CS 552
Wireless

R. Martin
Dept. of Computer Science
Rutgers University

Slides from B. Nath, R. Yang

Outline

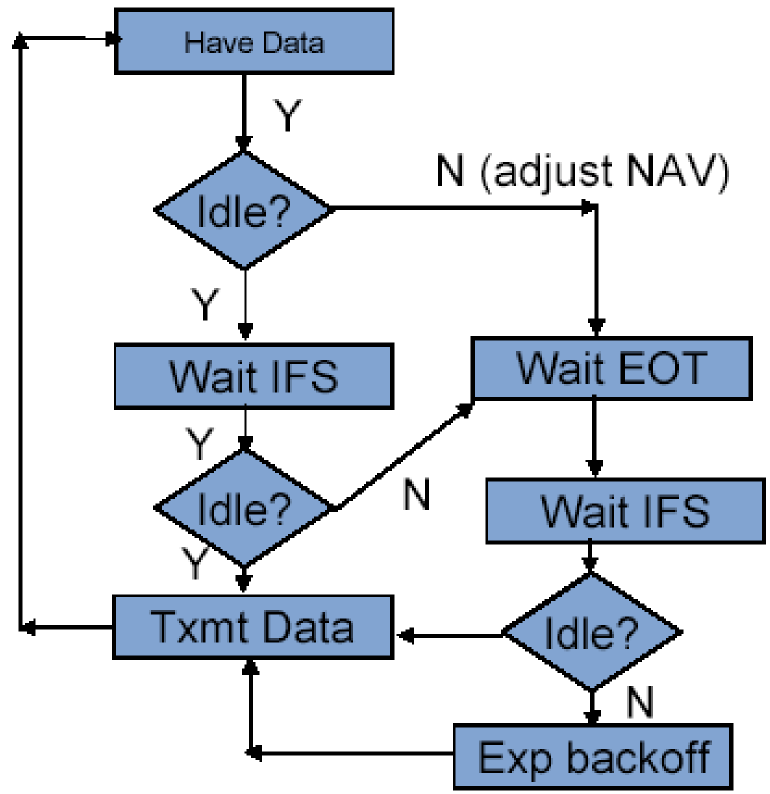
- 802.11
  - Bluetooth
  - ZigBee (802.15.4)
- I-TCP
- GPRSWeb

802.11 Concepts

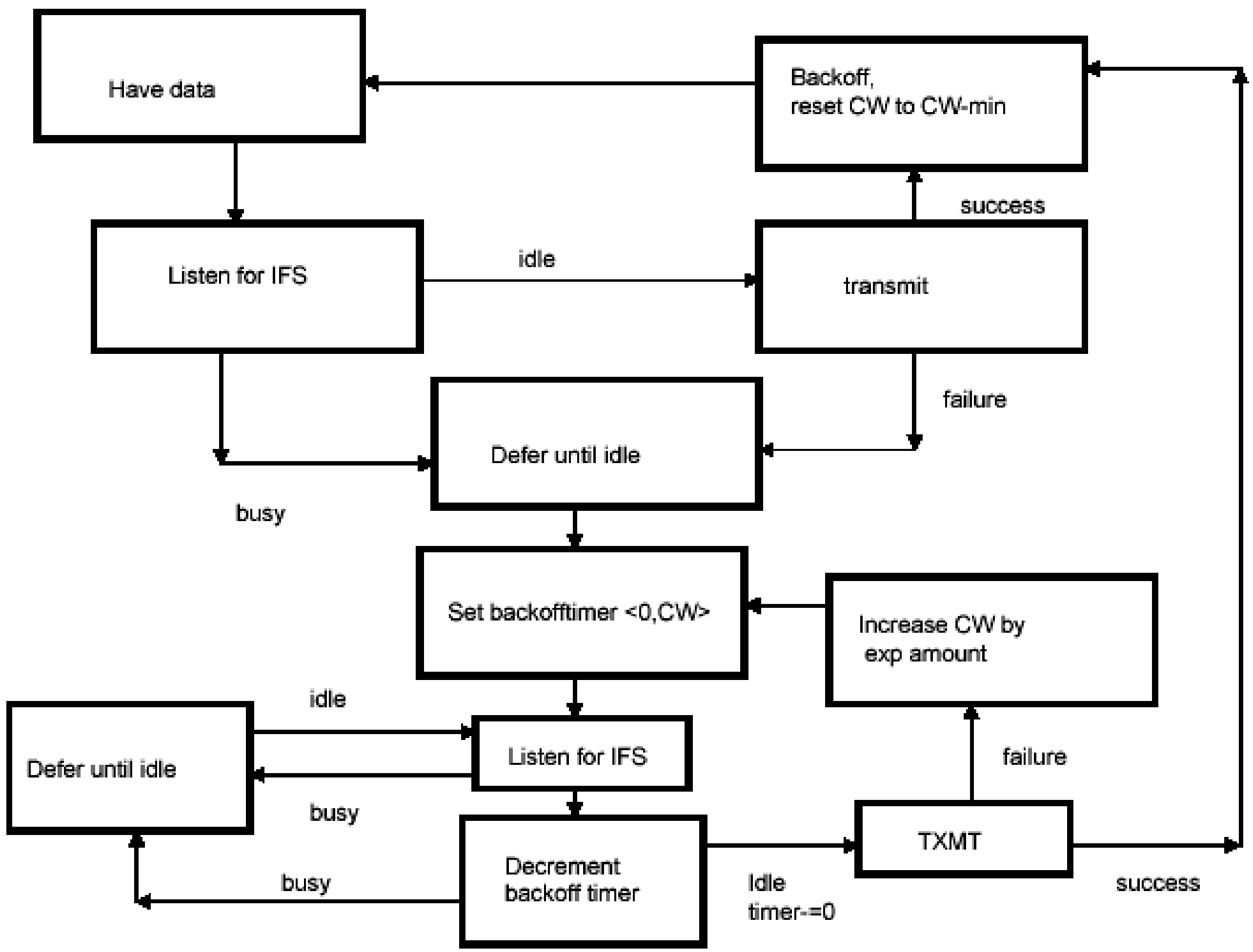
- Modes
  - Infrastructure Mode
  - Ad-Hoc Mode
- MAC
  - Distributed Coordination Function (DCF)
  - Point Coordination Function (PCF)
- Physical Layer
  - Frequency Hopping (FH)
  - Direct sequence (DS)
  - Orthogonal Frequency Division Muliplexing (OFDM)
  - Infrared (IR)

802.11 Access Control

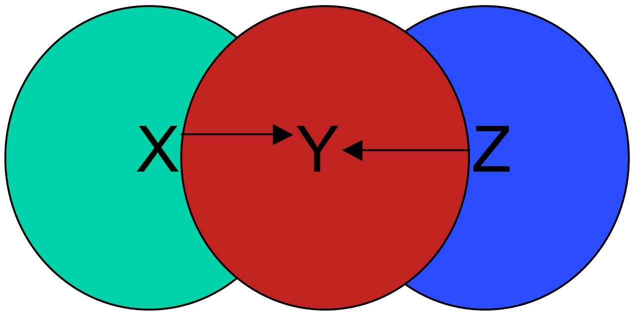
•Carrier Sense
•Is the medium idle? -> Wait for an amount of time
(IFS), if still idle transmit
  IFS = inter frame spacing
•Is the medium busy? ->
Wait until current txm
ends, wait (IFS), if idle wait for random amount of time, else wait until current txm ends and repeat
  (exponential backoff for collisions)
• ACKs and immediate response actions can be sent after SIFS (Short IFS) < PIFS<DIFS value used in multiple access control
 NAV=network allocation Data



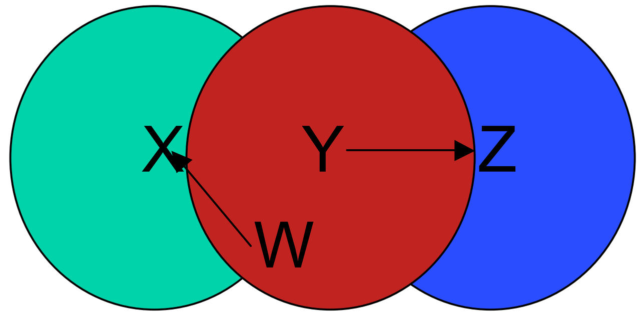
Flow Chart for CSMA/CA



Problems with Carrier Sensing



Hidden Terminal Problem



Exposed Terminal Problem

Hidden Terminal problem:

- Z can’t sense X; Tx to Y and collision with X
- No carrier does not always imply safe to send

Exposed terminal problem:

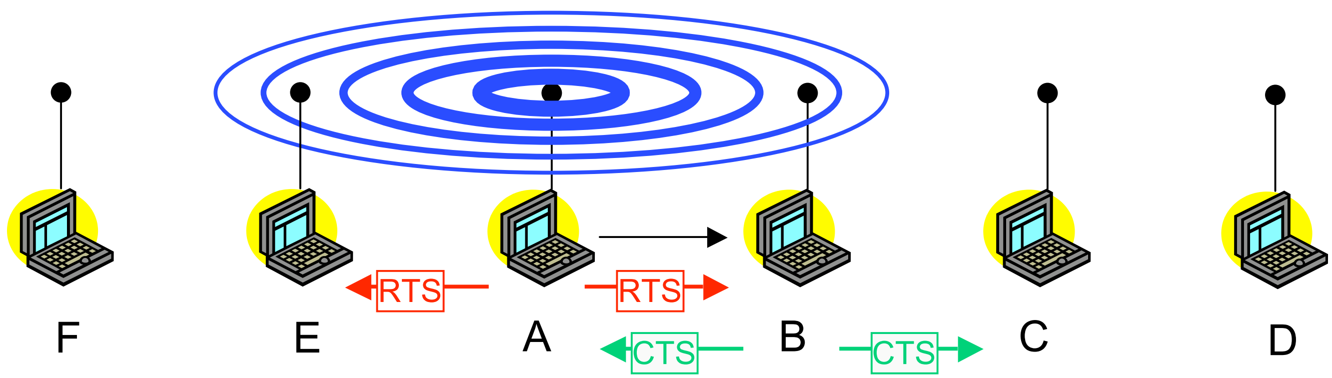
- W senses Y but can send to X
- Carrier may not imply unsafe to send

Problems of Wireless MAC

- Carrier sense
  - in Ethernet, we use carrier sense to avoid and detect potential collision
  - for wireless networks, the hidden-terminal, and the exposed-terminal problems make carrier sense (i.e., listen before talk) neither necessary nor sufficient
    - not detected transmission at the sender does not imply no current transmission to the receiver
    - detected transmission at the sender does not imply transmission will cause collision
- How to integrate random access “distributed coordination function (DCF) and taking turns “point coordination function (PCF)?

