TCP: Transmission Control Protocol which is a transport layer protocol mainly for providing end – to – end application communication.

Traditional TCP:

The Traditional TCP highlights several mechanisms of the transmission control protocol (TCP) that influence the efficiency of TCP in a mobile environment.

- Congestion Control
- Slow start
- Fast retransmit / Fast recovery, and
- Implications on Mobility.

❖ Congestion Control:

- Generally, TCP has been used for *fixed network s with fixed end systems,* where data transmission takes place using network adapters, fiber optics, copper wires, special hardware for routers etc.
- Since, the hardware and software was very matured it won't produce any *loss of packets* or *transmission errors.*
- Hence, a packet loss in a fixed network is mainly due to a situation named as *congestion.*

Definition:

- It is defined as a "a temporary overload, i.e. a 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 !ink". Thus, results in a dropping of packets.

Problem due to Congestion:

- *A* dropped *packet* is lost for the transmission, and the receiver notices a gap in the packet stream.
- The receiver intimates the sender about this loss by indirectly, not sending the acknowledgement for the missing packets.

Solution:

- By noticing the missing acknowledgement for the lost packet, the sender assumes this is due to *congestion.*
- To solve this problem, it simply retransmits the missing packets and also slows down the transmission rate.
- Thus, TCP plays a role in congestion control and also provides a *guaranteed delivery.*

❖ Slow Start:

- To get rid of congestion quickly, TCP follows a mechanism known as "*slow start*".
- When sender finds congestion, it 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).
- After receiving acknowledgements for the 2 packets the sender again adds 2 to the congestion window, one for each of the acknowledgements.

- Now the congestion window equals 4. This scheme doubles the congestion window every time the acknowledgements come back.
- This is called the exponential growth of the congestion window in the slow start mechanism.

Congestion Threshold:

- The exponential growth stops at a value named as **congestion threshold,** then will be a *linear increase by adding* 1 to the congestion window for each acknowledgements.
- This linear increase continues until timeout for the missing packets occurs.
- Again, the *exponential growth* starts until it reaches the new congestion threshold, then the window grows in linear fashion.

❖ **Fast retransmit / Fast recovery**

Two things lead to a reduction of the congestion threshold are,

(i) One is a sender receiving continuous acknowledgements for the same packet. This explains two things.

- One is that the receiver got all packets up to the acknowledged packet in sequence. In TCP, a receiver sends acknowledgements only if it receives any packets from the sender. Hence, the gap in the packet stream is not only due to congestion, it may also due to a transmission error.

Fast Retransmit:

The sender can now retransmit the missing packets before the timer expires. This is said to be fast retransmit.

Fast Recovery:

The receipt of acknowledgements shows that there is no congestion to justify a slow start.
The sender can continue with the current congestion window and performs a *fast recovery* from the packet loss.

(ii) The *other reason* for activating slow start is a time-out due to missing acknowledgement.

❖ **Implications on Mobility:**

Slow start on mobile networks:

Even though slow start is the most useful mechanism in fixed networks, it *decreases the efficiency of TCP,* when it is used with mobile networks due to wrong assumptions over missing acknowledgements, which always not only due to *congestion* but
also due to *packet loss.*

Mobility:

- 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.

Example: By using mobile IP, there could still be some packets in transit to the old foreign agent while the mobile node moves to the new foreign agent.

- The old foreign agent may not be able to forward those packets to The new foreign agent and hence results in loss of packet, which mainly due to problems of *rerouting* traffic.

Problem in TCP:

- The fundamental design problem in TCP is that, difficulty in identifying where the missing packets is due to,

  Transmission errors, or
  Congestion

- Standard TCP reacts with slow start if acknowledgements are missing, which doesn't help in the case of transmission errors over wireless links and which doesn't really help during handover.
- This behavior results in severe performance degradation of an unchanged TCP if used together with wireless links or mobile nodes.
- Thus to support mobility, we need several improvements in standard TCP.

