Advanced Computer Networks
263-3501-00

Wireless TCP

Patrick Stuedi
Spring Semester 2014
Outline

- Last week:
  - Cellular Networks
  - Mobile IP

- Today:
  - Wireless TCP
Remember: TCP congestion control

- Congestion control got added to TCP to in attempt to reduce congestion inside the network
- Must rely on indirect measures of congestion
- Implemented at the sender

Attempts to reduce buffer overflow inside the network
Remember: TCP Slow Start

- Congestion window (CW)
  - Number of bytes in TCP that can be transmitted without waiting for the ACK (CW always smaller than receiver window, flow ctrl)
  - Initially set to 1 TCP segment

- SSThresh
  - Initially set to 64 KB

- TCP congestion control:
  - After all ACKs corresponding to one CQ have been received (typically after one RTT), the window is doubled
    - slow start (actually quite fast)
  - If CW >= SSThresh increase CW by 1 after all ACKs corresponding to one CQ have been received
    - linear increase (congestion avoidance)
  - On a timeout: Set SSThresh to half of the current CW, then set CW back to 1K
Example: Slow start
Problems of TCP in Wireless Networks

- Congestion control algorithm has been designed for wired/fixed networks
  - In fixed networks a packet loss is an indication of congestion
  - In wireless networks packet lost due to transmission errors or mobility
  - TCP cannot distinguish between errors and congestion
    - TCP unnecessarily reduces window, resulting in low throughput and high latency

- Delay is often high
  - RTT can be very long and variable
  - TCP's timeout mechanism may not work well

- Links may be asymmetric
  - Delayed ACKs in the slow direction can limit throughput in fast direction
TCP in Wireless: Solutions

- Cannot change TCP fundamentally
  - TCP congestion control keep the Internet operable
  - Improvements have to be interoperable

- Possible Solutions:
  1) End-to-End
     - Fast retransmit, Selective Acknowledgments
  2) Split Connection
     - Separate wired path and wireless hop
  3) Link Layer
     - Error-correcting codes
     - Local retransmissions
     - Snooping
End-to-End: Fast retransmit

- Note: TCP sends an acknowledgement only after receiving a packet
- If a sender receives several acknowledgements for the same packet, this means
  - The receiver got all packets up to the acknowledged packet in sequence
  - The receiver is still receiving packets
  - The gap is most likely not due to congestion, try to avoid triggering slow start
- Fast retransmit:
  - If sender receives three duplicate ACKs for the same SeqNr he retransmits the missing packet (before the timeout occurs)
  - Reduce CW only to the half, and continue with linear increase
Fast Retransmit: Example

Single packet loss (within a RTT) can be handled with fast retransmit.
Fast Retransmit: Pros/Cons

- Advantages:
  - Simple, minor modifications in the mobile host's TCP stack (TCP Tahoe)
  - Correspondent node's stack does not have to be changed

- Disadvantages:
  - Inefficient: lost packets still have to cross the entire network between correspondent node and mobile host
Split-Connection: Indirect TCP

- Two TCP connections:
  - Fixed to Base: unmodified TCP connection
  - Base to mobile: optimized TCP connection
- Buffering at BS
- Independent flow and congestion control on the two connections
Indirect TCP: Pros/Cons

- Advantages:
  - Transmission errors on the wireless link do not propagate into the fixed network
  - The short delay on the mobile hop is known and therefore it is possible to use precise timeouts and fast retransmissions
  - It is possible to use a different transport layer protocol on the mobile hop

- Disadvantages:
  - Serious: Loss of end-to-end semantics
    ACK to the sender does not any longer mean that the receiver got a packet, FAs may crash
  - Problems during handover
Socket and State Migration during I-TCP handover

- Old proxy must forward buffered data to new proxy because it has already acknowledged the data with the CN
  - Migrate TCP buffer to new proxy
  - Migrate socket state (seqnbr, addresses, ports, etc)
Snooping TCP

- Snoop agent at the BS
  - Monitors TCP segments and ACKs
  - Caches segments until acknowledged
  - Detects packet loss and retransmits lost packets if cached

- Data transfer to the mobile host
  - Packet loss detected by snooping duplicated ACKs
  - Fast retransmission possible

- Data transfer from the mobile host
  - Packet loss detected by looking at sequence numbers
  - Snooping agent answers with NACK to the MH
Snooping TCP: pros/cons

**Advantages**

- End-to-end TCP semantics is preserved
- Changes of TCP only within the FA
  - the CN does need to change
- Handover can be more easily supported than with I-TCP
- Interoperable with FAs that do not support the enhancement

**Disadvantages**

- Won't work if TCP connection is encrypted
- Does not isolate the behaviour of the wireless link like I-TCP
  - As long as packets are not acknowledged end-to-end the corresponding node has a timer ready waiting to retransmit/slow start
Other improvements: Selective retransmission

- TCP acknowledgements are cumulative
  - ACK $N$ = correct and in-sequence receipt of packets up to $N$
  - If single packets are missing a whole packet sequence has to be retransmitted (go-back-$n$), thus wasting bandwidth

- Selective retransmission
  - RFC 2018 allows for ACKs of single packets (SACKs)
  - Sender can now retransmit only the missing packets

- Advantage
  - Much higher efficiency

- Disadvantage
  - More complex software and more buffer needed at the receiver
    - Trade-off memory, complexity vs performance
TCP over 3G: Best Practice

- Selective ACKs (TCP SACK)
- Increase the TCP’s initial window (from 1 to 2-4 segments)
- Larger TCP receive window (typically only 64KB)
  - Windows scale option (window gets shifted, allows for almost 1G size)
- Explicit Notification Schemes
  - Router marks ACK packets with ECN (congestion) or ELN (Loss) bit to allow end host to distinguish between congestion and loss
  - With ELN, end host will then not back off, but simply retransmit
  - Requires support from ECN capable routers
- Fast Retransmit
Longer RTT leads to higher Energy Consumption

- 3G consumes significantly more energy to upload a fixed size byte buffer than WiFi
- Energy cost mainly caused by longer RTT
  - Longer RTT → longer data transfer time → more energy used
TCP and 802.11 Power Save Mode (PSM)

- Experiment: transfer 500 KB over TCP using Wifi PSM
- First part of the transfer
  - TCP window is small: allow sender to sleep after finishing window
- Second part of transfer:
  - First Ack is received while still sending: sender won't sleep
- PSM may prolong total transfer time and increase energy consumption
References