Brown97\]](#))

This also uses the split connection method similar to I-TCP. The connection between the fixed host and the wireless host is broken up into two parts. The split connection method performs better than standard TCP because the base station (where the connection is split) is usually at one hop away from the wireless host and hence provides better control over the losses and also can adapt quickly to the dynamic mobile environment because of low RTT. Even if the base station sender makes its congestion window small in case of losses the window builds up fast when new acknowledgements arrive from wireless host because of small RTT.

TCP at the sender is not modified whereas the TCP at the base station is called SH-TCP (Supervisory Host TCP). The base station and the wireless host communicate using M-TCP. When the base station receives the data from the sender it forwards it to the wireless host but defers the acknowledgement (for the segment) to the sender until it receives an acknowledgement from the wireless host. If the size of advertised window at SH-TCP is  $W$  and it has received acknowledgement from wireless host for  $w \leq W$  then the SH-TCP acknowledges  $w-1$  bytes and defers ACK for the last byte. Now at this point if the wireless host undergoes a handoff or a period of data loss the base station sends the deferred acknowledgement and advertises the window size of zero, this causes the sender to go in a persist state. In this state all timers are frozen and does not close its congestion windows and does not back off its timers. When the wireless host regains the connection it sends a greeting packet to the SH. SH\_TCP send a duplicate acknowledgement with the window update packet to the sender so that it can resume transmitting data. This method provides a solution to the problem of periodic disconnection. M-TCP can be designed so that it is optimized for wireless links. Data compression can be used by M-TCP to make efficient use of wireless bandwidth. However the complexity at the base station is very high.

### 3.1.4 Delayed Duplicate Acknowledgements

This scheme attempts to imitate the behavior Snoop scheme described in 3.1.2. However in snoop scheme we needed to make modifications to the TCP whereas this solution is 'TCP Unaware' i.e. no changes in the TCP are needed. The base station needs to implement a mechanism to perform link level



retransmissions. This method tries to reduce the interference between TCP retransmissions and the link level retransmissions [Vaidya99]. When the TCP receiver receives out-of-order packets, it sends the duplicate ACKs for first two out-of-order packets but if it gets more of them  $p$  then it defers the ACKs for these packets for a time period of, say  $t$ . If during this time period  $t$ , it gets the next in-sequence packet then it discards the duplicate ACKs. If it does not get the in sequence packet during this time it sends all the deferred duplicate ACKs. If the packet was lost due to transmission error on wireless link and if the time  $t$  was chosen large enough for the link level retransmission to take place then the TCP receiver gets the packet before  $t$  and hence does not send the third duplicate ACK and hence the sender does not fast-retransmit. If the packet was lost due to congestion on the wired link, TCP receiver will delay the third duplicate ACK for time  $t$ . This can make this method perform worse than the standard TCP. This method performs better in a scenario when the packet losses are due to wireless transmission errors and performs bad when the losses are due to congestion on the wired network. The overall performance depends on the relative frequency of the two types of losses. The base station does not need to look at the TCP header so this method may be specially useful if the transmission is encrypted, the snoop approach fails in case of encrypted transmission.

## 3.2 Wireless Aware TCP at the fixed host

In this approach the fixed host (sender) is aware of the existence of the wireless link in the network and is able to distinguish the losses due to transmission error on wireless link from those due to congestion. The sender can avoid invoking congestion control algorithms when the losses are due to the wireless link. We now discuss TCP extensions, which are based on this approach.

### 3.2.1 Fast Retransmit

This method presents a solution to the problem where only one or two segments are lost. When a wireless host moves from one cell to other cell some packets are lost. The sender waits till a timeout occurs and treats this as congestion and reduces the window and retransmits. If the handoff completes quickly the wireless host has to wait for the timeout at the sender for it to retransmit the lost packets. This timeout period is fairly long and can affect the throughput. This solution overcomes this problem by making the wireless host resume the communication immediately after the handoff without waiting for the timeout at the sender. The wireless host sends a threshold number of duplicate ACKs to the sender. This prompts the TCP at the sender to reduce the window size to half and begin retransmissions [Caceres95].

