UNIT V

Transmission Control Protocol (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.

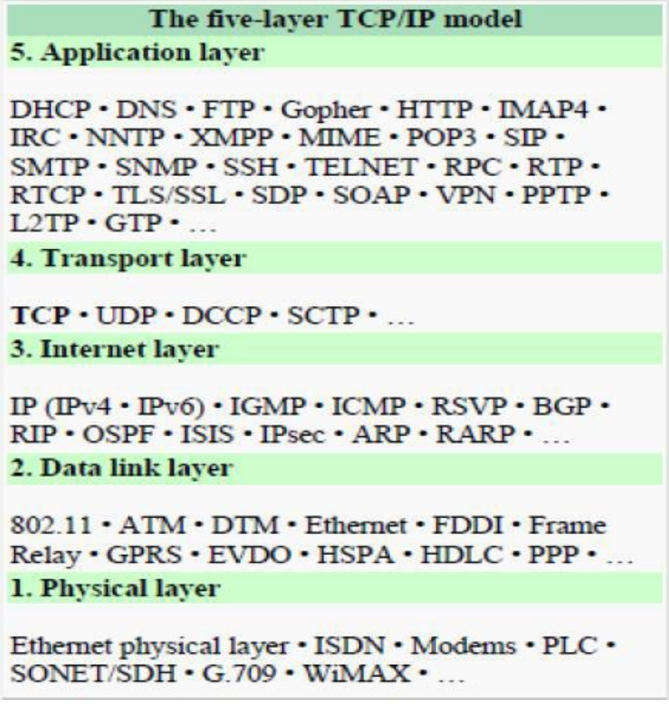
The major responsibilities of TCP in an active session are to:

• **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.



The five-layer TCP/IP model
5. Application layer
DHCP • DNS • FTP • Gopher • HTTP • IMAP4 • IRC • NNTP • XMPP • MIME • POP3 • SIP • SMTP • SNMP • SSH • TELNET • RPC • RTP • RTCP • TLS/SSL • SDP • SOAP • VPN • PPTP • L2TP • GTP • …
4. Transport layer
TCP • UDP • DCCP • SCTP • …
3. Internet layer
IP (IPv4 • IPv6) • IGMP • ICMP • RSVP • BGP • RIP • OSPF • ISIS • IPsec • ARP • RARP • …
2. Data link layer
802.11 • ATM • DTM • Ethernet • FDDI • Frame Relay • GPRS • EVDO • HSPA • HDLC • PPP • …
1. Physical layer
Ethernet physical layer • ISDN • Modems • PLC • SONET/SDH • G.709 • WiMAX • …

Congestion Control

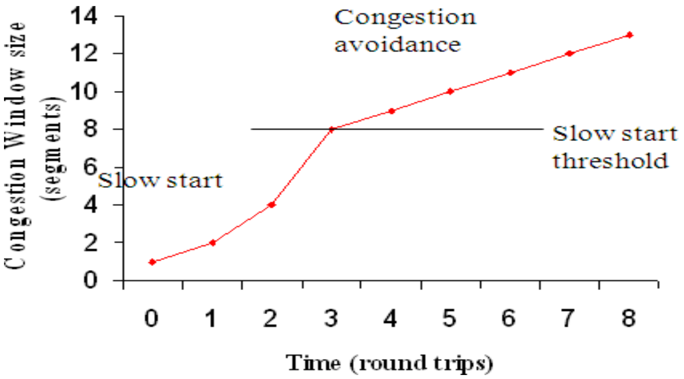
A transport layer protocol such as TCP has been designed for fixed networks with fixed end-systems. Congestion may appear from time to time even in carefully designed networks. The packet buffers of a router are filled and the router cannot forward the packets fast enough because the sum of the input 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. 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.

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.



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.

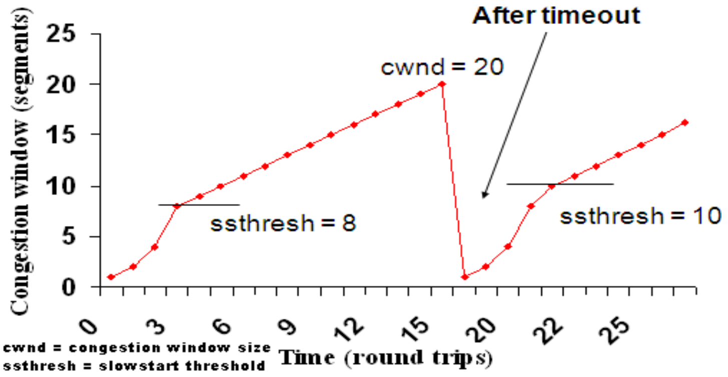
Fast retransmit/fast recovery

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 effects the efficiency when a CH transmits already delivered packets.



