Indirect TCP

- Two competing insights led to the development of indirect TCP (I-TCP) (Bakre, 1995).
- One is that TCP performs poorly together with wireless links; the other is that TCP within the fixed network cannot be changed. I-TCP segments a TCP connection into a fixed part and a wireless part.
- Figure 9.1 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.
- The correspondent node could also use wireless access.
- The following would then also be applied to the access link of the correspondent host.

- 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.
- Even an unchanged TCP can benefit from the much shorter round trip time, starting retransmission much faster.
- A good place for segmenting the connection between mobile host and correspondent host is at the foreign agent of mobile IP.
- The foreign agent 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.
- However, one can also imagine separating the TCP connections at a special server, e.g., at the entry point to a mobile phone network (e.g., IWF in GSM, GGSN in GPRS).

- The correspondent host in the fixed network does not notice the wireless link or the segmentation of the connection.
- The foreign agent acts as a proxy and relays all data in both directions.
- If the correspondent host sends a packet, the foreign agent acknowledges this packet and tries to forward the packet to the mobile host.
- If the mobile host receives the packet, it acknowledges the packet.
- However, this acknowledgement is only used by the foreign agent.

- If a packet is lost on the wireless link due to a transmission error, the correspondent host would not notice this.
- In this case, the foreign agent tries to retransmit this packet locally to maintain reliable data transport.

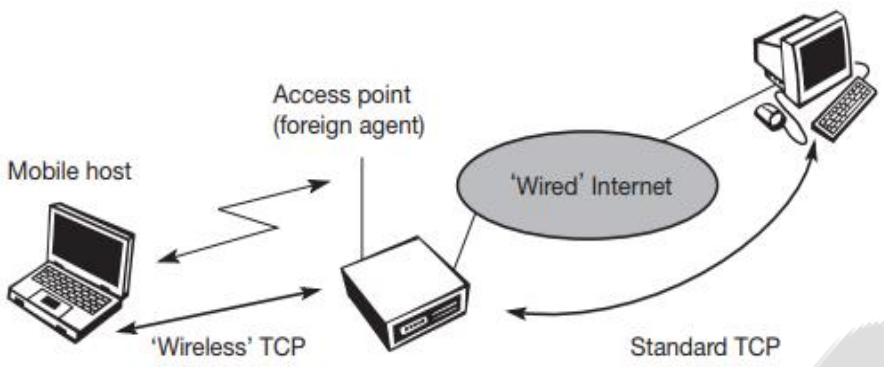
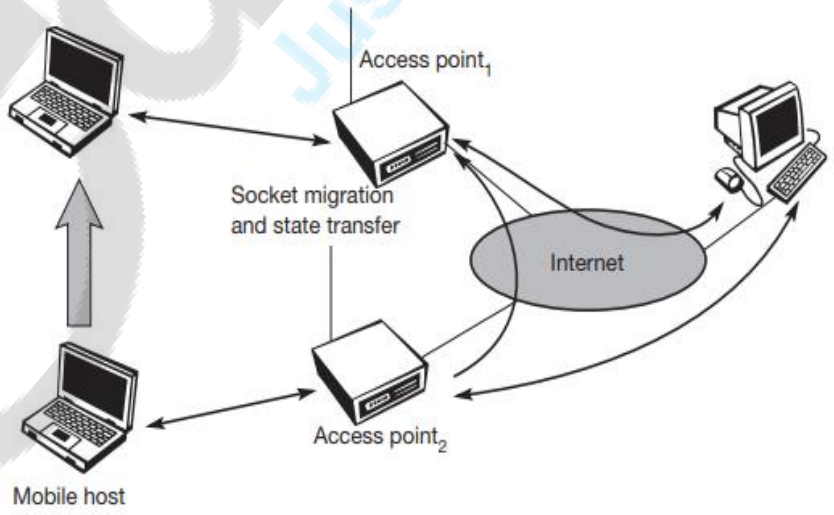