Basic Solution: Using RTS/CTS to Address the Carrier Sense Problem

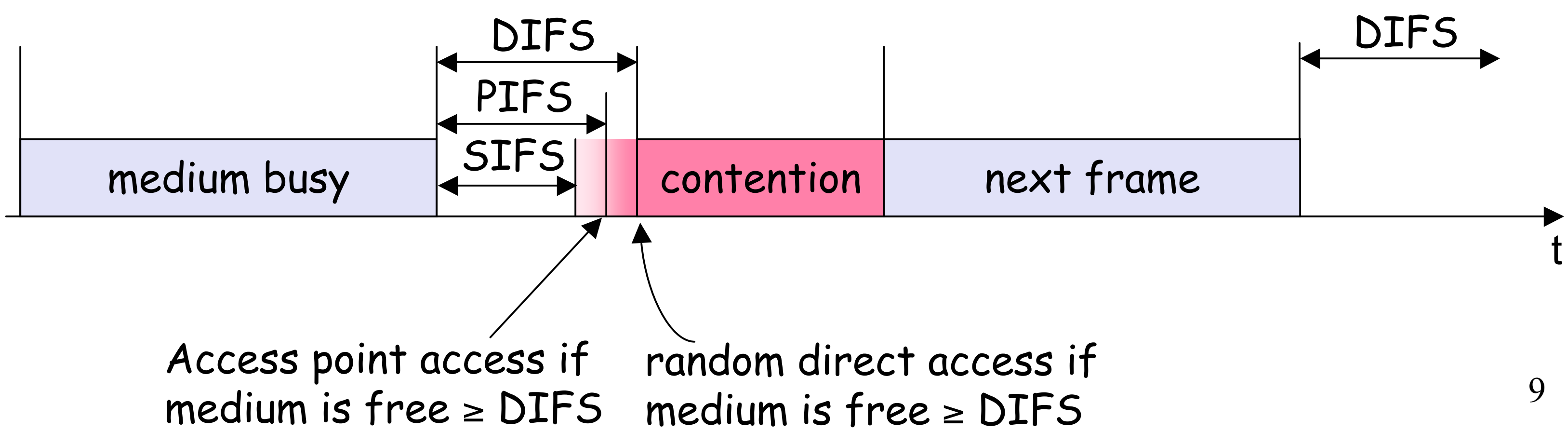
- Short signaling packets---virtual carrier sense
  - RTS (request to send) and CTS (clear to send)
    - to avoid collision at the receiver, any station who hears a CTS should not transmit
    - frames need to contain sender address, receiver address, transmission duration



Example: A sends to B

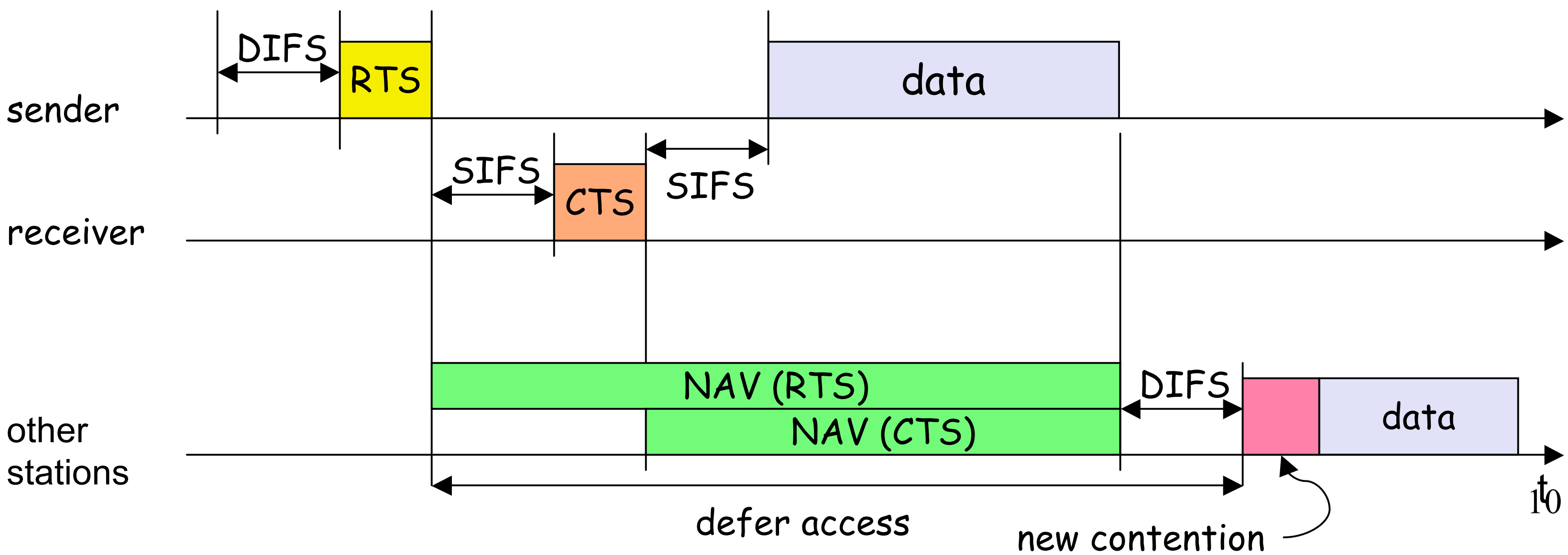
Basic Solution: Using Inter Frame Spacing to Prioritize Access

- Different inter frame spacing (IFS): if the required IFS of a type of message is short, the type of message has higher priority
  - SIFS (Short Inter Frame Spacing)
    - highest priority, for ACK, CTS, polling response
  - PIFS (Point Coordination Function Spacing)
    - medium priority, for time-bounded service using PCF
  - DIFS (Distributed Coordination Function Spacing)
    - lowest priority, for asynchronous data service



Basic Control Flow of RTS/CTS

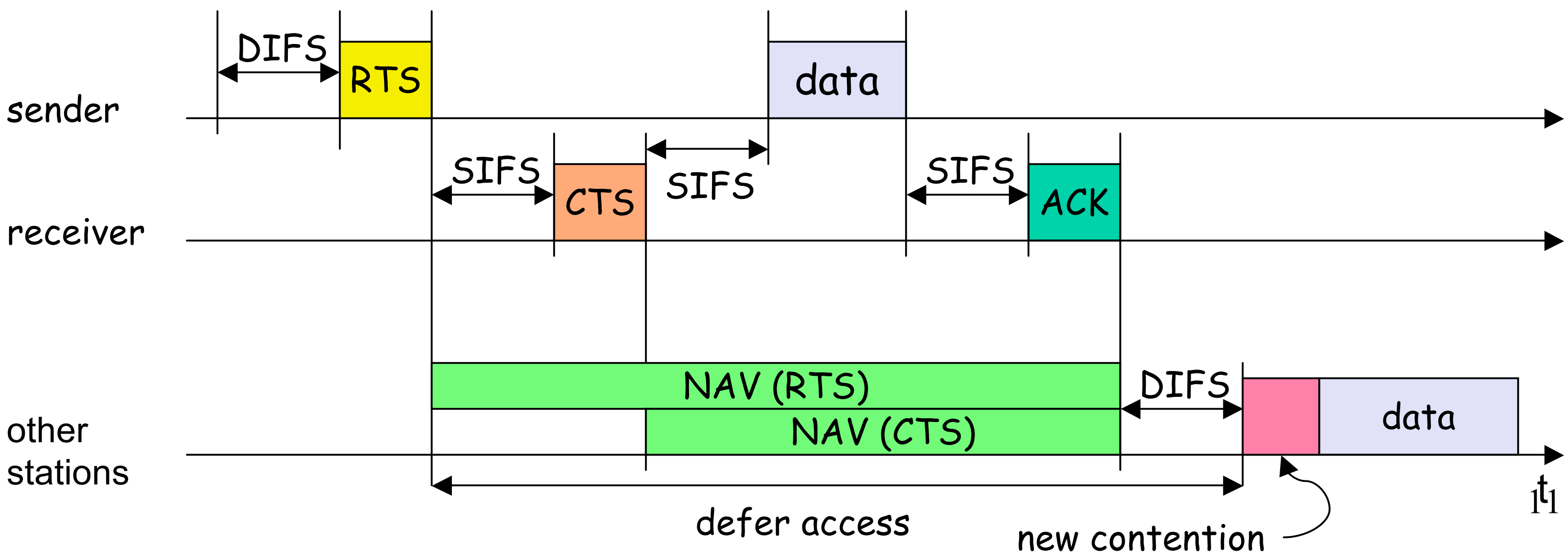
- Sender sends RTS with NAV (Network allocation Vector, i.e. reservation parameter that determines amount of time the data packet needs the medium) after waiting for DIFS
- Receiver acknowledges via CTS after SIFS (if ready to receive)
  - CTS reserves channel for sender, notifying possibly hidden stations;
  - any station hearing CTS should be silent for NAV
- Sender can now send data at once