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.

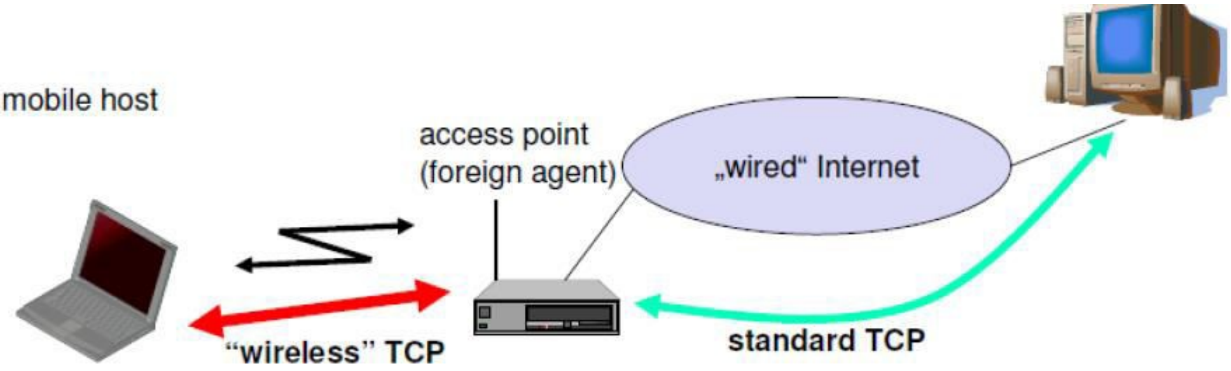
Mobility itself can cause packet loss. There are many situations where a soft handover from one access point to another is not possible for a mobile end-system.

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

Classical TCP Improvements:

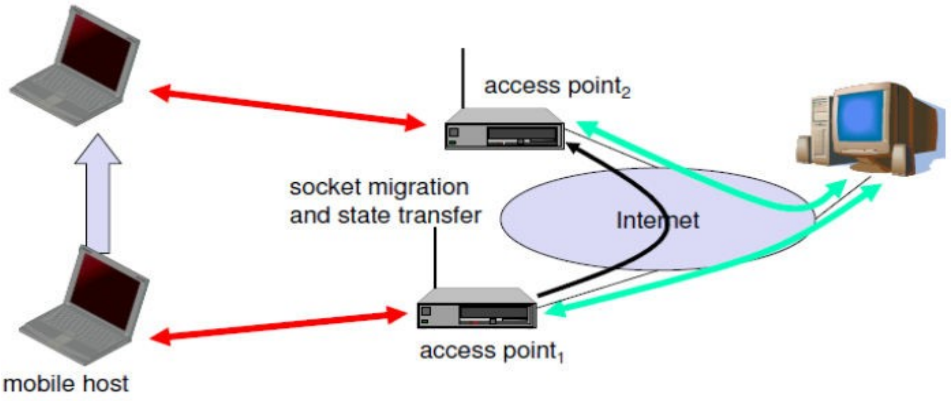
Indirect TCP (I-TCP)

Indirect TCP segments a TCP connection into a fixed part and a 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.



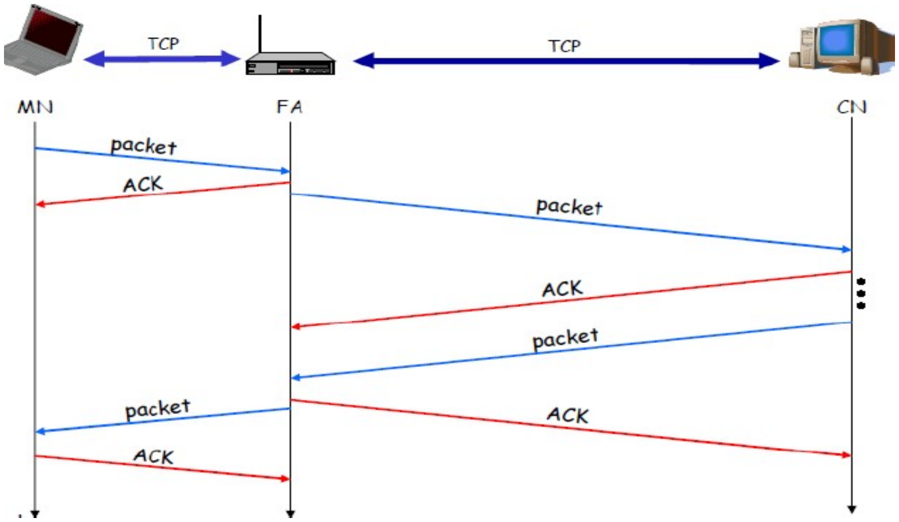
Standard TCP is used between the fixed computer and the access point. No computer in the internet recognizes any changes to TCP. Instead of the mobile host, the access point now terminates the standard TCP connection, acting as a proxy. This means that the access point is now seen as the mobile host 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.



