UNIT – 3 MOBILE TRANSPORT LAYER

TCP enhancements for wireless protocols - Traditional TCP: Congestion control, fast retransmit/fast recovery, Implications of mobility - Classical TCP improvements: Indirect TCP, Snooping TCP, Mobile TCP, Time out freezing, Selective retransmission, Transaction oriented TCP - TCP over 3G wireless networks.

3.1 TCP enhancements for wireless protocols

Supporting mobility only on lower layers up to the network layer is not enough to provide mobility support for application. Most applications rely on a transport layer, such as TCP (transmission control protocol) or UDP (user datagram protocol) in the case of the Internet. Two functions of the transport layer in the Internet are check summing over user data and multiplexing/ demultiplexing of data from /to applications. While the network layer only addresses a host, ports in UDP or TCP allow dedicated applications to be addressed. The connection less UDP does not offer much more than this addressing, so, the following concentrates on TCP. While UDP is connectionless and does not give certain guarantees about reliable data delivery, TCP is much more complex and, needs special mechanisms to be useful in mobile environments.

Mobility support in IP (such as mobile IP) is already enough for UDP to work. The main difference between UDP and TCP is that TCP offers connections between two applications. Within a connection TCP can give certain guarantees, such as in-order delivery or reliable data transmission using retransmission techniques. TCP has built-in mechanisms to behave in a 'network friendly' manner. If, for example, TCP encounters packet loss, it assumes network internal congestion and slows down the transmission rate. This is one of the main reasons to stay with protocols like TCP. One key requirement for new developments in the internet is **'TCP friendliness'**. UDP requires that applications handle reliability, in -order delivery etc. UDP does not behave in a network friendly manner, i.e., does not pull back in case of congestion and continues to send packets into an already congested network.

3.2 Traditional TCP

The **Transmission Control Protocol** (**TCP**) is one of the core protocols of the Internet protocol suite, often simply referred to as TCP/IP. TCP is reliable, guarantees in-order delivery of data and incorporates congestion control and flow control mechanisms

TCP supports many of the Internet's most popular application protocols and resulting applications, including the World Wide Web, e-mail, File Transfer Protocol and Secure Shell. In the Internet protocol suite, TCP is the intermediate layer between the Internet layer and application layer.

The major responsibilities of TCP in an active session are to:

- **Provide reliable in-order transport of data**: to not allow losses of data.
- **Control congestions in the networks**: to not allow degradation of the network performance,

•**Control a packet flow between the transmitter and the receiver**: to not exceed the receiver's capacity.

TCP uses a number of mechanisms to achieve high performance and avoid '**congestion collapse**', where network performance can fall by several orders of magnitude. These mechanisms control the rate of data entering the network, keeping the data flow below a rate that would trigger collapse. There are several mechanisms of TCP that influence the efficiency of TCP in a mobile environment. Acknowledgments for data sent, or lack of acknowledgments, are used by senders to implicitly interpret network conditions between the TCP sender and receiver.

` In mobile environment TCP applies several mechanisms to improve the efficiency. The traditional TCP mechanisms are

- **Congestion Control**
- **Slow Start**
- **Fast retransmit/ Fast recovery**
- **Implications of mobility**

3.2.1 Congestion Control

A transport layer protocol such as TCP has been designed for fixed networks with fixed end- systems. Congestion may appear from time to time even in carefully designed networks. The packet buffers of a router are filled and the router cannot forward the packets fast enough because the sum of the input rates of packets destined for one output link is higher than the capacity of the output link. The only thing a router can do in this situation is to drop packets. A dropped packet is lost for the transmission, and the receiver notices a gap in the packet stream. Now the receiver does not directly tell the sender which packet is missing, but continues to acknowledge all in-sequence packets up to the missing one.

The sender notices the missing acknowledgement for the lost packet and assumes a packet loss due to congestion. Retransmitting the missing packet and continuing at full sending rate would now be unwise, as this might only increase the congestion. To mitigate congestion, TCP slows down the transmission rate dramatically. All other TCP connections experiencing the same congestion do exactly the same so the congestion is soon resolved.

3.2.2 Slow start

TCP's reaction to a missing acknowledgement is quite drastic, but it is necessary to get rid of congestion quickly. The behavior TCP shows after the detection of congestion is called **slow start.**

The sender always calculates a **congestion window** for a receiver. The start size of the congestion window is one segment (TCP packet). The sender sends one packet and waits for acknowledgement. If this acknowledgement arrives, the sender increases the congestion window by one, now sending two packets (congestion window = 2). This scheme doubles the congestion window every time the acknowledgements come back, which takes one round trip time (RTT). This is called the exponential growth of the congestion window in the slow start mechanism.