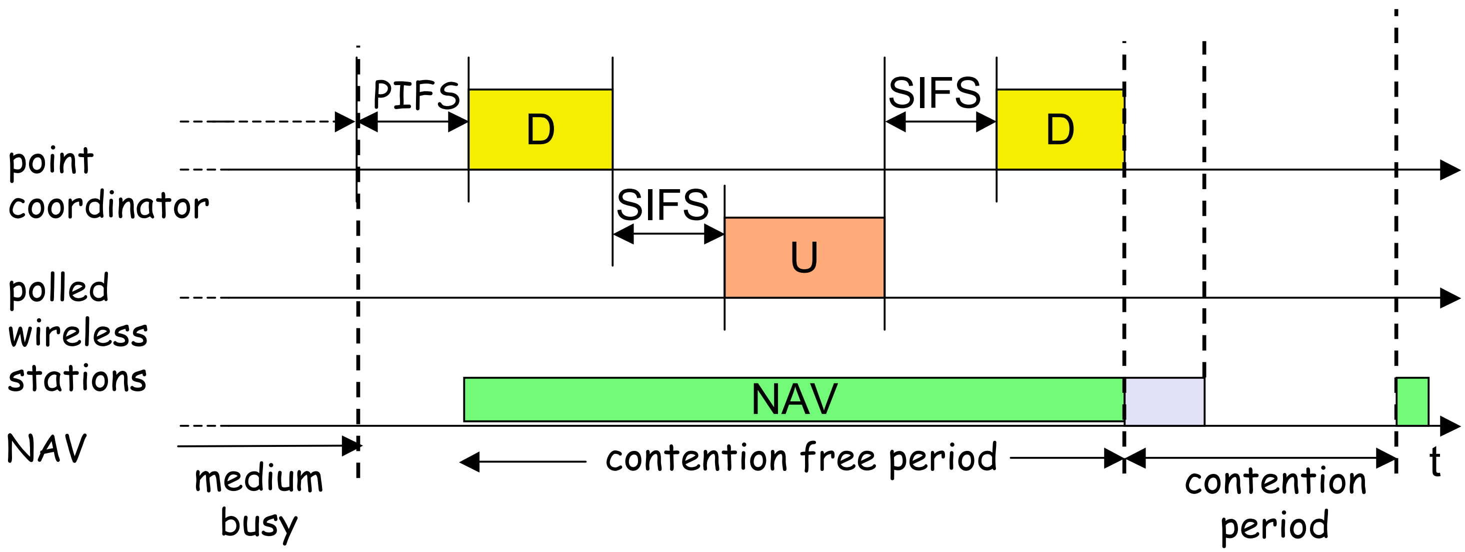
802.11: RTS/CTS + ACK, the Final Version

- 802.11 adds ACK in the signaling to improve reliability
  - implication: to avoid conflict with ACK, any station hearing RTS should not send for NAV
  - thus a station should not send for NAV if it hears either RTS and CTS

Note: RTS/CTS is optional in 802.11, and thus may not be always turned on---some network interface cards turn it on only when the length of a frame exceeds a given threshold



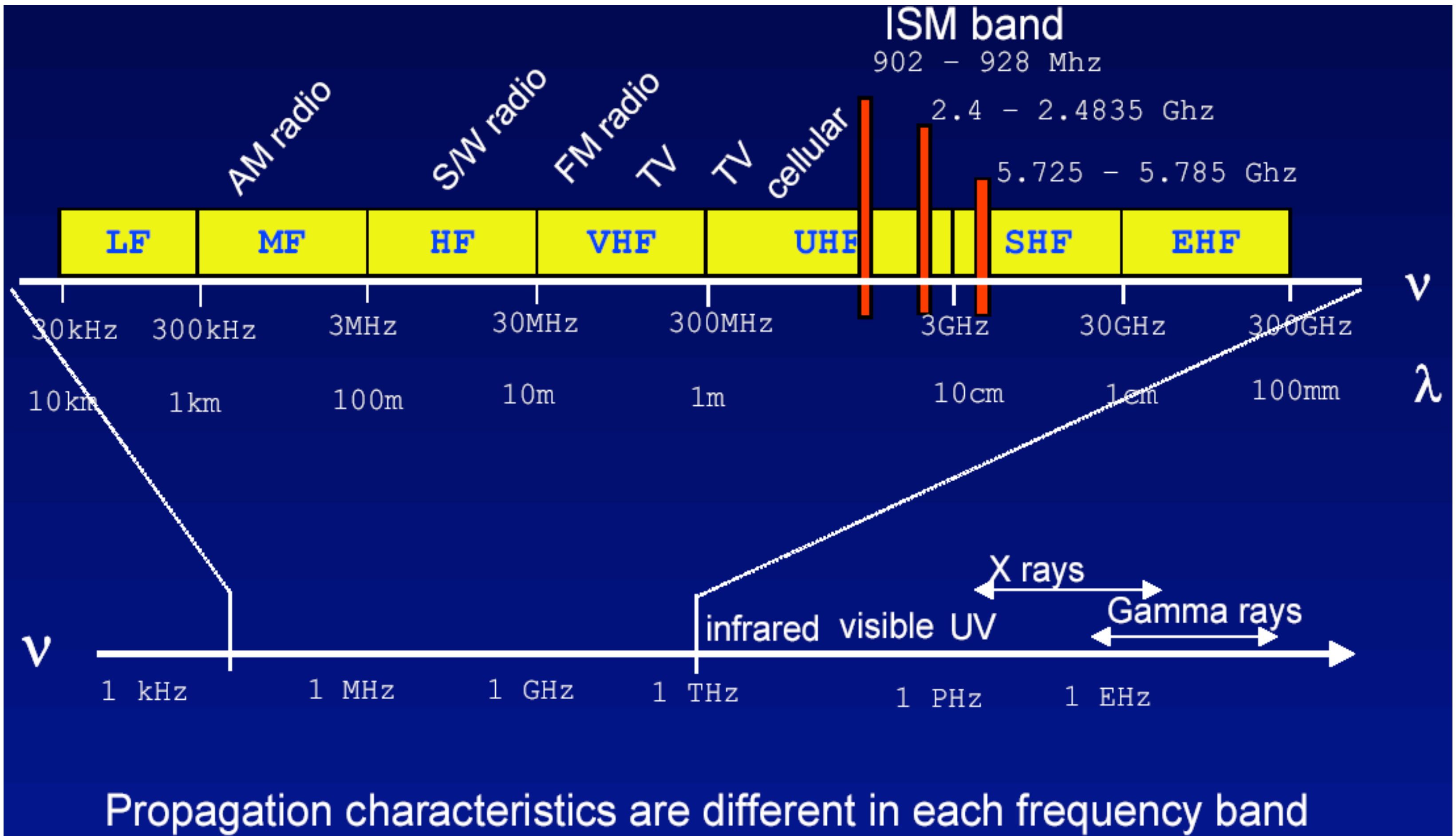
802.11: PCF for Polling



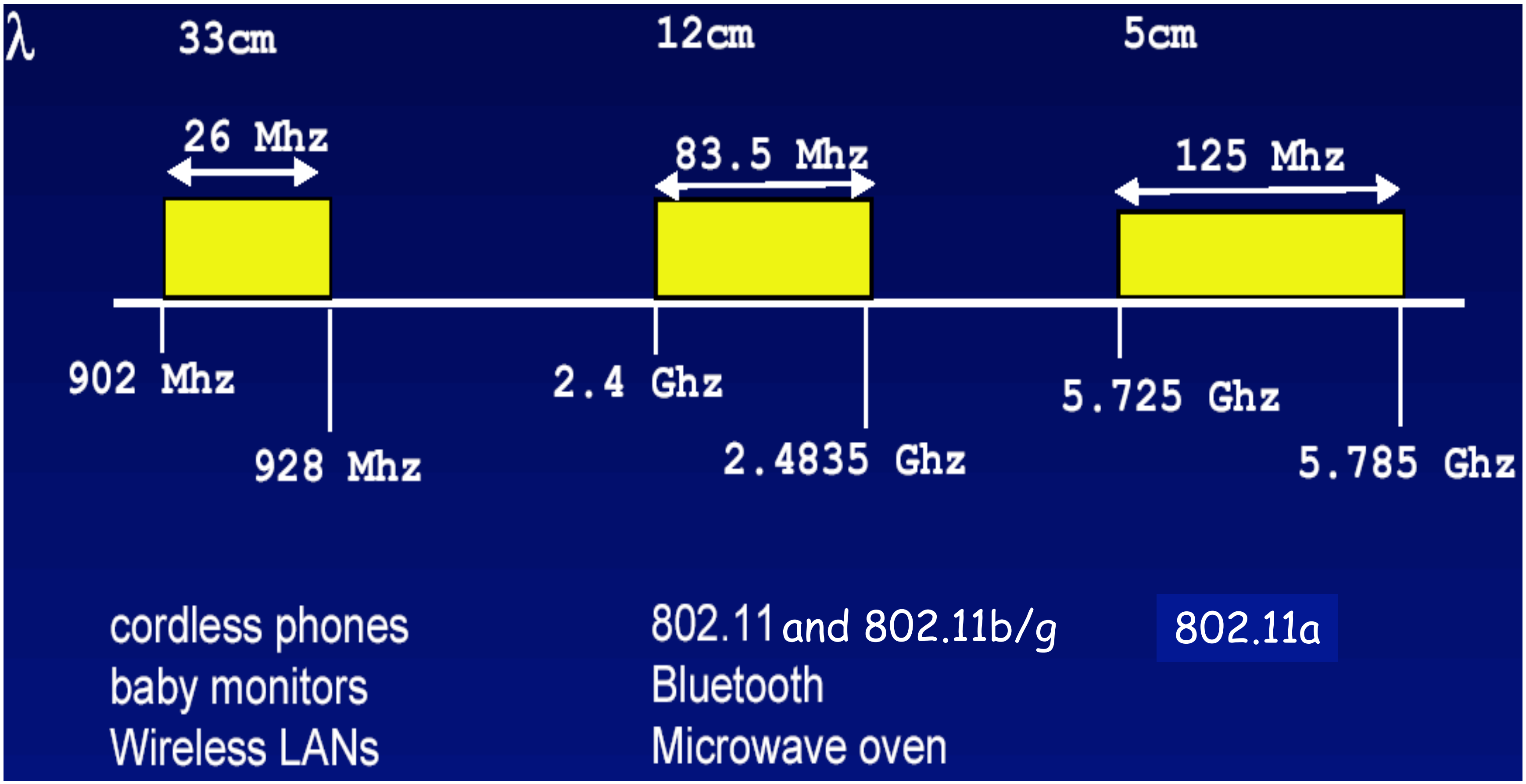
D: downstream poll, or data from point coordinator
U: data from polled wireless station