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:



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

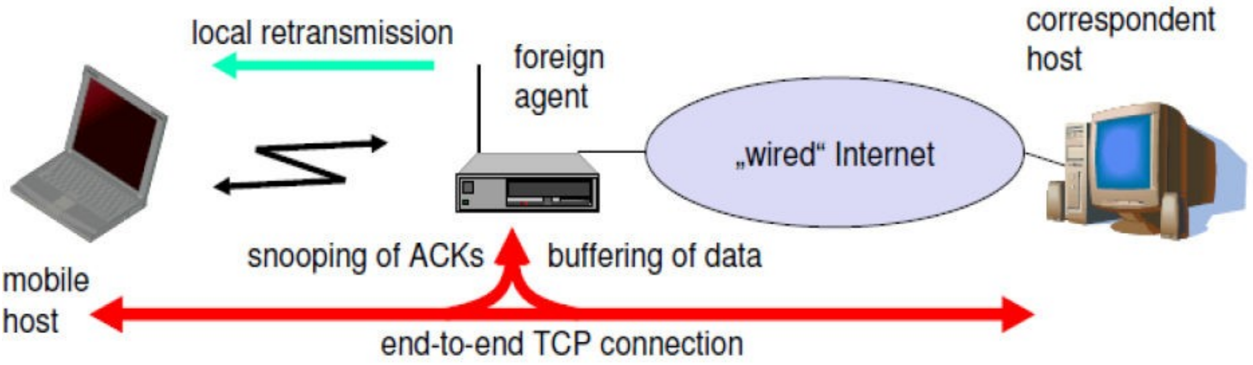
Loss of end-to-end semantics: - an acknowledgement to a sender no longer means that a receiver really has received a packet, foreign agents might crash.

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

Snooping TCP:

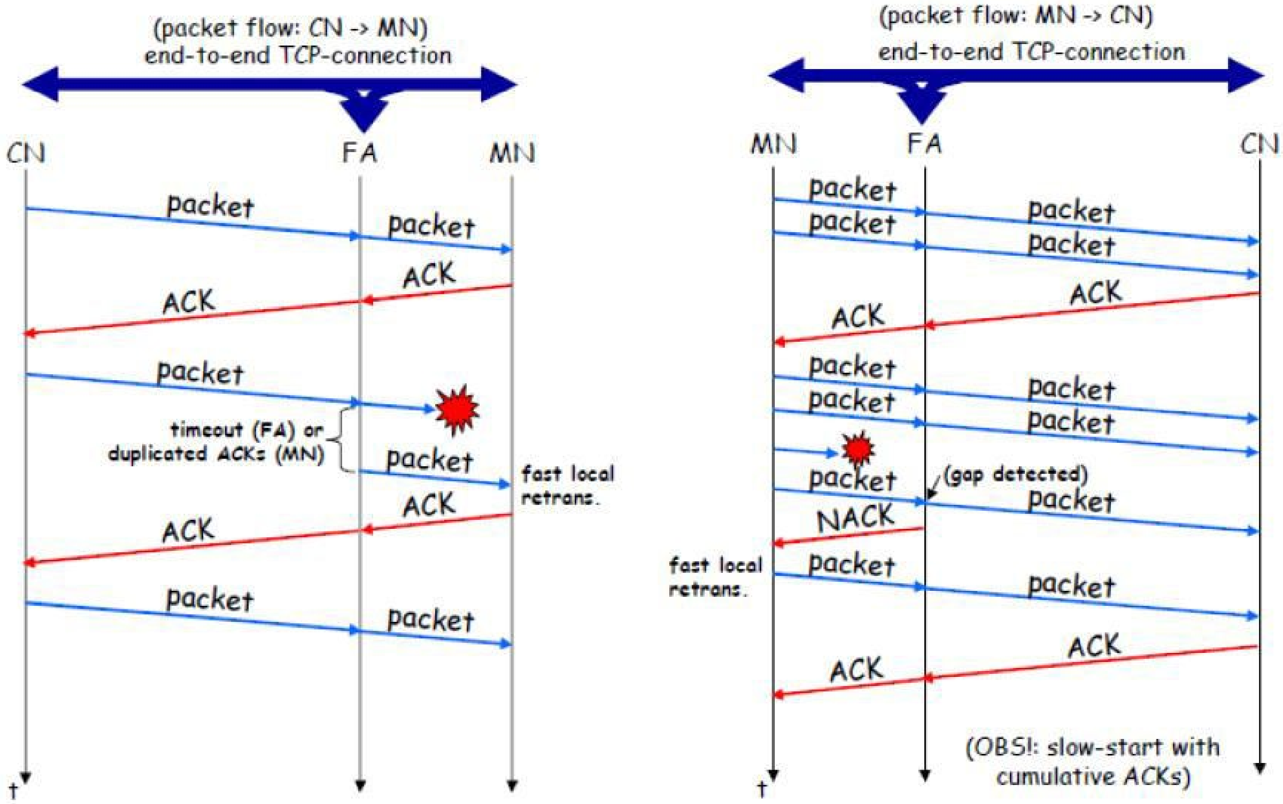
The main drawback of I-TCP is the segmentation of the single TCP connection into two TCP connections, which loses the original end-to-end TCP semantic. A new enhancement, which leaves the TCP connection intact and is completely transparent, is Snooping TCP. The main function is to buffer data close to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



