TCP over Wireless

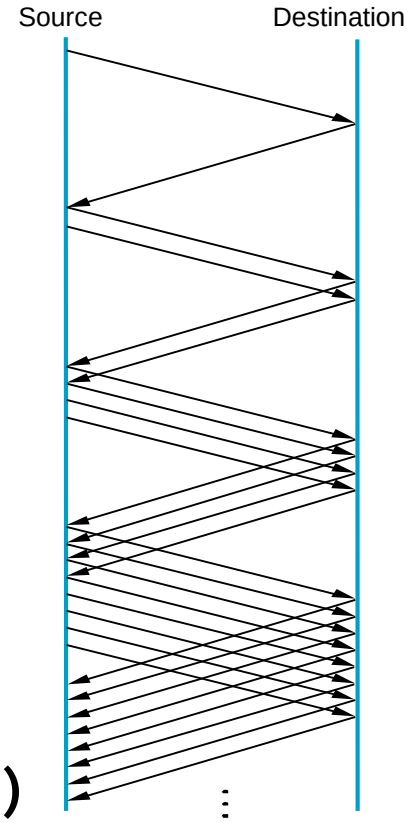
PROF. MICHAEL TSAI

2016/6/3

TCP Congestion Control (TCP Tahoe)

- **Only ACK correctly received packets**
- **Congestion Window Size:**
  Maximum number of bytes that can be sent without receiving acknowledgements.
- **Larger window size == faster rate**
- **Three major mechanisms:**
  - Slow start
  - Congestion avoidance
  - Fast retransmit

Slow start + congestion avoidance



- **Congestion window size = 1 initially.**
- **Ssthresh: slow start threshold**
- **Slow start:**
  For each received ACK (with a new seq. no.), the window size is exponentially increased (e.g., doubled).
  (when the window size is smaller than ssthresh)
- **Congestion avoidance:**
  After reaching ssthresh, the window size is linearly increased.
- **Additive Increase**

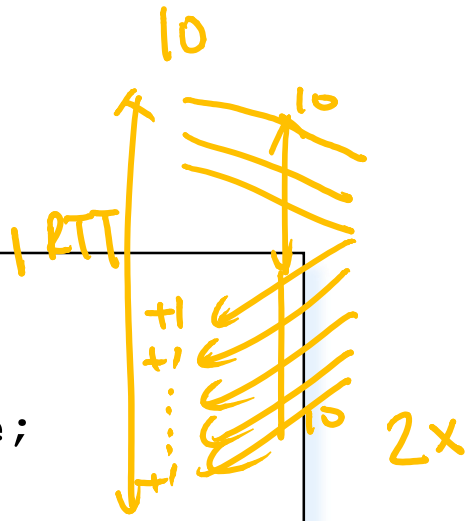
Slow start + congestion avoidance

$RTT = \alpha \cdot RTT + (1-\alpha) \cdot RTT\_new$

$0 < \alpha < 1$

- **TCP calculates expected round trip time (RTT) and its deviation**
- **Timeout: wait time > expected RTT + 4 * deviation**
- **Timeout: it signals a packet loss due to congestion (?)**
- **When a timeout happens, TCP Tahoe does the following:**
  - Dropping *ssthresh* into half the current window or 2 (multiplicative decrease)
  - Resetting its window size to 1 + enter slow start
  - Double round trip time timer

