# Transport Control Protocol/ Internet Protocol (TCP/IP) over Wireless Networks

Chapter 4



## Outline: TCP/IP over Wireless Networks

- Internet Protocol (IP)
- Basic Functionality of TCP
- Challenges
  - Channel Errors
  - Mobility
  - Asymmetry
- Split-Connection Solution – Indirect TCP
- Proxy-Based Solution - SNOOP

# Internet Protocol (IP)

- Provides connection-less, best-effort service for delivery of packets through the inter-network
- Two networks are connected through a **router**
- Best-effort: No error checking or tracking done for the sequence of packets (datagrams) being transmitted
- Upper layer should take care of sequencing
- Datagrams transmitted independently and may take different routes to reach same destination
- Fragmentation and reassembly supported to handle data links with different maximum – transmission unit (MTU) sizes



# Internet Control Message Protocol (ICMP)

- Companion protocol to IP
- Provides mechanisms for error reporting and query to a host or a router
- Query message used to probe the status of a host or a router
- Error reporting messages used by the host and the routers to report errors

# Basic Functionality of TCP

- TCP is a connection-oriented transport protocol.
- TCP aims to guarantee end-to-end reliable ordered delivery of data packets over wired networks.
- Basic functions include flow control, error control and congestion control.
- A sliding window protocol is used to implement flow control (prevent sender from sending too fast).

- Three windows are used
  - Congestion window
  - Advertised window
  - Transmission window

- Congestion window
  - Indicates the maximum number of segments (packets) a sender can transmit without congesting the network.
  - This number is determined by the sender based on feedback from the network.



# Basic Functionality of TCP

- Advertised window
  - Specified by the receiver in the acknowledgements.
  - Advertised window indicates to the sender the amount of data the receiver is ready to receive in the future.
  - Normally, it is equal to the available buffer size at the receiver in order to prevent buffer overflow.

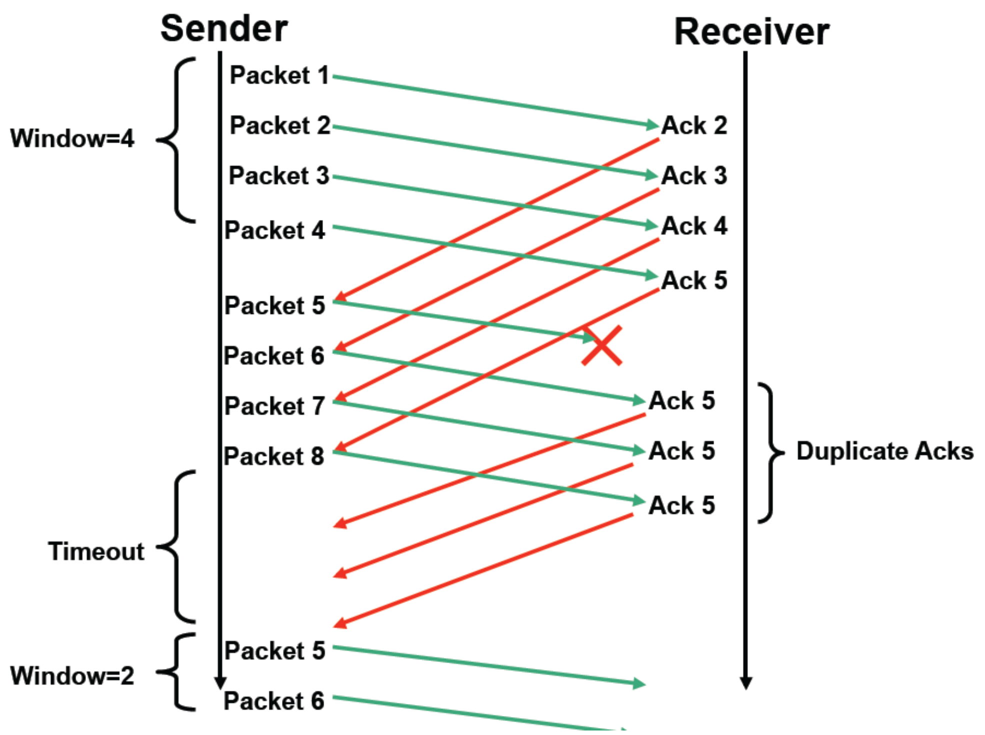
- Transmission window
  - Determines the maximum number of segments that the sender can transmit at one time without receiving any acknowledgements from the receiver.
  - To avoid network congestion and receiver buffer overflow, the size of the transmission window is determined as the minimum of the congestion window and the receiver's advertised window.