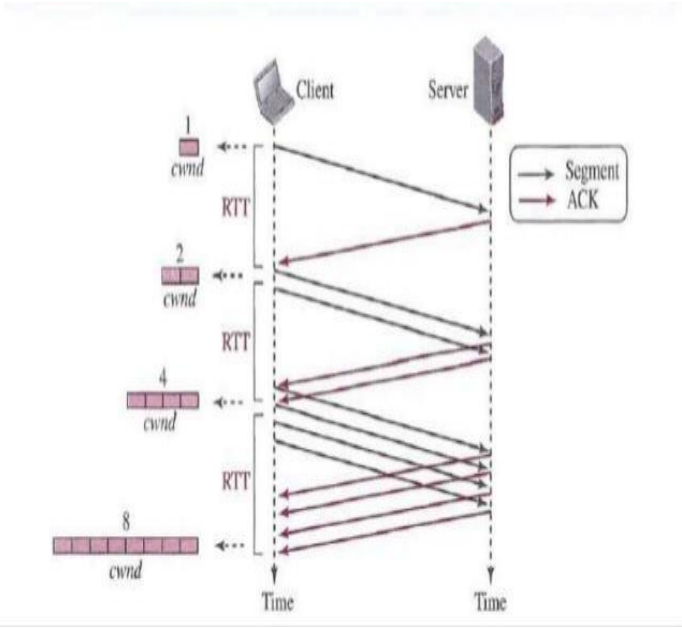


Fig1: **Packets in transit during slow start.**

But doubling the congestion window is too dangerous. The exponential growth stops at the **Congestion threshold**. As soon as the congestion window reaches the congestion threshold, further increase of the transmission rate is only linear by adding 1 to the congestion window each time the acknowledgements come back.

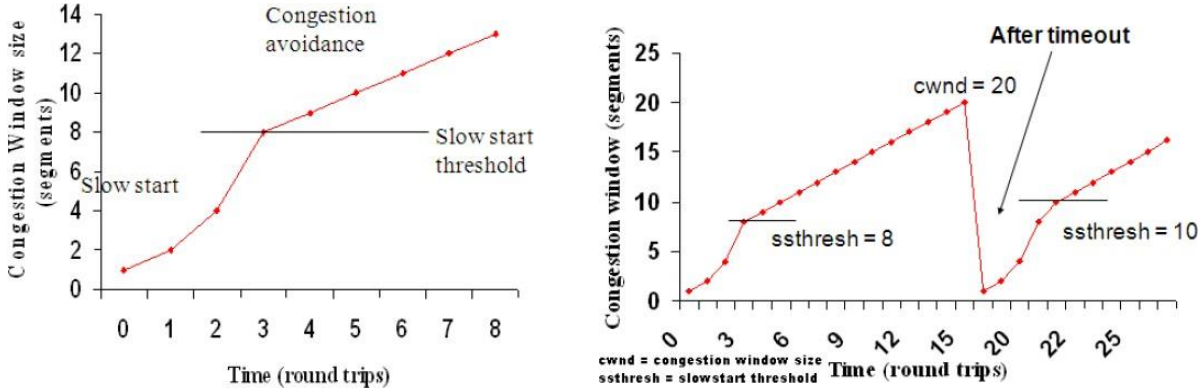


Fig2：**Congestion Window Trace**

Linear increase continues until a time-out at the sender occurs due to a missing acknowledgement, or until the sender detects a gap in transmitted data because of continuous acknowledgements for the same packet. In either case the sender sets the congestion threshold to half of the current congestion window. The congestion window itself is set to one segment and the sender starts sending a single segment. The exponential growth starts once more up to the new congestion threshold, then the window grows in linear fashion.

3.2.3 Fast retransmit/fast recovery

The congestion threshold can be reduced because of two reasons. First one is if the sender receives continuous acknowledgements for the same packet. It informs the sender that the receiver has got all the packets upto the acknowledged packet in the sequence and also the receiver is receiving something continuously from the sender. The gap in the packet stream is not due to congestion, but a simple packet loss due to a transmission error. The sender can now retransmit the missing packet(s) before the timer expires. This behavior is called **fast retransmit**.

It is an early enhancement for preventing slow-start to trigger on losses not caused by congestion. The receipt of acknowledgements shows that there is no congestion to justify a slow start. The sender can continue with the current congestion window. The sender performs a **fast recovery** from the packet loss. This mechanism can improve the efficiency of TCP dramatically. The other reason for activating slow start is a time-out due to a missing acknowledgement. TCP using fast retransmit/fast recovery interprets this congestion in the network and activates the slow start mechanism.