Pseudocode



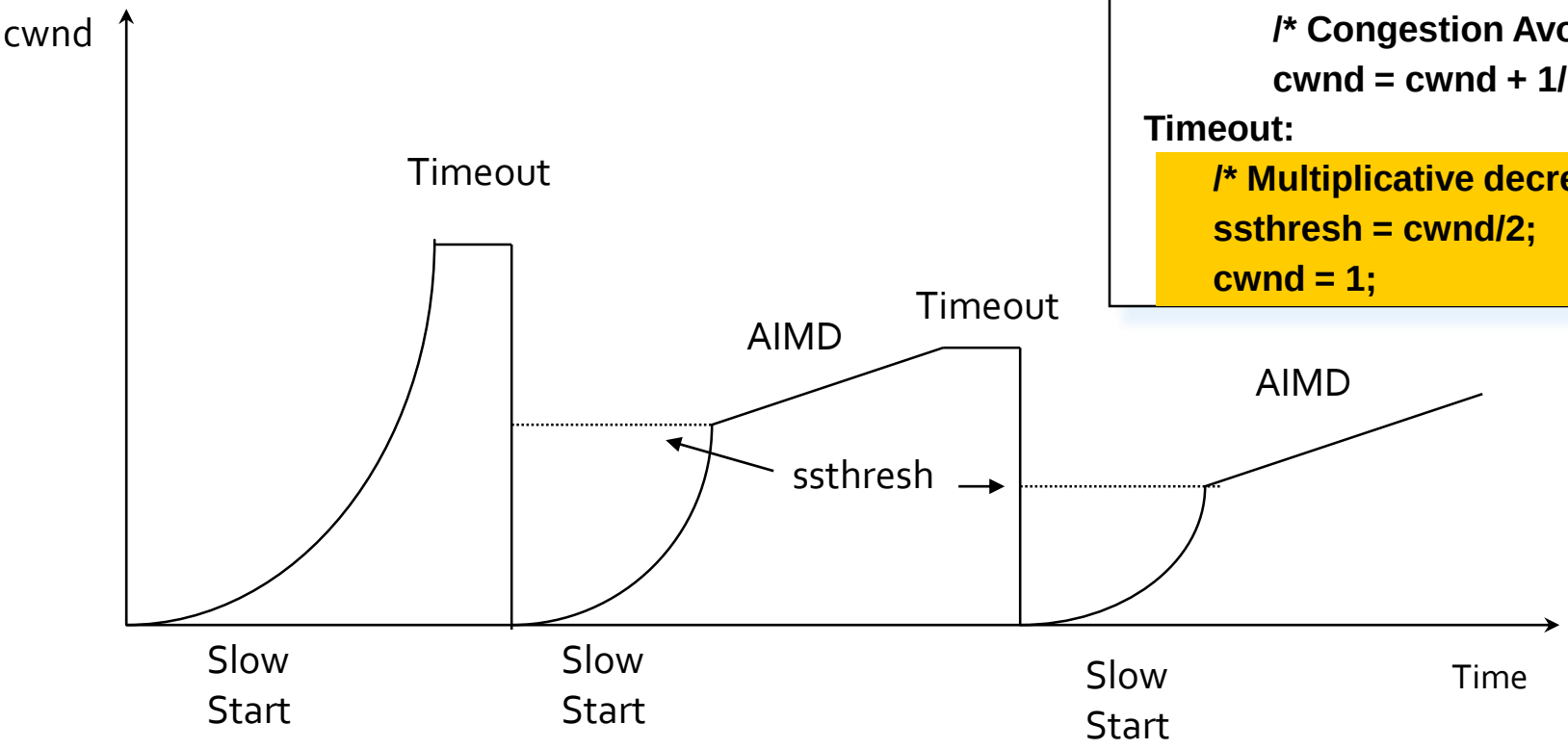

```
Initially:
   cwnd = 1;
   ssthresh = infinite;
New ack received:
   if (cwnd < ssthresh)
         /* Slow Start*/
         cwnd = cwnd + 1;
   else
         /* Congestion
   Avoidance */
        cwnd = cwnd + 1/cwnd;
Timeout:
   /* Multiplicative decrease
   */
   ssthresh = cwnd/2;
   cwnd = 1;
```

exponential

linear

The big picture (with timeouts)

