Lecture 12: More Wireless

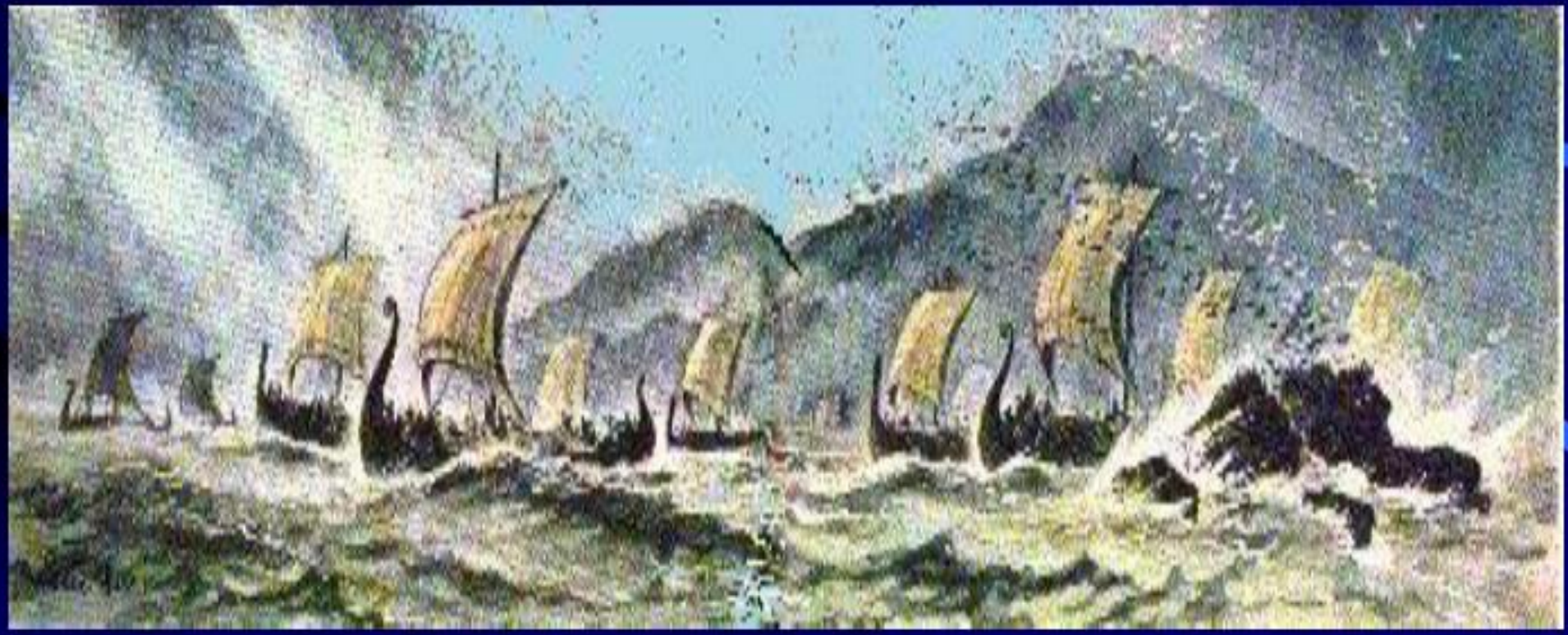
Schedule

- Mid-term 1: graded after thanksgiving
- Mid-term 2: Wed. 12/5
- Project 3:
  - Designing an efficient HTTP crawler
  - Released by 11/21, due 12/11 (Tue), or 12/13 if you use 2-day extension coupon

What We Learned So Far

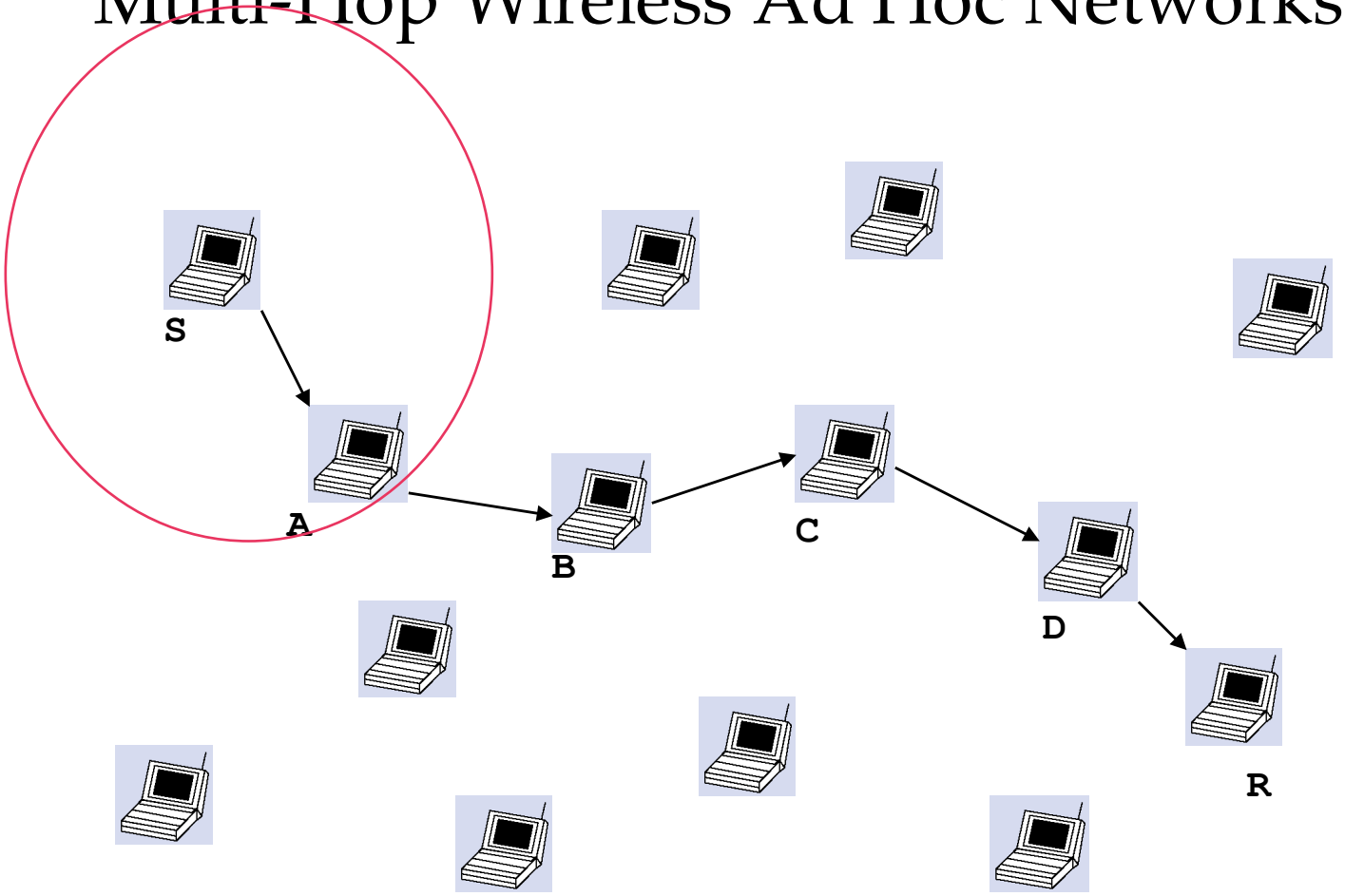
- Wireless is a tricky beast
  - Distributed multiple access problem
  - Hidden terminals
  - Exposed terminals
  - Current protocols sufficient, given overprovisioning
- Today:
  - Multihop even more complicated
  - TCP over Wireless?

TCP Over Wireless Ad-hoc Networks



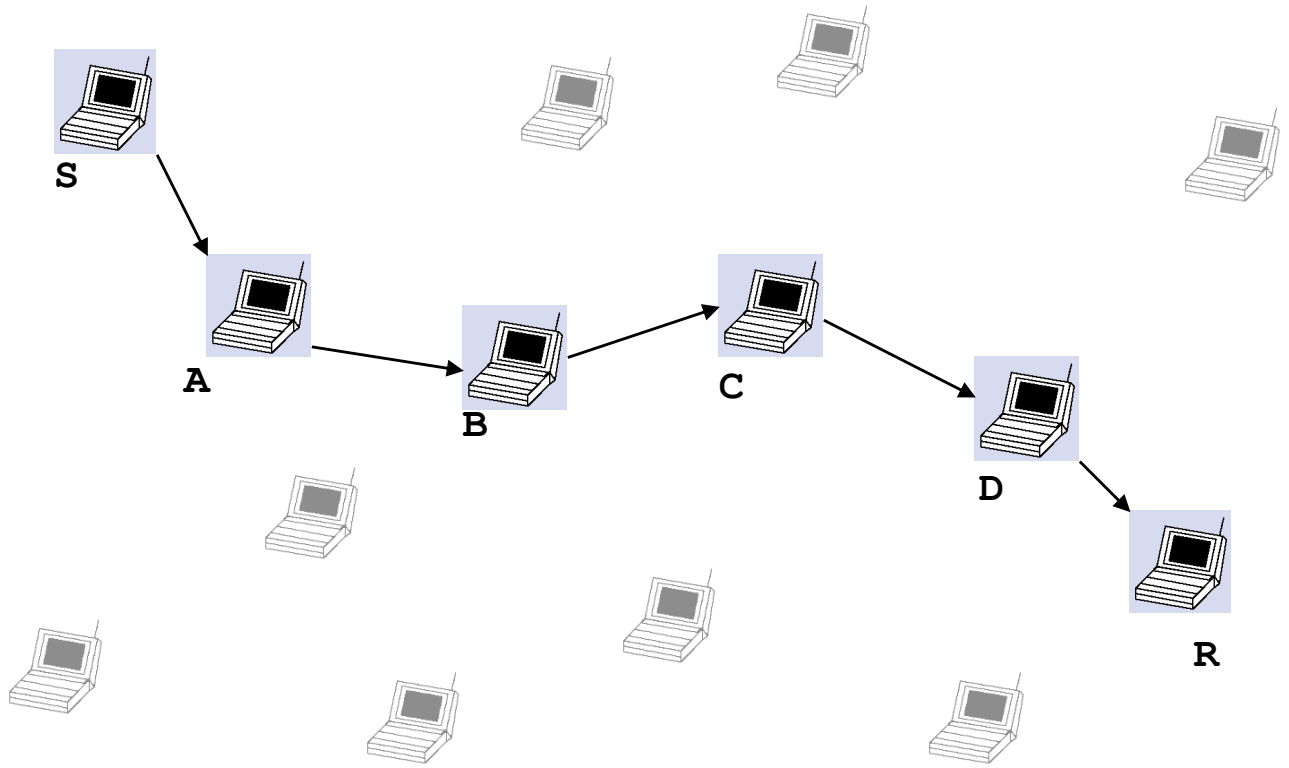
PART 4: MULTI-HOP WIRELESS ROUTING

Multi-Hop Wireless Ad Hoc Networks

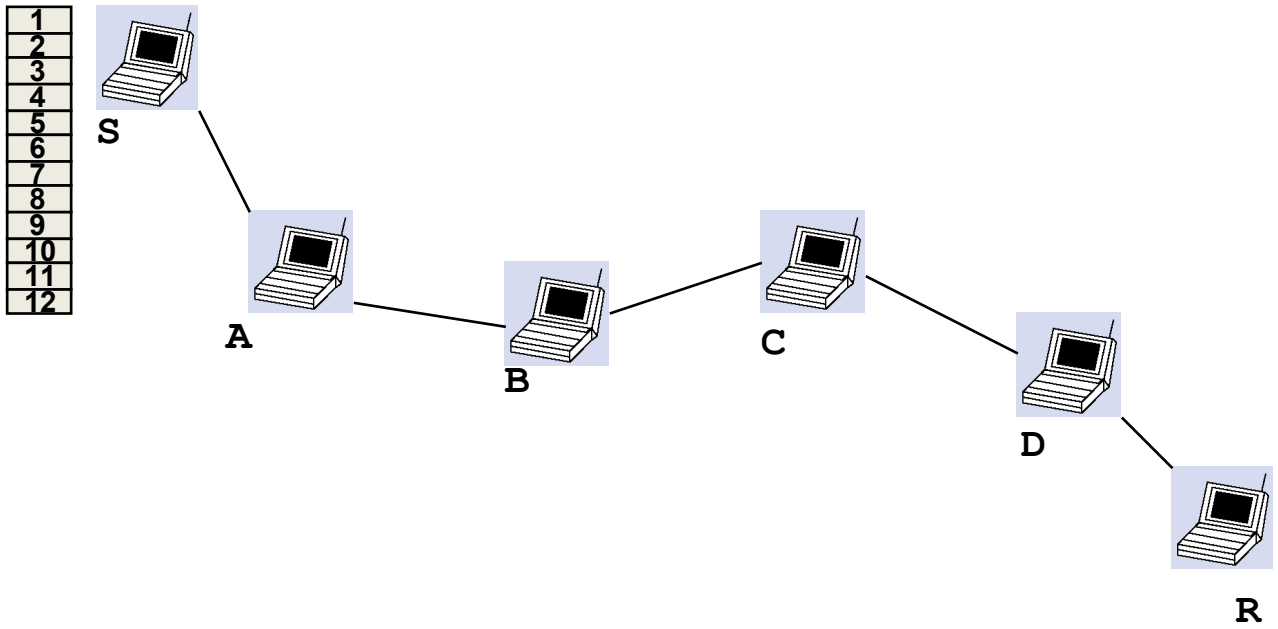


Courtesy of Tianbo Kuang and Carey Williamson University of Calgary)

Multi-Hop Wireless Ad Hoc Networks

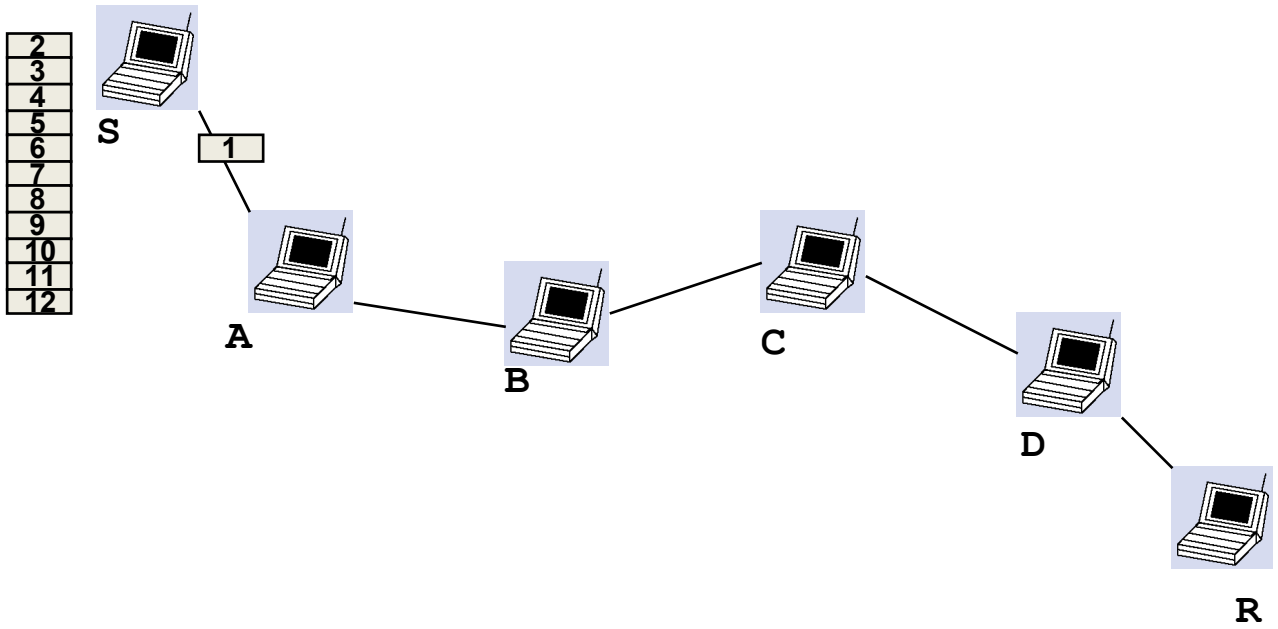


Multi-Hop Wireless Ad Hoc Networks

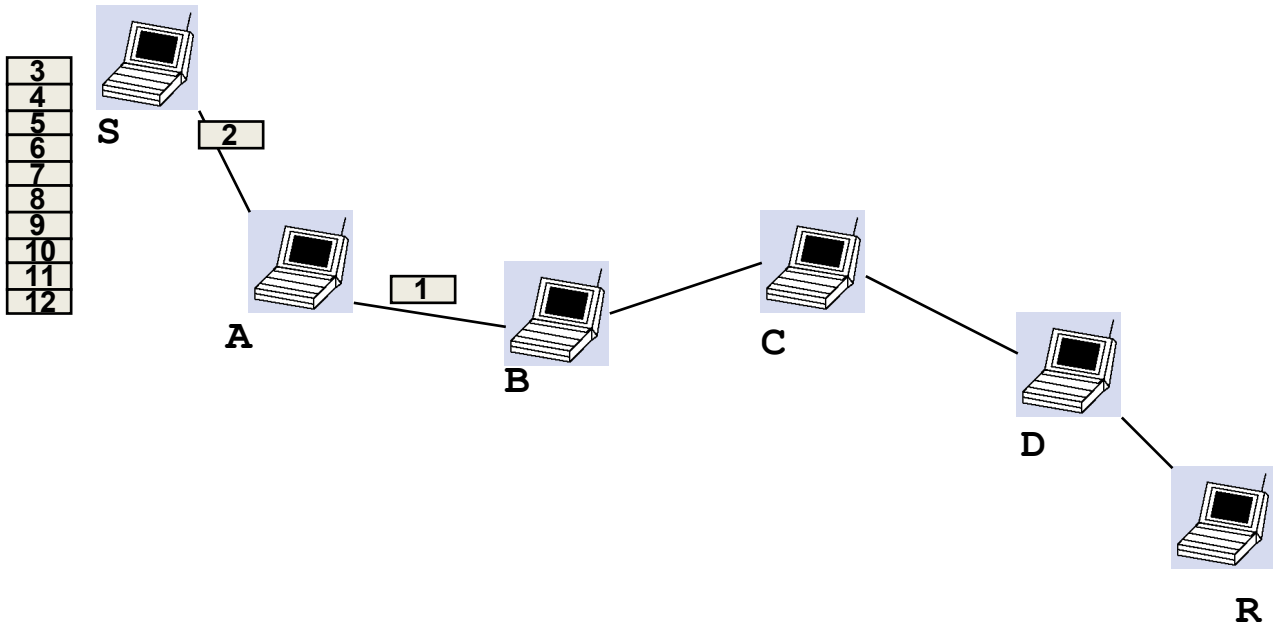


(Assume ideal world…)