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 foreignagent could receive a duplicate ACK which also shows the loss of a packet. Now, the FAretransmits the packet directly from the buffer thus performing a faster retransmissioncompared to the CH. For transparency, the FA does not acknowledge data to the CH, whichwould violate end-to-end semantic in case of a FA failure. The foreign agent can filter theduplicate acknowledgements to avoid unnecessary retransmissions of data from thecorrespondent host. If the foreign agent now crashes, the time-out of the correspondent hoststill 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.



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.

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.

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.

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.

Selective retransmission:

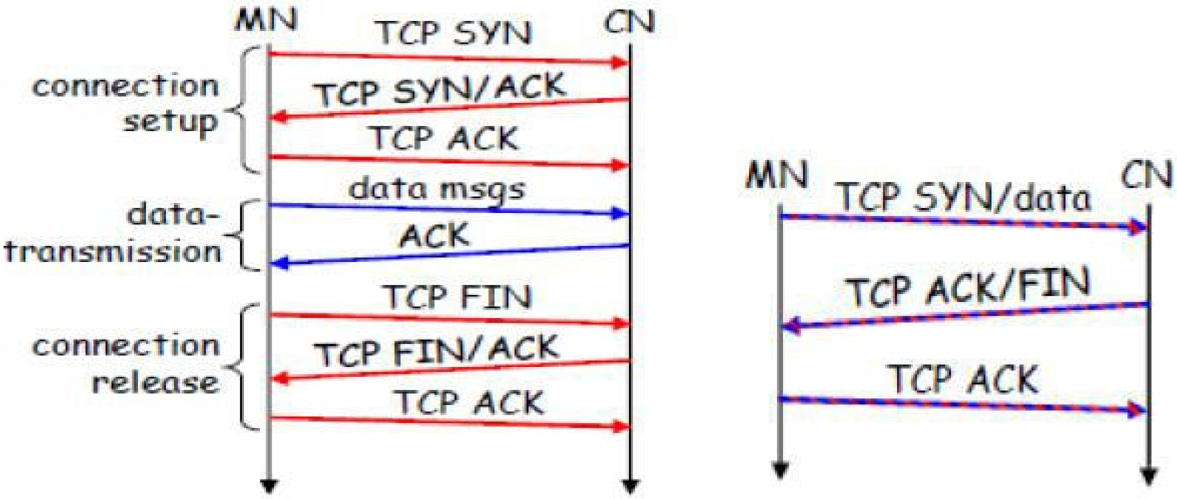
A very useful extension of TCP is the use of selective retransmission. TCP acknowledgements are cumulative, i.e., they acknowledge in-order receipt of packets up to a certain packet. A single acknowledgement confirms reception of all packets upto a certain packet. If a single packet is lost, the sender 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 a 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.

