Mobile Communications II
Chapter 8: Mobile Transport Layer

Mobile Transport Layer

- Motivation
- TCP mechanisms
- Classical approaches
  - Indirect TCP
  - Snooping TCP
  - Mobile TCP
  - PEPs in general
- Additional optimizations
  - Fast retransmit/recovery
  - Transmission freezing
  - Selective retransmission
  - Transaction oriented TCP
- TCP for 2.5G/3G wireless
TCP/IP-Stack als vereinfachter OSI-Stack

Prozess A

Application

TCP

IP

Access

Physical

Real-System

Internetwork-System

Prozess B

Anwendung

TCP

IP

Access

Physical

Prozess A

Application

TCP

IP

Access

Physical

Real-System

Internetwork-System

Prozess B

Anwendung

TCP

IP

Access

Physical
Motivation I

- Transport protocols typically designed for
  - Fixed end-systems
  - Fixed, wired networks
- Research activities for TCP in wireless networks:
  - Performance (scalability)
  - Congestion control (stability)
  - Efficient retransmissions (reliability)
- TCP congestion control behaviour:
  - packet loss in fixed networks typically due to (temporary) overload situations
  - router have to discard packets as soon as the buffers are full (or almost full depending on QoS protection rules)
  - TCP recognizes congestion only indirectly via missing acknowledgements, retransmissions unwise, they would only contribute to the congestion and make it even worse
  - slow-start algorithm as reaction (this is rather fast)
Motivation II

• TCP slow-start algorithm
  – sender calculates a congestion threshold for a receiver
  – start with a congestion window size equal to one segment
  – exponential increase of the congestion window up to the congestion threshold, then linear increase
  – missing acknowledgement causes the reduction of the congestion threshold to one half of the current congestion window
  – congestion window starts again with one segment
Slow Start Behaviour

Figure 110. TCP slow start and congestion avoidance behavior in action
Fast Retransmit/Recovery

- TCP sends a cumulative acknowledgement only after receiving a packet
- if a sender receives several acknowledgements for the same packet, this is due to a gap in received packets at the receiver
- however, the receiver got all packets up to the gap and is actually receiving packets
- therefore, packet loss is not due to congestion, continue with current congestion window (do not use slow-start)
Influences of mobility on TCP-mechanisms

- TCP assumes congestion if packets are dropped
  - typically wrong in wireless networks, here we often have packet loss due to *transmission errors in the wireless segment*
  - furthermore, *mobility* itself can cause packet loss, if e.g. a mobile node roams from one access point (e.g. foreign agent in Mobile IP) to another while there are still packets in transit to the wrong access point and forwarding is not possible or delayed

- The performance of an unchanged TCP degrades severely
  - however, TCP “cannot” be changed fundamentally due to the large base of installation in the fixed network, TCP for mobility has to remain compatible
  - the basic TCP mechanisms keep the whole Internet together
Minimal change and maximal use scenarios

Modified TCP

TCP → IP → LLC → MAC → PHY

Modified Link Layer

TCP → IP → LLC → MAC → PHY

IP-Proxi Support

TCP → IP → LLC → MAC → PHY

Split Connection

TCP → IP → LLC → MAC → PHY

Modified Mobile Only

TCP → IP → LLC → MAC → PHY
Initial i-Mode Architecture

Mobile Network

<table>
<thead>
<tr>
<th>mobile terminal</th>
<th>mobile gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>cHTML + tags</td>
<td>TCP</td>
</tr>
<tr>
<td>HTTP(S)</td>
<td>IP</td>
</tr>
<tr>
<td>TL</td>
<td>L2</td>
</tr>
<tr>
<td>PDC-P</td>
<td>L1</td>
</tr>
</tbody>
</table>

Specific transport protocol for wireless segment

Fixed Network

<table>
<thead>
<tr>
<th>gateway</th>
<th>content provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>HTTP(S)</td>
</tr>
<tr>
<td>IP</td>
<td>L2</td>
</tr>
<tr>
<td>L2</td>
<td>L1</td>
</tr>
</tbody>
</table>

Split connection between fixed and mobile network
i-mode protocol stack based on WAP 2.0

i-mode can use WAP 2.0/Internet protocols (example: i-mode in Germany over GSM/GPRS)

user equipment

<table>
<thead>
<tr>
<th>cHTML</th>
<th>HTTP</th>
<th>SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WTCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L1</td>
<td></td>
</tr>
</tbody>
</table>

Split connection gateway

<table>
<thead>
<tr>
<th>WTP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>IP</td>
</tr>
<tr>
<td>L2</td>
<td>L2</td>
</tr>
<tr>
<td>L1</td>
<td>L1</td>
</tr>
</tbody>
</table>

server

<table>
<thead>
<tr>
<th>cHTML</th>
<th>HTTP</th>
<th>SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WTCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L1</td>
<td></td>
</tr>
</tbody>
</table>