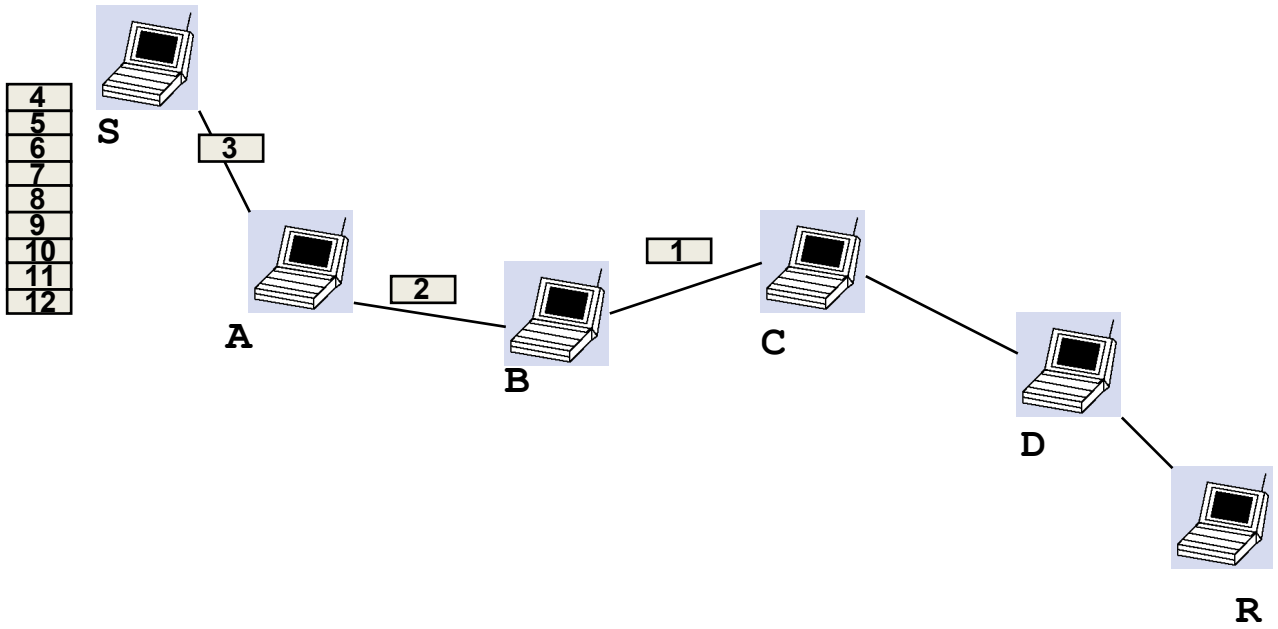
Multi-Hop Wireless Ad Hoc Networks



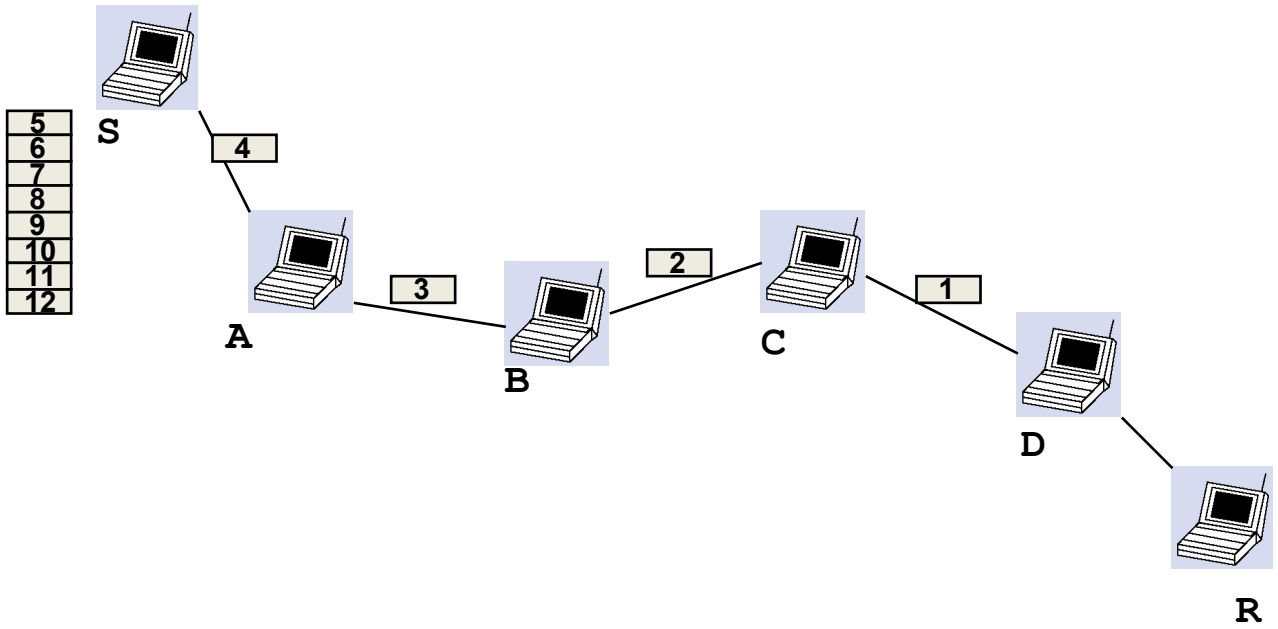
Multi-Hop Wireless Ad Hoc Networks



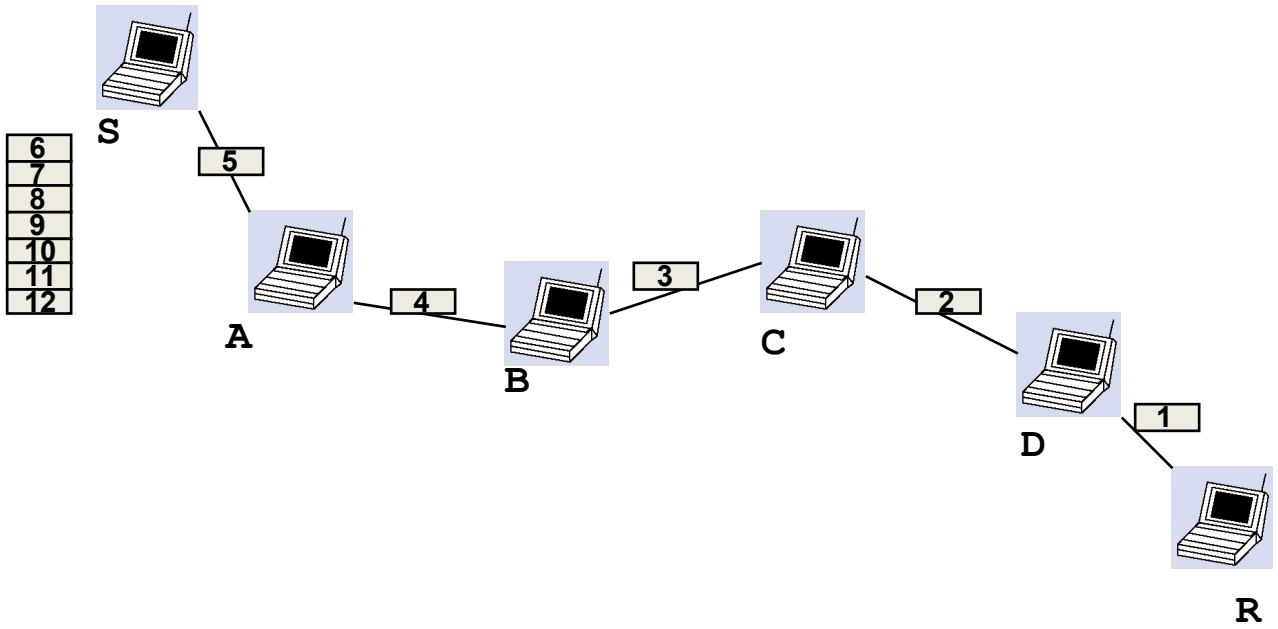
Multi-Hop Wireless Ad Hoc Networks



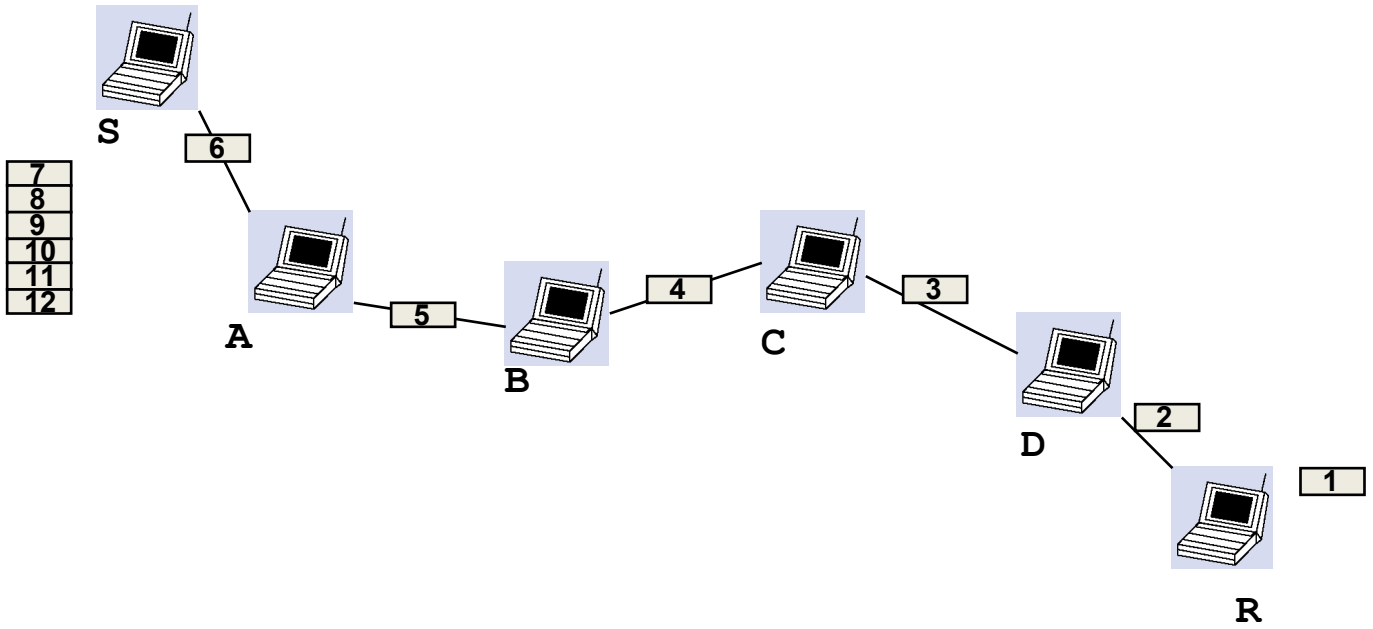
Multi-Hop Wireless Ad Hoc Networks



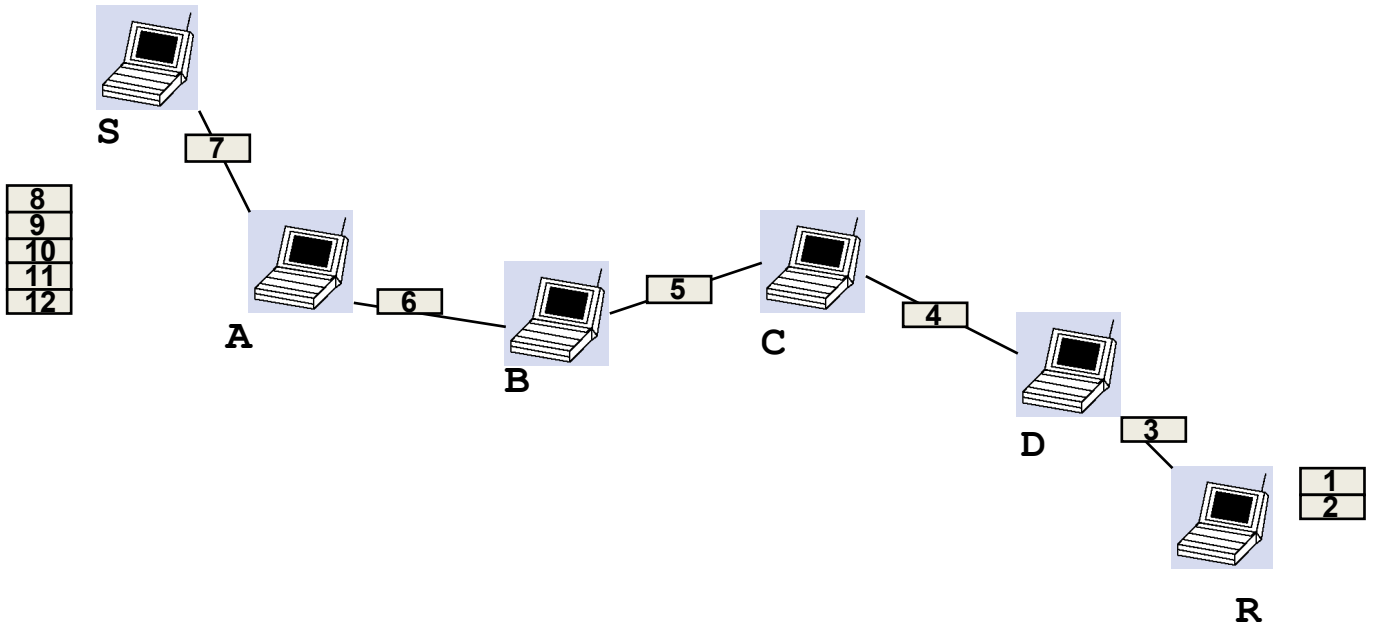
Multi-Hop Wireless Ad Hoc Networks



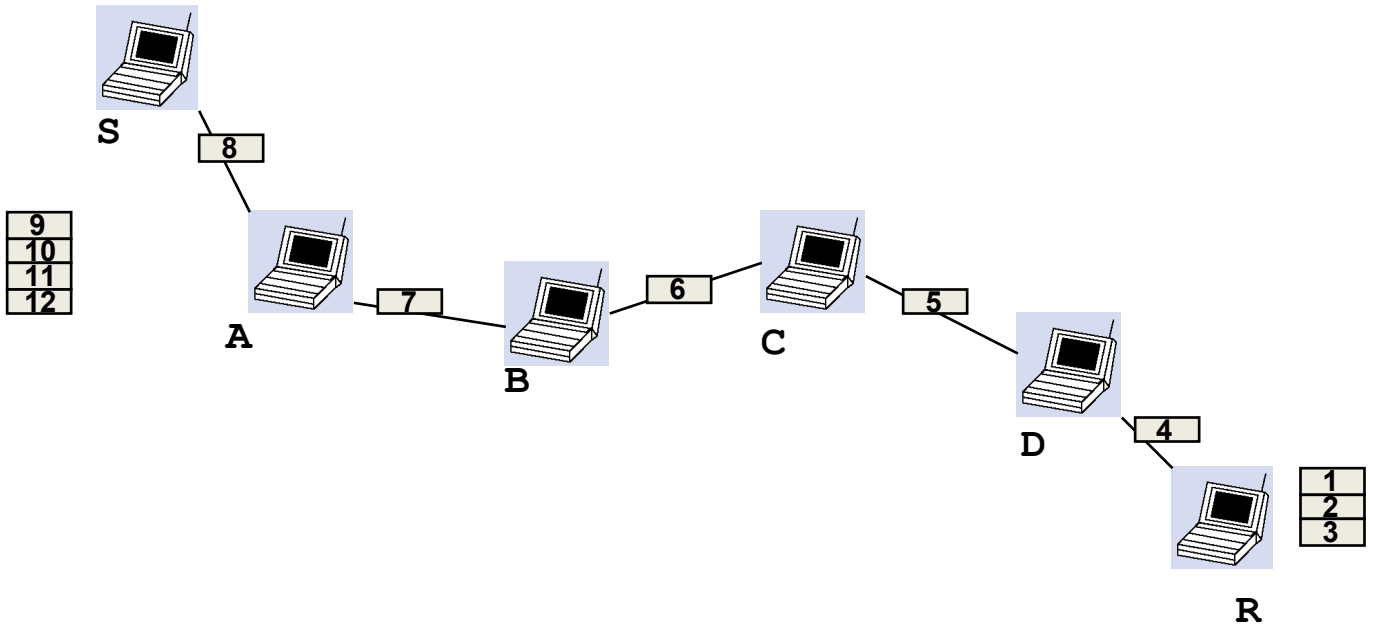
Multi-Hop Wireless Ad Hoc Networks



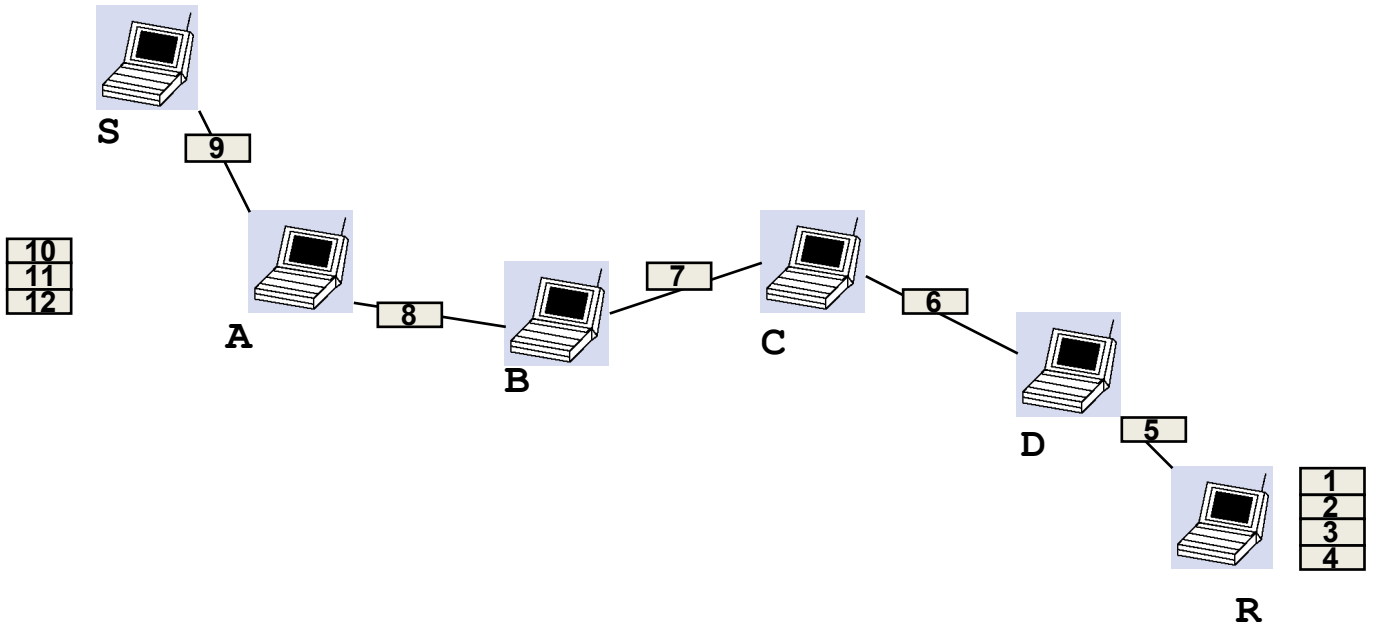
Multi-Hop Wireless Ad Hoc Networks



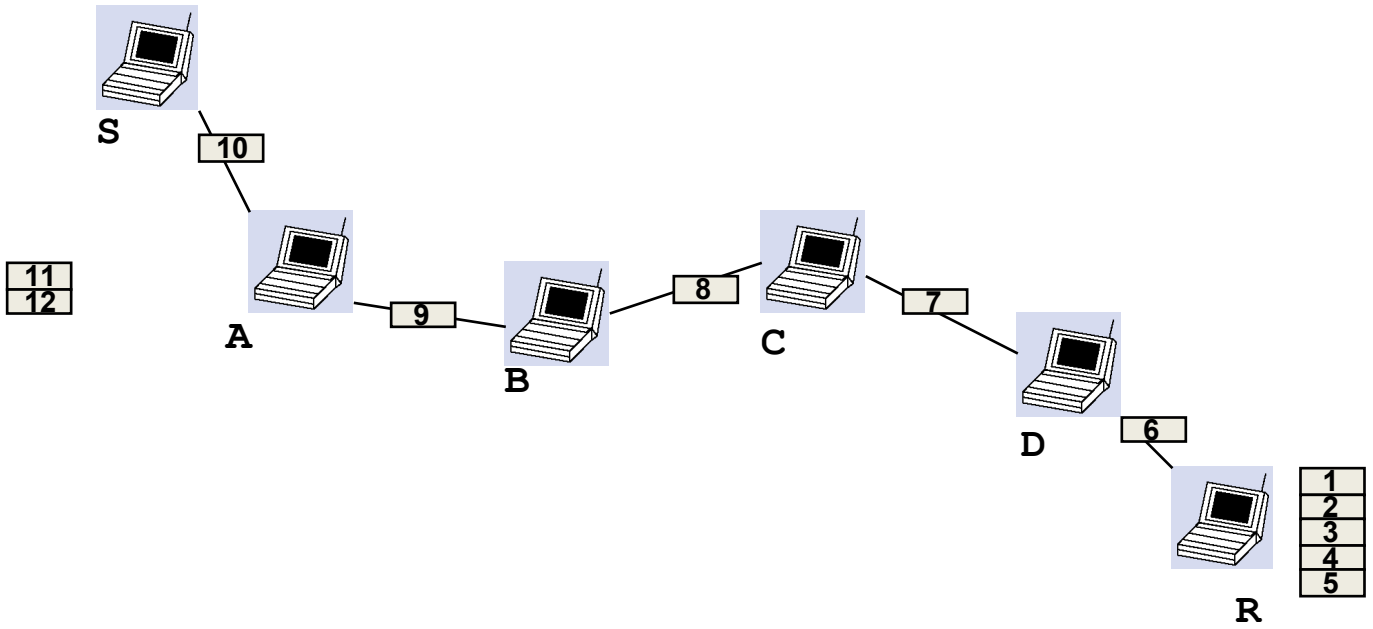
Multi-Hop Wireless Ad Hoc Networks