## Basic Functionality of TCP

- To notify the sender that data is correctly received, TCP employs a **cumulative acknowledgement** mechanism in which the receiver acknowledges the sequence number it expects next.
- Upon the receipt of an acknowledgement, the sender knows that all previously transmitted data segments with a sequence number less than the one indicated in the acknowledgement are correctly received at the receiver.
- In the case that an out-of-order segment (identified on the basis of sequence numbers) arrives at the receiver, a **duplicate acknowledgement** is generated and sent back to the sender.
- It is important to note that in wireline networks, an out-of-order delivery usually implies a packet loss.
- If three duplicate cumulative acknowledgements are received, the sender will assume that the packet is lost.
- A packet loss is also assumed if the sender does not receive an acknowledgement for the packet within a timeout interval called **retransmission timeout** (RTO), which is dynamically computed as the estimated **round-trip time** (RTT) plus four times the mean deviation.
- By retransmitting the lost packet, TCP achieves reliable data delivery.



## TCP: Cumulative and Duplicate Acks



# TCP: Selective Ack

- Selective Ack
  - Receiver can indicate missing segments
  - Handles "holes" efficiently
  - Co-exists with cumulative acks

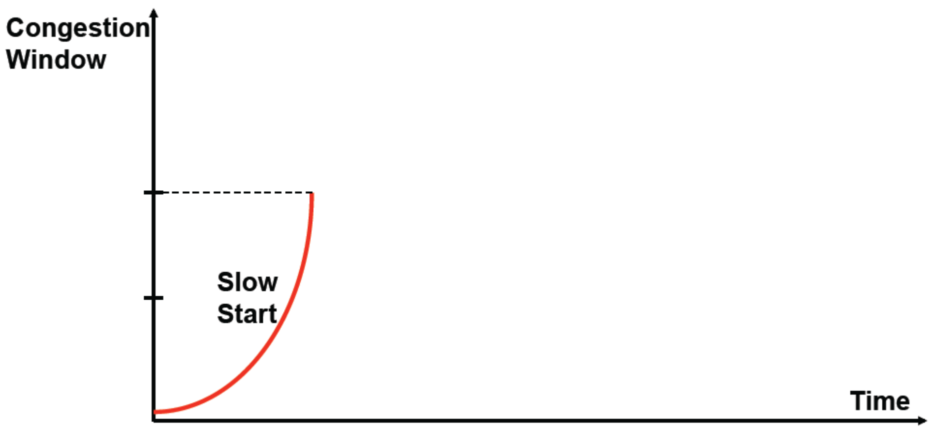
**TCP SACK**

| 0-99 | 100-199 | 200-299 | 300-399 | 400-499 |
|---|---|---|---|---|



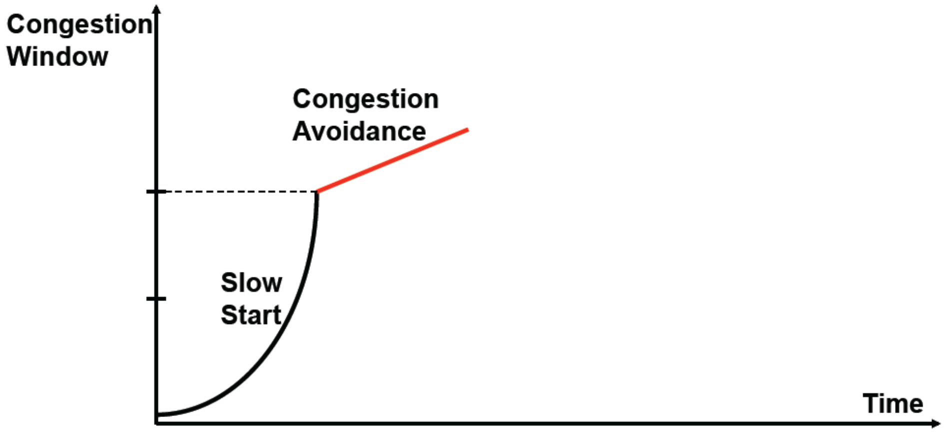
# TCP: Congestion Control

- Slow Start
  - Actually grows exponentially
  - Starts from small window size (e.g. 2 KB)
  - Multiplicatively increases window until threshold reached or packet loss occurs