ISM: Industry, Science, Medicine

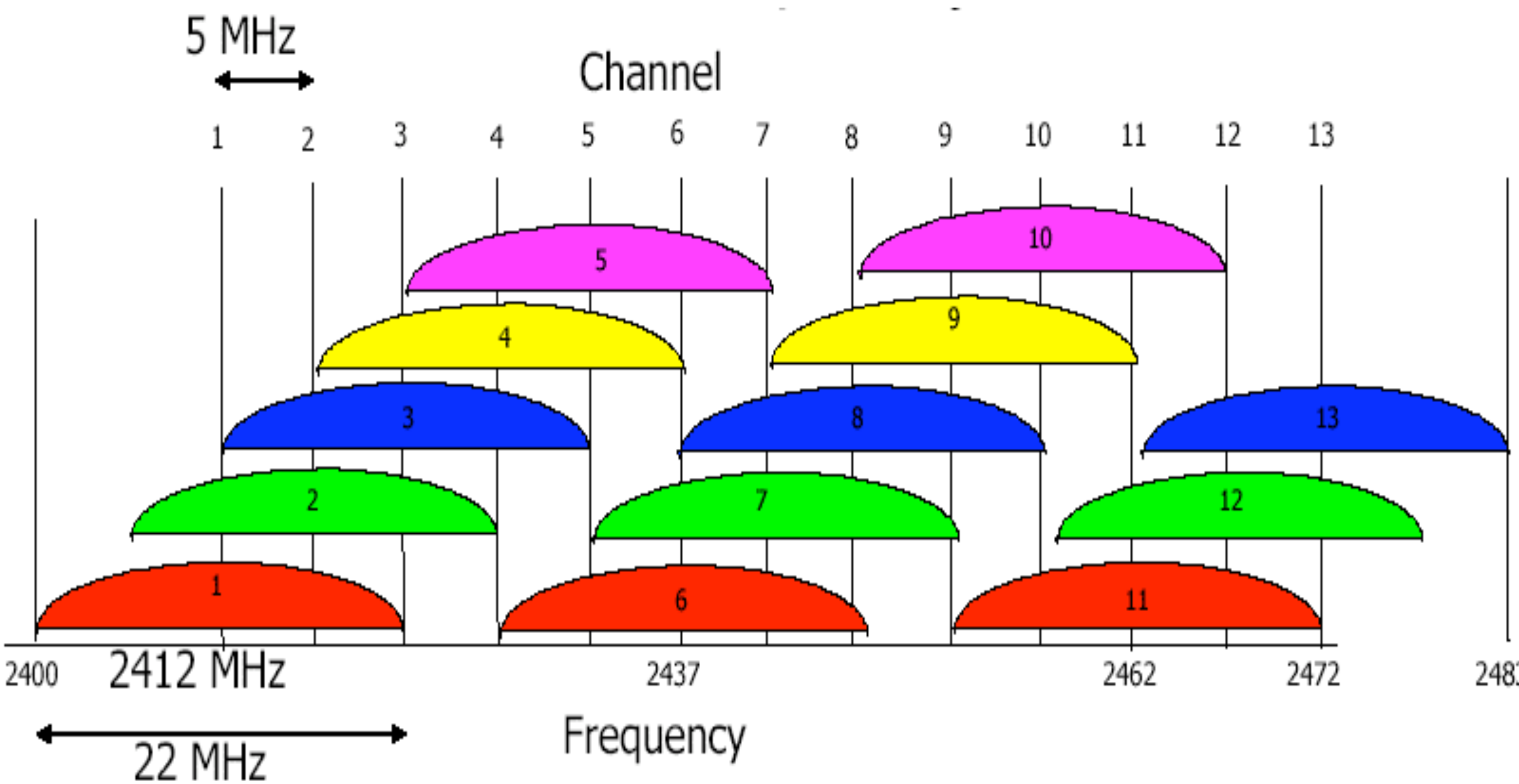
unlicensed frequency spectrum: 900Mhz, 2.4Ghz, 5.1Ghz, 5.7Ghz



IEEE 802.11 Frequency Band



802.11b/g Channels



IEEE 802.11 variants

	802.11a	802.11b	802.11g	802.11
Standard approved	Sep. 1999	Sep. 1999	June 2003	July 1997
Available bandwidth	300 MHZ	83.5 MHZ	83.5 MHZ	83.5 MHZ
freq. of operation	5.15-5.35G 5.725-5.825G	2.4-2.4835G	2.4-2.4835G	2.4-2.4835G
No. of non-overlapping Ch.	4	3	3	3
Rate per channel (Mbps)	6,12,24,36,48,54	1, 2, 5.5, 11	1, 2, 5.5, 11, 6, 9, 12, 18, 24, 36, 48, 54	1, 2
Range	~150 feet (indoor) 225 (outdoor)	~225 feet	~225 feet	??
Modulation	OFDM	DSSS/CCK	DSSS/CCK; DSSS/OFDM	DSSS, FHSS

FHSS: frequency hopping spread spectrum   DSSS: direct sequence spread spectrum
OFDM: orthogonal frequency division multiplexing

Bluetooth

A cable replacement

- Operates in the ISM band (2.4Ghz to 2.8 Ghz)
- Range is 10 cm to 10 meters can be extended to 100 meters by use of power control
- Data rates up to 1 Mbps (721Kbps)
- Supposed to be low cost, single chip radio
- Ideal for connecting devices in close proximity (piconet)
  - Phone and earpiece
  - Computer and printer
  - Camera and printer/fax etc
  -

Future Directions

- Wireless
  - Smaller, cheaper devices ($10)
  - Localization
- More bandwidth
  - Long haul
  - Last mile
- Cheaper computing at edges
  - ($100PC)

Future Networks: Social Issues

- Privacy
  - Big/Little brother
- Censorship/access control
- Intellectual Property/Fair use
- Dis-intermediation
- “Flat” world/globalization

Wireless TCP

- Packet loss in wireless networks may be due to
  - Bit errors
  - Handoffs
  - Congestion (rarely)
  - Reordering (rarely, except in ad-hoc networks (mobile))
- TCP assumes packet loss is due to
  - Congestion
  - Reordering (rarely)
- TCP's congestion responses are triggered by wireless packet loss but interact poorly with wireless nets

Impact of loss on TCP

- Random losses result in lower throughput
- Wireless loss is not due to congestion
- TCP cannot distinguish between link loss and congestion loss
- Wireless TCP needs to differentiate between the two
- Loss on wireless link means try harder, loss on wired means backoff
- How to reconcile between the two in an end-to-end transport mechanism
- A number of approaches
  - Link level, modified link level or link aware, transport level, explicit loss notification(ELN)

TCP congestion detection

- TCP assumes timeouts and duplicate acks indicate congestion or packet reordering (alternate paths)
- Timeout indicates packet or ack was lost
- Duplicate acks may indicate packet reordering
  - Receipt of duplicate acks means some data is still flowing
- Aggressive congestion control on loss but less aggressive on dup acks

Responses to congestion

- Basic timeout and retransmission
  - If sender receives no ack for data sent, timeout
  - Timeout value is sum of smoothed RTT delay and 4 X mean deviation
  - Exponential back-off
  - Timeout value based on mean and variance of RTT
- Congestion “avoidance” (really congestion control)
  - Uses congestion window (cwnd) for more flow control
  - Cwnd set to 1/2 of its value when congestion loss occurred
  - Sender can send up to minimum of advertised window and cwnd
  - Use additive increase of cwnd (at most 1 segment each RT)
  - Approach limit of the network capacity slowly
- Slow start, fast retransmit

Other problems in a wireless environment

- Burst errors due to poor signal strength or mobility (handoff)
  - More than one packet lost in TCP window
- Delay is often very high
  - RTT quite long
  - Tunneling, satellite
  - True in telephone networks providing data services that deploy fixed gateways (non-optimal routes)