The **advantage** of this method is its simplicity. Minor changes in the MH's software results in performance increase. No changes are required in FA or CH.

The **disadvantage** of this scheme is insufficient isolation of packet losses. It mainly focuses on problems regarding Handover. Also it affects the efficiency when a CH transmits already delivered packets.

3.2.4 Implications of mobility

Problems with Traditional TCP in wireless environments

- Slow Start mechanism in fixed networks decreases the efficiency of TCP if used with mobile receivers or senders.
- Error rates on wireless links are orders of magnitude higher compared to fixed fiber or copper links. This makes compensation for packet loss by TCP quite difficult.
- Mobility itself can cause packet loss. There are many situations where a soft handover from one access point to another is not possible for a mobile end-system.
- Standard TCP reacts with slow start if acknowledgements are missing, which does not help in the case of transmission errors over wireless links and which does not really help during handover. This behavior results in a severe performance degradation of an unchanged TCP if used together with wireless links or mobile nodes

3.3 Classical TCP Improvements

There are several mechanisms for the classical TCP improvements with the goal to increase TCP's performance in wireless and mobile environments. The classical TCP mechanisms are

- **Indirect TCP (I - TCP)**
- **Snooping TCP**
- **Mobile TCP (M - TCP)**
- **Fast retransmit/ Fast recovery**
- **Transmission/Time-out freezing**
- **Selective Retransmission**
- **Transaction-oriented TCP**

3.31 Indirect TCP (I-TCP)

The Traditional TCP had the problem of poor performance with a wireless links. Also the TCP available within a fixed network cannot be altered. Due to these reasons the Indirect TCP(I-TCP) emerged slowly.

Indirect TCP segments a TCP connection into two parts namely,

- **Fixed part**
- **Wireless part**

The following figure shows an example with a mobile host connected via a wireless link and an access point to the 'wired' internet where the correspondent host resides.

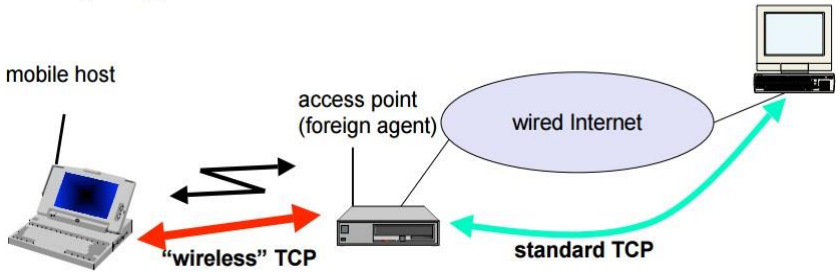


Fig 3: I-TCP segments a TCP connection into two parts

Standard TCP is used between the fixed computer and the access point. No computer in the internet recognizes any changes to TCP. Instead of the mobile host, the access point now terminates the standard TCP connection, acting as a proxy. This means that the access point is now seen as the mobile host for the fixed host and as the fixed host for the mobile host. Between the access point and the mobile host, a special TCP, adapted to wireless links, is used. However, changing TCP for the wireless link is not a requirement. A suitable place for segmenting the connection is at the foreign agent as it not only controls the mobility of the mobile host anyway and can also hand over the connection to the next foreign agent when the mobile host moves on.

The foreign agent acts as a proxy and relays all data in both directions. If CH(correspondent host) sends a packet to the MH, the FA acknowledges it and forwards it to the MH. MH acknowledges on successful reception, but this is only used by the FA. If a packet is lost on the wireless link, CH doesn't observe it and FA tries to retransmit it locally to maintain reliable data transport. If the MH sends a packet, the FA acknowledges it and forwards it to CH. If the packet is lost on the wireless link, the mobile hosts notice this much faster due to the lower round trip time and can directly retransmit the packet. Packet loss in the wired network is now handled by the foreign agent.