# TCP: Congestion Control

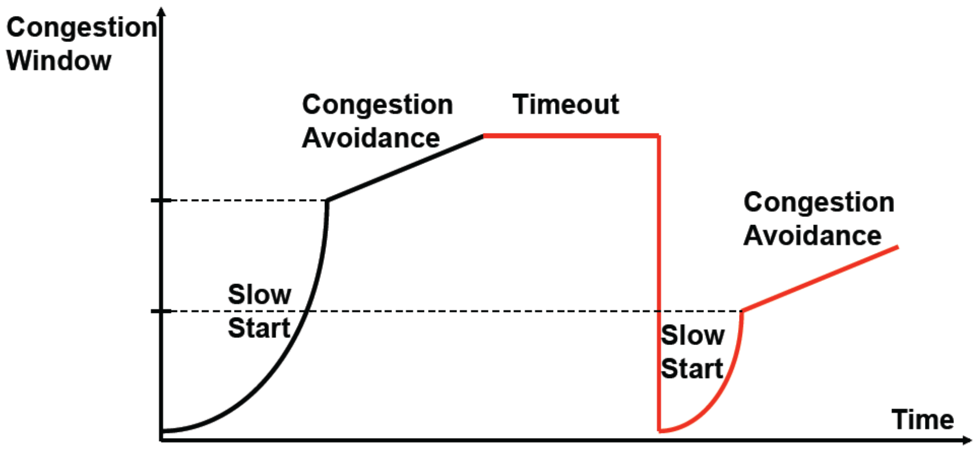
- Congestion Avoidance
  - Incrementally increases window after threshold reached





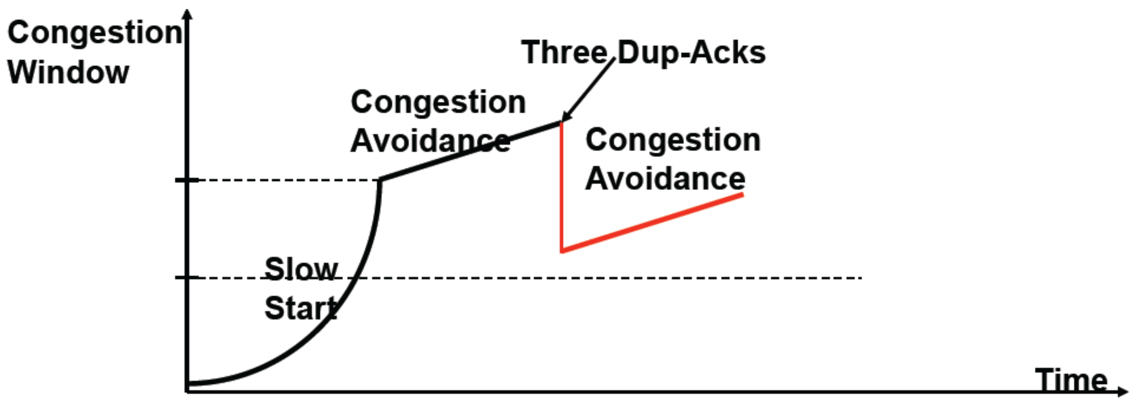

# TCP: Congestion Control

- Timeout (RTO)
  - Decreases window size to 1
  - Halves threshold
  - Timeout value = $f(Mean\_RTT, Variance\_RTT)$



# TCP: Congestion Control

- Fast Retransmit (TCP-Reno)
  - On three duplicate acks
    - Retransmits the lost segment right away (no need to wait for timeout)
    - Reduces window size to 0.5x*Window* + #*dup-acks* (fast recovery)
    - Enters congestion avoidance phase (fast recovery)






# Challenges of TCP over wireless

- Compared with wireline networks, wireless networks have some inherent adverse characteristics that will significantly deteriorate TCP performance if no action is taken.