### 3.2.2 Mobile TCP

One problem with TCP on wireless networks is that it can not distinguish the exact cause of the packet loss. The delay characteristics when a wireless host switches to a different network is different from when it moves from one cell to another in the same network and the data loss from these two reasons is different from data loss due to congestion in the wired network. TCP at the sender is not able to distinguish between these losses. The fixed host interprets these packet losses due to handoffs or interface switching as congestion and invokes the congestion control methods including reducing window size. This is not desirable. Mobile TCP [Stange198] distinguishes the packet losses due to handoff and those due to interface switching. It lets the base station tell the sender whether the loss is due

to handoff in the same network or if it is due to interface switching. The sender then marks the packets and retransmits them once the mobile host has completed handoff. In case of interface switch the wireless host enters a new network which may not have the same network characteristics as the previous one. So when the TCP at the sender knows about the interface switching it resets window size, ssthresh, estimates of RTT and RTO values and begins slow start. But if the wireless host has moved to a cell in the same network then the values of window size and ssthresh are halved and the RTT value remains the same. This algorithm performs well because it is able to know the cause of the packet loss. Though it handles the handoffs well it does not take care about the characteristics of the wireless link.

### 3.2.3 Multiple Acknowledgements

This method distinguishes the losses due to congestion or other errors on the wired link and those on the wireless link. This method uses two types of ACKs to isolate the wireless host and the fixed network [Biaz97].

**ACK<sub>p</sub>:** This partial acknowledgement carries a sequence number  $S$ . This tells the sender that packets till  $S-1$  have been received by the base station.

**ACK<sub>c</sub>:** This is the normal ACK in TCP.

Sender uses standard TCP with slow start, congestion avoidance, fast retransmit and fast recovery.

Let  $RTT$  be the end-to-end round trip time and  $RTT(w)$  be the round trip time between the base station and the wireless host. The sender retransmits after waiting  $rto$  time for end-to-end ACK.  $rto(w)$  is the maximum time between a packet is received at the base station and the time when it got a acknowledgement from the wireless host. When a sender gets a **ACK<sub>p</sub>** from a base station the sender understands that base station has received packets upto  $S-1$  but is having difficulty sending these to the wireless host i.e. it has not received the acknowledgement from the wireless host even after waiting for  $rto(w)$  time. So the sender updates  $rto$  to avoid end-to-end retransmissions. It marks the packets corresponding to this **ACK<sub>p</sub>**. On a timeout the sender checks if the packets are marked if they are not it acts as normal TCP would. If they are marked then it will not retransmit, and will not do any congestion control methods except backing off the timer.

When a base station gets a packet from sender it starts a  $rto(w)$  timer and sends it to the wireless host. If the timer goes off the base station sends a **ACK<sub>p</sub>** to the sender. If the base station receives an out of order packet which is not cached then it will try to deliver this packet to the wireless host and if the packet cannot be delivered for  $rto(w)$  time and if all earlier packets have been received by the base station then it sends a **ACK<sub>p</sub>** to the sender. The ACKs from the wireless host are forwarded only if they are needed by the TCP at sender and the ACKs that trigger unnecessary retransmissions are discarded.

### 3.2.4 Discriminating congestion losses from wireless losses

This method again uses the model where the network is wired and the last hop is wireless. It is also assumed that the wireless link is the bottleneck in the network. As the wireless link is the bottleneck the packets arrive at the base station early and they are queued at the base station. So most packets are transmitted back to back on the wireless link. The sender performs bulk transfers. These are the basic assumption used in this method. The packet inter-arrival time is defined as the time between arrival of

consecutive packets.