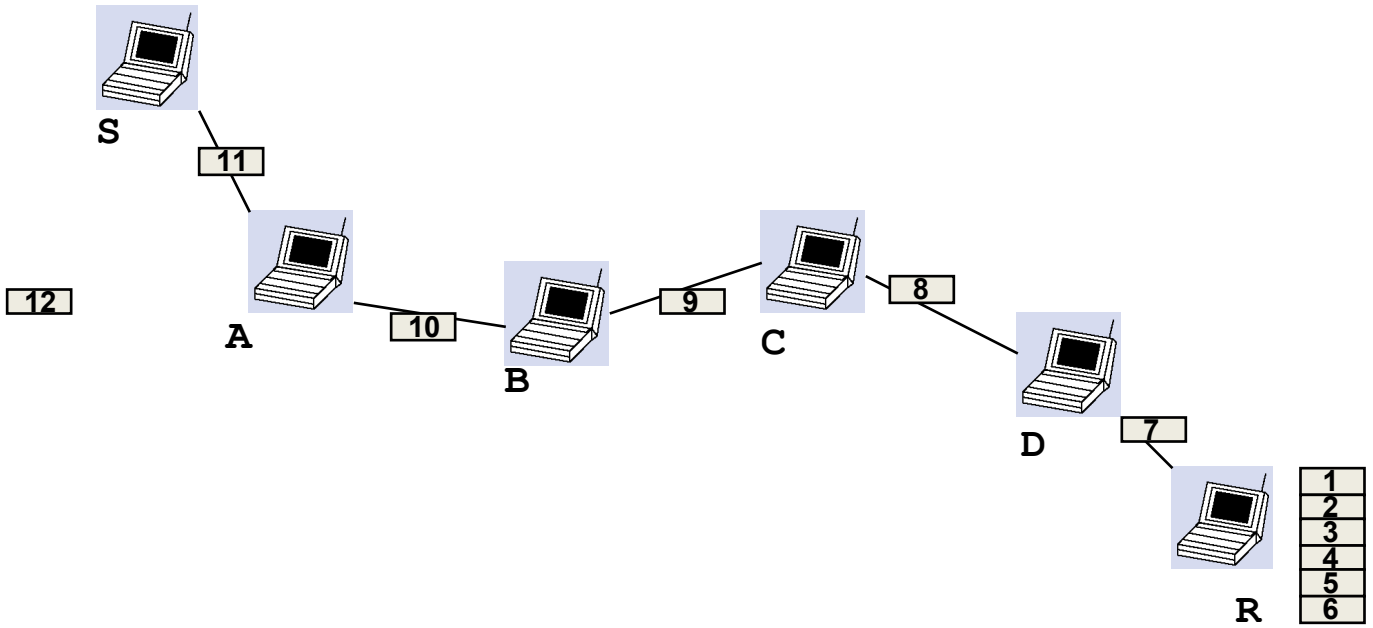
Multi-Hop Wireless Ad Hoc Networks



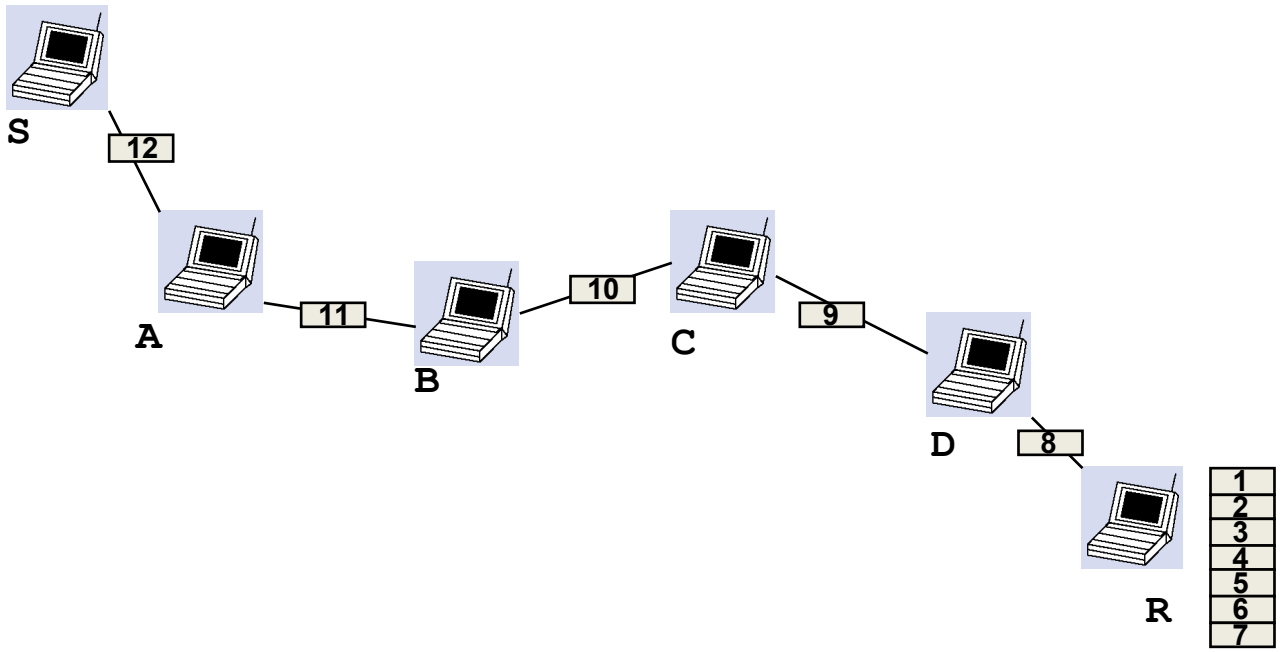
Multi-Hop Wireless Ad Hoc Networks



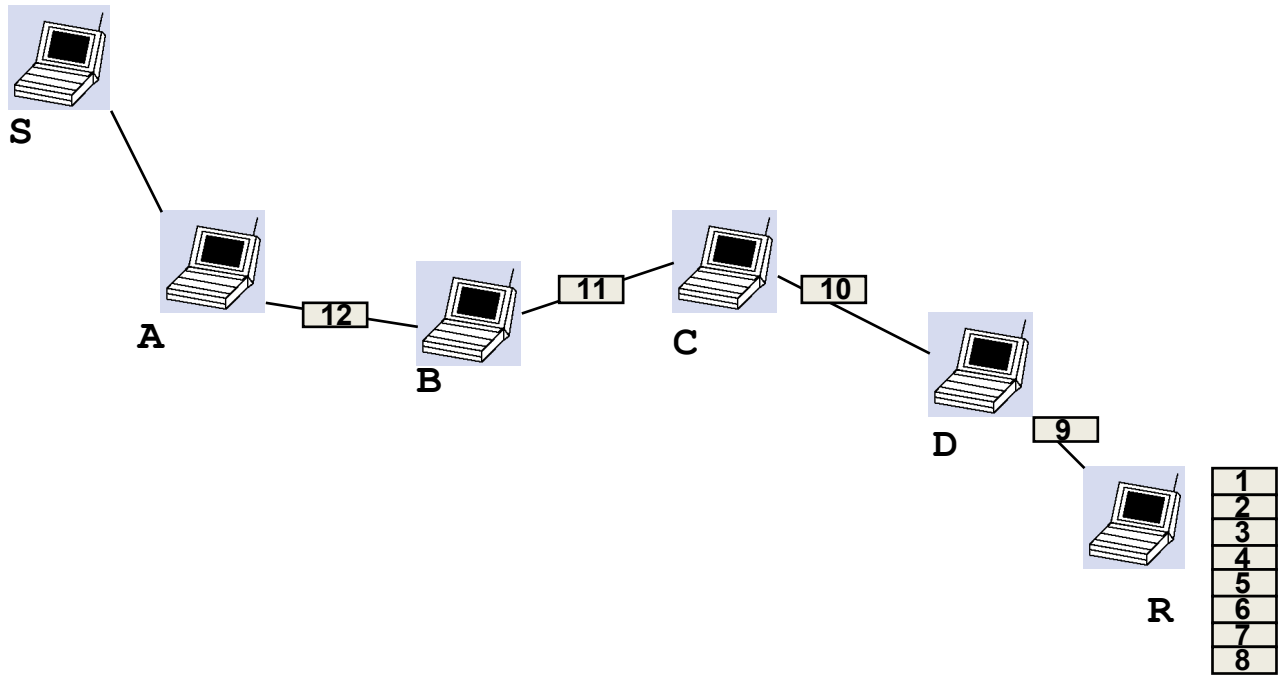
Multi-Hop Wireless Ad Hoc Networks



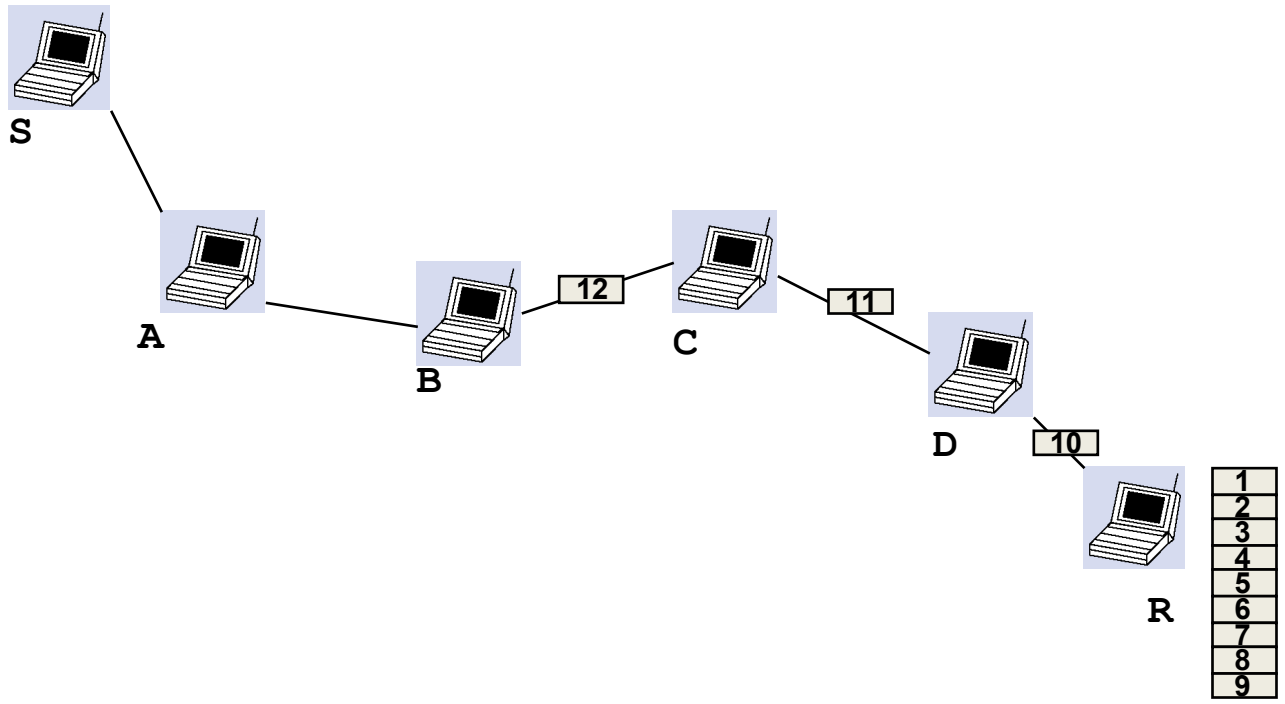
Multi-Hop Wireless Ad Hoc Networks



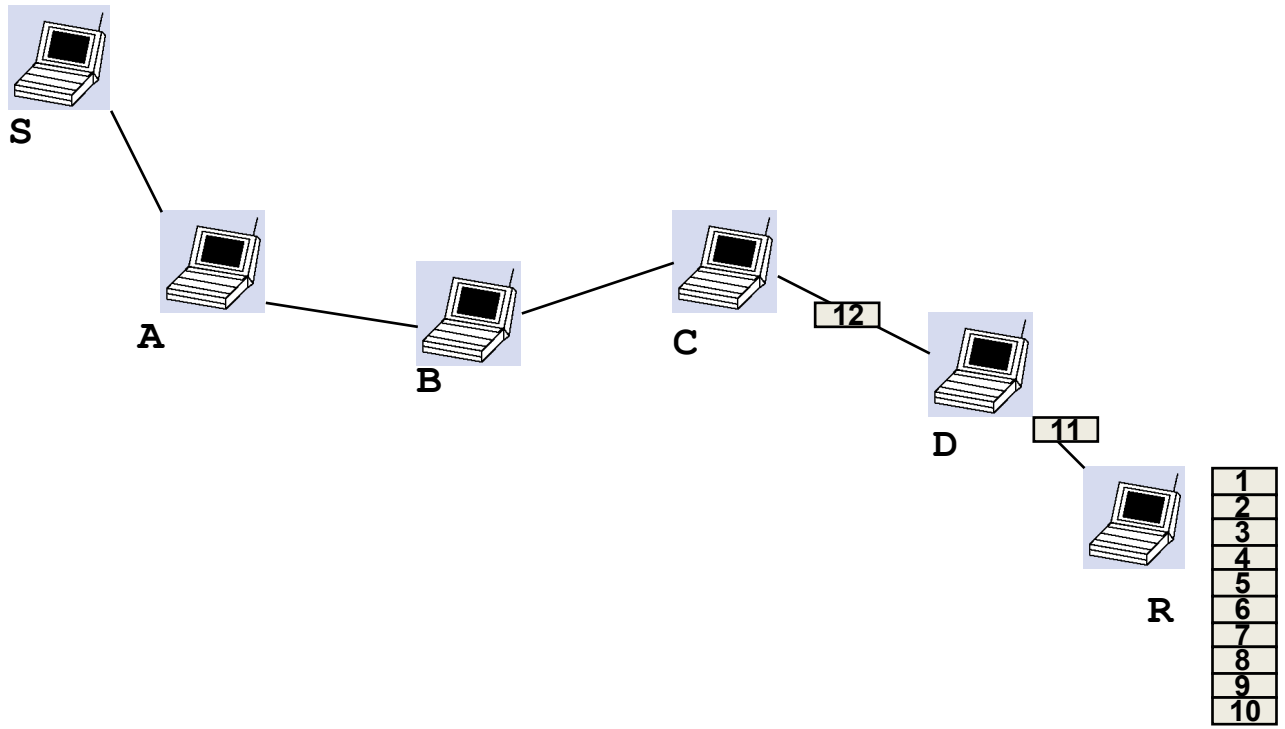
Multi-Hop Wireless Ad Hoc Networks



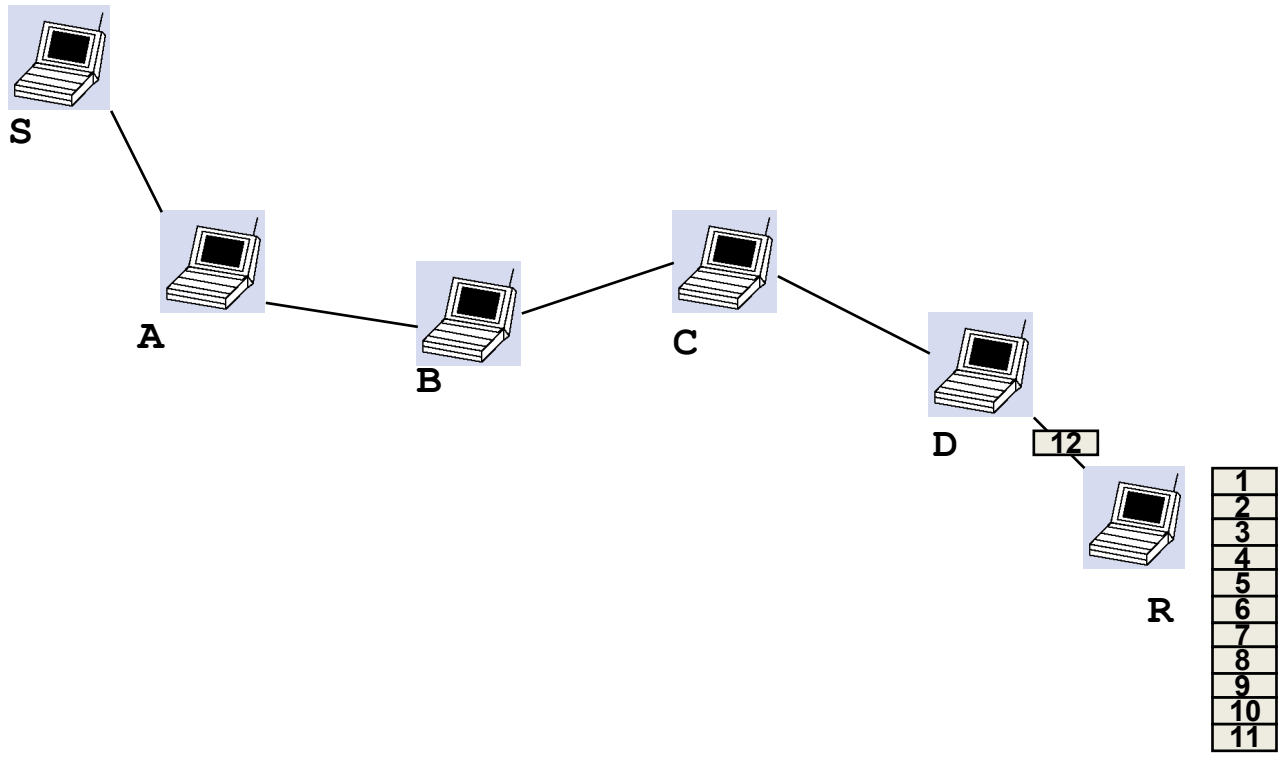
Multi-Hop Wireless Ad Hoc Networks



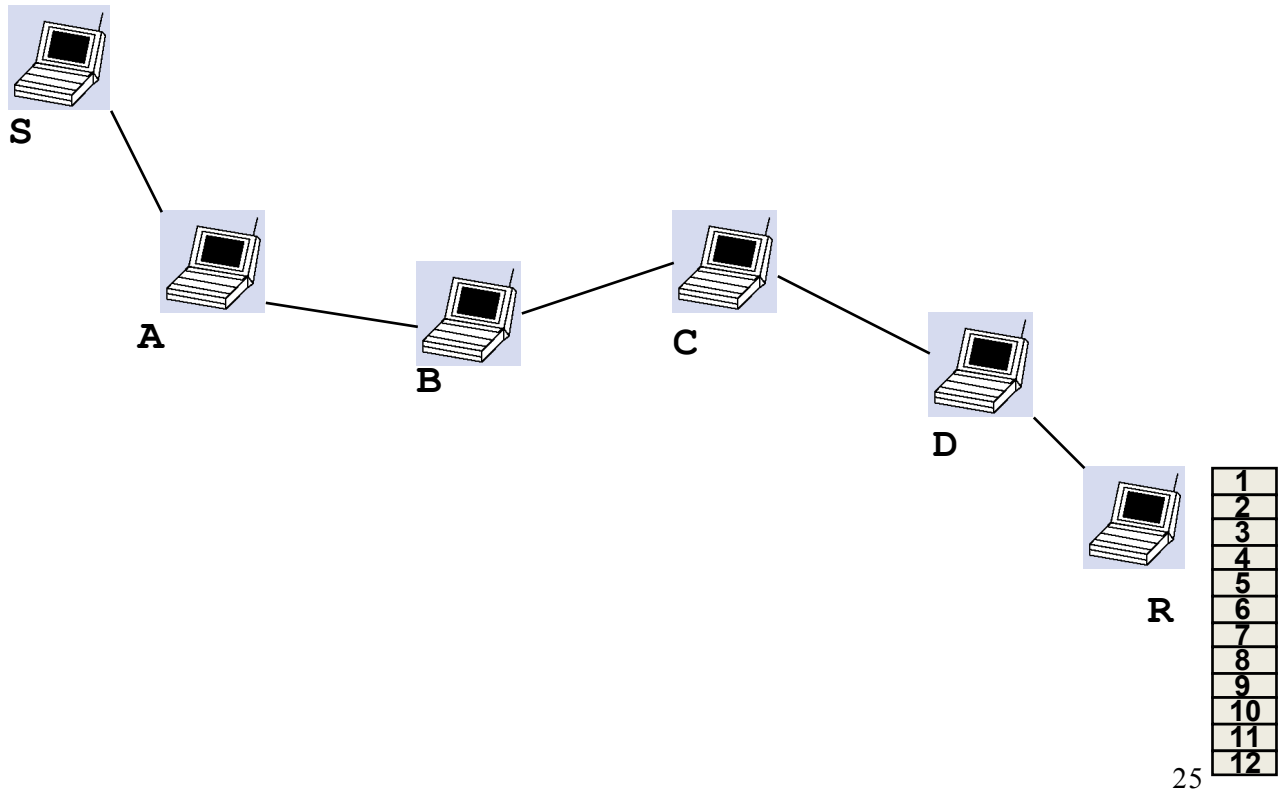
Multi-Hop Wireless Ad Hoc Networks



Multi-Hop Wireless Ad Hoc Networks