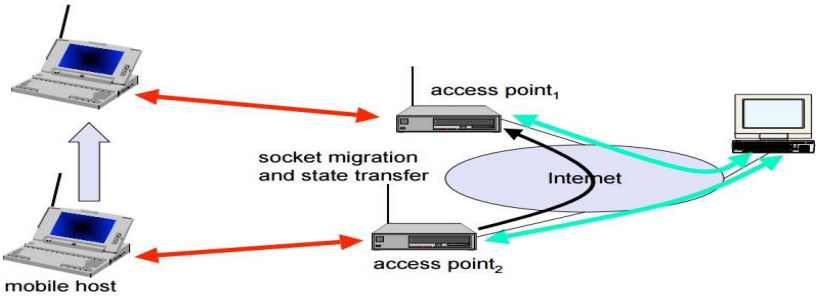


Fig4: Socket and state migration after handover of a mobile host

During handover, the buffered packets, as well as the system state (packet sequence number, acknowledgements, ports, etc), must migrate to the new agent. No new connection may be established for the mobile host, and the correspondent host must not see any changes in connection state. Packet delivery in I-TCP is shown below:

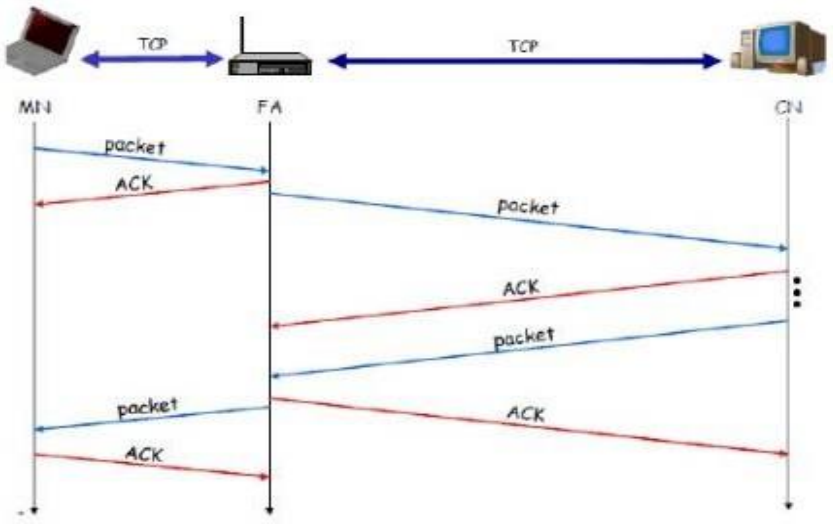


Fig 5: Packet delivery in I-TCP

Advantages of I-TCP

- No changes in the fixed network necessary, no changes for the hosts (TCP protocol)necessary, all current optimizations to TCP still work
- Simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host
    1. Transmission errors on the wireless link do not propagate into the fixed network
    2. Therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop s known
- It is always dangerous to introduce new mechanisms in a huge network without knowing exactly how they behave.
    - ❖ New optimizations can be tested at the last hop, without jeopardizing the stability of the Internet.
- It is easy to use different protocols for wired and wireless networks.

Disadvantages of I-TCP

- Loss of end-to-end semantics:- an acknowledgement to a sender no longer means that a receiver really has received a packet, foreign agents might crash.
- Higher latency possible:- due to buffering of data within the foreign agent and forwarding to a new foreign agent
- Security issue:- The foreign agent must be a trusted entity

3.3.2 Snooping TCP

The main drawback of I-TCP is the segmentation of the single TCP connection into two TCP connections, which loses the original end-to-end TCP semantic. A new enhancement, which leaves the TCP connection intact and is completely transparent, is Snooping TCP. The main function is to buffer data close to the mobile host to perform fast local retransmission in case of packet loss.

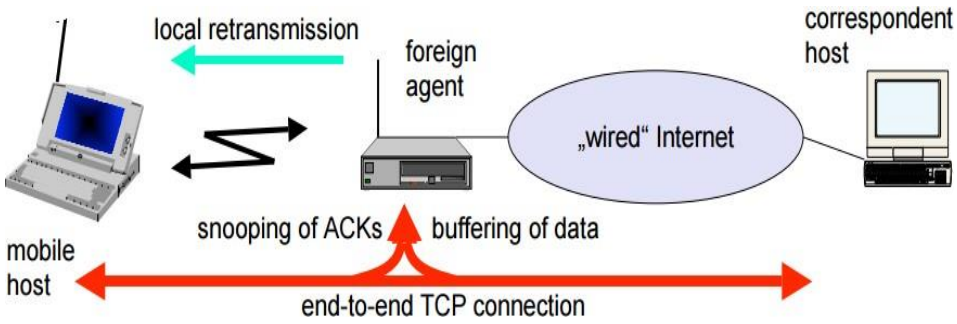


Fig6: Snooping TCP as a transparent TCP extension

Here, the foreign agent buffers all packets with **destination mobile host** and additionally ‘snoops’ the packet flow in both directions to recognize acknowledgements. The foreign agent buffers every packet until it receives an acknowledgement from the mobile host. If the FA does not receive an acknowledgement from the mobile host within a certain amount of time, either the packet or the acknowledgement has been lost. Alternatively, the foreign agent could receive a duplicate ACK which also shows the loss of a packet.