- **Channel Errors**
  - In wireless networks, relatively high bit error rate because of multipath fading and shadowing may corrupt packets in transmission, leading to losses of TCP data segments or acknowledgements.
  - If it cannot receive the acknowledgement within the retransmission timeout, the TCP sender immediately reduces its congestion window to one segment, exponentially backs off its RTO and retransmits the lost packets.
  - Intermittent channel errors may cause the congestion window size at the sender to remain small, thereby resulting in low TCP throughput.

## Challenges of TCP over wireless

- **Mobility**
  - Cellular networks are characterized by handoffs due to user mobility.
  - Normally, handoffs may cause temporary disconnections, resulting in packet losses and delay.
  - TCP will suffer a lot if it treats such losses as congestion and invokes unnecessary congestion control mechanisms.
  - The handoffs are expected to be more frequent in next generation cellular networks as the micro-cellular structure is adopted to accommodate more users.
  - TCP needs to handle handoff gracefully.
  - Similar problem may occur in WLANs as mobile users will also encounter communication interruptions if they move to the edge of the transmission range of the access point.



## Challenges of TCP over wireless

- **Asymmetry**
  - The wireless link between a base station and a mobile terminal is asymmetric in nature.
  - Compared with the base station, the mobile has limited power, processing capability and buffer space.
  - Another asymmetry comes from the vastly different characteristics of wireline and wireless links.
  - The former is reliable and has large bandwidth, while the latter is error-prone and has limited and highly variable bandwidth.
  - For example, the bandwidth of Ethernet is 1 Gbps or higher for Gigabit Ethernet, while the bandwidth for 4G networks is about 200 Mbps (usually lower in practice).
  - Therefore, the wireless link is very likely to become the bottleneck of TCP connections.

# TCP over Wireless Networks

## TCP Over Wireless Networks

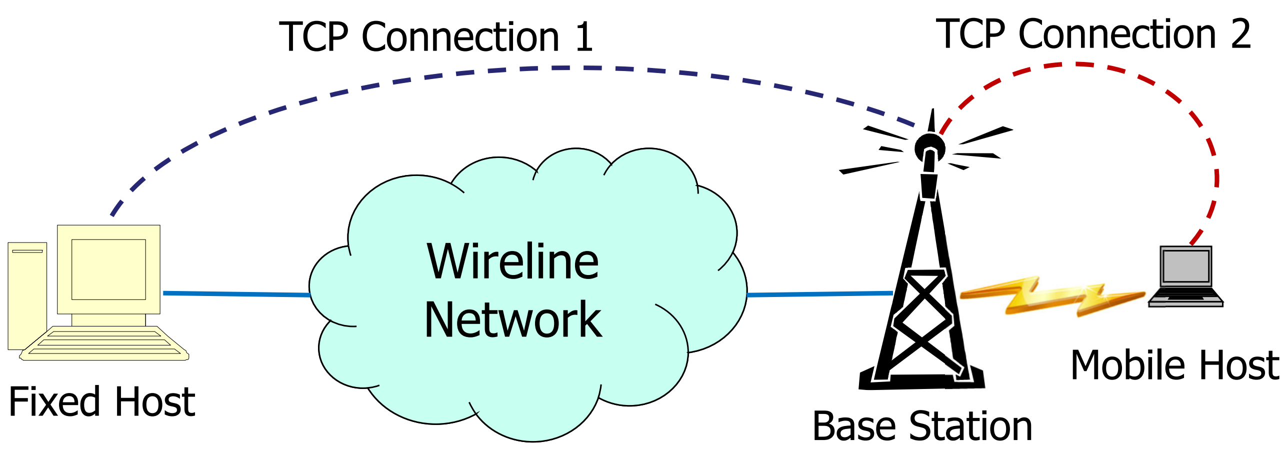
Split Connection Solution → Indirect TCP

Proxy-Based Solution → SNOOP



# 1. Split-Connection Solution– Indirect TCP

A. Bakre ; B.R. Badrinath, "I-TCP: Indirect TCP for mobile hosts", IEEE Int'l Conference on Distributed Computing Systems, 1995