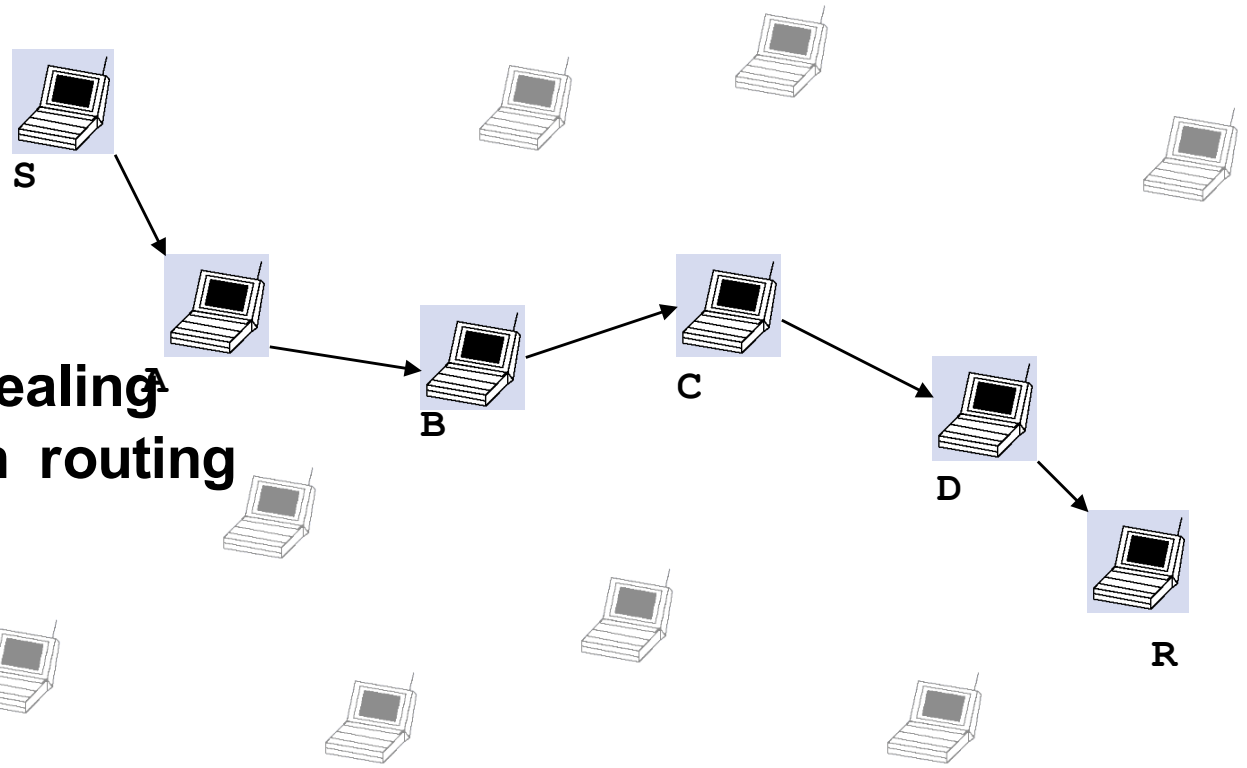
Multi-Hop Wireless Ad Hoc Networks



Wireless Multihop Networks

- Vehicular Networks
  - Delay Tolerant (batch) sending over several hops carry data to a base station
- Common in Sensor Network for periodically transmitting data
  - Infrastructure Monitoring
    - E.g., structural health monitoring of the Golden Gate Bridge

The end of phone companies & ISPs?

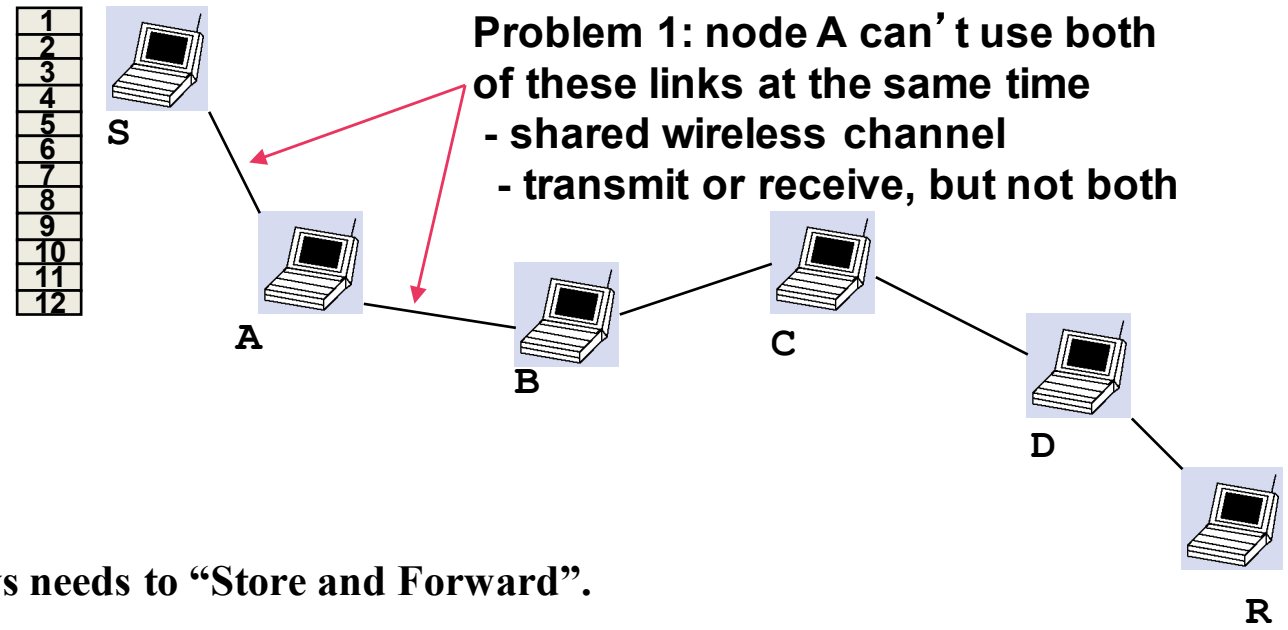


- **Self healing**
- **Multipath routing**

What Do YOU Think Really Happens?

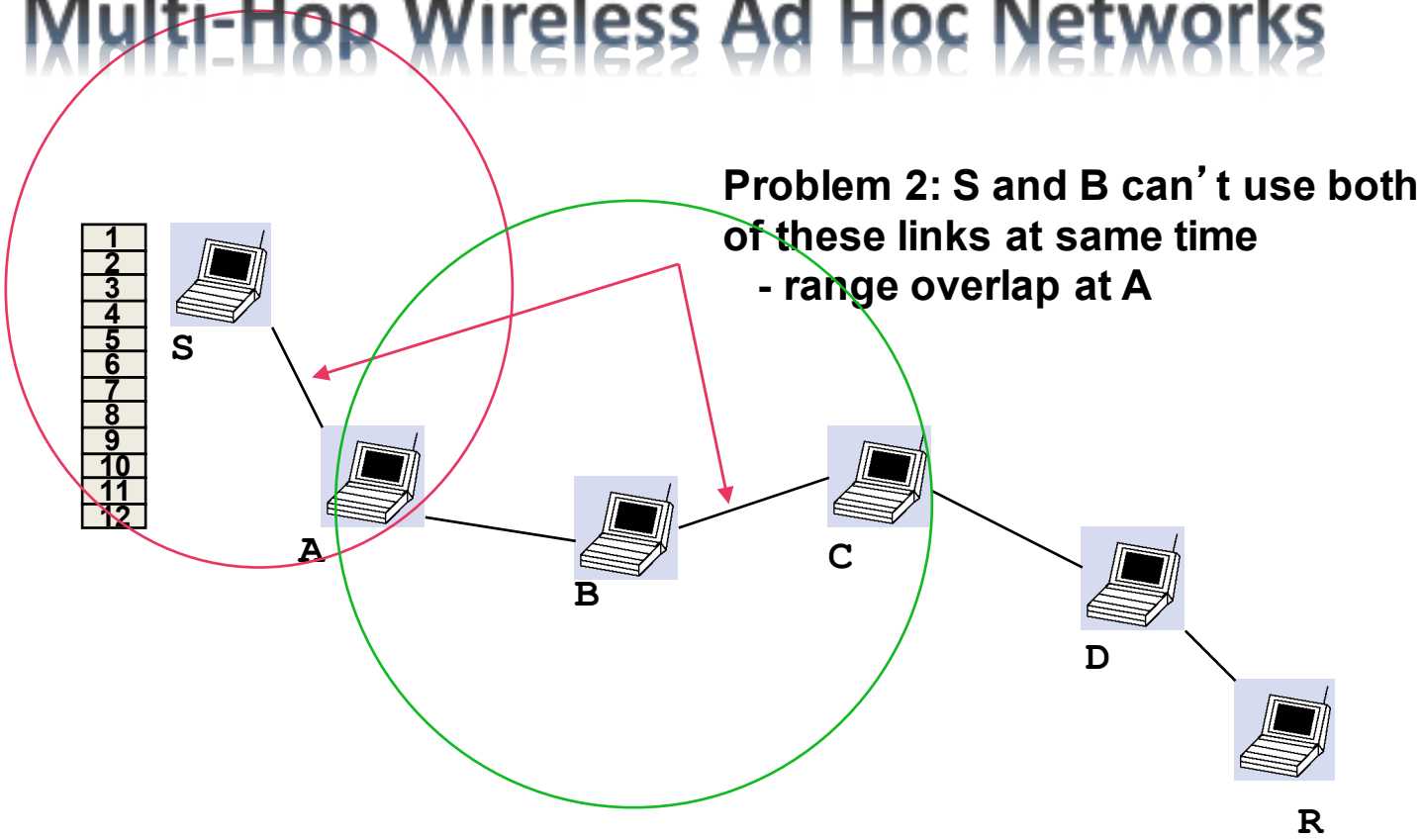
Multi-Hop Wireless Ad Hoc Networks

(Reality check…)