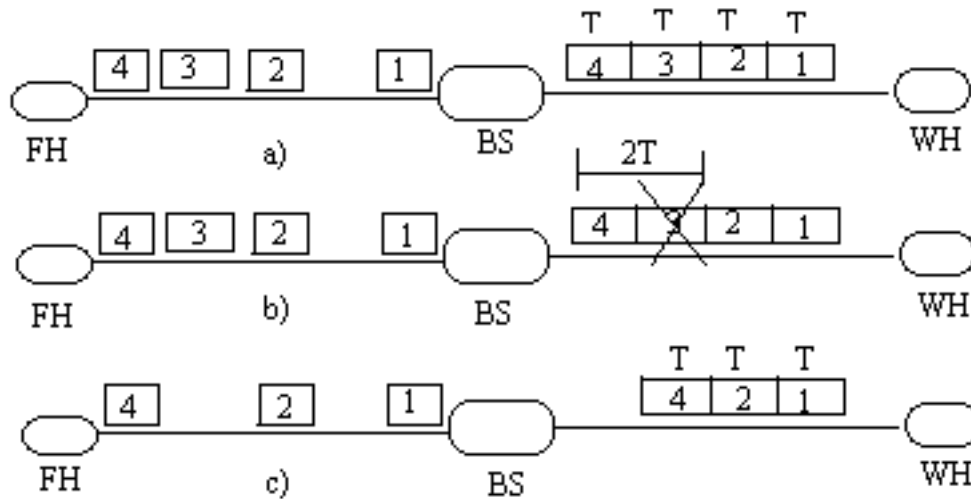


Figure 5. Inter-arrival gap

Consider the figure in case 'a' no packets are lost so packet inter-arrival time is same as time required to transfer the packet on the wireless network ( $T$ ). In case b the packet 3 is lost so the time of arrival of 4 after 2 is  $2T$  as packet 3 was on lost on the network before using  $T$  on the link. In case 'c' packet 3 is lost due to congestion so the inter-arrival time between packet 2 and packet 4 is  $T$ .

Let  $T_{min}$  be the minimum inter-arrival time observed so far.

$P_0$  denotes the out of order packet received by the receiver.  $P_i$  was the last in-sequence packet received before  $P_0$ .  $T_g$  is the time between arrivals of  $P_0$  and  $P_i$  and let  $n$  be the number of packets missing between  $P_i$  and  $P_0$ .

If  $(n+1)T_{min} \leq T_g \leq (n+2)T_{min}$  then  $n$  missing packets are assumed to be lost due to wireless transmission errors otherwise they are assumed to be lost due to congestion. Thus this method can be useful to distinguish losses due to congestion and losses due to transmission errors on the wireless link and the TCP at the ends can be designed to take advantage of this knowledge and improve performance [Biaz99].

### 3.2.5 Distinguish losses by making two connections

The basic model for this method is that the network is wired and the last hop from the base station to the wireless host is wireless. It is assumed that the wireless host receives all its packets from the base station. This method does not take care of the handoffs. The TCP at the sender is assumed to know that the destination host is a wireless host. The key idea in this algorithm is that the sender distinguishes between congestion on the wired network and the transmission errors on the wireless part and the wireless host assures that if the packets are dropped due to transmission error the sender retransmits them before its timeout [Banerjee97]. When a fixed host wants to communicate with a wireless host it opens two connections one with the base station and other with the wireless host. The connection between the sender and the base station is called control connection. This is used to estimate the congestion on the wired link. The packets on these two connections are expected to be routed in the same way and hence are affected same by the congestion. Sender sends packets on the control connection in regular intervals

sufficiently spaced so as not to cause overhead. The sender periodically compares the fraction of acknowledged packets on the two connections and checks if the packets are lost due to congestion or due to transmission error on the wireless link. If the acknowledged fraction is significantly different in these two cases then it concludes that the error in the wireless link is causing the packets to be dropped so the sender does not apply congestion control method and does not reduce the window size and continues the increment as before. If the two fractions are same then the congestion control methods are applied as in normal TCP. When the packets are lost the TCP at the sender has to wait for timeout after which it retransmits the packets. When the wireless host learns about this loss it sends duplicate ACKs. The sender on seeing the duplicates ACKs knows that its previous packets have been lost and immediately resends the packets without waiting for the timeout.

[\[Back to Table of Contents\]](#)

---

## 4. TCP extensions for satellite networks

Satellites are very useful for communication because of following reasons. They have natural broadcast capability. They can reach geographically remote areas and places which lack terrestrial communication infrastructure. This also makes them useful to reach mobile users. The satellite networks can be classified in following types [\[Allman99\]](#)

*Asymmetric satellite networks:* This is characterized by large data rate in one direction and very less in the other. These satellite systems are unidirectional and use non-satellite path in other direction( for ACKs). This is seen to be a problem for TCP.