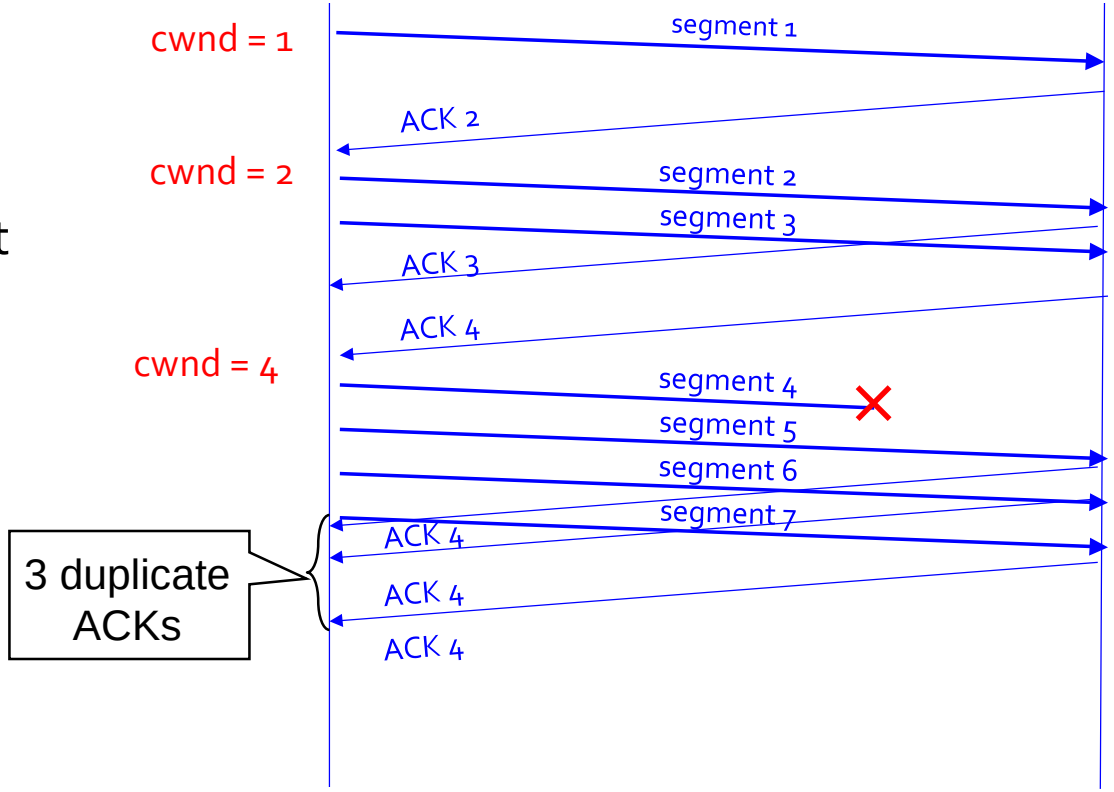
```
Initially:
    cwnd = 1;
    ssthresh = infinite;
New ack received:
    if (cwnd < ssthresh)
        /* Slow Start*/
        cwnd = cwnd + 1;
    else
        /* Congestion Avoidance */
        cwnd = cwnd + 1/cwnd;
Timeout:
    /* Multiplicative decrease */
    ssthresh = cwnd/2;
    cwnd = 1;
```



Fast Retransmit

- **Resend a segment after 3 duplicate ACKs**
  - Duplicate ACK means that an out-of sequence segment was received

- **Notes:**
  - ACKs are for next expected packet
  - Packet reordering can cause duplicate ACKs
  - Window may be too small to get enough duplicate ACKs



Big problems for TCP

- **High bit error rates:**
  up to 10^-5 bit error rate.
  Lots of lost packets due to bit errors.

10^{-10} 10^{-12} BER

- **Disconnections:**
  - Handoff: mobile devices move between base stations / access points.
  - Mobility: mobile devices move in and out of range of the transmitter.
  - Fading: signals blocked by buildings or other obstacles.
- **Round trip time could significantly vary.**
  (Indication of packet loss could be inaccurate)
- **Big problem: packet losses are not necessarily due to congestion! (the assumption of the original TCP)**