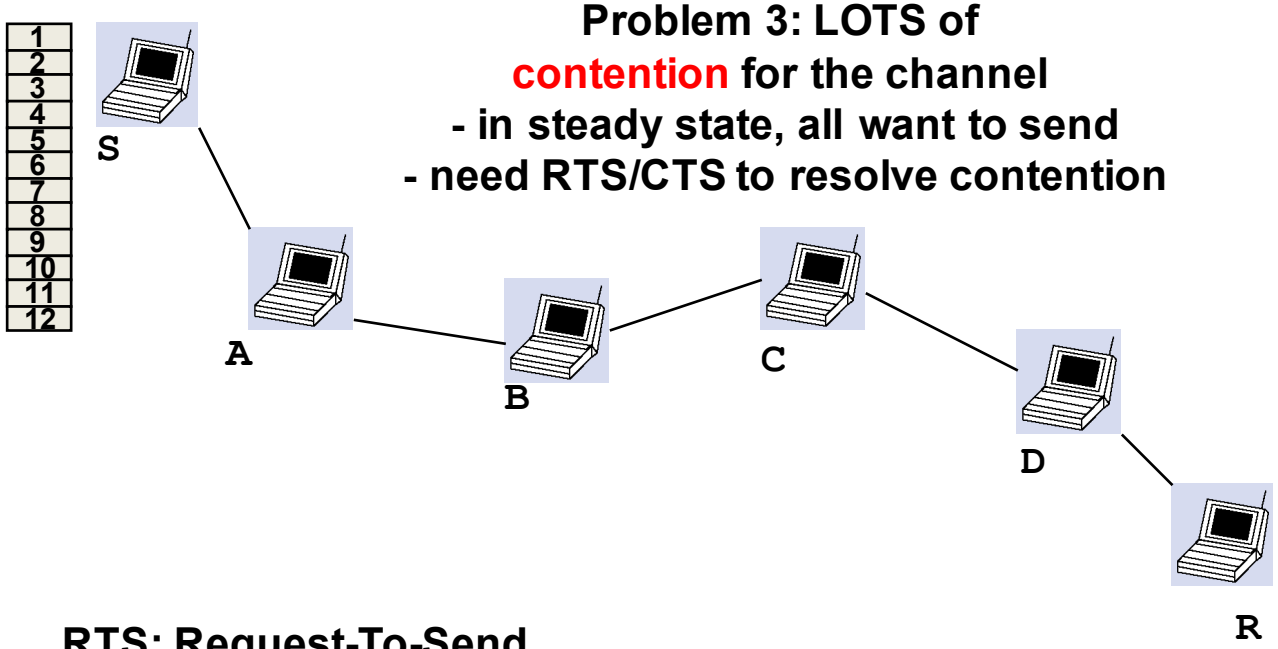
Relays needs to "Store and Forward".

Multi-Hop Wireless Ad Hoc Networks



Problem 2: S and B can't use both of these links at same time
- range overlap at A

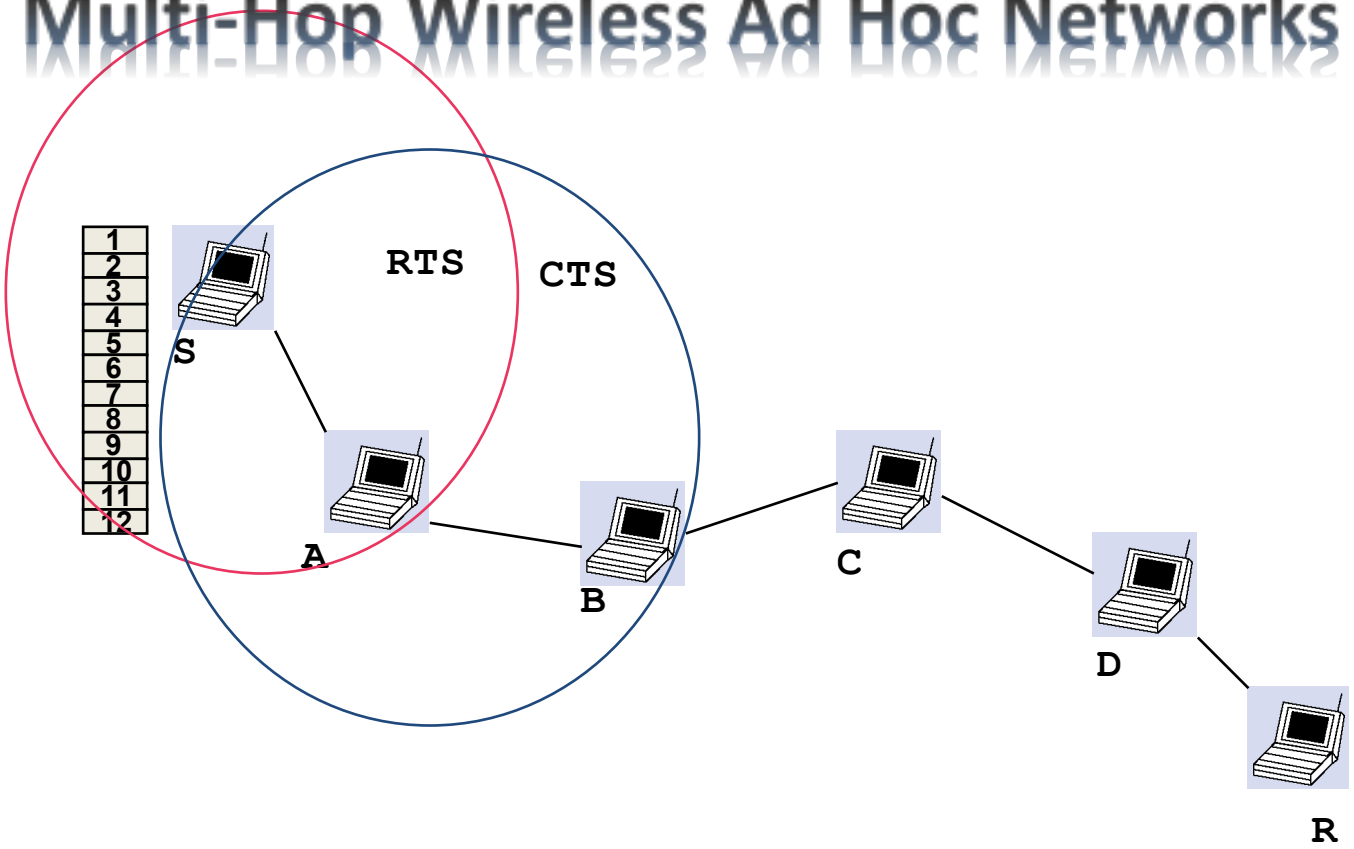
Multi-Hop Wireless Ad Hoc Networks



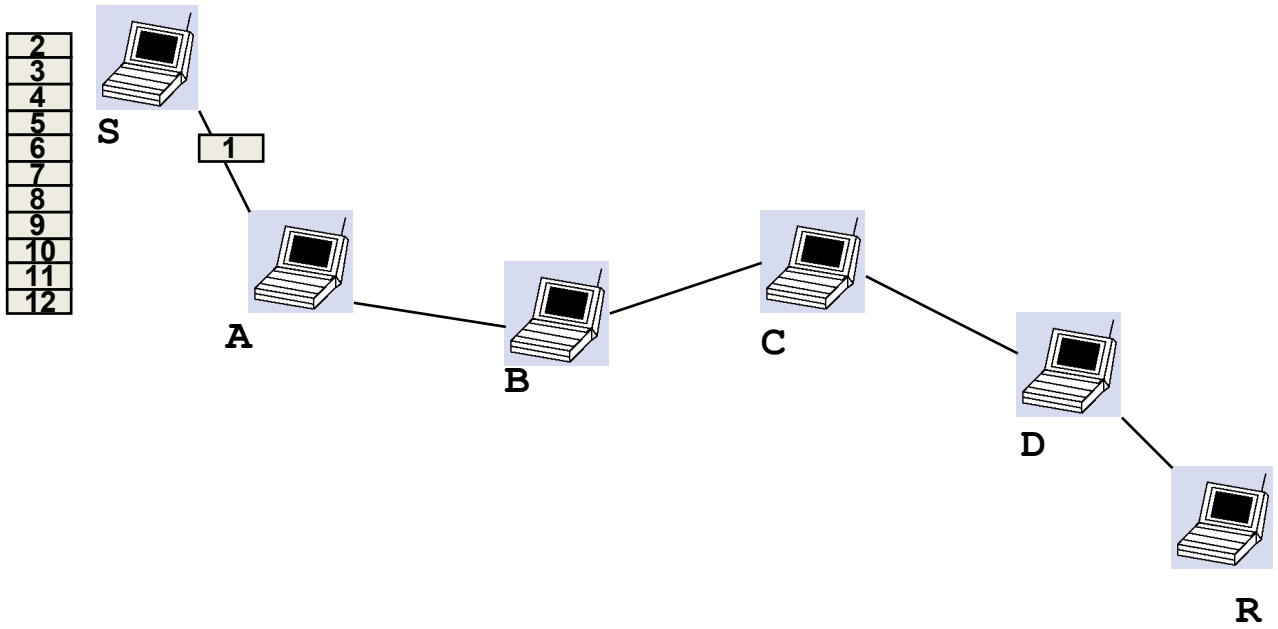
Problem 3: LOTS of
contention for the channel
- in steady state, all want to send
- need RTS/CTS to resolve contention