*Satellite link with last hop:* These provide service directly to the users. This can provide shared high speed downlink to users with low speed. This causes the network to be asymmetric as the return path may be terrestrial path.

*Hybrid satellite networks :* Satellite links are located at any point in a network and act as a link between two routers.

*Point-to point Satellite networks:* This is a pure satellite network with all hops going through satellites.

### 4.1 Issues

All satellite networks are inherently characterized by high delays because of finite speed of light and the altitude of the communication satellites. Many communication satellites are located at geostationary orbit (GEO) at an altitude of around 36000 km. This long distance causes the ground-to-satellite-to-ground propagation delay to be 239.6 milliseconds for a radio signal. For ground stations it turns out that the delay is 279ms. So the round trip delays for a message and reply could be as high as 558ms. It can be more than this if there are multiple satellite links. Sometime low earth orbit satellites(LEO) or medium earth orbit satellites(MEO) are used. These require a constellations of satellites so that they are constantly visible to the ground station, The propagation delay for LEO may be around 80ms.

The radio signal attenuates inversely in proportion of the distance. The large distance of satellites makes

the signal weak This causes the BER to be high. Another issue with satellite networks is of limited radio spectrum and there is a restricted amount of bandwidth available which is controlled by licenses.

TCP cannot make efficient use of satellite network because of the following characteristics of satellite networks [\[RFC2488\]](#).

Large delay\*bandwidth product: The delay bandwidth can be defined as amount of data on the link at any time to utilize the available channel bandwidth. This poses a problem for TCP.

Long feedback loop: Because of the long delay on the satellite links, it may take a long time for the sender to know if the packet was received at the destination. This is also bad for interactive applications like telnet and for congestion control algorithms.

Transmission errors: Satellites have high BER and the TCP at the sender cannot determine if the packets are dropped due to congestion or due to transmission errors. It interpretes this as congestion and reduces the window size which worsens the situation.

Variable round trip delays: In low earth orbit constellations the delays varies from time to time. This may or may not affect TCP performance

Handoffs: In non-geosynchronous satellites TCP connections need to be transferred from one satellite to other or from one ground station to another. This handoff affects the TCP performance adversely.

Asymmetric use: Some connections use satellite link for one direction and terrestrial link for opposite direction this poses problem for TCP.

## 4.2 Proposed Solutions

### 4.2.1 Slow start and congestion avoidance

Slow start algorithm wastes a lot of bandwidth on satellite networks. Because of large delay\*bandwidth product it takes a large time to increase the congestion window to fill the link and hence effectively utilize the bandwidth. Delayed ACKs also cause wasted bandwidth during this slow start phase. One method to deal with this is to increase the initial value of congestion window. It is suggested that following value of initial window yields good results

$\text{Min}( 4 * \text{MSS} , \text{max}( 2 * \text{MSS} , 4380 ) )$  [\[RFC2414\]](#)

This large window causes large packets to be sent in first round trip time so there will be more ACKs and the window size can be increased fast. To deal with the problem of delayed ACKs, 2 segments are sent initially instead of one so there is no need to wait for the delayed ACK timer. The initial window size gives an improvement of 3 RTT and a delayed ACK timeout.

### 4.2.2 Fast Retransmit

TCP uses timeouts to detect lost segments. When the timer expires the TCP retransmits the data and performs congestion control by setting ssthresh to half that of the window and making window 1 and then it starts the slow start. This algorithm uses three duplicate ACKs to trigger retransmission of the lost

segments of data and the sender can retransmit them without waiting for the timeout. This is called fast retransmit. Now it adjusts the window sizes which is called fast recovery. The value of `ssthresh` is set to half the value of the window, the window size is halved. The window size now is increased by one segment for each duplicate ACK it receives. When an ACK for the retransmitted packet is received the congestion window is restored to `ssthresh` which is equal to half the original value of the window size when the congestion was detected. Fast retransmit increases performance however in some cases it is seen to degrade performance [Floyd94]. In connections with large congestion windows, new segments are introduced during the recovery which can trigger multiple fast retransmits. This can reduce the congestion window multiple times for one loss event. It is suggested to use implementation of fast retransmit suggested in [RFC2001].