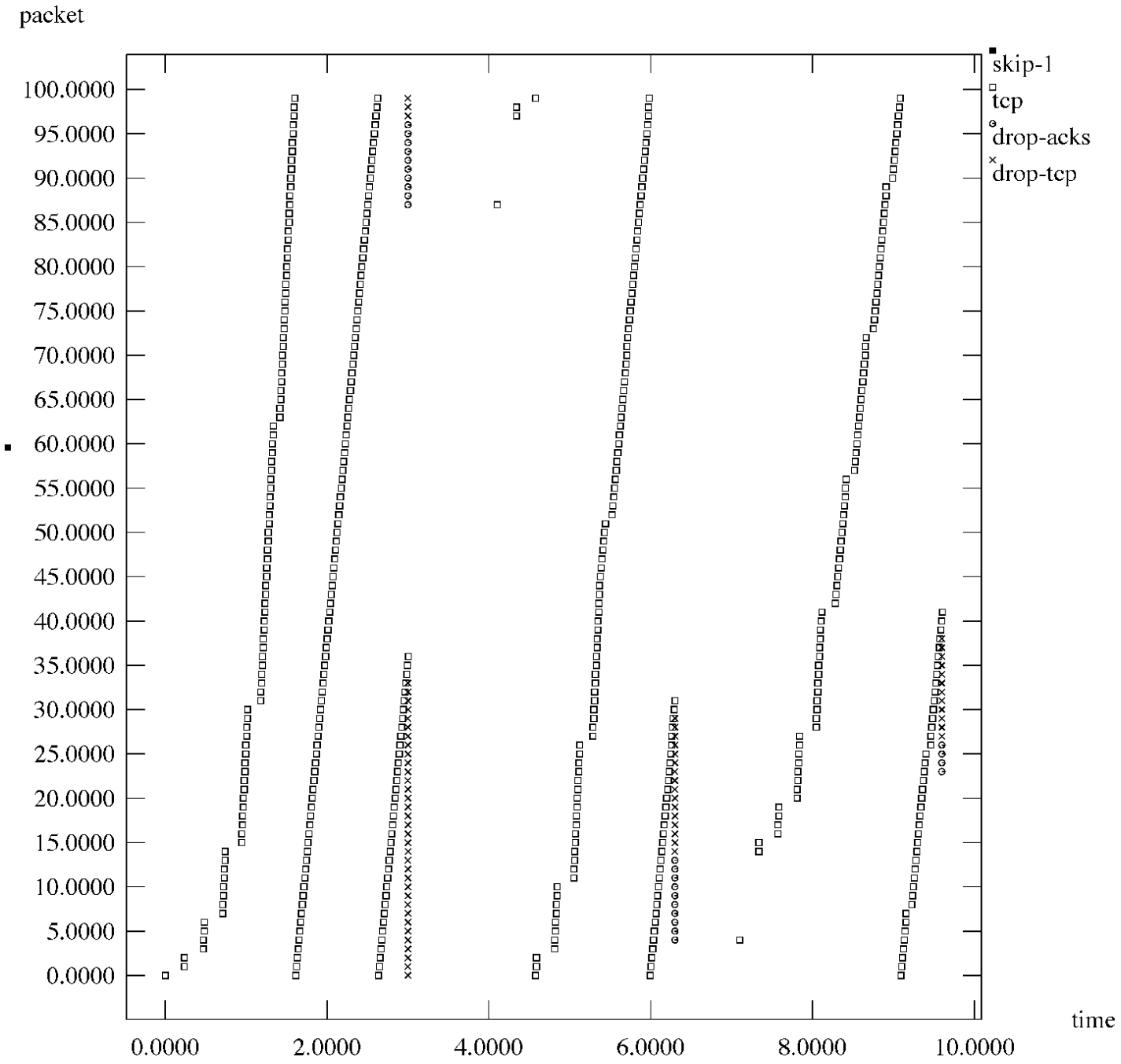