Wireless link

Wired link
Uplink: “Never Give Up” Strategy

- Insert new layer above MAC
  - Adjust data rate, fragmentation, probing dynamically

- TCP is a transport protocol: ensures reliable transport
  - Every lost packet is retransmitted by TCP eventually
  - Dropping packets in MAC is always a bad choice

- Never give up: continue with same packet indefinitely
  - If not possible, report lost connection to APP
  - TCP sees delays and jitter but not loss

- Vertical aspect: look into packets to detect TCP data
Improved Performance In Uplink

Chapter 8 – Page 356
Problems with „Never Give Up“

- Additional Layer required to distinguish between different forms of traffic (e.g. UDP, TCP)
- Adaptation layer must be involved in security related issues
- Timeout values are critical (retransmission of TCP layer and MAC layer can interfere)
- Vertical communication between protocol layers required (e.g. via management plane)
TCP_2 Processor (WI, MBE)

Block Diagram of TCP_2 Implementation
- power control via sleep mode possible

Chip Photo of TCP_2
- area: 54 mm²
- transistors: ~4,300,000
- memory: 62 kByte SRAM
- speed: 33 MHz (CardBus clock)
- package used: LQFP-256 / 244 pins
Integration of Crypto- and Communication Support

- Single Copy Architektur
  TCP/IP Hardware (Checksum) and AES (De-/Encryption) integrated into the Datapath
Indirect TCP I

- Indirect TCP (I-TCP) segments the connection
  - no changes to the TCP protocol for hosts connected to the wired Internet required
  - optimized TCP protocol for mobile hosts
  - splitting of the TCP connection at, e.g., the foreign agent into 2 TCP connections,
    - no real end-to-end connection any longer (but still invisible for user)
  - hosts in the fixed part of the net do not need to notice the characteristics of the wireless part
I-TCP socket and state migration

- Propagation of TCP packet might cause additional delay
Indirect TCP II

• Advantages
  – no changes in the fixed network necessary, no changes for the hosts (TCP protocol) necessary, all current optimizations to TCP still work
  – transmission errors on the wireless link do not propagate into the fixed network
  – simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host
  – therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop is known

• Disadvantages
  – loss of end-to-end semantics, an acknowledgement to a sender does now no longer mean that a receiver really got 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
    • might lead to TCP-timeout at host side
Snooping TCP I

• „Transparent“ extension of TCP within the foreign agent
  – buffering of packets sent to the mobile host within the foreign agent
  – lost packets on the wireless link (both directions!) will be retransmitted immediately by the mobile host or foreign agent, respectively (so called “local” retransmission)
  – the foreign agent therefore “snoops” the packet flow and recognizes acknowledgements in both directions, it also filters “duplicated” ACKs
  – ACKs received at correspondent host always come from mobile host
  – changes of TCP only within the foreign agent

![Diagram of mobile host, foreign agent, and correspondent host connected through a "wired" Internet, showing local retransmission, snooping of ACKs, buffering of data, and end-to-end TCP connection.]
Snooping TCP II

• Data transfer to the mobile host
  – FA buffers data until it receives ACK of the MH, FA detects packet loss via duplicated ACKs or time-out
  – fast retransmission possible, transparent for the fixed network
• Data transfer from the mobile host
  – FA detects packet loss on the wireless link via sequence numbers, FA answers directly with a NACK to the MH
  – MH can now retransmit data with only a very short delay
• Integration of the MAC layer
  – MAC layer often has similar mechanisms to those of TCP
  – thus, the MAC layer can already detect duplicated packets due to retransmissions and discard them
• Problems
  – snooping TCP does not isolate the wireless link as good as I-TCP
  – snooping might be useless depending on encryption schemes
Mobile TCP

- Special handling of lengthy and/or frequent disconnections
- M-TCP splits connection as I-TCP does but preserves E2E semantics
  - unmodified TCP fixed network to supervisory host (SH)
  - optimized TCP SH to MH
- Supervisory host
  - no caching, no retransmission
  - monitors all packets, if disconnection detected
    - set sender window size to 0
    - sender automatically goes into persistent mode
  - old or new SH reopen the window
- Advantages
  - maintains E2E semantics, supports disconnection, no buffer forwarding
- Disadvantages
  - loss on wireless link is propagated into fixed network
  - Higher packet error loss of wireless link is directly visible to fixed network communication node
  - adapted TCP on wireless link
Mobile TCP II

- Disadvantage:
  - Assumption of relative low error rate might be wrong
  - Different quality of transmission in wired and wireless network impacts directly the CN behavior and performance
Forced fast retransmit/fast recovery