Figure 9.1
Indirect TCP segments a TCP connection into two parts

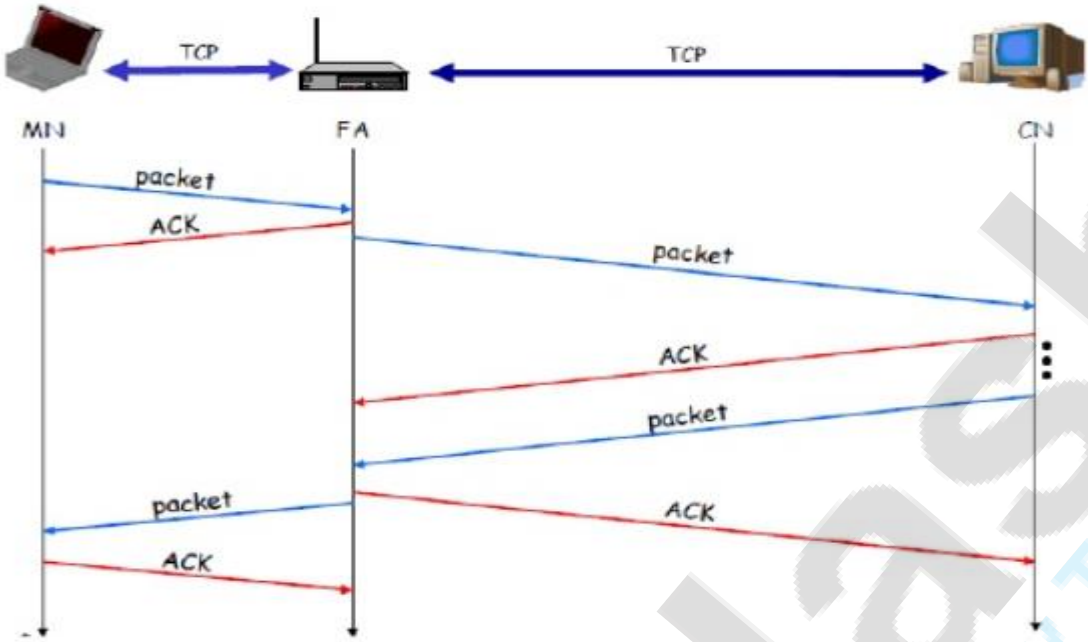
- Similarly, if the mobile host sends a packet, the foreign agent acknowledges this packet and tries to forward it to the correspondent host.
- If the packet is lost on the wireless link, the mobile hosts notice this much faster due to the lower round trip time and can directly retransmit the packet.
- Packet loss in the wired network is now handled by the foreign agent.

- 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.

Figure 9.2
Socket and state migration after handover of a mobile host



Packet delivery in I-TCP is shown below:



There are several advantages with I-TCP:

- I-TCP does not require any changes in the TCP protocol as well as it doesn't required any changes to the hosts (TCP protocol). All current optimizations for TCP still work between the foreign agent and the correspondent host.
- Simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host
  - Transmission errors on the wireless link do not propagate into the fixed network
  - Therefore, a very fast retransmission of packet is possible, the short delay on the mobile hops known
- It is always dangerous to introduce new mechanism in a huge network without knowing exactly how they behave.
  - New optimization can be tested at the last hop, without jeopardizing the stability of the Internet.
- It is easy to use different protocol for wired and wireless network.

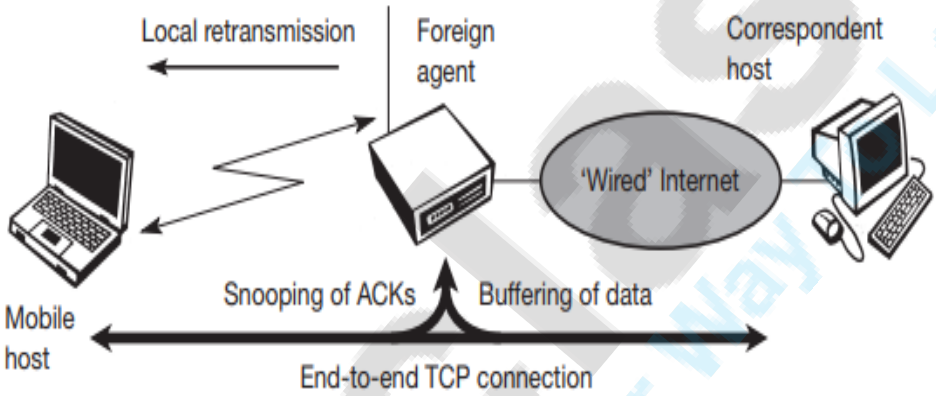
But the idea of segmentation in I-TCP also comes with some disadvantages:

- The loss of the end-to-end semantics: an acknowledgment to a sender no longer means that a received a packet, foreign agent 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.