Poor interaction with TCP

- Cumulative ack scheme not good with bursty losses
  - Missing data detected one segment at a time
  - Duplicate acks take a while to cause retransmission
  - TCP Reno may suffer coarse time-out and enter slow start!
  - TCP New Reno still only retransmits one packet per RTT
- Packet loss due to noise or hand-offs indicated by dup acks
  - Enter congestion control
  - Slow increase of cwnd
- Bursts of packet loss and hand-offs indicated by timeouts
  - Timeout
  - Enter slow start (start from cwnd = 1)

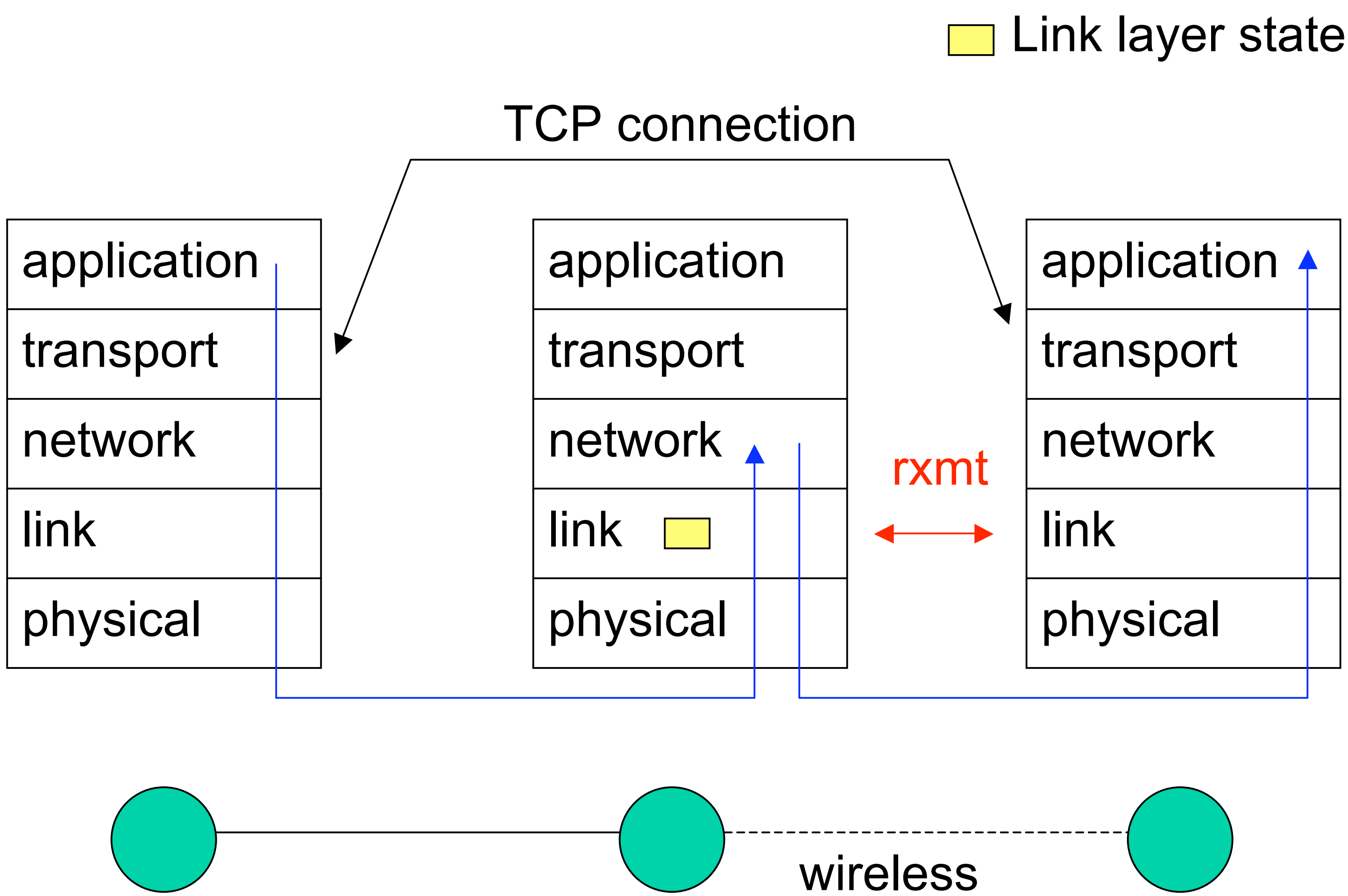
Multiple losses in window

- Assume cwnd of 8
- 1st and 4th packets lost
- 3rd duplicate ack causes retransmit of 1st packet
- Also sets cwnd to 4 + 3 = 7, ssthresh= 4
- Further duplicate acks increment cwnd by 1
- Ack for retransmit of packet 1 is a partial ack since packet 4 is also lost
- In TCP Reno this results in an exit out of fast retransmit
- reset congestion window to 4 but 8 packets were already sent

Approaches

- Link layer enhancements (FEC, retransmissions)
  - Interacts with RTT, higher variance may still lead to timeouts
  - Not a problem with coarse grain timeouts
  - But a problem in slow wireless links, as RTO estimates may be high
  - Interested see (Reiner Ludwig’s paper at Infocom)
- Transport layer I-TCP [BakreBadri95]
- TCP aware Link layer aware (Snoop)[Hari et al 96]
- Explicit Loss Notification schemes

Link Level Retransmissions



Link Level Retransmissions Issues

- How many times to retransmit at the link level before giving up?
  - Finite bound -- semi-reliable link layer
  - No bound -- reliable link layer
- What triggers link level retransmissions?
  - Link layer timeout mechanism
  - Link level acks (negative acks, dupacks, sacks)
- How much time is required for a link layer retransmission?
  - Small fraction of end-to-end TCP RTT
  - Large fraction/multiple of end-to-end TCP RTT
- Should the link layer deliver packets as they arrive, or deliver them in-order?
  - Link layer may need to buffer packets and reorder if necessary so as to deliver packets in-order

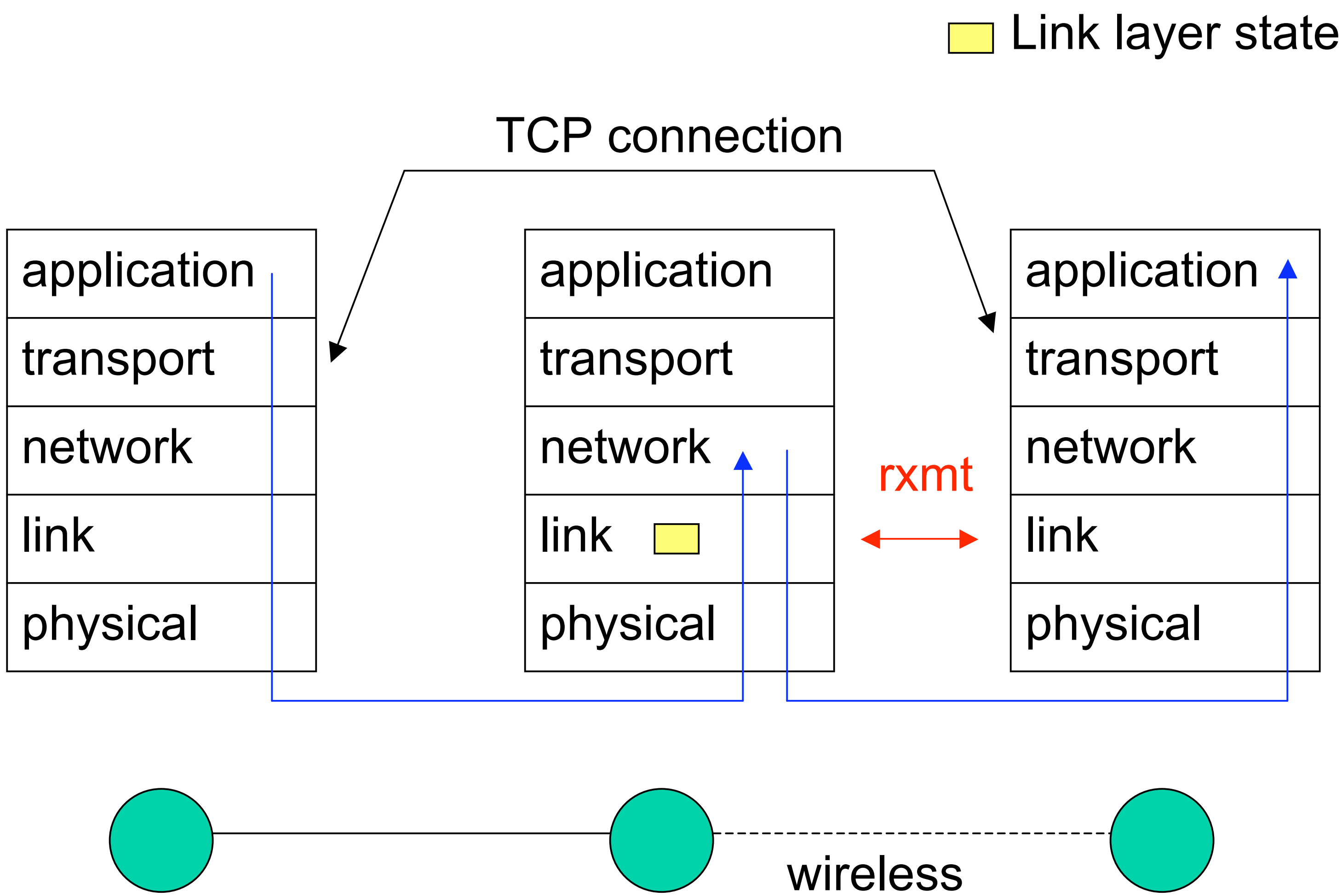
Link Layer Schemes applicability

- When is a reliable link layer beneficial to TCP performance?
- if it provides almost in-order delivery and
- TCP retransmission timeout large enough to tolerate additional delays due to link level retransmits
- Another headache, link layer packets may be smaller than MSS of TCP packets
- GSM protocol an example