Now, the FA retransmits the packet directly from the buffer thus performing a faster retransmission compared to the CH. For transparency, the FA does not acknowledge data to the CH, which would violate end-to-end semantic in case of a FA failure. The foreign agent can filter the duplicate acknowledgements to avoid unnecessary retransmissions of data from the correspondent host. If the foreign agent now crashes, the time-out of the correspondent host still works and triggers a retransmission. The foreign agent may discard duplicates of packets already retransmitted locally and acknowledged by the mobile host. This avoids unnecessary traffic on the wireless link.

For data transfer from the mobile host with **destination correspondent host**, the FA snoops into the packet stream to detect gaps in the sequence numbers of TCP. As soon as the foreign agent detects a missing packet, it returns a negative acknowledgement (NACK) to the mobile host. The mobile host can now retransmit the missing packet immediately. Reordering of packets is done automatically at the correspondent host by TCP.

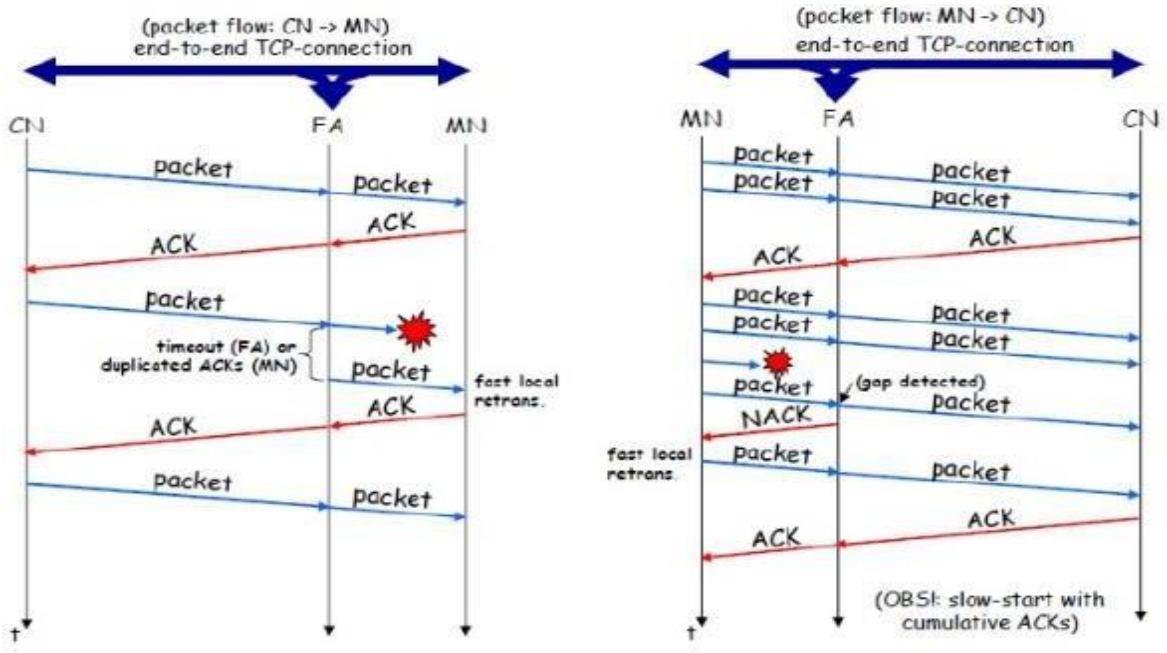


Fig7: Snooping TCP: Packet delivery

Advantages of snooping TCP:

- The end-to-end TCP semantic is preserved.
- Most of the enhancements are done in the foreign agent itself which keeps correspondent host unchanged.
- Handover of state is not required as soon as the mobile host moves to another foreign agent. Even though packets are present in the buffer, time out at the CH occurs and the packets are transmitted to the new COA.
- No problem arises if the new foreign agent uses the enhancement or not. If not, the approach automatically falls back to the standard solution.

Disadvantages of snooping TCP

- Snooping TCP does not isolate the behavior of the wireless link as well as I-TCP. Transmission errors may propagate till CH.
- Using negative acknowledgements between the foreign agent and the mobile host assumes additional mechanisms on the mobile host. This approach is no longer transparent for arbitrary mobile hosts.
- Snooping and buffering data may be useless if certain encryption schemes are applied end-to-end between the correspondent host and mobile host. If encryption is used above the transport layer, (eg. SSL/TLS), snooping TCP can be used.