Snooping TCP

- The main drawbacks of I-TCP is the segmentation of the single TCP connection into two TCP connections. This loses the original end-to-end TCP semantic.
- A new TCP enhancement, which leaves the TCP end-to-end connection intact and is completely transparent, is Snooping TCP.
- The main function of the enhancement is to buffer data close to the mobile host to perform fast local retransmission in case of packet loss.
- A good place for the enhancement of TCP could be the foreign agent in the Mobile IP context (see Figure 9.3).

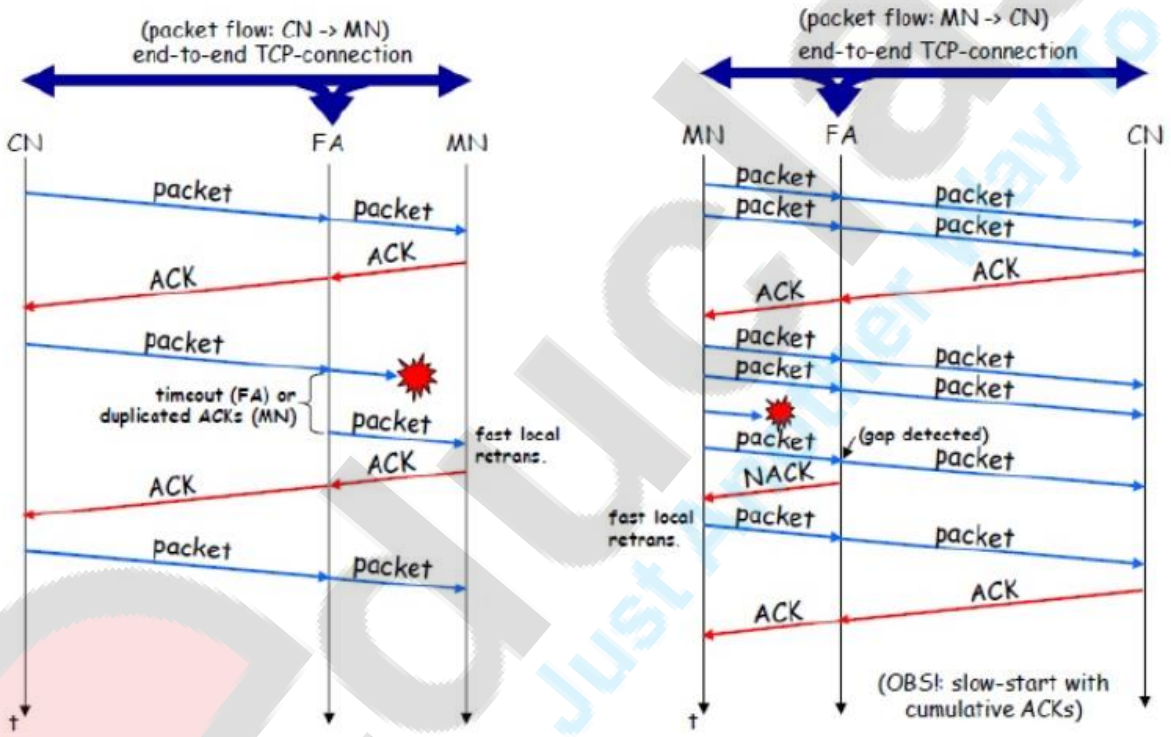
Figure 9.3 Snooping TCP as a transparent TCP extension



- In this approach, 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 foreign agent 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 foreign agent retransmits the packet directly from the buffer, performing a much faster retransmission compared to the correspondent host.
- For transparency, the foreign agent does not acknowledge data to the correspondent host which would violate end-to-end semantic in case of a FA failure.
- However, 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.

- Data transfer from the mobile host with destination correspondent host works as follows.
- The foreign agent 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.



Snooping TCP: Packet delivery

Extending the functions of a foreign agent with a 'snooping' TCP has several advantages:

- 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 corresponding host occurs and the packets are transmitted to the new care-of address.

- No problem arises if the next foreign agent uses the enhancement or not. If not, the approach automatically falls back to the standard solution.

However, the simplicity of the scheme also results in some disadvantages:

- Snooping TCP does not isolate the behavior of the wireless link as well as ITCP. Transmission errors may propagate till corresponding host.
- 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 (e.g., SSL/TLS) snooping TCP can be used.

Mobile TCP

- Both I-TCP and Snooping TCP does not help much, if a mobile hosts get 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.
- The supervisory host is responsible for exchanging data between both parts similar to the proxy in ITCP (see Figure 9.1).
- 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.