RTS: Request-To-Send
CTS: Clear-To-Send

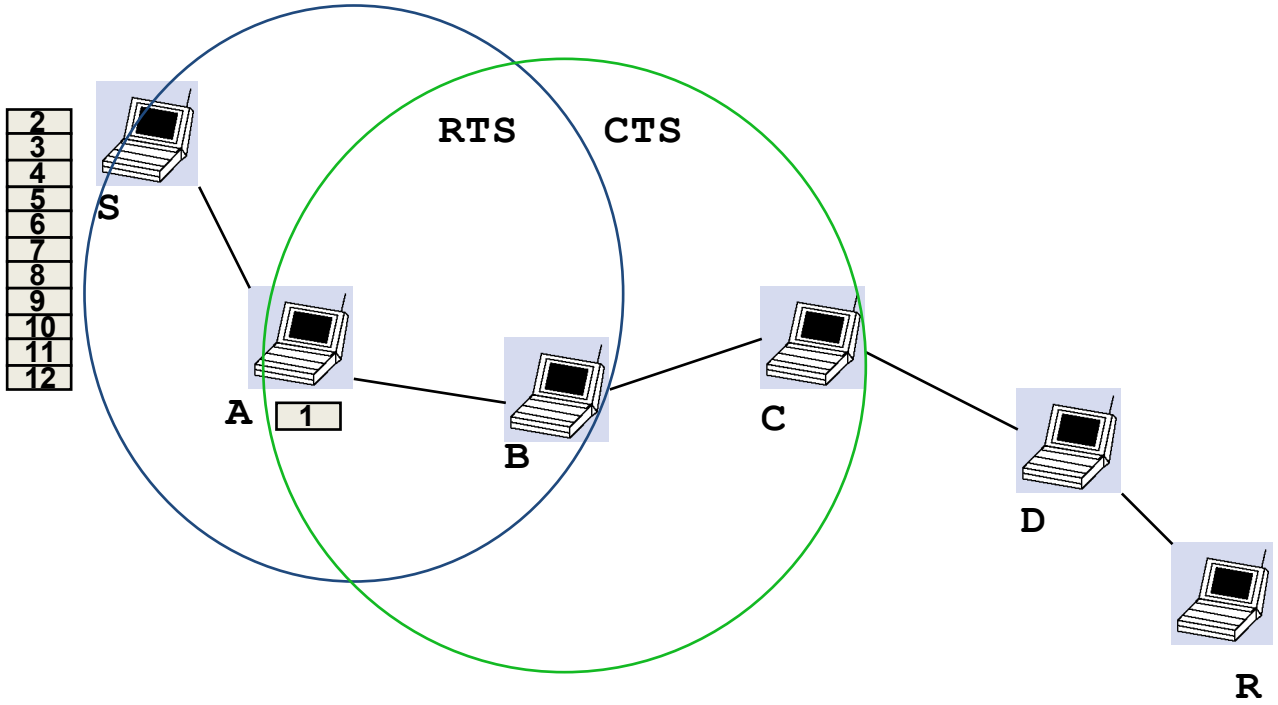
Multi-Hop Wireless Ad Hoc Networks



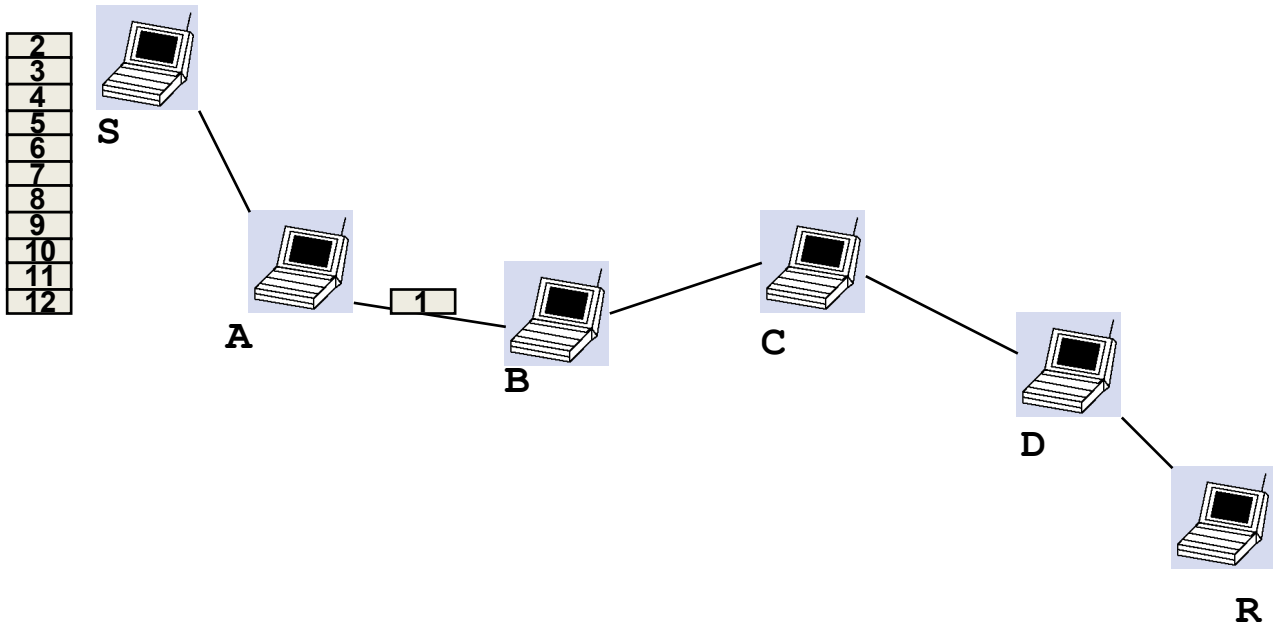
Multi-Hop Wireless Ad Hoc Networks



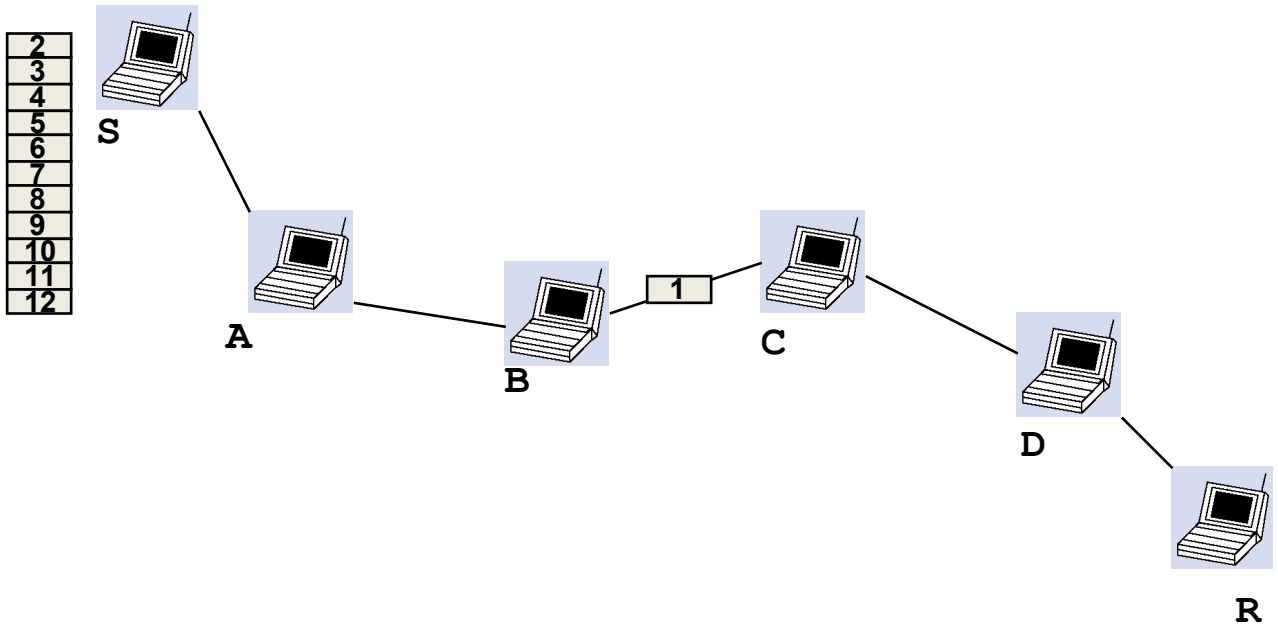
Multi-Hop Wireless Ad Hoc Networks



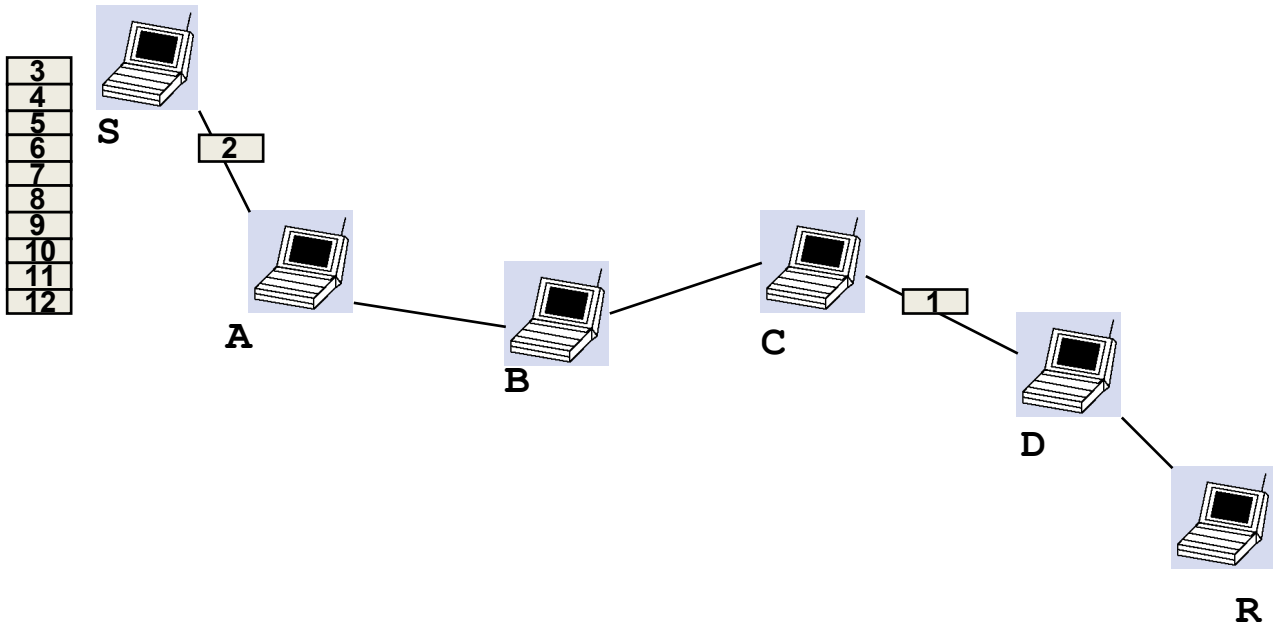
Multi-Hop Wireless Ad Hoc Networks



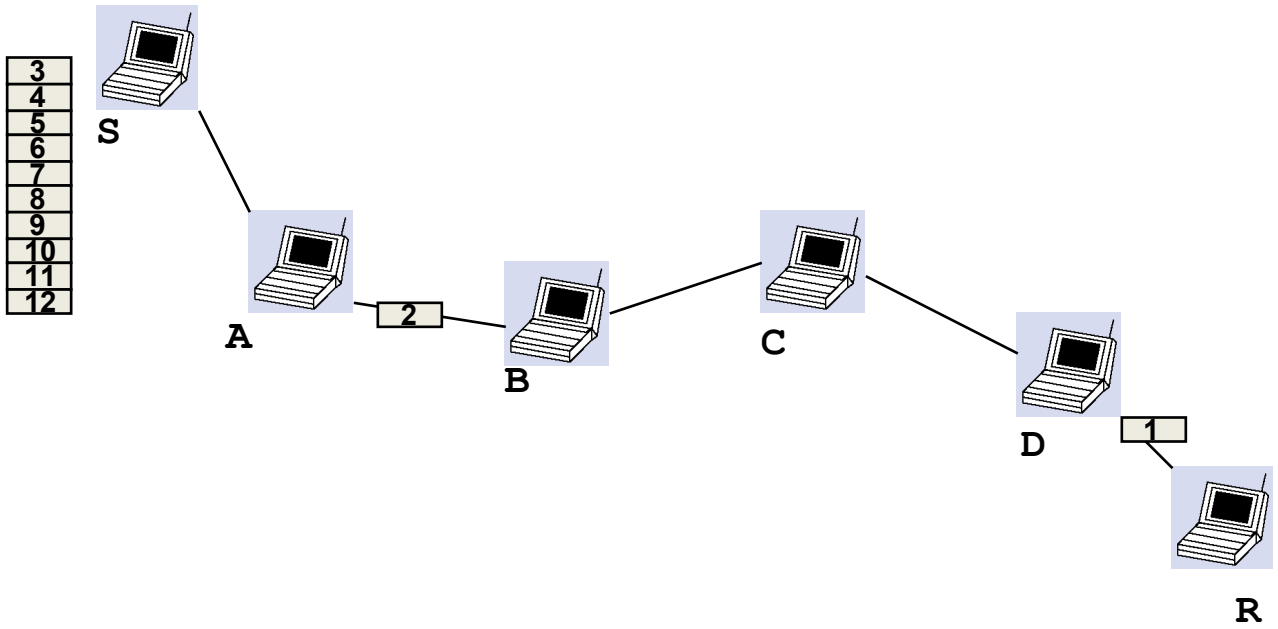
Multi-Hop Wireless Ad Hoc Networks



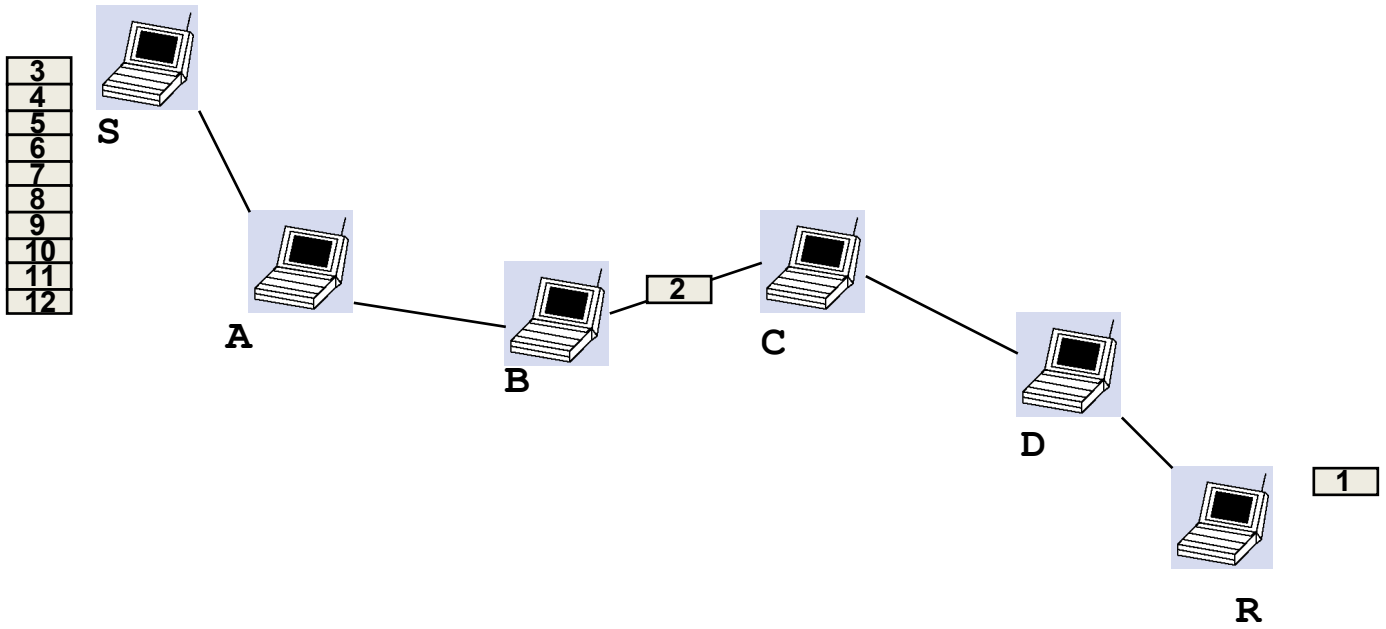
Multi-Hop Wireless Ad Hoc Networks



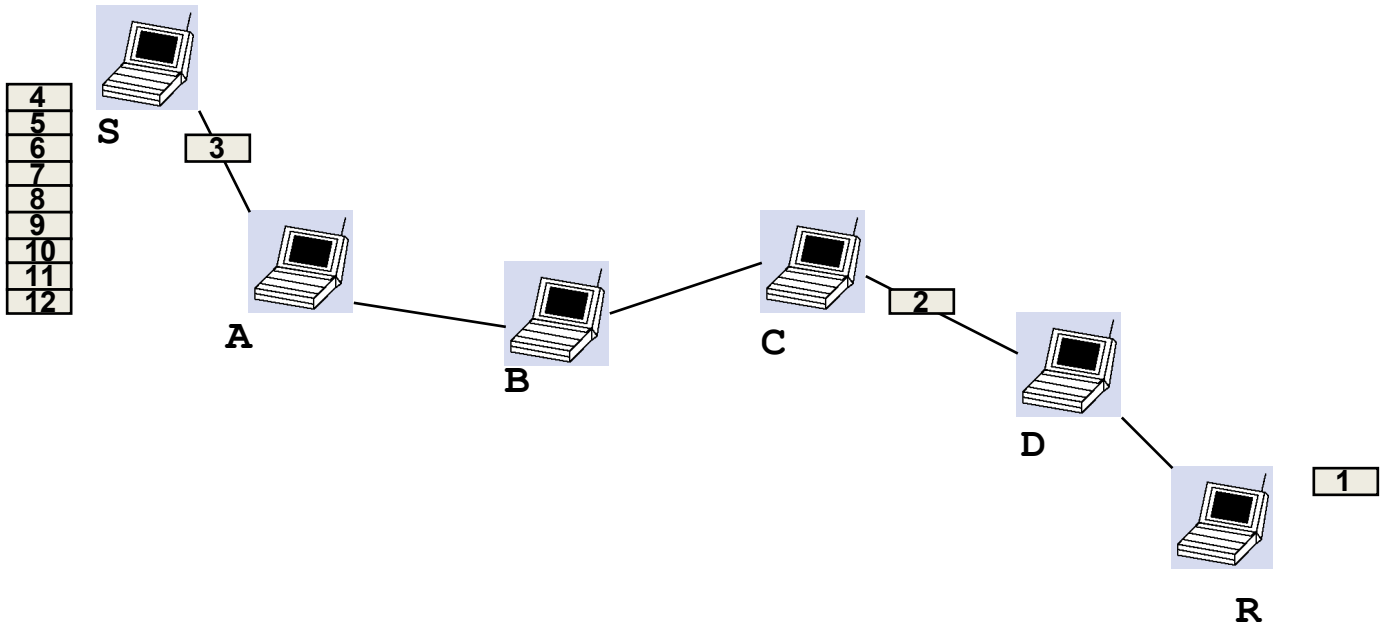
Multi-Hop Wireless Ad Hoc Networks



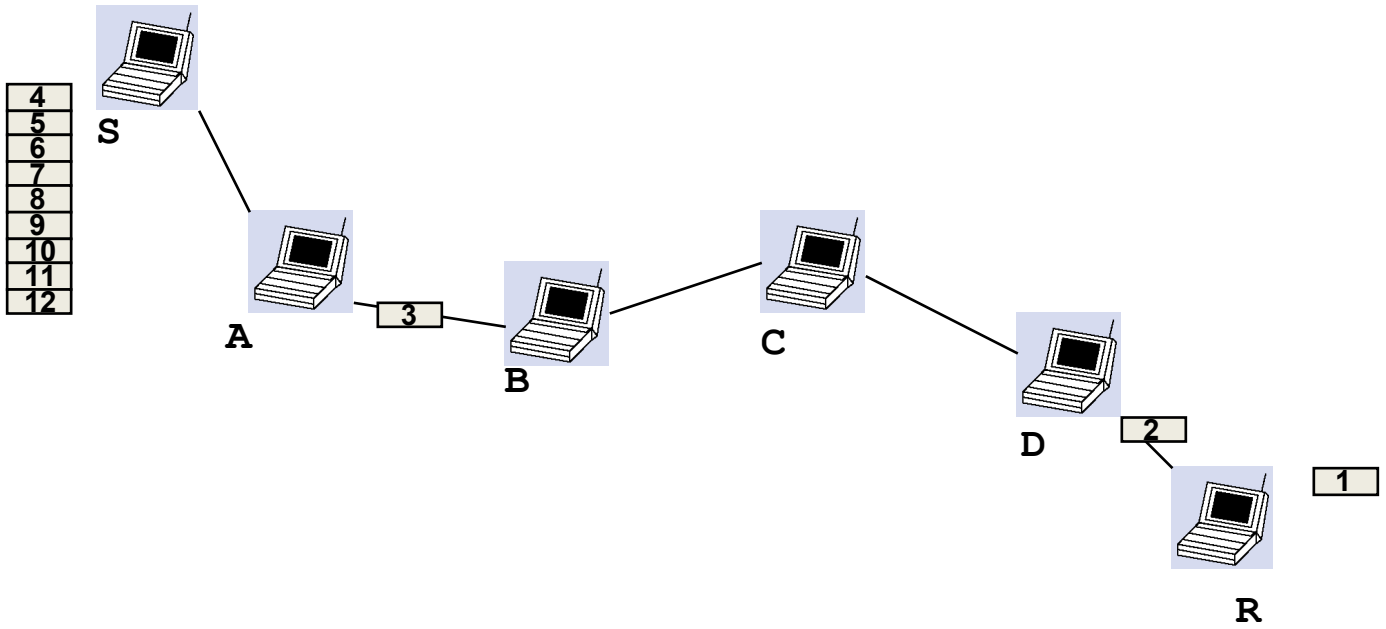
Multi-Hop Wireless Ad Hoc Networks