3.3.3 Mobile TCP

Both I-TCP and Snooping TCP does not help much, if a mobile host gets disconnected. The **M-TCP (mobile TCP)** approach has the same goals as I-TCP and snooping TCP: to prevent the sender window from shrinking if bit errors or disconnection but not congestion cause current problems. M-TCP wants to improve overall throughput, to lower the delay, to maintain end-to-end semantics of TCP, and to provide a more efficient handover. Additionally, M-TCP is especially adapted to the problems arising from lengthy or frequent disconnections.

M-TCP splits the TCP connection into two parts as I-TCP does. An unmodified TCP is used on the standard host-**supervisory host (SH)** connection, while an optimized TCP is used on the SH-MH connection.

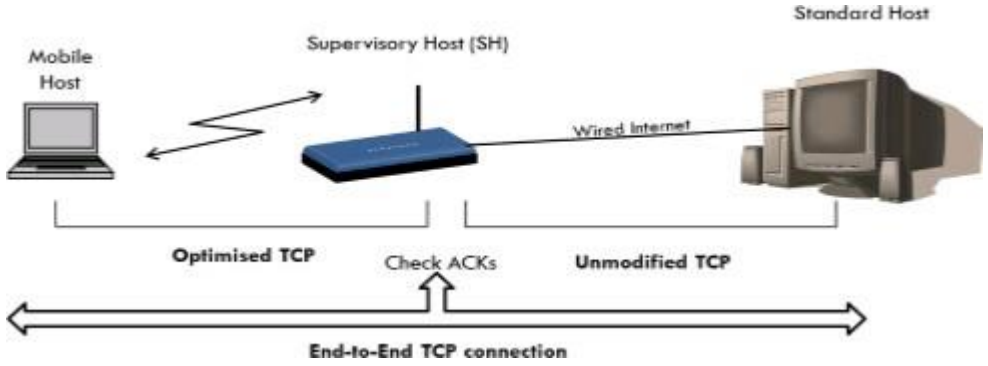


Fig 8: Mobile-TCP

The **supervisory host (SH)** is responsible for exchanging data between both parts similar to the proxy in the I-TCP. The M-TCP approach assumes a relatively low bit error rate on the wireless link. Therefore, it does not perform caching/retransmission of data via the SH. If a packet is lost on the wireless link, it has to be retransmitted by the original sender. This maintains the TCP end-to-end semantics.

The SH monitors all packets sent to the MH and ACKs returned from the MH. If the SH does not receive an ACK for some time, it assumes that the MH is disconnected. It then chokes the sender by setting the sender's window size to 0. Setting the window size to 0 forces the sender to go into **persistent mode**, i.e., the state of the sender will not change no matter how long the receiver is disconnected. This means that the sender will not try to retransmit data. As soon as the SH (either the old SH or a new SH) detects connectivity again, it reopens the window of the sender to the old value. The sender can continue sending at full speed. This mechanism does not require changes to the sender's TCP. The wireless side uses an adapted TCP that can recover from packet loss much faster. This modified TCP does not use slow start, thus, M-TCP needs a **bandwidth manager** to implement fair sharing over the wireless link.

Advantages of M-TCP

- It maintains the TCP end-to-end semantics. The SH does not send any ACK itself but forwards the ACKs from the MH.
- If the MH is disconnected, it avoids useless retransmissions, slow starts or breaking connections by simply shrinking the sender's window to 0.
- As no buffering is done as in I-TCP, there is no need to forward buffers to a new SH. Lost packets will be automatically retransmitted to the SH.

Disadvantages of M-TCP:

- As the SH does not act as proxy as in I-TCP, packet loss on the wireless link due to bit errors is propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption.
- A modified TCP on the wireless link not only requires modifications to the MH protocol software but also new network elements like the bandwidth manager.

3.3.4 Fast Retransmit/Fast Recovery

Moving to a new foreign agent can cause packet loss or time out at mobile hosts or corresponding hosts. TCP concludes congestion and goes into slow start, although there is no congestion. The mechanisms of fast recovery/fast retransmit in traditional TCP a host can use after receiving duplicate acknowledgements, thus concluding a packet loss without congestion.

But the idea on Classical TCP Fast retransmit/ Fast recovery is to artificially force the fast retransmit behavior on the mobile host and correspondent host side. As soon as the mobile host registers at a new foreign agent using mobile IP, it starts sending duplicated acknowledgements to correspondent hosts. The proposal is to send three duplicates. This force the corresponding host to go into fast retransmit mode and not to start slow start, i.e., the correspondent host continues to send with the same rate it did before the mobile host moved to another foreign agent.