Split Connection Using Indirect TCP

## Split-Connection Solution – Indirect TCP

- Indirect-TCP is a split-connection solution.
- It suggests that any TCP connection from a mobile host (MH) to a fixed host (FH) in the wireline network should be split into two separate connections.
- One of the connections is between the MH and its base station (BS) over the wireless medium.
- The other connection is between the BS and the FH over the wireline network.
- A packet which is sent to the MH is first received by the BS.
- It then sends an acknowledgement to the FH and then forwards the packet to the MH.
- If the MH moves to a different cell while communicating with a FH, the whole connection information maintained at the current BS is transferred to the new BS and the new BS takes over thereafter.
- The FH is unaware of this redirection and is not affected when this switch occurs.
- Since the end-to-end connection is split, the TCP connection over the wireless link can use some wireless-link-aware TCP variation, which may be tailored to handle wireless channel errors and handoff disruptions.



## Split-Connection Solution – Indirect TCP

- Indirect-TCP separates the congestion control functionality on the wireless link from that on the wireline network, which enables the two kinds of links to identify different reasons for packet losses and then take corresponding actions.
- Since the TCP connection is broken into two, it is possible for a mobile host to use some lightweight transport protocol instead of a full TCP/IP suite to communicate with the base station and access the wireline network through the base station.
- This feature is desirable since a mobile host has limited battery and processing power.
- However, Indirect-TCP violates the end-to-end semantics of TCP acknowledgements, as both the wireline part and wireless part of a connection have their own acknowledgements.
- The control overhead is also considerable as the base station needs to maintain a significant amount of state information for each TCP connection and all the state information needs to be transferred to the new base station in the event of a handoff, which could lead to long delay.