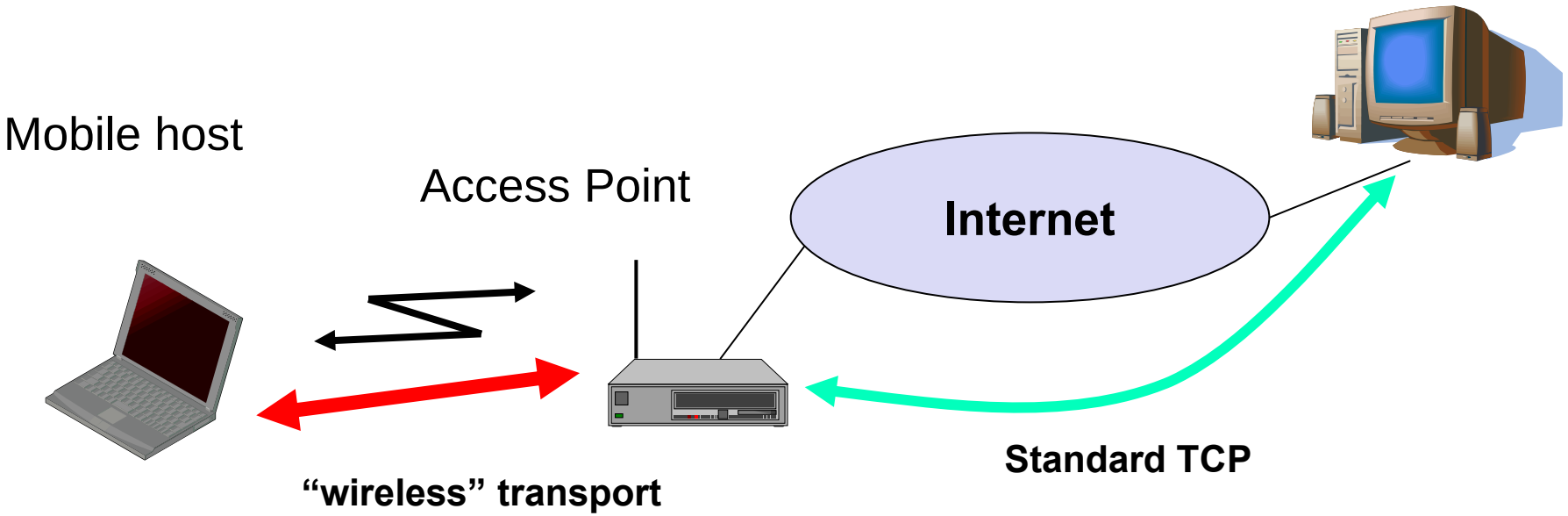