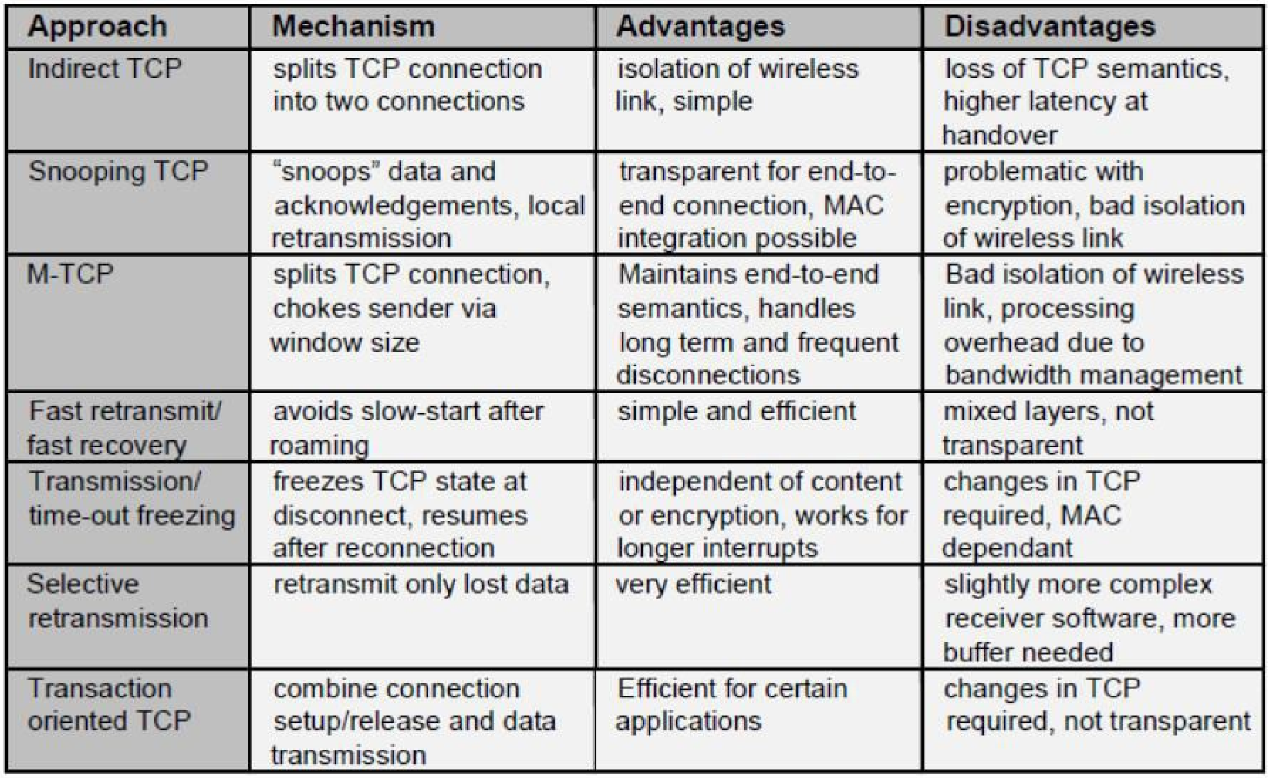
Transaction-oriented TCP

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

Features of windows CE

Applications and Services Development

Describes the operating system functionality that is available in Windows CE for developing applications and services.

Applications - End User

Describes the operating system functionality that is available for developing end user applications.

Communication Services and Networking

Describes the networking and communications capabilities in Windows CE that enable devices to connect and communicate with other devices and people over both wireless and wired networks.

Core OS Services

Describes the core operating system (OS) services that are available in Windows CE. Core OS services contain information on the Windows CE kernel and other features common to all Windows CE OS designs. The core OS services enable low-level tasks such as process, thread, and memory management.

File Systems and Data Store

Provides an overview of the file systems and data store architecture in Windows CE.

Fonts

Provides an overview of fonts and font technologies that are supported in Windows CE. Describes how you can replace fonts, specify a directory from which the OS should load fonts, and change the font size for the Help system. Also describes how you can enable ClearType, antialiased fonts, linked fonts, end-user-defined-characters (EUDC), and line breaking for Asian fonts.

Graphics and Multimedia Technologies

Describes the graphics and multimedia technologies that are supported in Windows CE. Includes detailed descriptions of the audio, graphics, and media support in Windows CE.

International

Describes the International support in Windows CE. The International technologies in Windows CE are comprised of a collection of functionality that provides general locale services and locale-specific support for certain key capabilities.

Internet Client Services

Describes the support for Internet client services in Windows CE. Windows CE provides support for browser applications, technologies that enable you to create custom browsers, and run-time engines for parsing and translating scripting languages.

Security

Provides an overview of the security technologies that enable you to enhance the security of your devices or applications.

Shell and User Interface

Provides a description of the shell and user interface technologies in Windows CE. These include the functionality that is necessary for a user to interact with a Windows CE-based device and the underlying OS.

Voice over IP Phone Services

Describes the technologies that are available in Windows CE to build IP phone devices.