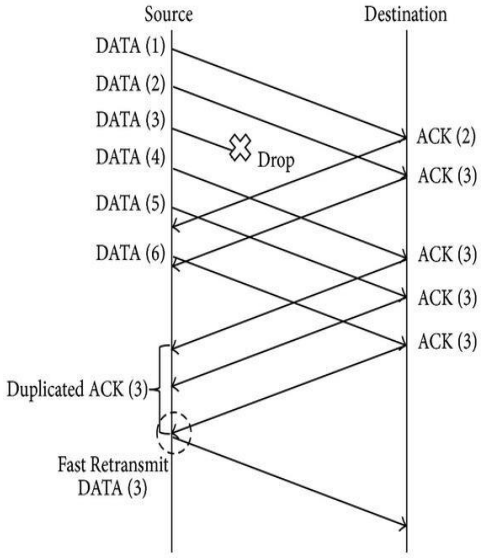


Fig 9: Fast Retransmit/Fast Recovery

As the mobile host may also go into slow start after moving to a new foreign agent, this approach additionally puts the mobile host into fast retransmit. The mobile host retransmits all unacknowledged packets using the current congestion window size without going into slow start.

Advantages of Fast Retransmit/Fast Recovery:

- Only minor changes in the mobile host's software already result in a performance increase. No foreign agent or correspondent host has to be changed.

Disadvantages of Fast Retransmit/Fast Recovery:

- Insufficient isolation of packet losses.
- Forcing fast retransmission increases the efficiency, but retransmitted packets still have to cross the whole network between correspondent host and mobile host. If the handover from one foreign agent to another takes a longer time, the correspondent host will have already started retransmission. So packet losses due to handover.
- It requires more cooperation between the mobile IP and TCP layer making it harder to change one without influencing the other.

3.3.5 Transmission/time-out freezing

Often, MAC layer notices connection problems even before the connection is actually interrupted from a TCP point of view and also knows the real reason for the interruption. The MAC layer can inform the TCP layer of an upcoming loss of connection or that the current interruption is not caused by congestion. TCP can now stop sending and 'freezes' the current state of its congestion window and further timers. If the MAC layer notices the upcoming interruption early enough, both the mobile and correspondent host can be informed. With a fast interruption of the wireless link, additional mechanisms in the access point are needed to inform the correspondent host of the reason for interruption. Otherwise, the correspondent host goes into slow start assuming congestion and finally breaks the connection.

As soon as the MAC layer detects connectivity again, it signals TCP that it can resume operation at exactly the same point where it had been forced to stop. For TCP time simply does not advance, so no timers expire.

Advantages:

- It offers a way to resume TCP connections even after long interruptions of the connection.
- It can be used together with encrypted data as it is independent of other TCP mechanisms such as sequence no or acknowledgements

Disadvantages:

- Lots of changes have to be made in software of MH, CH and FA.

3.3.6 Selective retransmission

A very useful extension of TCP is the use of selective retransmission. TCP acknowledgements are cumulative, i.e., they acknowledge in-order receipt of packets up to a certain packet. A single acknowledgement confirms reception of all packets upto a certain packet. If a single packet is lost, the sender has to retransmit everything starting from the lost packet (go-back-n retransmission). This obviously wastes bandwidth, not just in the case of a mobile network, but for any network.

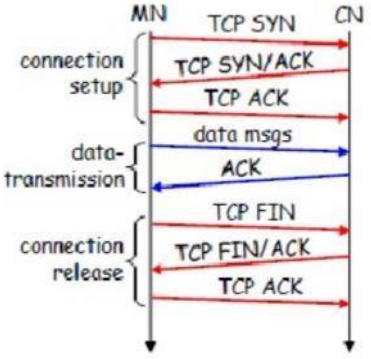
Using selective retransmission, TCP can indirectly request a selective retransmission of packets. The receiver can acknowledge single packets, not only trains of in-sequence packets. The sender can now determine precisely which packet is needed and can retransmit it.

The **advantage** of this approach is obvious: a sender retransmits only the lost packets. This lowers bandwidth requirements and is extremely helpful in slow wireless links.

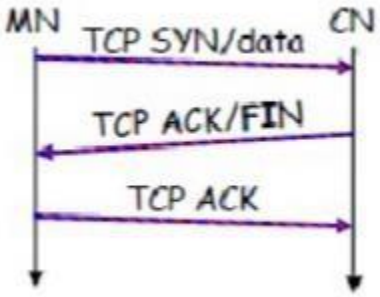
The **disadvantage** is that more complex software on the receiver side is needed. Also more buffer space is needed to resequence data and to wait for gaps to be filled.