### 4.2.3 Large window sizes

The standard TCP allows a maximum window size of 65535 bytes. TCP throughput is limited by:

$$\text{throughput} = \text{window size} / \text{RTT}$$

Geosynchronous satellite have a channel RTT of 560 ms [Kru95]. The maximum throughput that can be achieved by standard TCP over satellites is limited to:

$$\text{throughput} = 65,535 \text{ bytes} / 560 \text{ ms} = 117,027 \text{ bytes/second}$$

therefore TCP should be extended for larger windows. This help improve performance of TCP on satellites. [Allman99]

### 4.2.4 Delayed Acknowledgements

Standard TCP generates an ACK for each incoming segment. In this method, an ACK is generated for every second full-sized segment, or if a second full-size segment does not arrive within a given timeout (which must not exceed 500 ms). The congestion window is increased based on the number of incoming ACKs and delayed ACKs reduce the number of ACKs being sent by the receiver. Therefore, the congestion window grows much more slowly when using delayed ACKs compared to the case when the receiver sends an acknowledgement for each incoming segment [Allman98].

One might wonder if this can be solved by turning the mechanism off and let the receiver ACK each incoming segment. However, this is not recommended. And increasing the number of ACKs by a factor of two in a shared network may have consequences that are not yet understood. Therefore, disabling delayed ACKs is still a research issue and thus, at this time TCP receivers should continue to generate delayed ACKs, per [RFC1122].

### 4.2.5 Selective Acknowledgements

When multiple packets of data are lost, the TCP at sender waits for the timeout and determines which segments have to be retransmitted. During this time, the data segments and their acknowledgments are lost from the network. As there are no ACKs coming in (which are used to clock new segments in network) the sender invokes slow start and restarts transmission which is very time consuming and degrades the performance of TCP. TCP receivers can inform senders which packets have arrived using

Selective acknowledgments (SACKs) [\[RFC2018\]](#). This way TCP recovers more quickly from the lost segments and avoids unnecessary retransmissions. With SACKs, the sender can determine which segments need to be retransmitted in the first RTT after loss detection. This way the sender can send segments at an appropriate rate, sustain the ACK clock and avoid the slow start too.

## 4.2.6 TCP for transaction

TCP uses three-way handshake to establish a connection. This requires 1-1.5 round-trip times and this time can be high. This method (T/TCP) [\[RFC1644\]](#) eliminates this startup time. The two hosts need to set up connection only once. Next time T/TCP can bypass this three-way handshake and the sender can transmit the data in the first segment itself with the SYN. This can enhance performance in case of short request-response type of connections between the sender and the receiver. This method requires changes at both sender and receiver side.

## 4.2.7 Explicit congestion notification

In this approach the intermediate node in the network (routers) inform TCP senders about congestion in the network by using Explicit Congestion Notification (ECN). This can be classified in two parts:

1. Backward Explicit Congestion Notification (BECN): The router sends messages directly to the sender and informs it about the congestion. This can be done by using ICMP Source Quench message. The sender on getting this message reduces its window size (it reduces its rate of data transfer).
2. Forward Explicit Congestion Notification (FECN): In this approach when the routers see the congestion they mark the packets with a special tag before forwarding the packet. The data receiver on receiving this marked packet sends this congestion information to the sender in the ACK packet. The sender then invokes the congestion control methods.

Senders transmit segments with an "ECN-Capable Transport" bit set in the IP header of each packet [\[RFC2481\]](#). If a router employing an active queuing strategy, such as Random Early Detection (RED), would otherwise drop this segment, it instead of dropping this segment sets a "Congestion Experienced" bit in the IP header. When the receiver gets this packet it informs the sender about the congestion by setting a bit in the TCP header. TCP at sender reacts to this by decreasing the congestion window.

Currently TCP assumes that all the losses are due to congestion and does not distinguish between losses due to satellite link and those due to congestion. As the satellite networks have higher bit-error rates than terrestrial networks, determining whether a segment was lost due to congestion or wireless link may allow TCP to achieve better performance in high BER environments than currently possible. Adding ECN support to TCP may help in achieving this.