Classical TCP Improvement:

To increase TCP performance in wireless and mobile environments, several research projects were started. They are,

- Indirect TCP
- Snooping TCP
- Mobile TCP
- Fast Retransmit / Fast Recovery
- Transmission / Timeout Freezing
- Selective Retransmission
- Transaction oriented TCP

Indirect TCP:

Need for I-TCP:

Two reasons for the development of Indirect - TCP are,

(i) TCP performance is very poor with wireless links.

(ii) The TCP within fixed network cannot be changed.

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

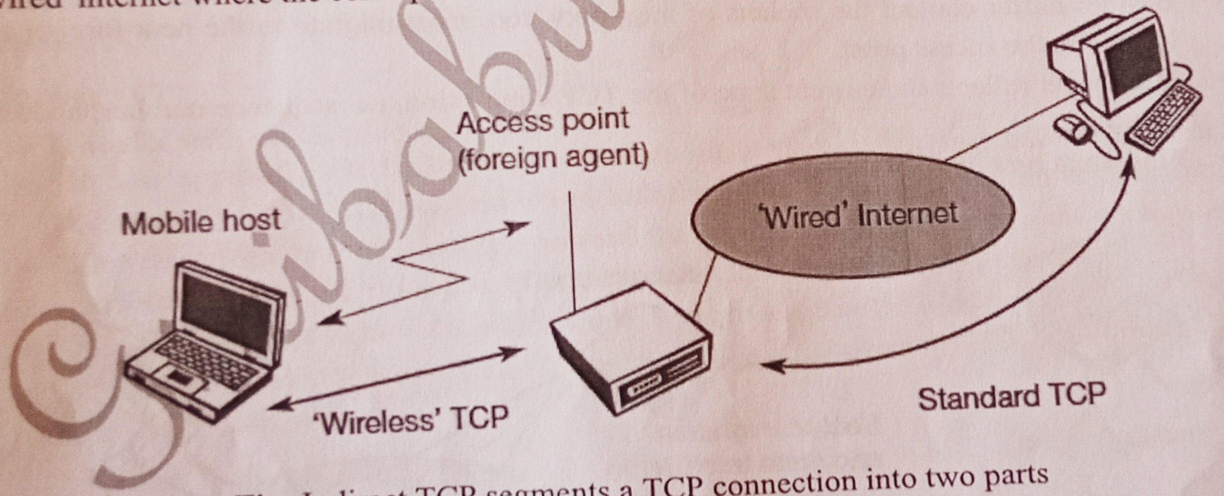


Fig: Indirect TCP segments a TCP connection into two parts

- From the above fig the *standard TCP* is used between *fixed computer* and the *access point (AP)*.
- Instead of mobile host, now AP terminates the connection, which acts as a proxy.
- This means that *AP* is now *mobile host* for *fixed host,* and fixed host for the mobile host.
- A good place for segmenting both the Mobile Host (MH) and the Correspondent Host (CH) is at the foreign agent of mobile IP.

Correspondent Host and Mobile Host:

Correspondent Host (CH):

- The CH in the fixed network doesn't notice the wireless link or the segmentation of the connection.
- The AP acts as a proxy and relays all data in both directions.
- If CH sends a packet, the foreign agent acknowledges it and tries to forward the packet to the MH.
- If MH receives the packet, it acknowledges the packet, which in turn only used by the foreign agent.

Packet Loss

The packet loss in the wireless link due to transmission errors, won't be known by CH, the foreign agent itself will retransmit the packet.

Mobile Host (MH):

Similarly, if the MH sends a packet, the foreign agent acknowledges this packet and tries to forward it to the CH.

Packet Loss

If the packet is lost on *wireless link, the* MH notice this much faster due to lower RTT and can directly retransmit the packet. Packet loss in the wired network is now handled by the foreign agent.