3.3.7 Transaction-oriented TCP

Assume an application running on the mobile host that sends a short request to a server from time to time, which responds with a short message and it requires reliable TCP transport of the packets. For it to use normal TCP, it is inefficient because of the overhead involved. Standard TCP is made up of three phases: setup, data transfer and release.



First, TCP uses a three-way handshake to establish the connection. At least one additional packet is usually needed for transmission of the request, and requires three more packets to close the connection via a three-way handshake. So, for sending one data packet, TCP may need seven packets altogether. This kind of overhead is acceptable for long sessions in fixed networks, but is quite inefficient for short messages or sessions in wireless networks. This led to the development of transaction-oriented TCP (T/TCP).T/TCP can combine packets for connection establishment and connection release with user data packets. This can reduce the number of packets down to two instead of seven.



The obvious **advantage** for certain applications is the reduction in the overhead which standard TCP has for connection setup and connection release.

Disadvantage is that it requires changes in the software in mobile host and all correspondent hosts. This solution does not hide mobility anymore. Also, T/TCP exhibits several security problems.

CLASSICAL ENHANCEMENTS TO TCP FOR MOBILITY: A comparison

Approach	Mechanism	Advantages	Disadvantages
Indirect TCP	splits TCP connection into two connections	isolation of wireless link, simple	loss of TCP semantics, higher latency at handover
Snooping TCP	"snoops" data and acknowledgements, local retransmission	transparent for end-to-end connection, MAC integration possible	problematic with encryption, bad isolation of wireless link
M-TCP	splits TCP connection, chokes sender via window size	Maintains end-to-end semantics, handles long term and frequent disconnections	Bad isolation of wireless link, processing overhead due to bandwidth management
Fast retransmit/ fast recovery	avoids slow-start after roaming	simple and efficient	mixed layers, not transparent
Transmission/ time-out freezing	freezes TCP state at disconnect, resumes after reconnection	independent of content or encryption, works for longer interrupts	changes in TCP required, MAC dependant
Selective retransmission	retransmit only lost data	very efficient	slightly more complex receiver software, more buffer needed
Transaction oriented TCP	combine connection setup/release and data transmission	Efficient for certain applications	changes in TCP required, not transparent

3.4 TCP over 3G on Wireless Networks

The focus on 3G for transport of internet data is important as already more than 1 billion people use mobile phones and it is obvious that the mobile phone systems will also be used to transport arbitrary internet data.

The following characteristics have to be considered when deploying applications over 3G wireless links:

- **Data Rates:** Data rates for 3G is around 64 kbit/s uplink and 115–384 kbit/s downlink. Data rates are asymmetric as it is expected that users will download more data compared to uploading. Uploading is limited by the limited battery power.

- **Latency:** All wireless systems comprise elaborated algorithms for error correction and protection, such as forward error correction (FEC), check summing, and interleaving. FEC and interleaving let the round trip time (RTT) grow to several hundred milliseconds up to some seconds. The current GPRS standard specifies an average delay of less than two seconds for the transport class with the highest quality

- **Jitter:** Wireless systems suffer from large delay variations or 'delay spikes'. Reasons for sudden increase in the latency are: link outages due to temporal loss of radio coverage, blocking due to high-priority traffic, or handovers.

- **Packet loss:** Packets might be lost during handovers or due to corruption. Link-level retransmissions the loss rates of 3G systems due to corruption are relatively low. However**,** recovery at the link layer appears as jitter to the higher layers.

Based on these characteristics, suggests the following configuration parameters to adapt TCP to wireless environments:

- **Large windows:** TCP should support large enough window sizes based on the bandwidth delay product experienced in wireless systems. A larger initial of 2 to 4 segments may increase performance particularly for short transmissions.

- **Limited transmit:** Use small amounts of data are to be transmitted is particularly useful for Fast Retransmit/Fast Recovery.

- **Large MTU:** The larger the MTU (Maximum Transfer Unit) the faster TCP increases the congestion window. Link layers fragment PDUs for transmission anyway according to their needs and large MTUs may be used to increase performance.

- **Selective Acknowledgement (SACK)**

- **Explicit Congestion Notification (ECN):** ECN allows a receiver to inform a sender of congestion in the network by setting the ECN-Echo flag on receiving an IP packet that has experienced congestion. This is used to distinguish packet loss due to transmission errors from packet loss due to congestion.

- **Timestamp:** With the help of timestamps higher delay spikes can be tolerated by TCP without experiencing a spurious timeout. The effect of bandwidth oscillation is also reduced.

- **No header compression:** Header compression is not compatible with TCP options such as SACK or Timestamps.