Link Layer Schemes: Classification

- Hide wireless losses from TCP sender
- Link layer modifications needed at both ends of wireless link
  - TCP need not be modified

Link Level Retransmissions



Link Level Retransmissions
Issues

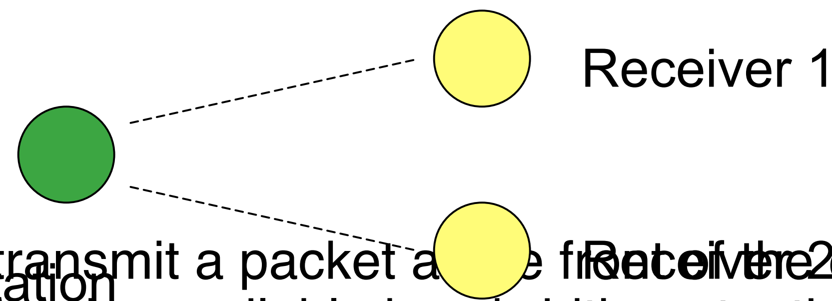
- How many times to retransmit at the link level before giving up?
  - Finite bound -- semi-reliable link layer
  - No bound -- reliable link layer
- What triggers link level retransmissions?
  - Link layer timeout mechanism
  - Link level acks (negative acks, dupacks, …)
  - Other mechanisms (e.g., Snoop, as discussed later)
- How much time is required for a link layer retransmission?
  - Small fraction of end-to-end TCP RTT
  - Large fraction/multiple of end-to-end TCP RTT

Link Level Retransmissions Issues

- Should the link layer deliver packets as they arrive, or deliver them in-order?
  - Link layer may need to buffer packets and reorder if necessary so as to deliver packets in-order

Link Level Retransmissions Issues

- Retransmissions can cause congestion losses

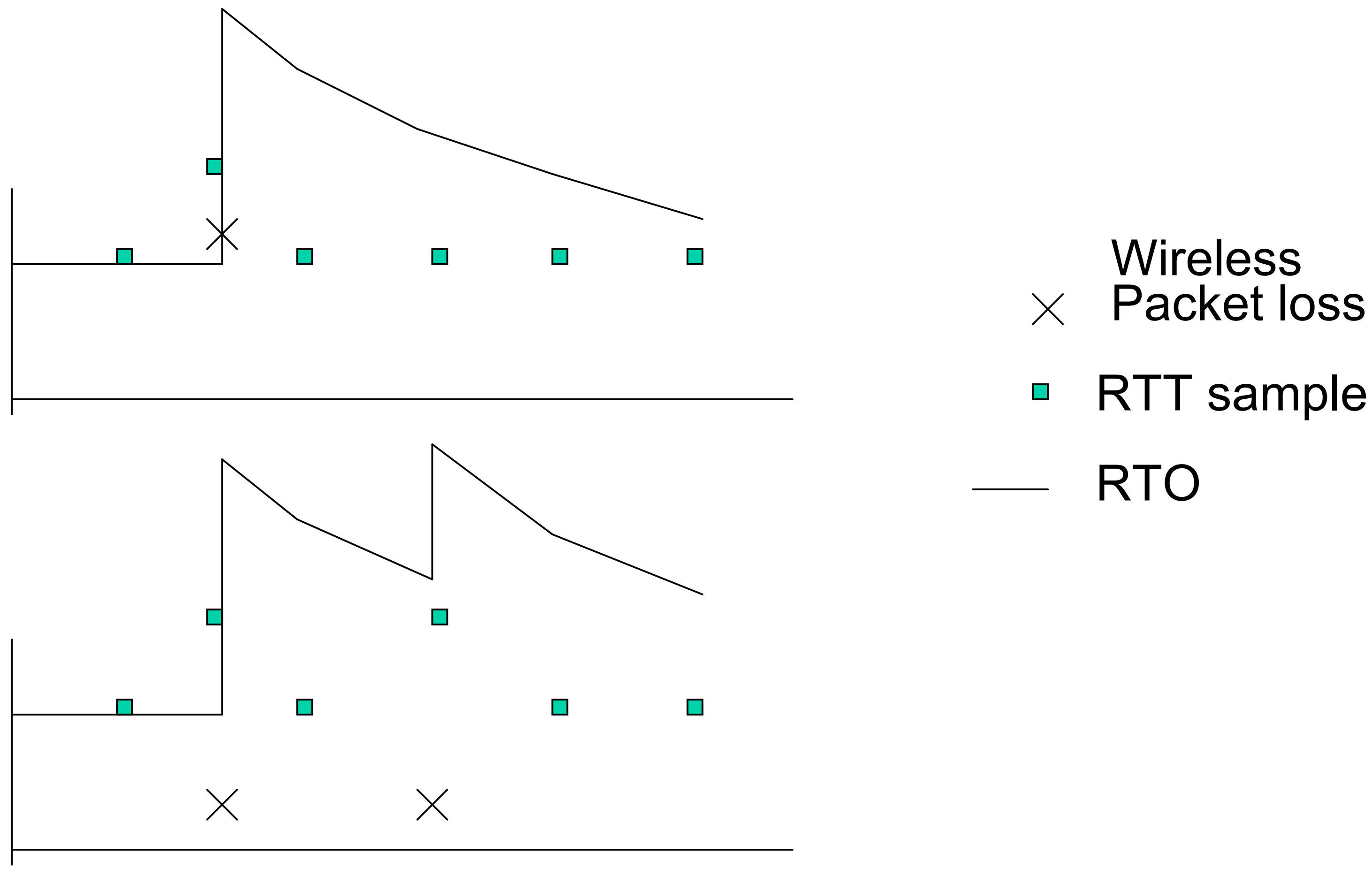


- Attempting to retransmit a packet at the front of the queue, effectively reduces the available bandwidth, potentially making the queue at base station longer
- If the queue gets full, packets may be lost, indicating congestion to the sender
- Is this desirable or not ?

Link Level Retransmissions
An Early Study [DeSimone93]

- The sender's Retransmission Timeout (RTO) is a function of measured RTT (round-trip times)
  - **Link level retransmits increase RTT, therefore, RTO**
- **If errors not frequent**, RTO will **not** account for RTT variations due to link level retransmissions
  - When errors occur, the sender may timeout & retransmit before link level retransmission is successful
  - Sender and link layer both retransmit
  - Duplicate retransmissions (interference) waste wireless bandwidth
  - Timeouts also result in reduced congestion window

RTO Variations



A More Accurate Picture

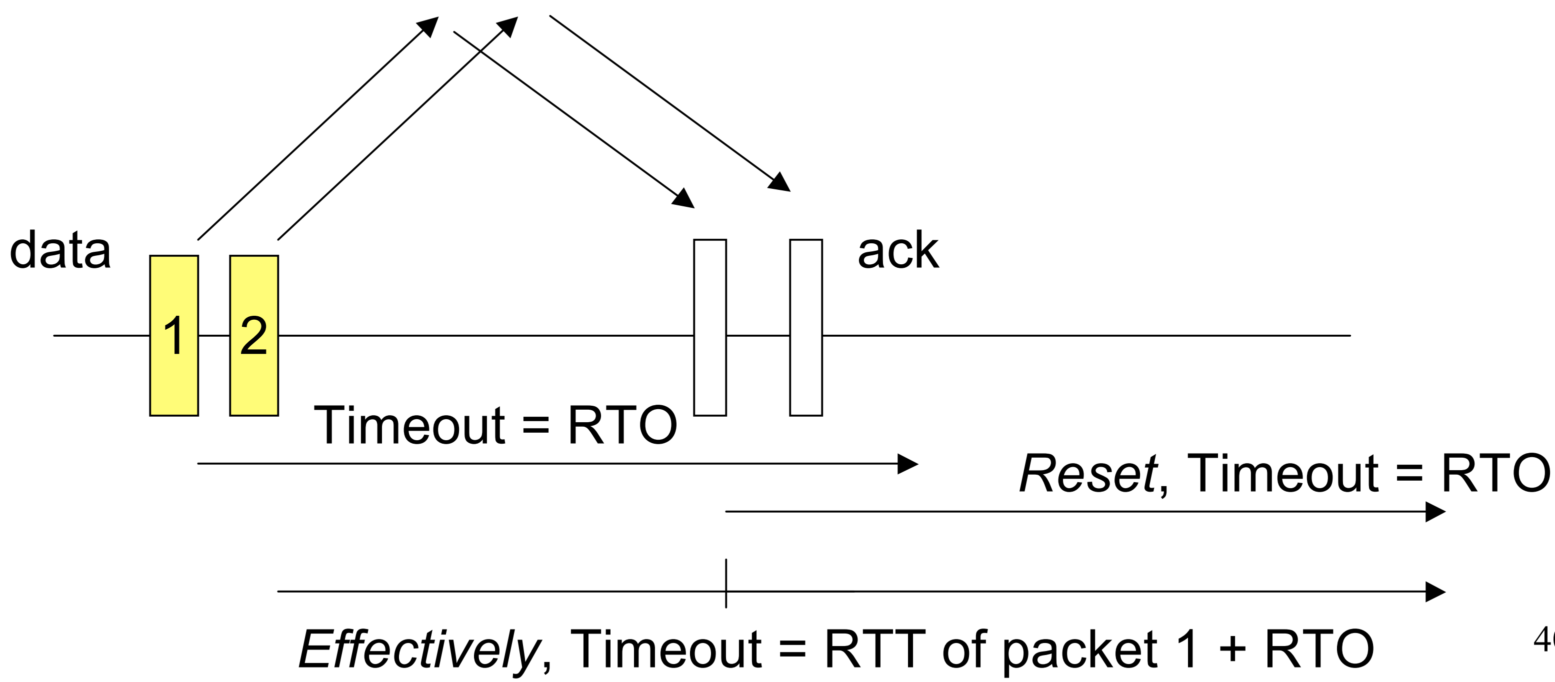
- Analysis in [DeSimone93] does not accurately model real TCP stacks
- With large **RTO granularity**, interference is unlikely, if time required for link-level retransmission is small compared to TCP RTO [Balakrishnan96Sigcomm]
  - Standard TCP RTO granularity is often large
  - Minimum RTO (2*granularity) is large enough to allow a small number of link level retransmissions, if link level RTT is relatively small
  - Interference due to timeout not a significant issue when wireless RTT small, and RTO granularity large [Eckhardt98]