Windows CE Error Reporting

Describes the Windows CE Error Reporting technology. Windows CE Error Reporting allows a device to save key information about the state of the machine at the time of a program crash.

PalmOS

Palm OS uses multitasking, but only one task is for applications. The user uses one application at a time, one application program must finish before the next can be selected. This constraint allows the operating system to devote full attention to the application that is open. The space needed by the system for any application that is running is kept in dynamic, reusable random access memory (RAM). The application and its related database are kept in what is called permanent storage, but here the permanent storage is RAM (rather than a hard disk) that cannot be reused as the dynamic RAM can. Palm OS divides an application into runnable code and different types of data elements, such as user interface elements and icons. The data elements can be easily changed without necessarily having to rewrite code.

Palm OS comes with these applications built-in: Dates, Address Book, To Do List, Memo Pad, Calculator, and Password Protection. New applications can be written and added using several facilities that accelerate development.

Palm supports Metrowerks' CodeWarrior as the official software development kit (SDK), using a Macintosh or Windows environment. UNIXplatform users can use a kit called GCC, which is available through the Free Software Foundation. Programmers can use C, C++, assembler, or scripting. The Palm user interface is emulated within a window in the desktop environment, encouraging rapid application development. Simpler applications can be developed using Palm's forms interface.

Palm OS comes with communication interfaces to infrared transmissiondevices, TCP/IP (for Web connection through wireless or wireline devices), and, optionally, barcode recognition scanners.

WWW

The World Wide Web abbreviated as WWW or W3, commonly known as the Web) is a system of interlinked hypertextdocuments that are accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

Wireless Application Protocol

WAP, Wireless Application Protocol aims to provide Internet content and advanced telephony services to digital mobile phones, pagers and other wireless terminals. The protocol family works across different wireless network environments and makes web pages visible on low-resolution and low-bandwidth devices. WAP phones are "smart phones" allowing their users to respond to e-mail, access computer databases and to empower the phone to interact with Internet-based content and e-mail.

WAP specifies a Wireless application Environment and Wireless Protocols. The Wireless application environment (WAE) is based on WSP (Wireless Session Protocol) and WTP (Wireless Transaction Protocol).

The basic construction of WAP architecture can be explained using the following model. The order of the independent levels – which are a hierarchy - has the advantage that the system is very flexible and can be scaled up or down. Because of the different levels – or stacks - this is called the "WAP Stack", which is divided into 5 different levels.

- Application Layer: Wireless Application Environment (WAE).
- Session Layer: Wireless Session Protocol (WSP).

- Transaction Layer: Wireless Transaction Protocol (WTP).
- Security Layer: Wireless Transport Layer Security (WTLS).

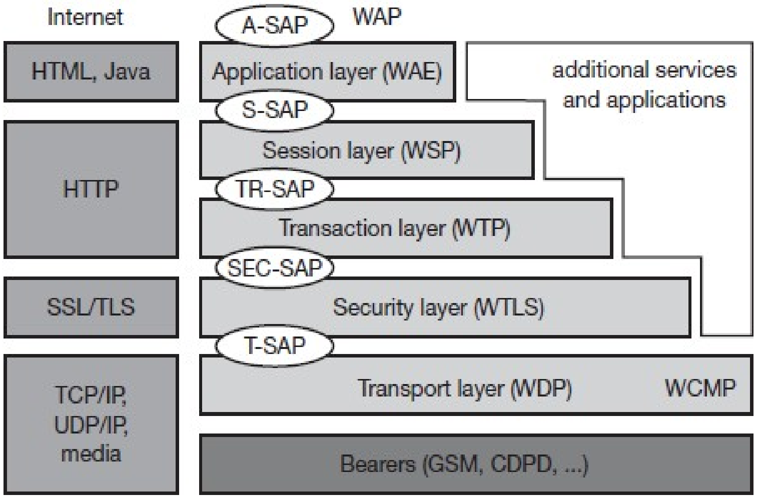


Figure 10.9 Components and interface of the WAP 1.x architecture

- Transport Layer: Wireless Datagram Protocol (WDP).

Each stack overlaps with the stack below. This stack architecture makes it possible for software manufacturers to develop applications and services for certain stacks. They may even develop services for stacks which are not specified yet.

