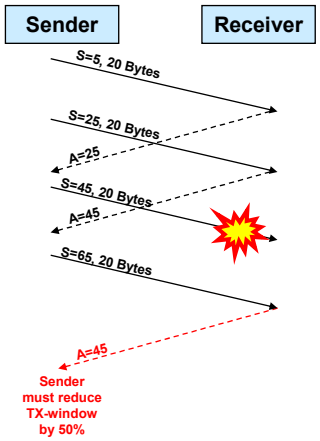
TCP in Wireless Environments

Problems and Solutions

TCP Design Basics

- TCP's algorithms have been designed for wired communications
  - Slow start, CA, window-based flow control (Fast retransmit, fast recovery)
- Assume BER through queue drops!!!
  - Duplicate ACKs are sent, sender throttles down
- Wireless: High BER through interferences!!!
  - Duplicate ACKs are counter-productive
  - Only little throughput improvement when MTU increased beyond 500 Bytes



TCP Data/ACK Collisions

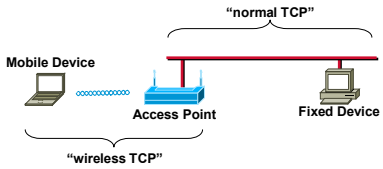
- Rule of thumb
  - The higher the total frame rate in a WLAN the higher the collision rate
- TCP causes ACKs
  - Continuous "co-propagating" frames increase collision probability
  - Doubles the total frame rate
- Average MAC retransmission rates
  - TCP: 4-7% (UDP: 0.4-1%)
  - Disabling MAC-layer retransmission degrades throughput by 23%

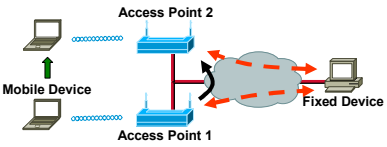
MAC Rate Adaptation Problem

- BW varies dynamically at long distances
  - Variability in packet transmission duration arrival rates
- BW probing mechanisms based on
  - Feedback of packet losses and packet loss ratio and monitoring packet RTTs
- Prefer TCP-friendly algorithms
  - Increase rate until packet loss ratio *p* exceeds threshold *t*
  - "Additive-increase-multiplicative-decrease" (AIMD): if (p<t) then (rate+=Δ) else (rate*=α), with α <1
  - Or TCP rate equation:
    rate = (1.22 × MTU) / (RTT × sqrt(p))
- But most WLAN products use "multiplicative-increase-multiplicative-decrease (MIMD)" bandwidth probing algorithms
  - Indeed faster—on Layer 2 only
  - But not TCP friendly:
    - Temporary deadlocks (especially if receiver uses delayed Ack)
    - Network thrashing and battery power waste!

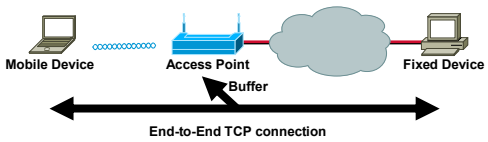
Indirect TCP

- AP is "TCP proxy"
  - Buffers all data until mobile device has sent all ACKs
- During an AP handover, old AP must send
  - All buffered data to new AP
  - As well as socket and TCP/IP state information (SeqNr, …)
- No changes in the TCP code of end-devices needed
  - Except deactivation of slow start would be helpful
- Disadvantages
  - Slow AP handover
  - Large buffers needed on AP
  - Security: AP must be trusted device





Snooping TCP



- Enhanced TCP code
- AP buffers data until ACKs from wired node seen
  - To provide immediate retransmissions
- Therefore AP listens (snoops) TCP stream to detect missing ACKs
  - Can also filter ACKs and segments
- After detection of a void within the sequence numbers
  - AP sends out NACK to mobile device
- Doesn't work if end-to-end encryption/integrity active

Mobile TCP

- Similar to Indirect TCP, Mobile TCP also splits TCP connection in two components
  - Normal TCP (on wired interface)
  - Optimized TCP (on radio interface; no slow start)
- AP is called "supervisory host"
  - Does NOT buffer any data
  - Observes TCP stream
- If ACKs are missing for some longer period (interferences, client disconnect)
  - AP sets *sending window* to zero
  - Sender will NOT retransmit packets
  - If client again visible: AP opens the sending window
- Advantage: No buffers needed
- Problem: BW must be managed somehow at radio interface since slow start not available
  - "BW administrator" algorithm

Other Tricks

- Fast Recovery
  - After a mobile device connects to a new AP: *Three fake duplicated ACKs* are sent to the peer
  - Peer uses Fast Recovery and *continuous* sending with same speed
  - Only small code patch needed
- Selective Retransmissions
  - Use SACK
- Transaction-oriented TCP
  - Designed for short connections
  - No 3-way handshake: Connection establishment, data transaction, and connection clearance requires 2 segments only