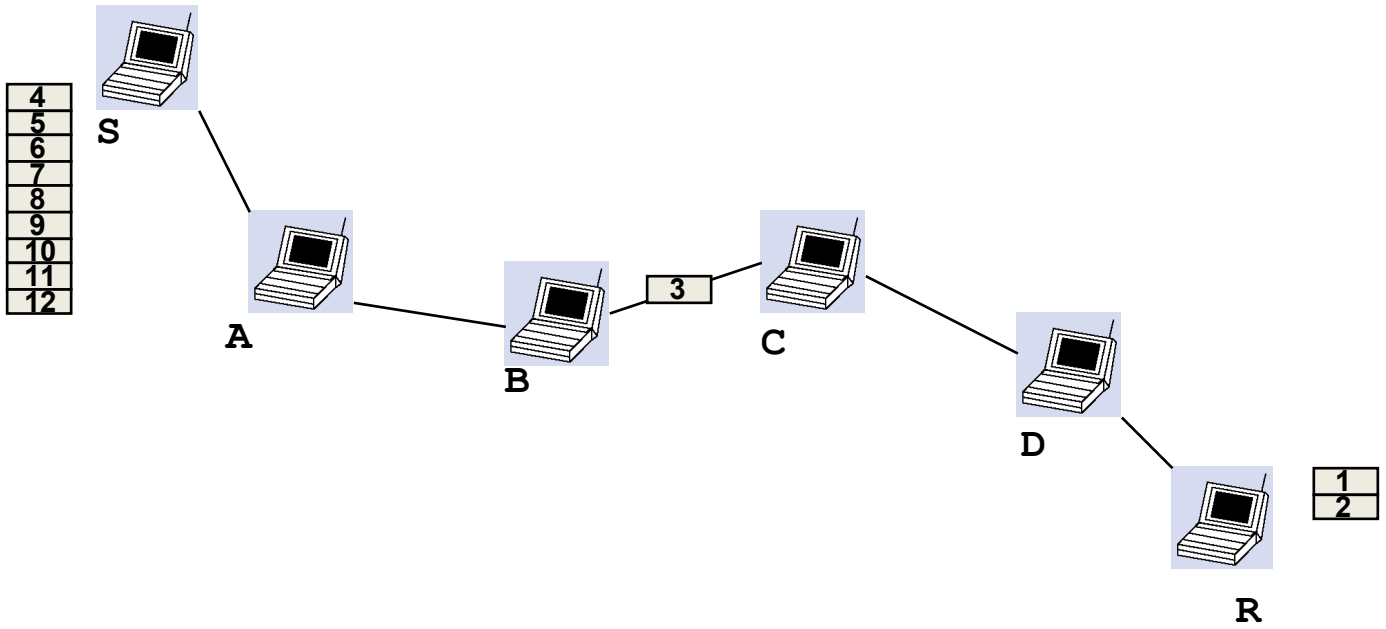
Multi-Hop Wireless Ad Hoc Networks



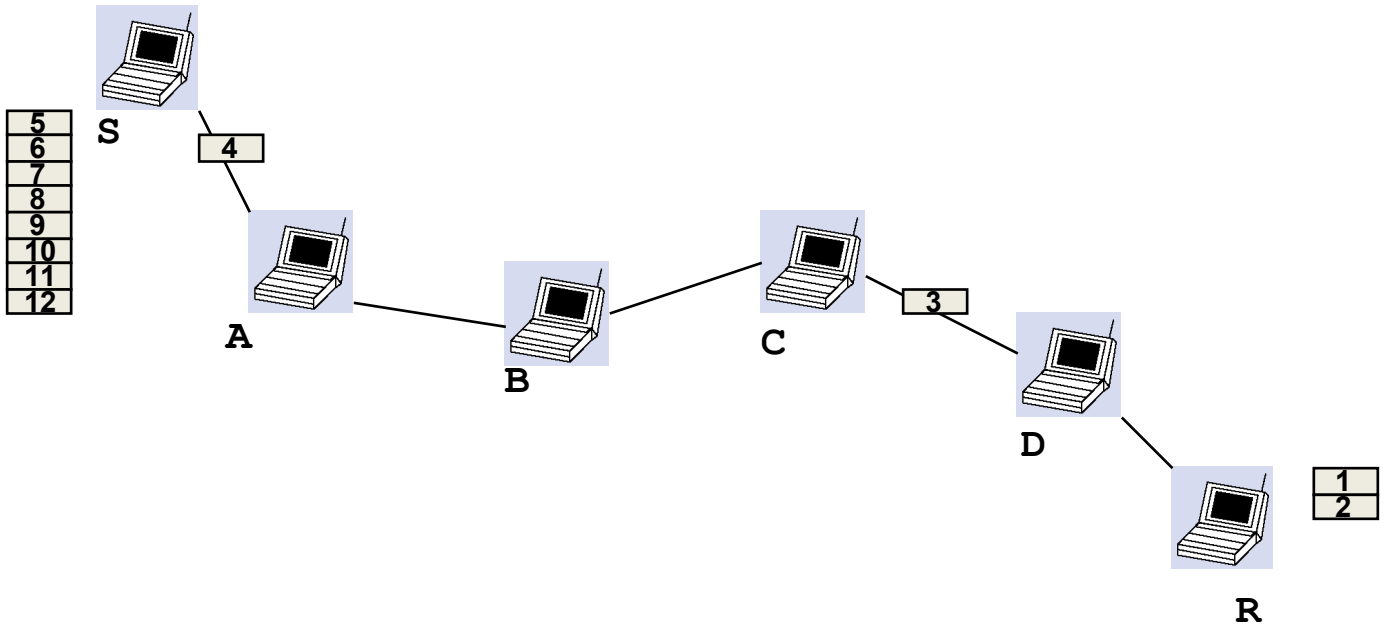
Multi-Hop Wireless Ad Hoc Networks



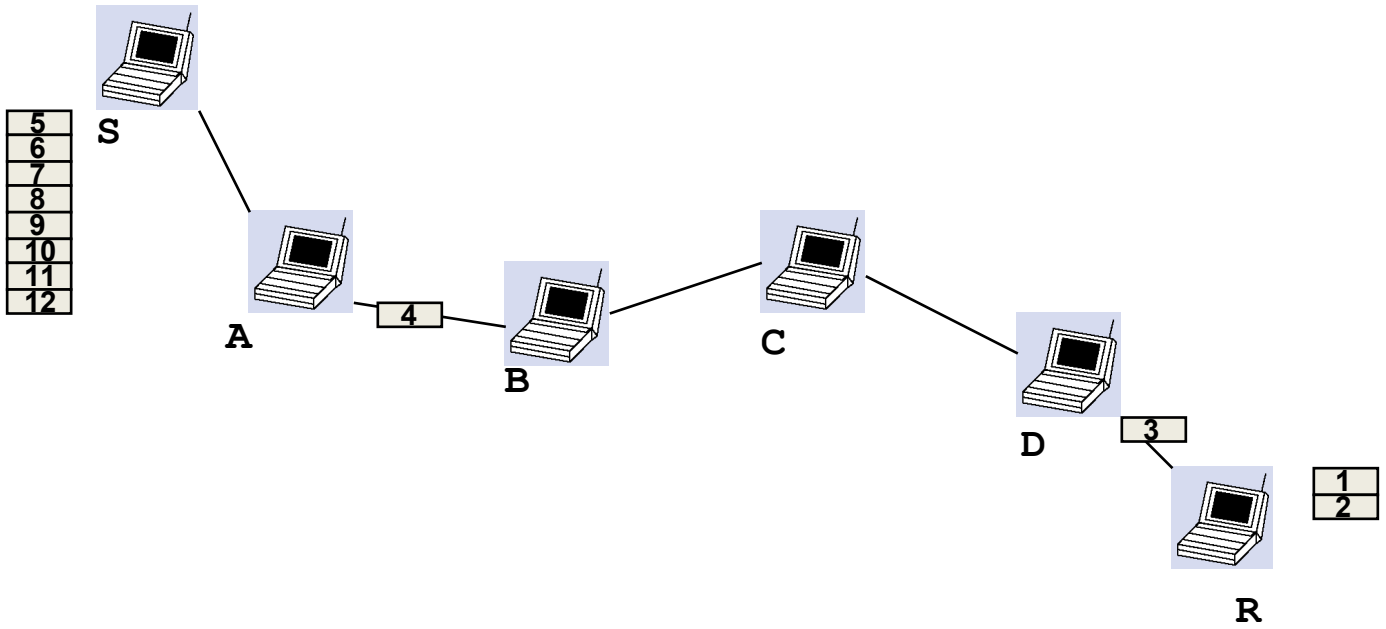
Multi-Hop Wireless Ad Hoc Networks



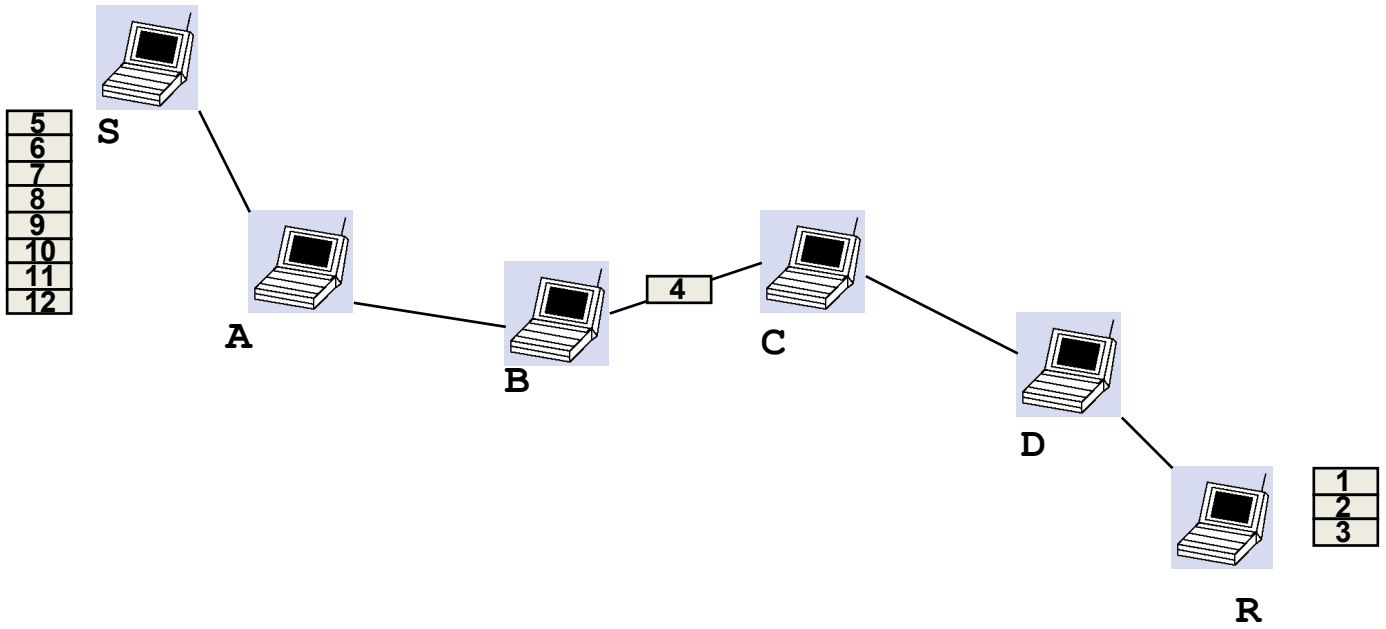
Multi-Hop Wireless Ad Hoc Networks



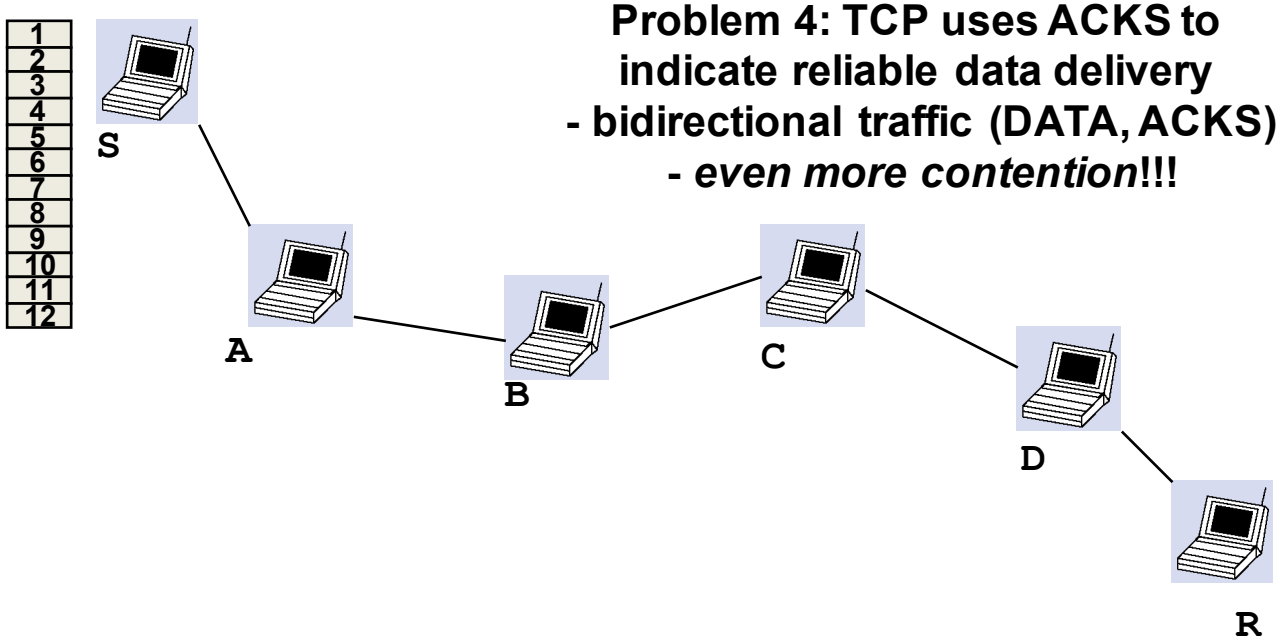
Multi-Hop Wireless Ad Hoc Networks



Multi-Hop Wireless Ad Hoc Networks

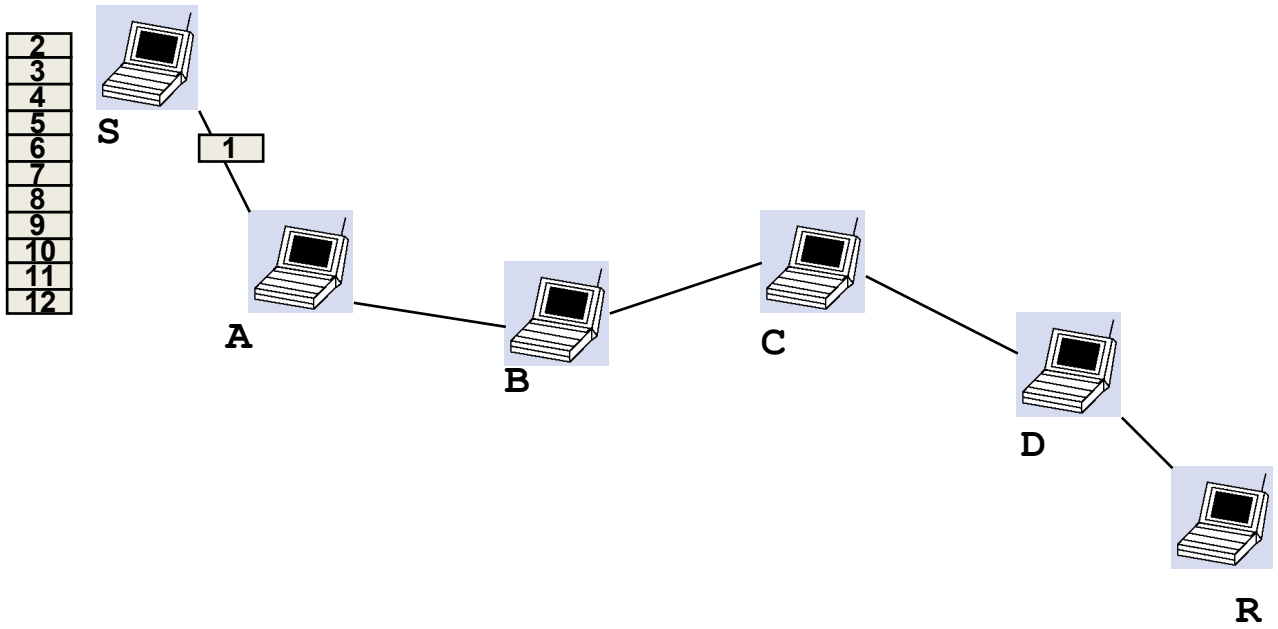


Multi-Hop Wireless Ad Hoc Networks

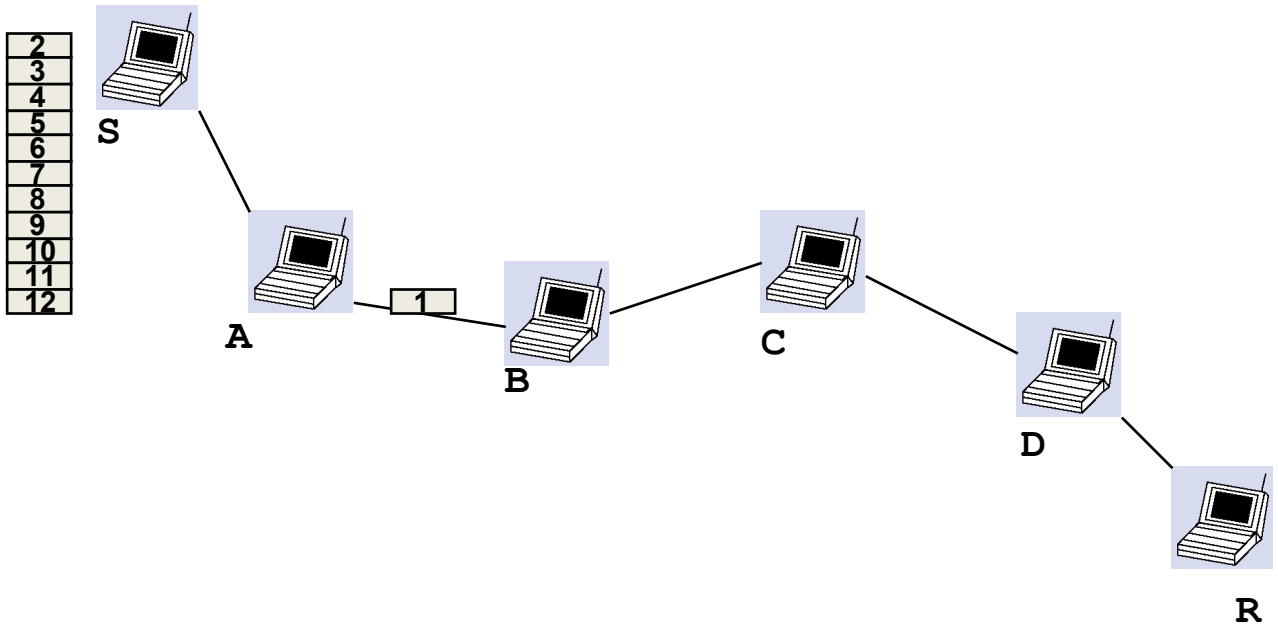


Problem 4: TCP uses ACKS to indicate reliable data delivery
- bidirectional traffic (DATA, ACKS)
- *even more contention*!!!