[\[Back to Table of Contents\]](#)

---

# 5. Conclusion

In this paper I have presented several schemes to improve the performance of TCP in wireless networks. These solutions were divided in two types:- first where TCP at the sender is unaware of the wireless link in the network and second when it is aware of the wireless link. According to results presented in [\[Bala97\]](#) TCP aware link layer protocol with selective acknowledgements performs best, split connection approach does not result in good performance and the scheme with selective acknowledgement performs well in case of bursty lossy links.

Schemes for improvement of TCP on satellite networks are also discussed. These include large initial windows, fast retransmit, delayed acknowledgements, selective acknowledgements and explicit congestion notification.

[\[Back to Table of Contents\]](#)

---

# 6. References

1. **[Liu99]** Liu J "Transport layer protocols for wireless networks" Ph.D. thesis University of South Carolina, 1999, 117 pages *Discusses schemes for improving TCP performance for mobile ad hoc networks*
2. **[Katz95]** Balakrishna H, Katz R. et al "Improving TCP/IP performance over wireless networks" Proc. 1st ACM Int'l Conf on Mobile Computing and Networking (Mobicom), November 95, 10 pages. <http://www.cs.berkeley.edu/~ss/papers/mobicom95/html/mobicom-final.html> *Discusses improvement in TCP performance for wireless by using a split-connection approach (Berkeley Snoop Module)*
3. **[Bala95]** Balakrishna H, Katz, Seshan S "Improving reliable Transport and handoff performance in cellular wireless networks" Wireless Networks,1(4), Dec 95, 19 pages <http://www.cs.berkeley.edu/~ss/papers/winet/html/winet.html> *Discusses improvement in TCP performance for wireless by using a split-connection approach (Berkeley Snoop Module)*
4. **[Bala97]** Balakrishna H, Katz, Seshan S "A Comparison of Mechanisms for Improving TCP Performance over Wireless links" ACM SIGCOMM, August 1996, 20 pages. <http://HTTP.CS.Berkeley.EDU/~hari/papers/ton.ps> *Experimentally compares different schemes for TCP over wireless*
5. **[Brown97]** Brown K, Singh S, "M-TCP: TCP for mobile cellular networks" INFOCOM96, 1996, 26 pages <ftp://ftp.ece.orst.edu:/pub/users/singh/papers/mtcp.ps.gz> *This paper suggests a new architecture for cellular wireless networks and describes schemes to improve TCP performance for this architecture*
6. **[Vaidya99]** Vaidya N, Mehta M "Delayed duplicate acknowledgements: A TCP-unaware approach to improve performance of TCP over wireless" Texas A&M University, Technical Report 99-003, February 1999, 16 pages. <http://www.cs.tamu.edu/faculty/vaidya/papers/mobile-computing/99-003.ps> *Discusses a*