The advantages of M-TCP are the following:

- 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.
- Since it does not buffer data in the SH as I-TCP does, it is not necessary to forward buffers to a new SH. Lost packets will be automatically retransmitted to the new SH.

The lack of buffers and changing TCP on the wireless part also has some disadvantages:

- 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.

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 approach is its simplicity. Only minor changes in the mobile host's software result in a performance increase. No changes are required in foreign agent or correspondent host.
- The main disadvantage of this scheme is the insufficient isolation of packet losses. This approach mainly focuses on loss due to handover. Also it effects the efficiency when a CH transmits already delivered packets.

Transmission/time-out freezing

- Quite often, the MAC layer has noticed connection problems, before the connection is actually interrupted from a TCP point of view.
- Additionally, the MAC layer knows the real reason for the interruption and does not assume congestion, as TCP would.
- 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.

- The **advantage** of this approach is that it offers a way to resume TCP connections even after longer interruptions of the connection.
- It is independent of any other TCP mechanism, such as acknowledgements or sequence numbers, so it can be used together with encrypted data.

- However, this scheme has some severe disadvantages.
- Lots of changes have to be made in software of MH, CH and FA.

Freezing the state of TCP does not help in case of some encryption schemes that use time-dependent random numbers. These schemes need resynchronization after interruption.

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.
- 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 RFC 2018, 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.

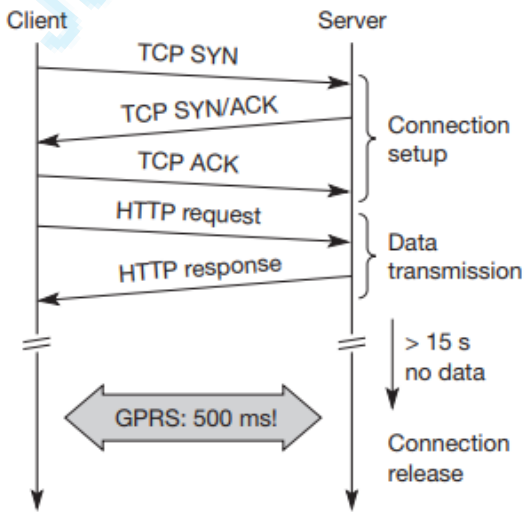
- The advantage of this approach is obvious: a sender retransmits only the lost packets.
- This lowers bandwidth requirements and is extremely helpful in slow wireless links.
- The gain in efficiency is not restricted to wireless links and mobile environments.
- Using selective retransmission is also beneficial in all other networks.
- However, there might be the minor disadvantage of more complex software on the receiver side, because now more buffer is necessary to resequence data and to wait for gaps to be filled.
- But while memory sizes and CPU performance permanently increase, the bandwidth of the air interface remains almost the same.
- Therefore, the higher complexity is no real disadvantage any longer as it was in the early days of TCP.

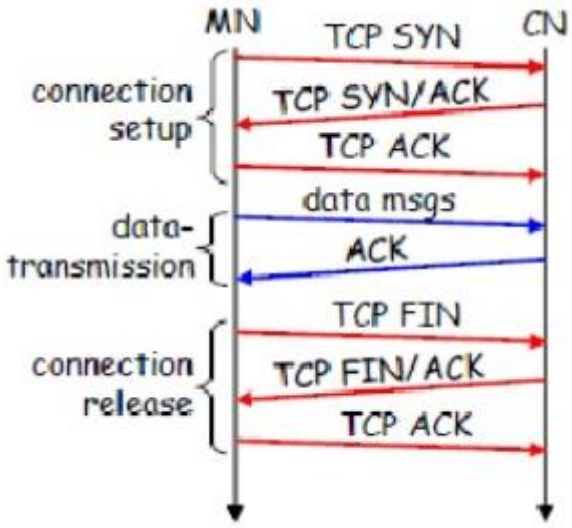
Transaction-oriented TCP

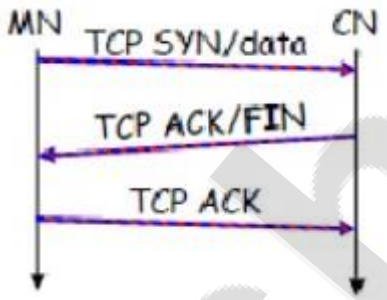
- Assume an application running on the mobile host that sends a short request to a server from time to time, which responds with a short message.
- If the application requires reliable transport of the packets, it may use TCP (many applications of this kind use UDP and solve reliability on a higher, application-oriented layer).