Actions of Indirect TCP during Handover:

- The AP acts as a *proxy* buffering packets for retransmission.
- After handover, the old proxy must forward buffered packets to a new proxy, because it has already acknowledged the data.
- Besides buffer content the sockets of the proxy, too, must migrate to the new foreign agent located in the access point.
- The socket reflects the current state of the TCP connection, i.e. sequence number, addresses, ports etc.
- This situation is shown as,

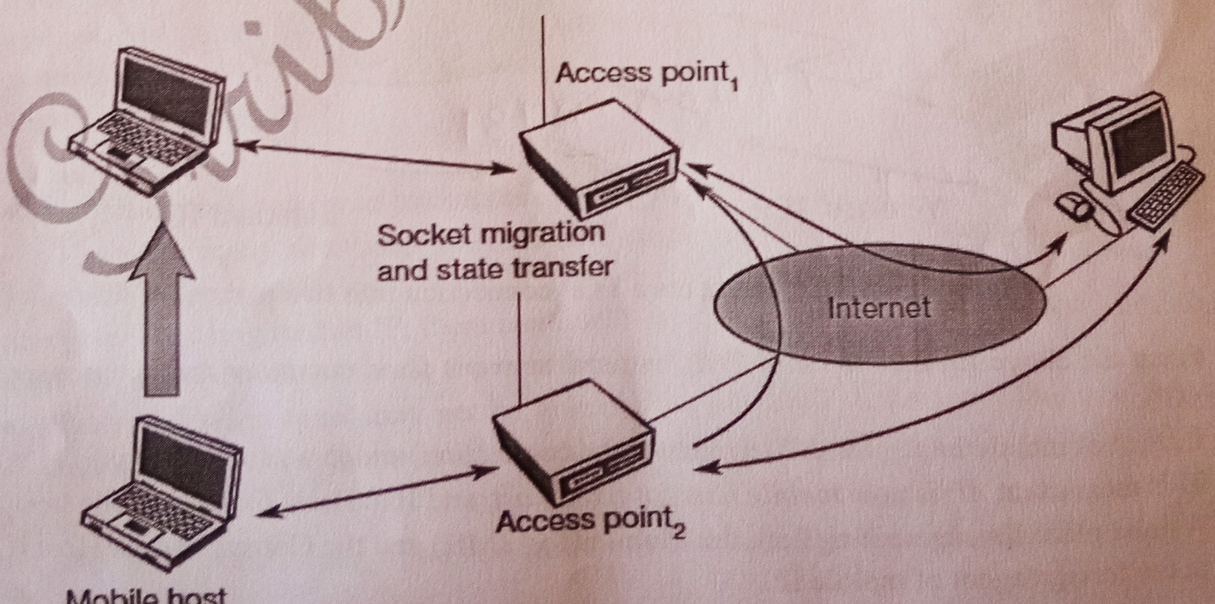


Fig: Socket and state migration after handover of a mobile host

Advantages of Indirect TCP:

There are several advantages of I-TCP. They are,

(i) It *doesn't* require *any changes* in the TCP protocol.

(ii) Due to strict partitioning into wireless and fixed networks, transmission errors on the wireless link *cannot propagate* into fixed network.

(iii) TCP performance can be improved using new mechanisms, with I-TCP only between MH and foreign agent allows for testing different solutions *without jeopardizing* the stability of the network.

(iv) Partitioning into two connections also allows the use of a different transport layer protocol between the foreign agent and the mobile host or the use of compressed headers etc.

(v) Delay on wireless links will be higher than the delay on wired networks. It will be reduced by Indirect TCP.

Disadvantages of Indirect TCP:

Some of the disadvantages of I-TCP are,

(i) If the foreign agent partitioning the TCP connection crashes, there will be a loss of the end to end semantics, it is a great problem.

(ii) Practically, increased handover latency may be much more problematic.

(iii) The foreign agent must be a trusted entity because, the TCP connections end this point, thus the foreign agent has to be integrated into all security mechanisms.

Snooping TCP:

Drawbacks of I-TCP:

The one of the demerit of I-TCP is the segmentation of single TCP connection into two TCP connections. This loses the original end – to – end semantic. This problem can be solved using Snooping TCP.