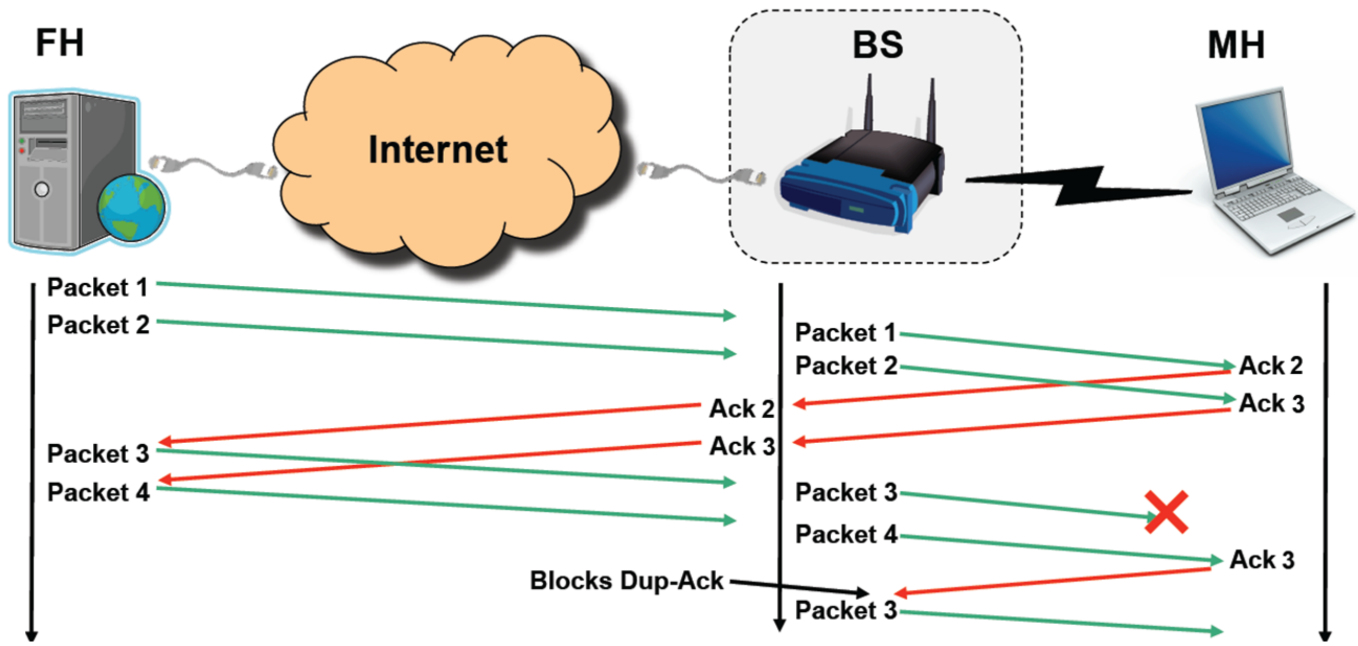
# 2. Proxy-Based Solution - SNOOP

H. Balakrishnan, et al. Improving TCP/IP Performance over Wireless Networks, 1995

- SNOOP sought to improve TCP performance by modifying the network-layer software at a BS while preserving end-to-end TCP semantics.
- It adds a Snooping module to the network layer, which monitors every packet that passes a BS in either direction.
- If TCP packets are sent from a FH to a MH, the SNOOP module caches each packet that has not yet been acknowledged by the MH.
- Meanwhile, the SNOOP module also keeps track of all acknowledgements sent from the mobile host.
- The SNOOP module determines that a packet loss occurs by detecting if it receives a duplicate acknowledgement or its local timer times out.
- In this case, the lost packet is retransmitted if it has been cached.
- The duplicate acknowledgements, if any, are suppressed.
- In this way, unnecessary congestion control mechanism invocations are avoided since packet losses due to wireless channel errors are hidden from the FH.



# Illustration of SNOOP



# Proxy-Based Solution - SNOOP

- In the case that packets are transmitted from a MH to a FH, the MH cannot tell whether a packet loss is due to errors on the wireless link or due to congestion in the wireline network.
- Thus, the TCP selective acknowledgement option is used.
- A selective-acknowledgement-enabled sender can retransmit multiple lost packets in one RTT instead of detecting only one lost packet in each RTT.
- At the BS, when the SNOOP module notices a gap in the inbound sequence numbers of the packets sent from the MH, selective acknowledgements are sent to the MH.
- Upon receiving such selective acknowledgements, the MH will retransmit the lost packet for local loss recovery.



# Proxy-Based Solution - SNOOP

- SNOOP is also designed to handle handoffs.
- Several BSs near a MH will form a multicast group and buffer some latest packets from the FH.
- Prior to a handoff, the MH will send control messages to determine that a BS with the strongest signal should be the one forwarding packets to the MH, and all other BSs just buffer packets.
- Therefore, both handoff delay and packet losses are reduced.
- SNOOP improves TCP performance through performing local retransmissions across the wireless link without affecting end-to-end TCP semantics.
- Although TCP performance during handoffs may be improved, considerable overhead is incurred for maintaining the multicast BS group and state transfer from one BS to another.

- Special care is needed to handle the interaction of the SNOOP module retransmission and TCP end-to-end retransmission because SNOOP is similar to link-level approaches over the wireless links.

# Other 'TCP over wireless' enhancements

- TCP-SACK
  - Selective Acknowledgement and Selective Retransmission.
  - Sender can retransmit missing data due to random errors/mobility
- WTCP Protocol
  - Separate flows for wired (Sender to BS) and wireless (BS to MS) segments of TCP connections
  - Local Retransmission for mobile link breakage
  - BS sends ACK to sender after Timestamp modification to avoid change in round trip estimates
- Freeze-TCP Protocol
  - Mobile detects impending handoff
  - Advertises Zero Window size, to force the sender into Zero Window Probe mode
- Explicit Band State Notification (EBSN)
  - Local Retransmission from BS to shield wireless link errors
  - EBSN message from BS to Source during local recovery
  - Source Resets its timeout value after EBSN



# Other 'TCP over wireless' enhancements (cont.)

- Fast Retransmission approach
  - Tries to reduce the effect of MS handoff
  - MS after handoff sends certain number of duplicate ACKs
  - Avoids coarse time-outs at the sender, Accelerates retransmission
- Transport Unaware Link Improvement Protocol (TULIP)
  - Provides Service Aware Link Layer
  - Reliable Service for TCP packets, unreliable service for UDP Packets
- AIRMAIL Protocol
  - Asymmetric Reliable Mobile Access in Link Layer
  - Uses combination of FEC and ARQ for loss recovery

**The End**

Questions?