Link Level Retransmissions
A More Accurate Picture

- **Frequent errors** increase RTO significantly on slow wireless links
  - RTT on slow links large, retransmissions result in large variance, pushing RTO up
  - Likelihood of interference between link layer and TCP retransmissions smaller
  - But congestion response will be delayed due to larger RTO
  - When wireless losses do cause timeout, much time wasted

Link-Layer Retransmissions
A More Accurate Picture [Ludwig98]

- Timeout interval may actually be larger than RTO
  - Retransmission timer reset on an ack
  - If the ack'd packet and next packet were transmitted in a burst, next packet gets an additional RTT before the timer will go off

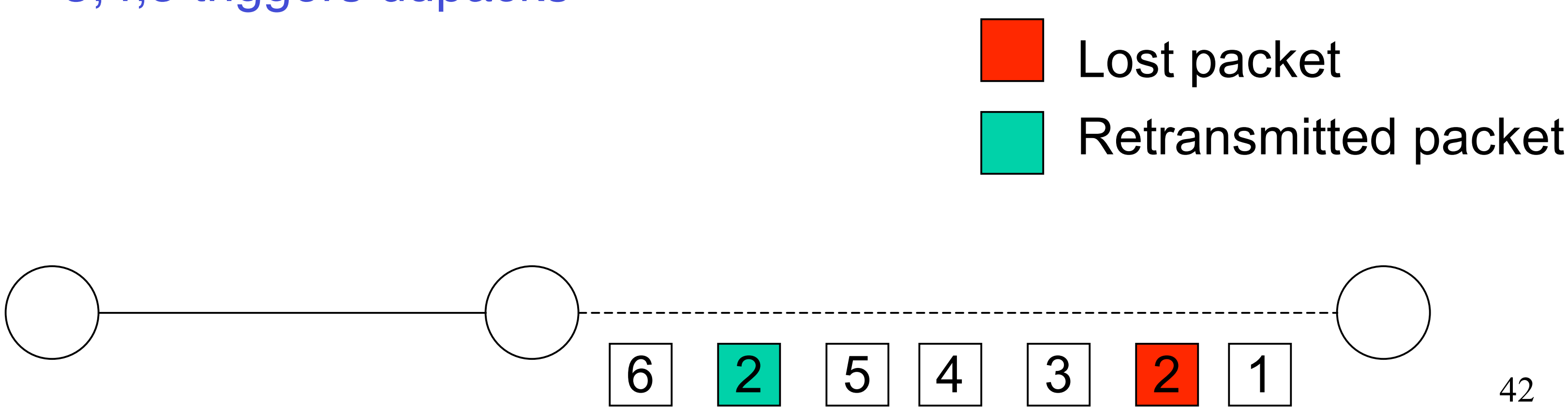


Large TCP Retransmission Timeout Intervals

- Good for reducing interference with link level retransmits
- Bad for recovery from congestion losses
- Need a timeout mechanism that responds appropriately for both types of losses
  - Open problem

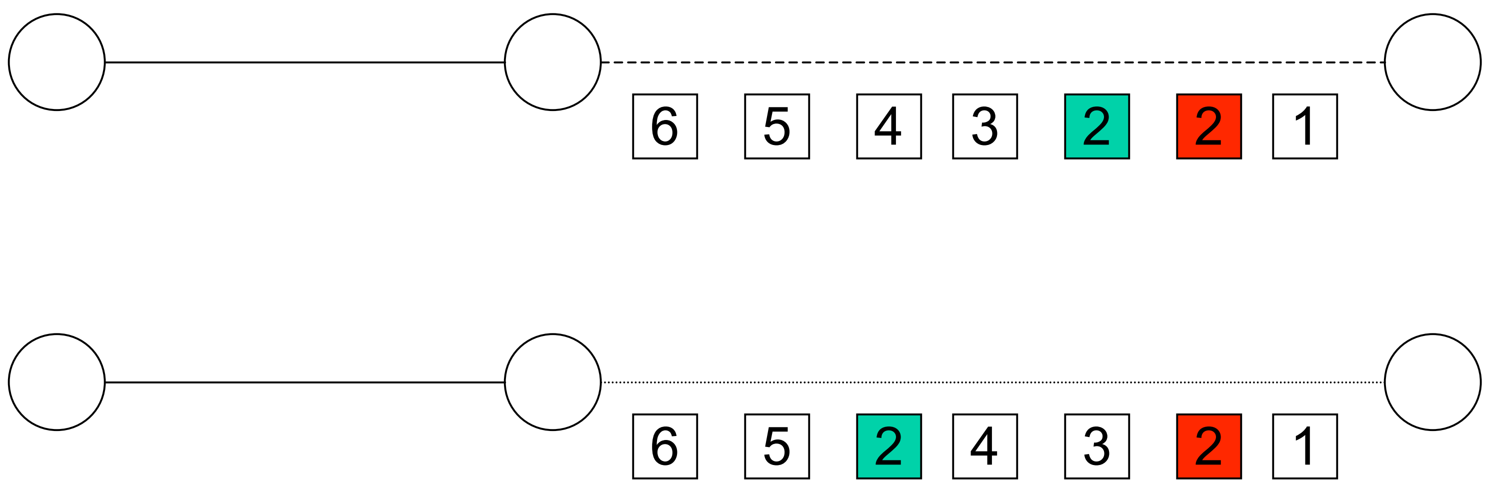
Link Level Retransmissions

- Selective repeat protocols can deliver packets out of order
- Significantly out-of-order delivery can trigger TCP fast retransmit
  - Redundant retransmission from TCP sender
  - Reduction in congestion window
- Example: Receipt of packets 3,4,5 triggers dupacks



Link Level Retransmissions
In-order delivery

- To avoid unnecessary fast retransmit, link layer using retransmission should attempt to deliver packets "almost in-order"



Link Level Retransmissions In-order delivery

- Not all connections benefit from retransmissions or ordered delivery
  - audio
- Need to be able to specify requirements on a per-packet basis
  - Should the packet be retransmitted? How many times?
  - Enforce in-order delivery?

Link Layer Schemes: Summary

When is a reliable link layer beneficial to TCP performance?

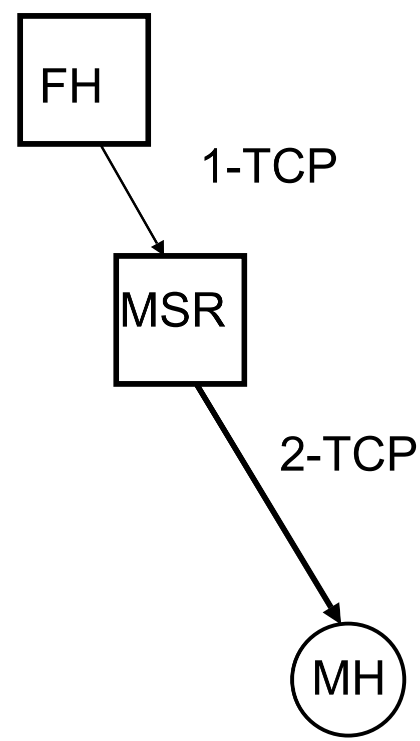
- if it provides almost in-order delivery

and

- TCP retransmission timeout large enough to tolerate additional delays due to link level retransmits