Snooping TCP:

Data Transfer with destination as Mobile Host:

- 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 reason for buffering packets toward the mobile node is to enable the foreign agent to perform a *local retransmission* in case of packet loss on the wireless link.
- The foreign agent buffers the packets until it receives an acknowledgement.
- It can also be able to identify and filter the *duplicate acknowledgements* to avoid unnecessary retransmission of data.

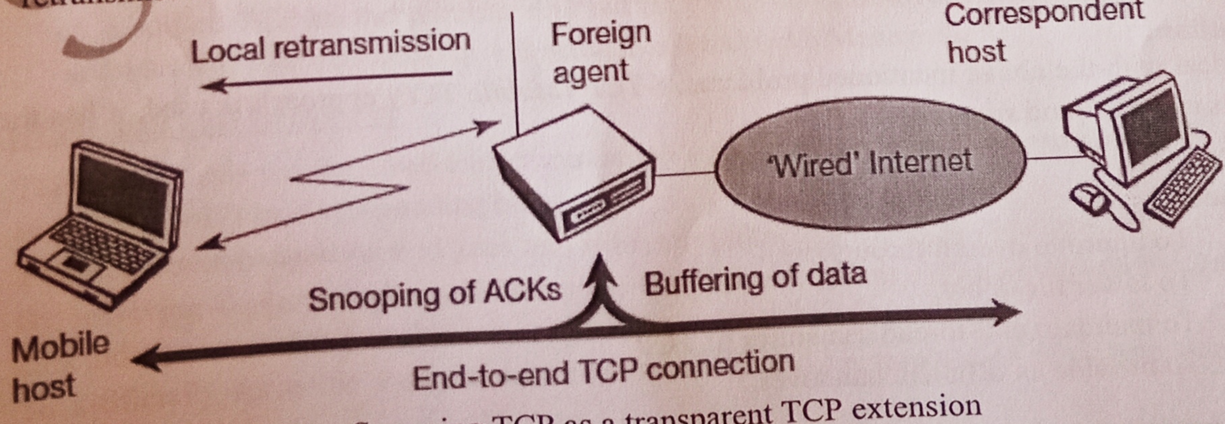


Fig: Snooping TCP as a transparent TCP extension

Data transfer with destination as Correspondent Host:

The data transfer from mobile 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 *NACK* (Negative ACK) to the mobile host.
- Thereby, the mobile host can now retransmit the missing packet immediately, and reordering of packets can rakes place automatically.

Advantages of Snooping TCP:

Several advantages of this snooping TCP mechanism are,

(i) The end-to-end TCP semantic is *preserved.*

(ii) The correspondent host *doesn't* need to be *changed* i.e., most of the enhancements are in the foreign agent.

(iii) *No need* for the h*andover* of state as soon as the mobile host moves to another foreign agent.

(iv) It doesn't matter if the next foreign agent uses the enhancement or not. If not, the approach automatically falls back to the standard solution. This is one of the problems of *I-TCP.*

Disadvantages of Snooping TCP:

Some of the disadvantages of this scheme are,

(i) Snooping TCP doesn't isolate the behavior of the wireless link as well as I-TCP.

(ii) Using NACK between the foreign agent and the Mobile Host assumes additional mechanisms on the Mobile Host.

(iii) All efforts for snooping and buffering data may be useless if certain encryption schemes are applied end to end between the correspondent host and mobile host.

Mobile TCP:

Problem with I - TCP and Snooping TCP:

- Not only the *dropping packets* due to a handover or higher bit error rates is the problem in the wireless links, the occurrence *of lengthy* and/or *frequent disconnections* is another problem.
- If the mobile is disconnected,

(i) In the case of *I-TCP,* the proxy has to buffer more and more data, so the longer the period of disconnection, the *more buffers* is needed.

(ii) The *snooping* approach also suffers from being disconnected. The mobile will not be able to send ACKs so, snooping cannot help in this situation.