Fig. 2. Effect of mobile disconnection on TCP.

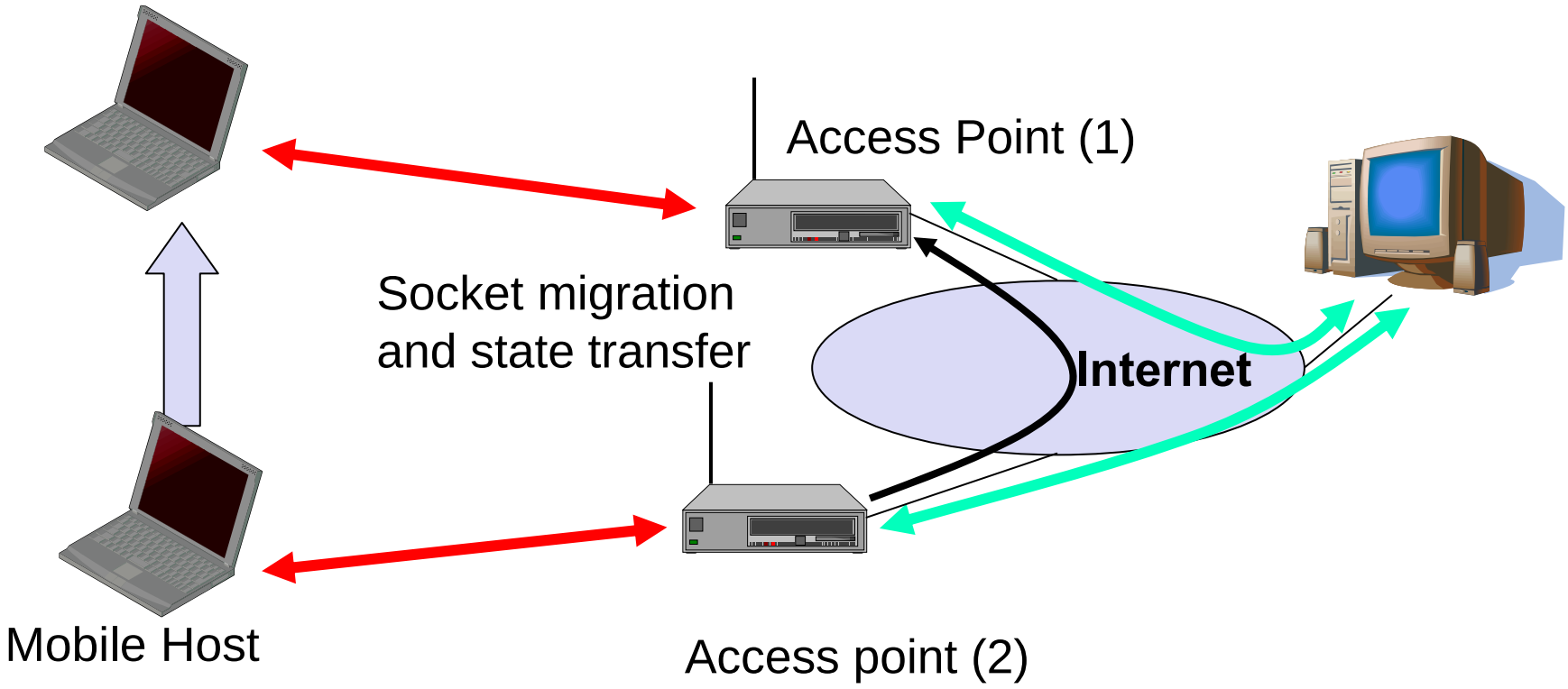
Classification of Solutions

- **Link layer**
  - Goal: improve the quality of the link layer, hide the losses from TCP
  - "Local problems should be solved locally"
- **End-to-end**
  - Additional considerations are added to TCP to improve its performance
  - Limited performance improvement
  - No modification on the hosts is required
- **Split-connection**
  - Wireless → usually the last link to the mobile host
  - Isolate that wireless link with the rest of the path (wired)
  - Two TCP connections, bridged at the wireless gateway
    - Use regular TCP for wired links
    - Use a special protocol designed specifically for that wireless link

Indirect TCP (I-TCP)



Indirect TCP (I-TCP)