The WAP stack is an entity of protocols which cover the wireless data transfer. The diagram above shows the order of the different stacks and their protocols. This includes the stacks responsible for the layout as well as the stacks resposible for the actual data transfer. The highest level or stack is the one which deals with the layout. A lower stack is responsible for the transfer and the security through WTLS (Wireless Transport Layer Security). All stacks lower than this one are being called network stack. Due to this hierarchy of stacks any changes made in the network stacks will have no influence over the stacks above

Application Layer (WAE and WTA)

The environment for wireless applications (Wireless Application Environment WAE) and the application for wireless phones (Wireless Telephony Application WTA) are the highest layer in the hierarchy of WAP architechture. These two are the main interface to the client device, which gives and controls the description language, the script language of any application and the specifics of the telephony. WAE and WTA have only a few easy functions on the client device, like the maintenance of a history list, for example.

Session Layer (Wireless Session Protocol WSP)

The Wireless Session Protocol (WSP) has all the specifications for a session. It is the interface between the application layer and the transfer layer and delivers all functions that are needed for wireless connections. A session mainly consists of 3 phases: start of the session, transfering information back and forth and the end of the session. Additionally, a session can be interrupted and started again (from the point where it was interrupted.)

Transaction Layer (Wireless Transaction Protocol WTP)

The specifications for the transfer layer are in the Wireless Transaction Protocol (WTP). Like the User Datagramm Protocol (UDP), the WTP runs at the head of the datagramm service. Both the UDP and the WTP are a part of the standard application from the TCP/IP to make the simplified protocol compatible to mobile terminals. WTP supports chaining together protocol data and the delayed response to reduce the number of transmissions. The protocol tries to optimize user interaction in order that information can be received when needed.

Wireless Transport Layer Security WTLS

The Wireless Transport Layer Security (WTLS) is a optional layer or stack which consists of description devices. A secure transmission is crucial for certain applications such as e-commerce or WAP-banking and is a standard in these days. Furthermore WTLS contains a check for data integrity, user authentification and gateway security.

Transport Layer (Wireless Datagram Protocol WDP)

The Wireless Datagram Protocol (WDP) represents the transfer or transmission layer and is also the interface of the network layer to all the above stacks/layers. With the help of WDP the transmission layer can be assimilated to the specifications of a network operator. This means that WAP is completely independent from any network operator. The transmission of SMS, USSD, CSD, CDPD, IS-136 packet data and GPRS is supported. The Wireless Control Message Protocol (WCMP) is an optional addition to WAP, which will inform users about occurred errors.

WTLS

Wapforum version 11/99

Wireless Transport Layer Security is a protocol based on the TLS protocol. It is used with the WAP transport protocols and has been optimised for use over narrow-band communication channels. The WTLs layer is above the transport protocol layer. The required security layer of the protocol determines whether it is used or not. It provides a secure transport service interface that preserves the transport service interface below; additionally it provides an interface for managing secure connections. WTLS aims to provide privacy, data integrity and authentication between two communication applications. Among its features are datagram support, optimised handshaking and dynamic key refreshing. It is optimised for low-bandwidth bearer networks with relatively long latency.

The WTLS Record Protocol is a layered protocol. The Record Protocol takes messages to be transmitted, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, and decompressed, then delivered to higher-level clients. Four record protocol clients are described in the WTLS standard; the change cipher spec protocol, the handshake protocol, the alert protocol and the application data protocol. If a WTLS implementation receives a record type it does not understand, it ignores it. Several records can be concatenated into one transport SDU. For example, several handshake messages can be transmitted in one transport SDU. This is particularly useful with packet-oriented transports such as GSM short messages.

Handshake protocols	Alert Protocol	Application Protocol	Change Cipher Spec Protocol
Record protocol			

The handshake protocol is made up of 3 sub-protocols. All messages are encapsulated in a plaintext structure.

WTP

The Wireless Transaction Protocol provides the services necessary for interactive browsing applications. During a browsing session the client requests information from a server and the server responds with the information. This is referred to as a transaction. WTP runs on a datagram service and possible a security service.

Advantages of WTP include:

- Improved reliability over datagram services
- Imported efficiency over connection oriented services
- As a message oriented protocol, it is designed for services oriented towards transactions.

Main features:

- 3 kinds of transaction services.
    - Class 0 Unreliable invoke messages with no result messages
    - Class 1: Reliable invoke messages with no result messages
    - Class 2: Reliable invoke messages with exactly one reliable result message.

- Reliability achieved by using unique transaction identifiers, acknowledgements, duplicate removal; and retransmissions.
- No explicit set up or tear down phases.
- Optional user-to-user reliability.
- Optionally the last acknowledgement of the transaction may contain out-of-band information.
- Concatenation may be used to convey multiple PDUs in one service data unit of the datagram transport.
- The basic unit of interchange is an entire message, not a stream of bytes.
- Mechanisms are provided to minimize the number of transactions replayed as a result of duplicate packets.
- Abort of outstanding transactions.
- For reliable invoke messages, both success and failure reported.
- Asynchronous transactions allowed.