Solution:

To deal with the above mentioned problem, *M-TCP (Mobile TCP)* approach is used, it has the same goals as *I-TCP* and *snooping TCP.*

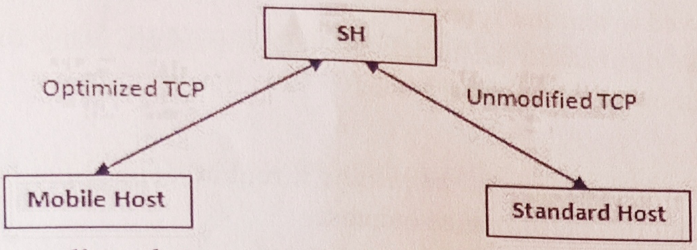
M-TCP:

It wants,

To improve overall throughput,
To lower the delay,
To maintain end-to-end semantics of TCP
To provide an efficient handover

Methodology:

- M-TCP splits the TCP connection into parts as I-TCP does.
- An un*modified TCP* is used between the standard host and supervisory host (SH) connection, while an optimized TCP is used on the *Supervisory host(SH) and mobile host(MH).*
- The SH is responsible for exchanging data between both parts similar to the proxy in I-TCP.



- The SH monitors all *packets* sent to the *MH* and *ACKs returned* from the *MH.* If a packet is lost, it has to be retransmitted by the original sender; hence, no buffering by SH is needed.
- If the SH doesn't receive an ACK for some time, it assumes the MH is disconnected, and forces the sender to go in to *persistent mode, by* making the *window size to O.*
- This state of the sender will avoid the continuous retransmission of data, and it will continue till the SH detects the connectivity again.
- After the connection is found, the sender can continue sending at full speed. This mechanism *doesn't* 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 manages* to implement fair sharing over the wireless link.

Advantages of M-TCP:

Several advantages of M-TCP are,

(i) It maintains TCP *end-to-end* semantics.

(ii) If the MH is disconnected, it avoids *useless retransmissions, slow starts* or *breaking connections by* simply shrinking the sender's window to 0.

(iii) Since no buffering of packets in SH as I-TCP, there is also no *need to forward* packets to a new SH.

Disadvantages of M-TCP:

Few disadvantages of M - TCP are,

(i) As SH doesn't act as proxy as in I·TCP, the packet loss on the wireless link due to bit errors in propagated to the sender. M-TCP assumes low bit error rates, which is not always a valid assumption.

(ii) A modified TCP on the wireless link not only requires modification to the MH protocol software but also new network elements like Bandwidth Manager.

Fast Retransmit / Fast Recovery:

- The mechanisms we discussed earlier moving to a new foreign agent can cause *packet loss* or *time out* at MH's or corresponding hosts.
- TCP concludes congestion and goes into *slow start,* although there is no congestion.
- By receiving duplicate acknowledgements, the Fast retransmit / Fast recovery mechanism conclude it as a packet *loss* without *congestion.*
- To artificially force the fast retransmit on the MH and correspondent host side, canceres proposed an idea,

(i) As soon as the mobile host registers at a new foreign agent using mobile IP, it starts sending duplicated acknowledgements to correspondent hosts.

(ii) The proposal is to send *three duplicates.*

(iii) 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.

(iv) The mobile host *retransmits* all unacknowledged packets using the current congestion window size without going into slow start.

Advantages:

- The advantage of this approach is its *simplicity,* it requires only minor changes in the MH's software already result in a performance increase.
- *No* foreign *agent* or *correspondent host* has to be changed

Disadvantage:
The main disadvantage of this mechanism is the *insufficient isolation* of packet losses.

Transmission / Time out Freezing:

All mechanisms described so far can handle short interruptions of the connection either due to handover or transmission errors on the wireless link.

Methodology:

- The MAC layer has already 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 MH and CH can be informed.
- Otherwise, the CH goes into *slow start* assuming congestion and finally *breaks* the connection.

Advantage:
It offers a way to *resume TCP* connections even after longer interruptions of the connection.