I-TCP

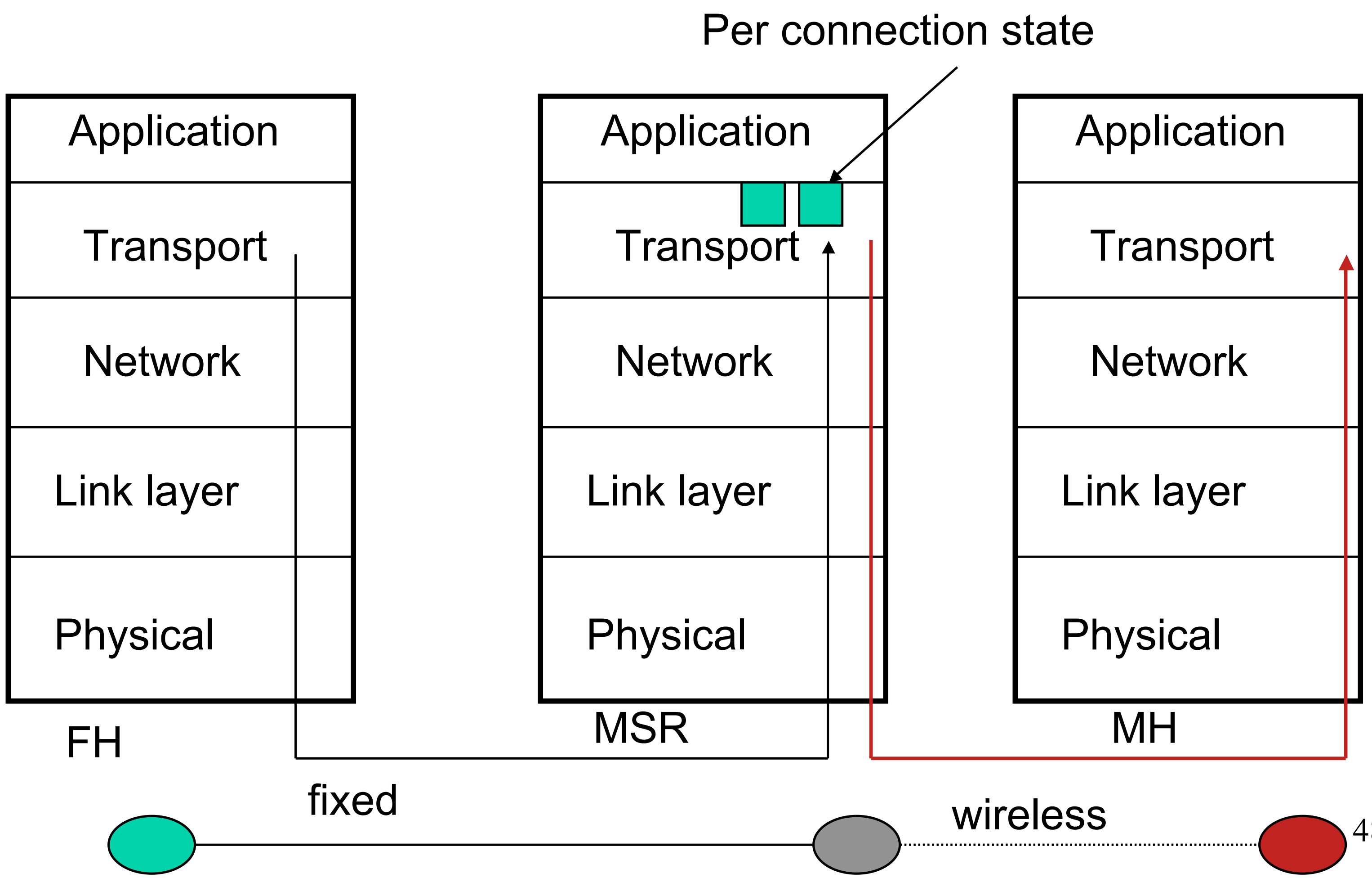
- Uses a split connection
  - End-to-end connection is broken into one connection for the wired part and another connection for the wireless part
  - Wireless part of the TCP can be optimized for wireless
  - TCP optimization close to where it is needed



Split connection approach

- Split connection results in two independent flows. Hence, independent decision of what do with packet loss
- On wireless, loss ➔try harder
- On fixed, loss ➔ backoff
- Tune TCP stack to get this behavior

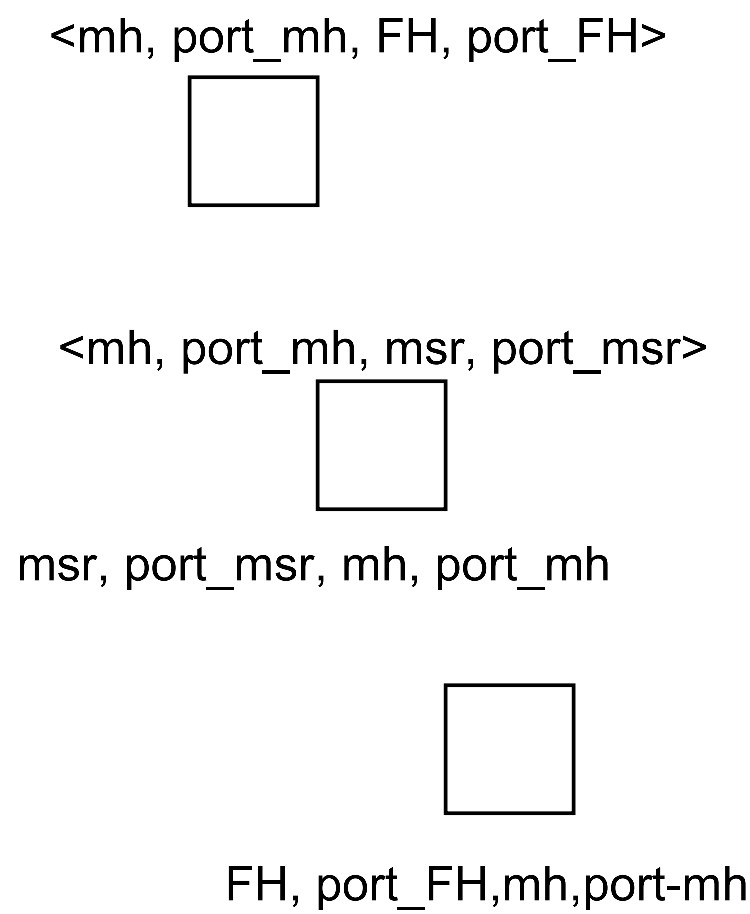
Transport level solution



Establishing TCP connections

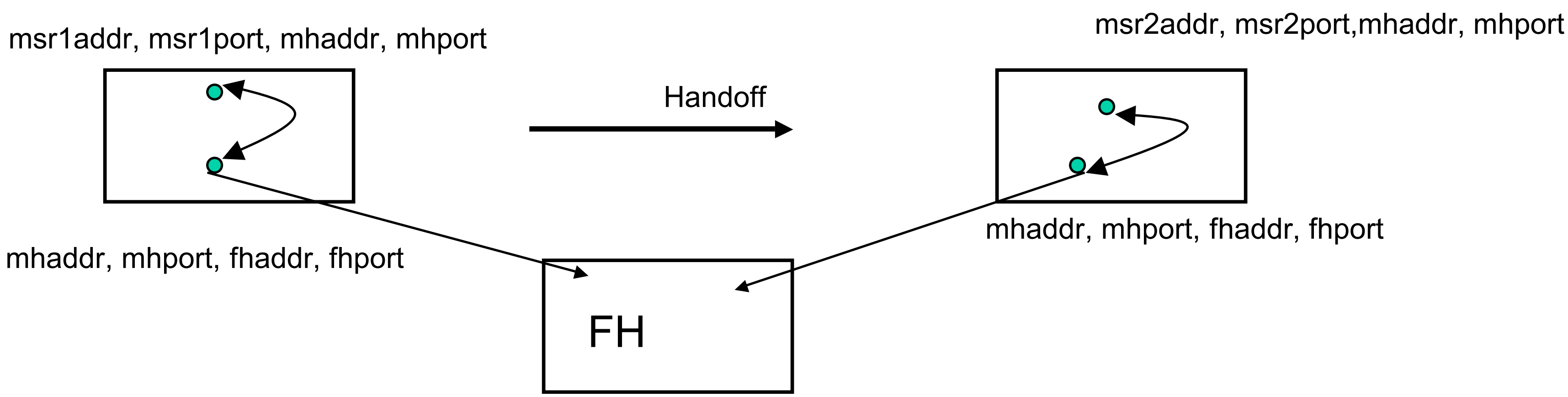
- FH should see a TCP connection coming from MH and not from MSR
- MH should open a TCP connection to FH and should not be aware that the connection is going to MSR
- MH has a I-TCP library that intercepts connection requests and opens a connection to MSR
- MSR opens a connection to FH

  but with the <address of MH and port #> sent by FH



I-TCP handoff

- When a MH moves to a new location, it establishes a connection with the new MSR
- The new MSR get the TCP state from the old MSR and continues the TCP connection



I-TCP features

- Hides packet loss due to wireless from sender
- Wireless TCP can be independently optimized
- Good performance in case of wide-area networks
- Retransmission occurs only on the bad link
- Faster recovery due to relatively short RTT for wireless link
- Handoff requires state transfer
- Buffer space needed, extra copying at MSR
- End-to-end semantics violation needs to be augmented by application level actions
- Base station (MSR) failure may cause loss of TCP state

I-TCP : Advantages

- BS-MH connection can be optimized independent of FH-BS connection
  – Different flow / error control on the two connections
- Local recovery of errors
  – Faster recovery due to relatively shorter RTT on wireless link
- Good performance achievable using appropriate BS-MH protocol
  – Standard TCP on BS-MH performs poorly when multiple packet losses occur per window (timeouts can occur on the BS-MH connection, stalling during the timeout interval)
  – Selective acks improve performance for such cases

I-TCP disadvantages

- End-to-end semantics violated
  - ack may be delivered to sender, before data delivered to the receiver
  - May not be a problem for applications that do not rely on TCP for the end-to-end semantics
- BS retains hard state BS failure can result in loss of data (unreliability)
  - Acked packets from BS, sender assumes that packet actually reached the receiver

GPRSweb

- Study of HTTP over GPRS
  - Problems with wired compared to wireless links
- Similar approach : proxies
  - Operate at a level higher than TCP
  - Adds proxies on both the client and server

GPRS

I-TCP : Disadvantages

- BS retains per-connection state
- Bufferred packets at BS must be transferred to new BS
- Hand-off latency increases due to state transfer
- Buffer space needed at BS for each TCP connection
  - BS buffers tend to get full, when wireless link slower (one window worth of data on wired connection could be stored at the base station, for each split connection)
- Extra copying of data at BS
  - copying from FH-BS socket buffer to BS-MH socket buffer
  - increases end-to-end latency
- May not be useful if data and acks traverse different paths (both do not go through the base station)
  - Example: data on a satellite wireless links