*scheme for TCP over wireless with unmodified TCP at the sender*

7. **[Biaz99]** Vaidya N, Biaz S Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver, IEEE Symposium ASSET'99, Richardson, TX, USA, March 1999, 8 pages. <http://www.cs.tamu.edu/faculty/vaidya/papers/mobile-computing/asset99.ps>  
*Discusses a scheme for TCP over wireless with modified TCP at the sender*
8. **[Biaz97]** Biaz S, Vaidya N et al "TCP over wireless networks using multiple acknowledgements" Texas A&M University, Technical Report 97-001, January 1997, 10 pages. <http://www.cs.tamu.edu/faculty/vaidya/papers/mobile-computing/97-001.ps> *Discusses a scheme for TCP over wireless with modified TCP at the sender*
9. **[Caceres95]** Caceres R, Iftode L "Improving the performance of reliable transport protocols in mobile computing environments" IEEE Trans. on Selected Area in Comm. pages 850-857, June 1995. *Discusses a scheme for TCP over wireless based on fast retransmit*
10. **[Badrinath95]** Badrinath B R, Bakre A "I-TCP :Indirect TCP for mobile hosts" Proc. 15th Int'l Conference on Distributed Computing, May 1995, 18 pages. <ftp://paul.rutgers.edu/pub/badri/itcp-tr314.ps.Z> *Discusses a split-connection approach for TCP over wireless*
11. **[Floyd94]** Floyd S, "TCP and Successive Fast Retransmits" Technical Report, October 1994 <ftp://ftp.ee.lbl.gov/papers/fastretrans.ps> *Discusses situations where TCP performance degrades in case of fast retransmit*
12. **[Mate98]** Mate A "Transport layer extensions for mobile fading environment" MS thesis, IISC, India, 1998, 80 pages *Discusses different solutions for TCP and UDP over wireless*
13. **[Banerjee97]** Banerjee S, Goteti J "Extending TCP for wireless networks" University of Maryland, College Park, 7 pages. <http://www.cs.umd.edu/users/suman/docs/711s97/711s97.html>  
*Distinguishes between types of losses in wireless networks by making two connections*
14. **[RFC1644]** Braden R "T/TCP -- TCP extensions for transaction: functional specification" , July 1994, RFC 1644, 38 pages. <http://www.ietf.org/rfc/rfc1644.txt> *Discusses TCP for transaction*
15. **[Stangel98]** Stangel M, Bhargavan V "Improving TCP performance in mobile computing environments" Proc. IEEE ICC'98, 1998, 6 pages. *Discusses a wireless aware TCP scheme known as Mobile TCP*
16. **[Allman98]** Allman M "On the generation and use of TCP acknowledgements" ACM Computer Communication Review, October 1998, 17 pages *Presents a simulation study of TCP acknowledgement generation and utilization techniques.*
17. **[RFC2414]** Allman M "Increasing TCP's initial window" RFC2414, September 1998, 14 pages. <http://www.ietf.org/rfc/rfc2414.txt> *Discusses pros and cons of increasing TCP's initial window size.*
18. **[Allman99]** Allman M et al " Ongoing TCP research related to satellites" Internet Draft, September 1999, 38 pages <http://www.ietf.org/internet-drafts/draft-ietf-tcpsat-res-issues-12.txt>  
*Discusses current work related to TCP for satellite networks.*
19. **[RFC2488]** Allman M et al "Enhancing TCP over satellite channels using standard mechanisms" RFC2488, January 1999, 19 pages. <http://www.ietf.org/rfc/rfc2488.txt> *Discusses using standard mechanisms for improving TCP performance for satellite networks.*

20. **[RFC1122]** Braden R "Requirements for Internet Hosts – Communication Layers" RFC1122, October 1989, 116 pages. <http://www.ietf.org/rfc/rfc1122.txt> *Discusses the requirements for host system implementations of the Internet protocol suite*
21. **[RFC2018]** Mathis M, Mahdavi J et. Al." TCP Selective Acknowledgement Options", RFC 2018, October 1996, 11 pages <http://www.ietf.org/rfc/rfc2018.txt> *This memo proposes an implementation of SACK and discusses its performance and related issues.*
22. **[RFC2481]** Ramakrishnan k K, Floyd S, " A proposal to add Explicit Congestion Notification to IP" January 1999, RFC 2481, 23 pages <http://www.ietf.org/rfc/rfc2481.txt> *Describes Explicit Congestion Notification for IP.*
23. **[RFC2001]** Stevens W, "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms",RFC2001, January 1997, 5 pages. <http://www.ietf.org/rfc/rfc2001.txt> *Formally describes slow start, congestion avoidance, fst retransmit and fast recovery algorithms.*

[\[Back to Table of Contents\]](#)

---

## List Of Acronyms:

FH - Fixed Host  
WH - Wireless Host  
BS - Base Station  
BER - Bit Error Rate  
SH-TCP- Supervisory Host TCP  
I-TCP - Indirect TCP  
T/TCP - Transaction TCP  
RTT - Round Trip Time  
MSS - Maximum Segment Size  
SACK - Selective Acknowledgement  
FECN - Forward Explicit Congestion Notification  
BECN - Backward Explicit Congestion Notification  
RED - Random Early Detection  
GEO - GeoStationary Orbit  
LEO - Low Earth Orbit  
MEO - Medium Earth Orbit

[\[Back to Table of Contents\]](#)

---

Date Last Modified: 11/18/99

Note: This paper is available on-line at [http://www.cis.ohio-state.edu/~jain/cis788-99/tcp\\_wireless/index.html](http://www.cis.ohio-state.edu/~jain/cis788-99/tcp_wireless/index.html)