Disadvantages:
Few disadvantages of this scheme is that,

(i) Not only does the software on the MH have to be changed, to be more effective the correspondent host cannot remain unchanged.

(ii) *All mechanisms* rely on the capability of the *MAC layer* to detect future interruptions.

(iii) *Freezing the state* of TCP doesn't help in case of some *encryption schemes* that use time dependent random numbers.

Selective Retransmission:

- Most useful extension of TCP is the use of *selective retransmission.*
- TCP acknowledgements are cumulative i.e., they acknowledge in order receipt of packets upto a certain packet.

- For a loss of single packet, the TCP has to retransmit everything starting from the lost packet which is called as *go-back-n* retransmission.

Problem:

This results in wastage of bandwidth.

Solution:

It can be overcomed by retransmitting only selective packets i.e. the packets which get lost. This type of retransmission is termed as "selective retransmission".

Advantage:

Reduce Bandwidth wastages by retransmitting only a lost packet.

Disadvantage:

More complex software on the receiver side, because now more buffers is necessary to *rearrange data* and to *wait for gaps* to be filled.

Transaction Oriented TCP:

- For Example, if 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 needs a reliable communication, it may use TCP.
- Using TCP may now require several packets over the wireless links.

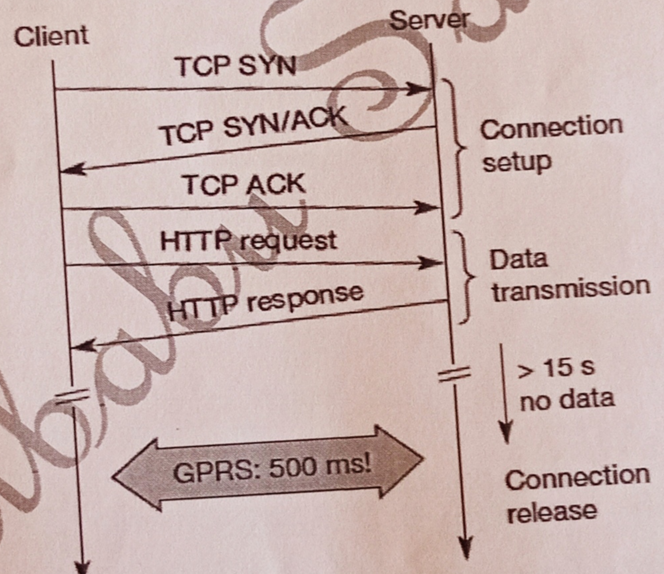


Fig: Example TCP connection setup overhead

Problem:

- 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.
- If the communication is for long duration, this overhead is minimal. If it is for sending one data packet, it seems to be hard, requires seven packets.

Solution:

- To avoid the above mentioned overhead, the transaction oriented TCP has been developed.

- It is denoted as T / TCP, can combine packets for connection establishment and release with user data packets.
- Thus, it reduces the number of packets down to two instead of seven.

Advantage:

Its merit is the reduction in the overhead which standard TCP has for connection setup and release.

Disadvantage:

- It requires changes in the mobile host and all the correspondent hosts.
- It no longer hides mobility.
- It exhibits several security problems.
- An additional scheme that can be used to reduce TCP overhead is header compression.

Overview of classical enhancements to TCP for mobility

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, security problems
Snooping TCP	Snoops data and acknowledgements, local retransmission	Transparent for end-to-end connection, MAC integration possible	Insufficient isolation of wireless link, security problems
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, security problems
Fast retransmit/ fast recovery	Avoids slow-start after roaming	Simple and efficient	Mixed layers, not transparent
Transmission/ time-out freezing	Freezes TCP state at disconnection, resumes after reconnection	Independent of content, works for longer interruptions	Changes in TCP required, MAC dependent
Selective retransmission	Retransmits only lost data	Very efficient	Slightly more complex receiver software, more buffer space needed
Transaction-oriented TCP	Combines connection setup/release and data transmission	Efficient for certain applications	Changes in TCP required, not transparent, security problems