- Change of foreign agent often results in packet loss
  - TCP reacts with slow-start although there is no congestion
- Forced fast retransmit
  - as soon as the mobile host has registered with a new foreign agent, the MH sends duplicated acknowledgements on purpose
  - this forces the fast retransmit mode at the communication partners
  - additionally, the TCP on the MH is forced to continue sending with the actual window size and not to go into slow-start after registration
- Advantage
  - simple changes result in significant higher performance
- Disadvantage
  - further mix of IP and TCP, no transparent approach
Transmission/time-out freezing

- Mobile hosts can be disconnected for a longer time
  - no packet exchange possible, e.g., in a tunnel, disconnection due to overloaded cells or multiplexing with higher priority traffic
  - TCP disconnects after time-out completely (several minutes)
- TCP freezing
  - MAC layer is often able to detect interruption in advance
  - MAC can inform TCP layer of upcoming loss of connection
  - TCP stops sending, but does not assume a congested link
  - MAC layer signals again if reconnected
- Advantage
  - scheme is independent of TCP data and can be used also in encrypted environments
- Disadvantage
  - TCP on mobile host has to be changed, mechanism depends on MAC layer
Selective retransmission

- TCP acknowledgements are often cumulative
  - ACK n acknowledges correct and in-sequence receipt of packets up to n
  - if single packets are missing quite often a whole packet sequence beginning at the gap has to be retransmitted (go-back-n), thus wasting bandwidth

- Selective retransmission as one solution
  - RFC2018 allows for acknowledgements of single packets, not only acknowledgements of in-sequence packet streams without gaps
  - sender can now retransmit only the missing packets

- Advantage
  - much higher efficiency

- Disadvantage
  - more complex software in a receiver, more buffer needed at the receiver
E.g. HTTP (used by web services) typically uses TCP
- Reliable transport between client and server required

TCP
- Stream oriented, not transaction oriented
- Network friendly: time-out → congestion → slow down transmission

Well known – TCP guesses quite often wrong in wireless and mobile networks
- Packet loss due to transmission errors
- Packet loss due to change of network

Result
- Severe performance degradation
- TCP not particularly suited for asymmetric short message transport

**Transaction Oriented Layer**

![Diagram showing the process of establishing a TCP connection and data transmission over GPRS with a delay of 500ms and an idle period of more than 15 seconds.]
Transaction Oriented TCP

- TCP phases
  - connection setup, data transmission, connection release
  - using 3-way-handshake needs 3 packets for setup and release, respectively
  - thus, even short messages need a minimum of 7 packets!
- Transaction oriented TCP
  - RFC1644, T-TCP, describes a TCP version to avoid this overhead
  - connection setup, data transfer and connection release can be combined in a single message
  - thus, only 2 or 3 packets are needed
- Advantage
  - efficiency
- Disadvantage
  - requires changed TCP
  - mobility not longer transparent
Comparison of different approaches for a “mobile” TCP

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

- Actual research work
  - Indirect TCP, Snoop TCP, M-TCP, T/TCP, SACK, Transmission/time-out freezing, ...
- TCP over 2.5/3G wireless networks
  - Fine tuning today’s TCP
  - Learn to live with
    - Data rates: 64 kbit/s up, 115-384 kbit/s down; asymmetry: 3-6, but also up to 1000 (broadcast systems), periodic allocation/release of channels
    - High latency, high jitter, packet loss
  - Suggestions
    - Large (initial) sending windows, large maximum transfer unit, selective acknowledgement, explicit congestion notification, time stamp, no header compression
• **Performance enhancing proxies (PEP, RFC 3135)**
  – Transport layer
    • Local retransmissions and acknowledgements
  – Additionally on the application layer
    • Content filtering, compression, picture downscaling
    • E.g., Internet/WAP gateways
    • Web service gateways?
  – Big problem: breaks end-to-end semantics
    • Disables use of IP security
    • Choose between PEP and security!

• **More open issues**
  – RFC 3150 (slow links)
    • Recommends header compression, no timestamp
  – RFC 3155 (links with errors)
    • States that explicit congestion notification cannot be used
  – In contrast to 2.5G/3G recommendations!
Conclusion

• TCP/IP is a near optimal solution for transport layer protocols for wired network
• For wireless or hybrid networks problems due to high error rates become a challenge
• Several solutions for improving have been presented which all do not solve the problem in an ideal way
• A change of TCP with a richer variety of error modes have to be provided that can deal with different errors in a flexible way
• New projects in the international research community are working on new Internet Solutions that also should solve the problem of the transport layers
Example IP-based 4G/Next G/… network

IP-based core

PSTN, CS core

SS7 signalling

GSM

UMTS

BSC

MSC

RNC

SGSN

Internet

GGSN

server farm, gateways, proxies

firewall, GGSN, gateway

broadcast

access points

private WPAN

private WLAN

public WLAN

Chapter 8 – Page 376