The protocol data unit (PDU) consists of the header and data (if present). The header contains a fixed part and a variable part; The variable parts are carried in the Transport Information Item (TPI). Each PDU has its own fixed header (the fixed headers vary slightly in structure). As an example, the structure of the invoke PDU fixed header appears below:

<table>
<tr><td>1</td><td colspan="4">2-5</td><td colspan="2">6</td><td>7</td><td>8</td></tr>
<tr><td>Con</td><td colspan="4">PDU Type</td><td colspan="2">GTR</td><td>TTR</td><td>RID</td></tr>
<tr><td colspan="9">TID</td></tr>
<tr><td colspan="2">Version</td><td>TIDnew</td><td>U/P</td><td colspan="2">RES</td><td>RES</td><td colspan="2">TCL</td></tr>
</table>

ONcontinueflag (1bit):
The continue flag indicates the presence of any TPIs in the variable part. If the flag is set, there are one or more TPIs in the variable portion of the header. If the flag is clear, the variable part of the header is empty. This flag is also used as the first bit of a TPI, and indicates whether the TPI is the last of the variable header. If the flag is set, another TPI follows this TPI. If the flag is clear, the octet after this TPI is the first octet of the user data.

PDUtype
The PDU type determines the length and structure of the header and dictates what type of WTP PDU the PDU is (Invoke, Ack, etc). This provides information to the receiving WTP provider as to how the PDU data should be interpreted and what action is required.

The following PDU types are defined:

PDU Code	**PDU Type**
0x01	Invoke
0x02	Result
0x03	Ack
0x04	Abort
0x05	Segmented Invoke
0x06	Segmented Result
0x07	Negative Ack

Group trailer (GTR) and Transmission trailer (TTR) flag (2 bit):
When segmentation and re-assembly is implemented, the TTR flag is used to indicate the last packet of the segmented message. The GTR flag is used to indicate the last packet of a packet group.

GTR/TTR flag combinations:

GTR TTR	**Description**
1	Not last packet
2	Last packet of message
10	Last packet of packet group
11	Segmentation and Re-assembly NOT supported.

The default setting should be GTR=1 and TTR=1, that is, WTP segmentation and re-assembly not supported.

RIDRe-transmissionIndicator (1bit):
Enables the receiver to differentiate between packets duplicated by the network and packets re-transmitted by the sender. In the original message the RID is clear. When the message gets re-transmitted the RID is set.

TIDTransactionidentifier (16bit):
The TID is used to associate a packet with a particular transaction.

Version
The current version is 0X00

TIDnew flag
This bit is set when the Initiator has wrapped the TID value, i.e. set it to be lower than the previous TID value.

U/P
When this flag is set it indicates that the Initiator requires a User acknowledgement from the server WTP user. The WTP user confirms every received message.

RES
This is a reserved bit and its value should be set to 0.

TCL
The transaction class shows the desired transaction class in the invoke message.

WSP

WAP WSP 5/11/99

The Session layer protocol family in the WAP architecture is called the Wireless Session Protocol, WSP. WSP provides the upper-level application layer of WAP with a consistent interface for two session services. The first is a connection-mode service that operates above a transaction layer protocol WTP, and the second is a connectionless service that operates above a secure or non-secure datagram transport service.

The Wireless Session Protocols currently offer services most suited for browsing applications. WSP provides HTTP 1.1 functionality (it is a binary form of HTTP) and incorporates new features such as long-lived sessions, a common facility for data push, capability negotiation and session suspend/resume. The protocols in the WSP family are optimized for low-bandwidth bearer networks with relatively long latency. Requests and responses can include both headers and data. WSP provides push and pull data transfer WSP functions on the transaction and datagram services.

Messages can be in connection mode or connectionless. Connection mode messages are carried over WTP. In this case the protocol consists of WTP protocol messages with WSP PDUs as their data. Connectionless messages consist only of the WSP PDUs.

The general structure of the WSP PDU is as follows:

1 bite1 bite	

TID/PIDPDU Type	Type Specific Contents

TID/PID
Transaction ID or Push ID. The TID field is used to associate requests with replies in the connectionless session service. The presence of the TID is conditional. It is included in the connectionless WSP PDUs, and is not included in the connection-mode PDUs. In connectionless WSP, the TID is passed to and from the session user as the "Transaction Id" or "Push Id" parameters of the session primitive

PDU type
The Type field specifies the type and function of the PDU. The type numbers for the various PDUs are defined below. The rest of the PDU is type-specific information, referred to as the contents.