- Using TCP now requires several packets over the wireless link.
- 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.
- Assuming connections with a lot of traffic or with a long duration, this overhead is minimal.
- But in an example of only one data packet, TCP may need seven packets altogether.
- Figure 9.4 shows an example for the overhead introduced by using TCP over GPRS in a web scenario.
- Web services are based on HTTP which requires a reliable transport system.
- In the internet, TCP is used for this purpose. Before a HTTP request can be transmitted the TCP connection has to be established. This already requires three messages.
- If GPRS is used as wide area transport system, one-way delays of 500 ms and more are quite common. The setup of a TCP connection already takes far more than a second.
- This led to the development of a transaction-oriented TCP.

Figure 9.4
Example TCP connection setup overhead







- 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

TCP over 2.5/3G wireless networks

The current internet draft for TCP over 2.5G/3G wireless networks (Inamura, 2002) describes a profile for optimizing TCP over today's and tomorrow's wireless WANs such as GSM/GPRS, UMTS, or cdma2000. The configuration optimizations recommended in this draft can be found in most of today's TCP implementations so this draft does not require an update of millions of TCP stacks. The focus on 2.5G/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 2.5G/3G wireless links:

- **Data rates:** While typical data rates of today's 2.5G systems are 10–20 kbit/s uplink and 20–50 kbit/s downlink, 3G and future 2.5G systems will initially offer data rates around 64 kbit/s uplink and 115–384 kbit/s downlink. Typically, 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. In cellular networks, asymmetry does not exceed 3–6 times, however, considering broadcast systems as additional distribution media (digital radio, satellite systems), asymmetry may reach a factor of 1,000. Serious problems that may reduce throughput dramatically are bandwidth oscillations due to dynamic resource sharing. To support multiple users within a radio cell, a scheduler may have to repeatedly allocate and deallocate resources for each user. This may lead to a periodic allocation and release of a high-speed channel.

- **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 (see chapter 4).

- **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. Handovers are quite often only virtually seamless with outages reaching from some 10 ms (handover in GSM systems) to several seconds (intersystem handover, e.g., from a WLAN to a cellular system using Mobile IP without using additional mechanisms such as multicasting data to multiple access points).

- **Packet loss:** Packets might be lost during handovers or due to corruption. Thanks to link-level retransmissions the loss rates of 2.5G/3G systems due to corruption are relatively low (but still orders of magnitude higher than, e.g., fiber connections!). However, recovery at the link layer appears as jitter to the higher layers.

Based on these characteristics, (Inamura, 2002) 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. With the help of the windows scale option (RFC 1323) and larger buffer sizes this can be accomplished (typical buffer size settings of 16 kbyte are not enough). A larger initial window (more than the typical one segment) of 2 to 4 segments may increase performance particularly for short transmissions (a few segments in total).

- **Limited transmit:** This mechanism, defined in RFC 3042 (Allman, 2001) is an extension of Fast Retransmission/Fast Recovery (Caceres, 1995) and is particularly useful when small amounts of data are to be transmitted (standard for, e.g., web service requests).

- **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. MTU path discovery according to RFC 1191 (IPv4) or RFC 1981 (IPv6) should be used to employ larger segment sizes instead of assuming the small default MTU.

- **Selective Acknowledgement (SACK):** SACK (RFC 2018) allows the selective retransmission of packets and is almost always beneficial compared to the standard cumulative scheme.

- **Explicit Congestion Notification (ECN):** ECN as defined in RFC 3168 (Ramakrishnan, 2001) 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 mechanism makes it easier to distinguish packet loss due to transmission errors from packet loss due to congestion. However, this can only be achieved when ECN capable routers are deployed in the network.

- **Timestamp:** TCP connections with large windows may benefit from more frequent RTT samples provided with timestamps by adapting quicker to changing network conditions. 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:** As the TCP header compression mechanism according to RFC 1144 does not perform well in the presence of packet losses this mechanism should not be used. Header compression according to RFC 2507 or RFC 1144 is not compatible with TCP options such as SACK or timestamps.

It is important to note that although these recommendations are still at the draft-stage, they are already used in i-mode running over FOMA as deployed in Japan and are part of the WAP 2.0 standard (aka TCP with wireless profile).