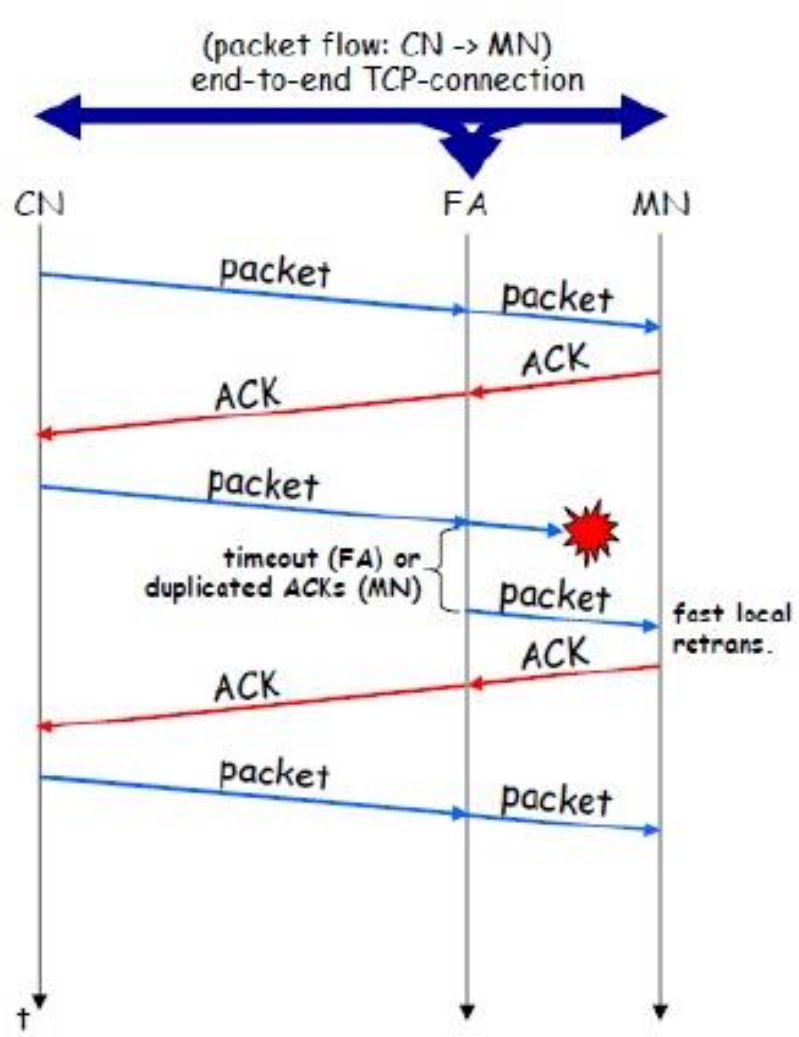
I-TCP

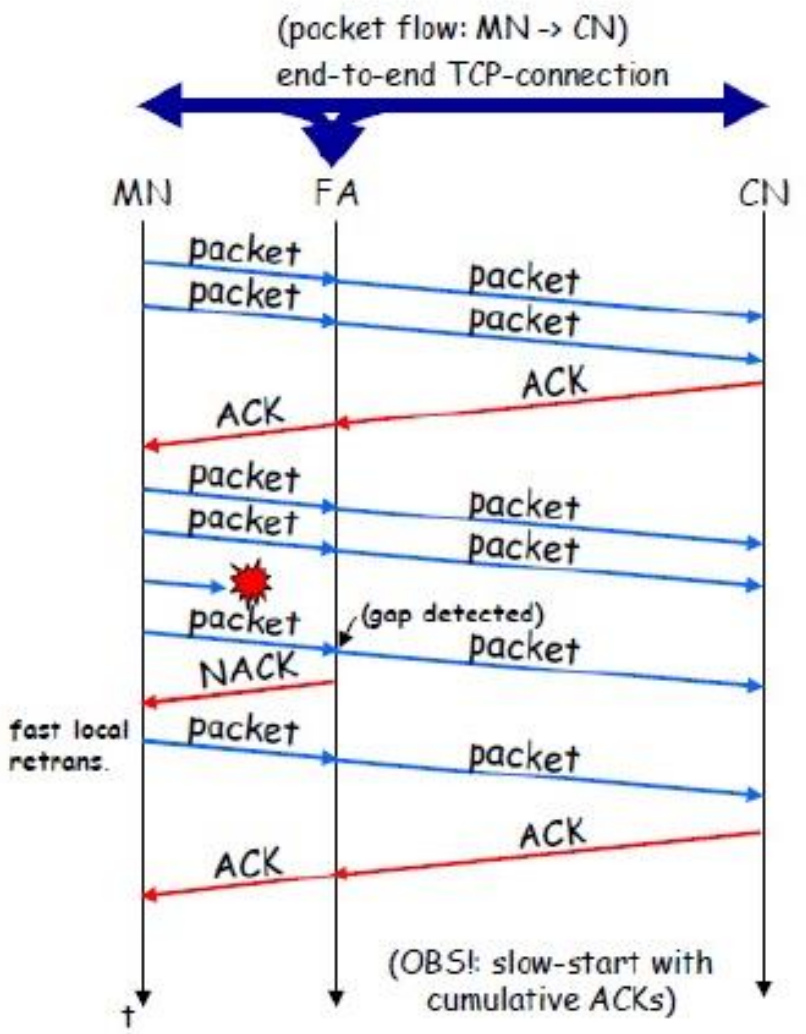
- **Advantages**
  - No changes in the fixed network / hosts (TCP)
  - Wireless transmission errors do not propagate to the wire-line network
  - Simple and effective
- **Disadvantages**
  - End-to-end semantics become less clear
  - Higher end-to-end delay due to buffering and forwarding at the gateway

Snooping TCP

- **Transparent extension of TCP within the gateway**
- **Hides wireless losses from wired host**
  - Buffer packets sent to the mobile host
  - Local retransmission:
    lost packets on the wireless link, for both directions, are retransmitted immediately by the mobile host or foreign agent
- **Wireless gateway snoops the packet flow so that it can cover up signs of packet loss**
  - Recognizes ACK in both directions and suppresses duplicate ACKs

Snooping TCP





Snooping TCP

- **Data transfer to the mobile host**
  - FA buffers data until it receives ACK from the MH
  - FA detects packet loss via duplicated ACKs or timeout
- **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**
  - Similar mechanisms often exist in MAC

Snooping TCP