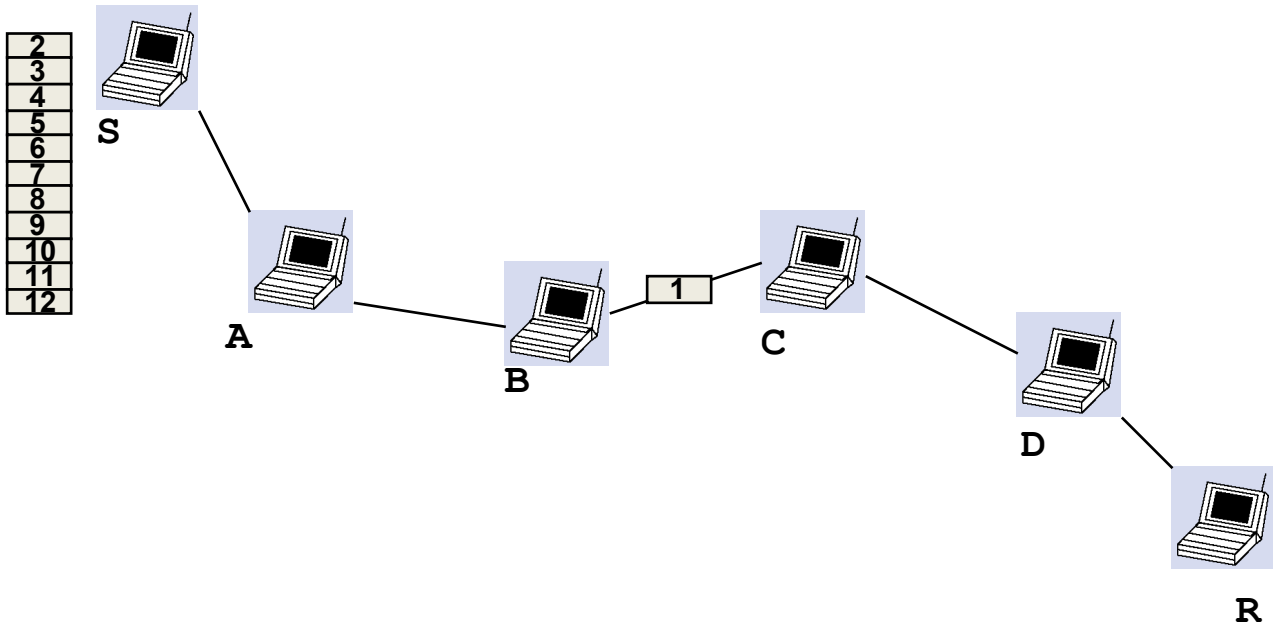
Multi-Hop Wireless Ad Hoc Networks



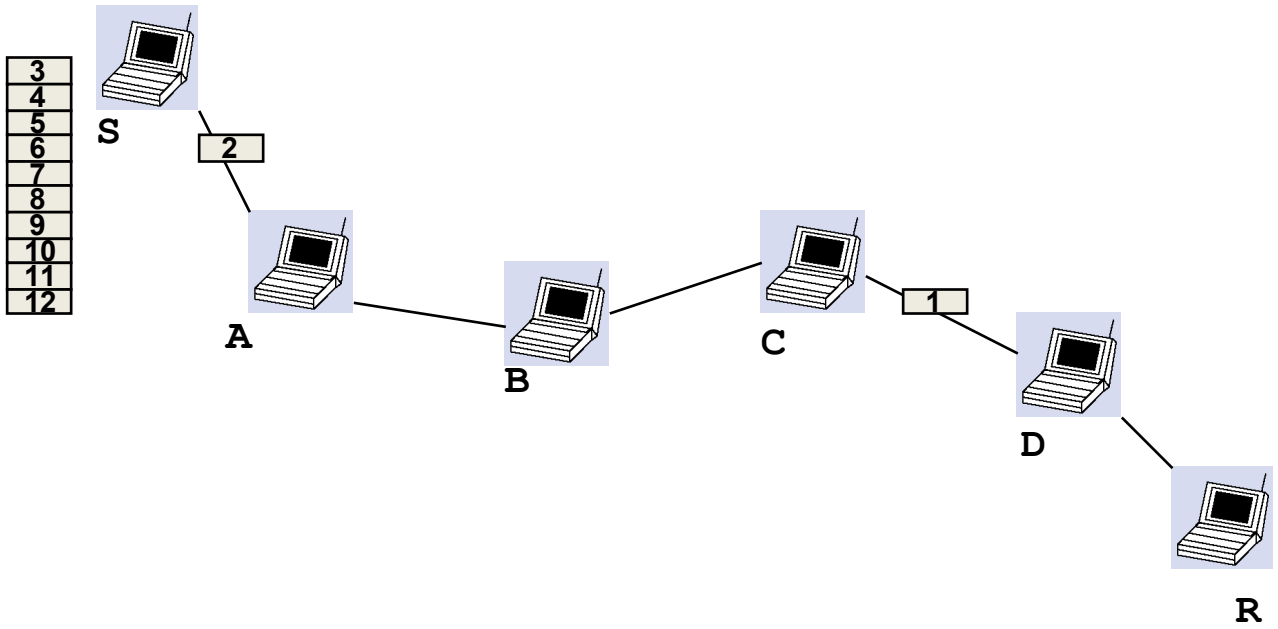
Multi-Hop Wireless Ad Hoc Networks



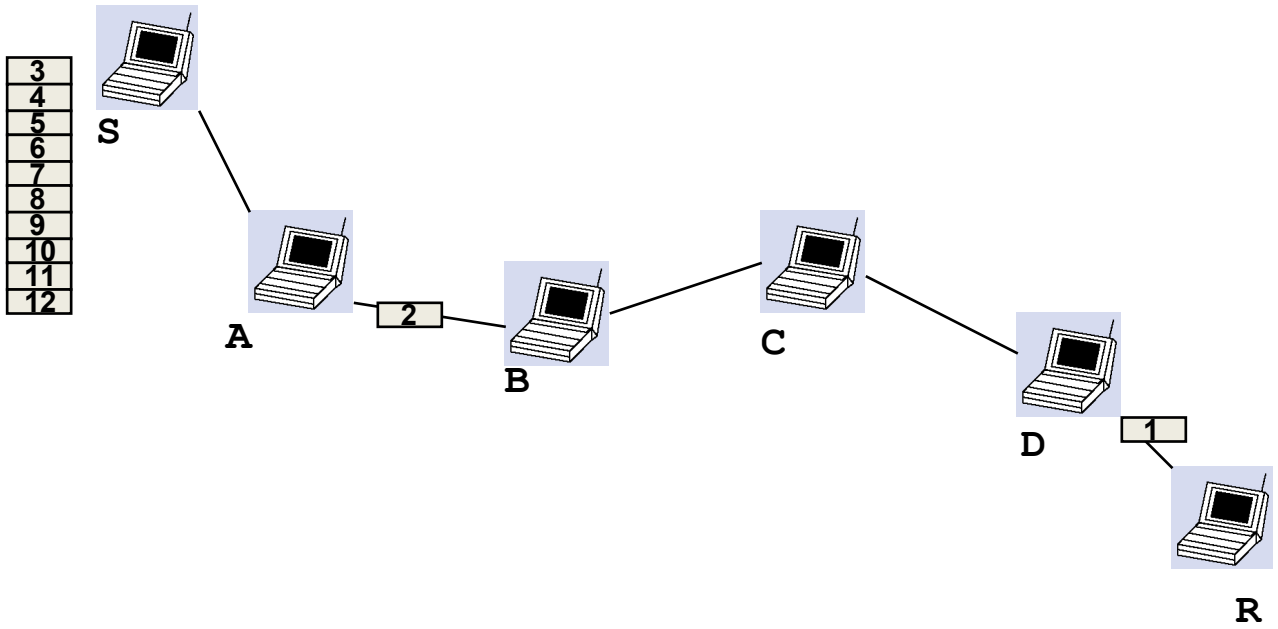
Multi-Hop Wireless Ad Hoc Networks



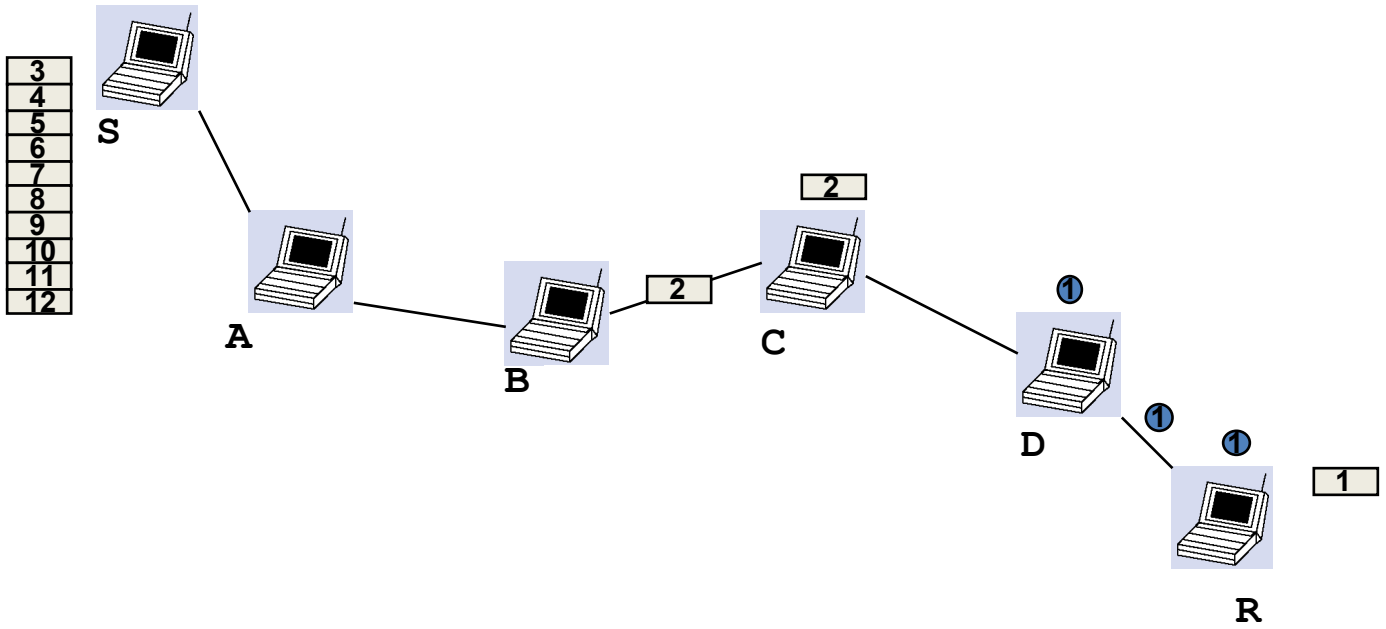
Multi-Hop Wireless Ad Hoc Networks



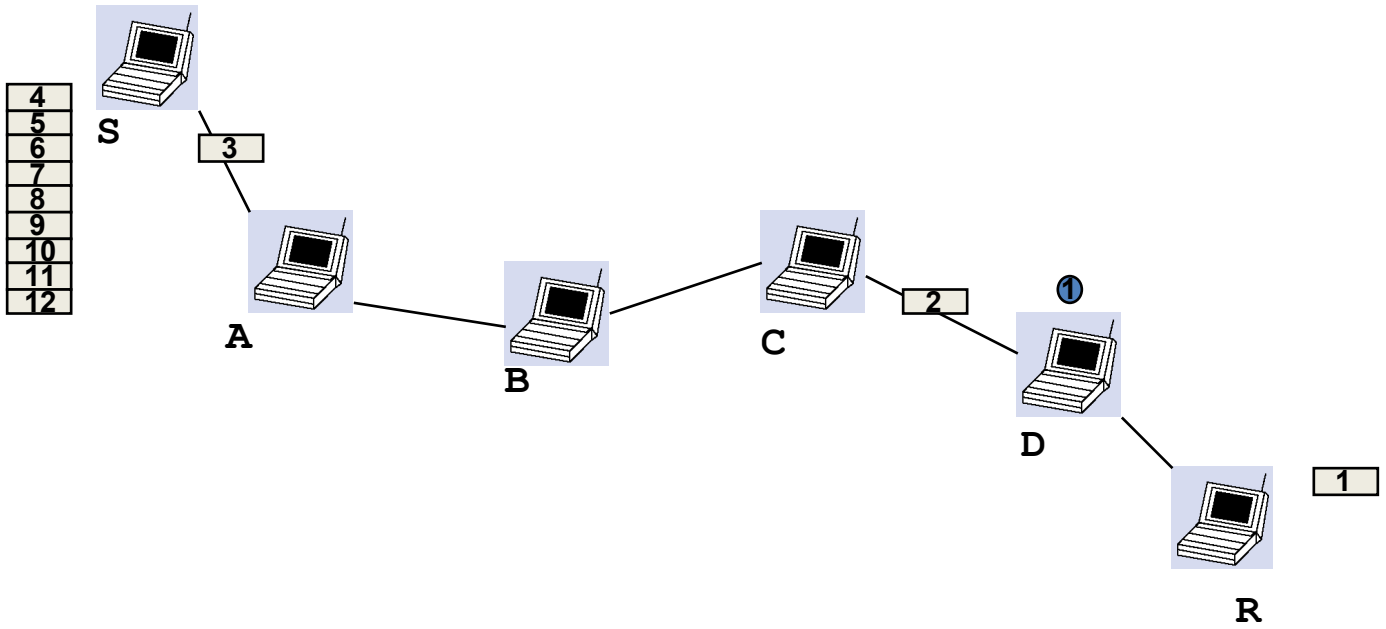
Multi-Hop Wireless Ad Hoc Networks



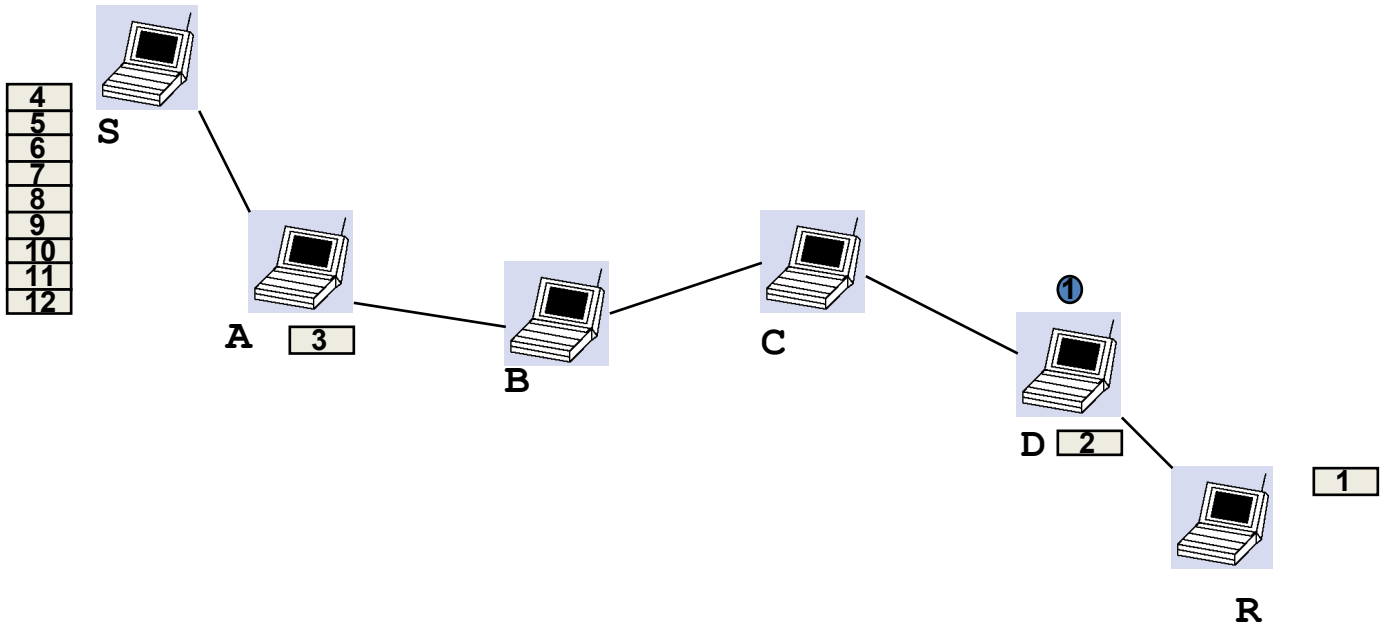
Multi-Hop Wireless Ad Hoc Networks



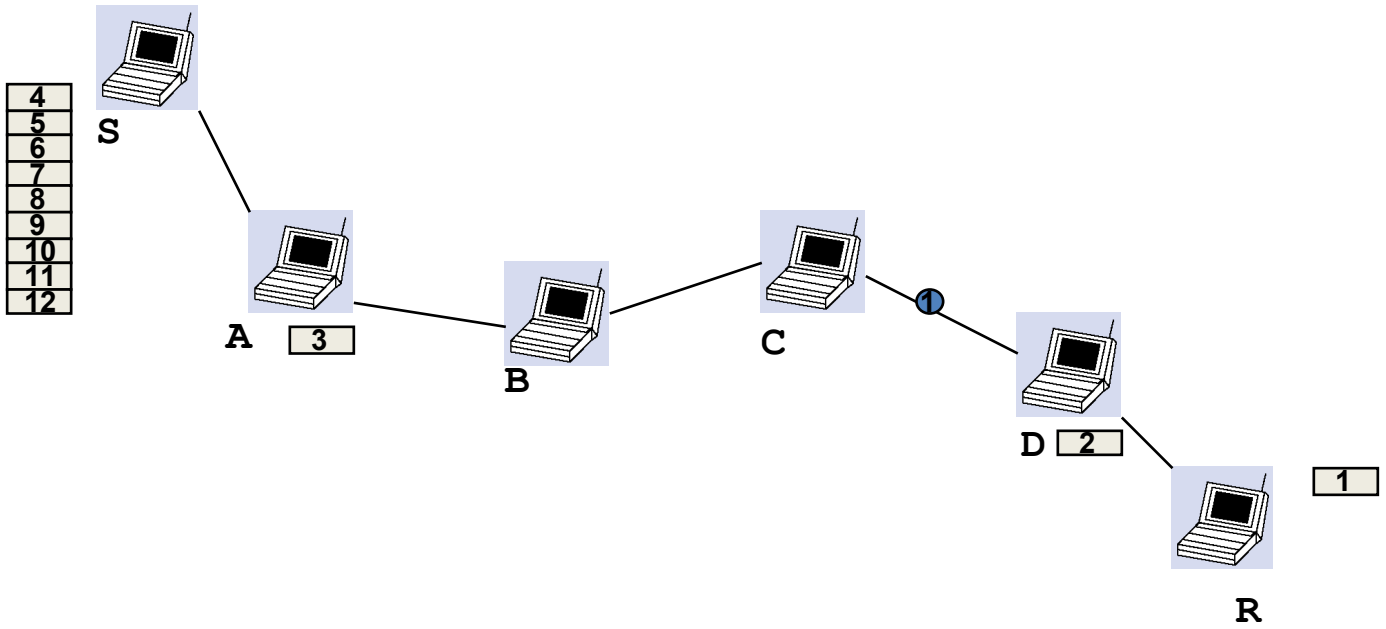
Multi-Hop Wireless Ad Hoc Networks



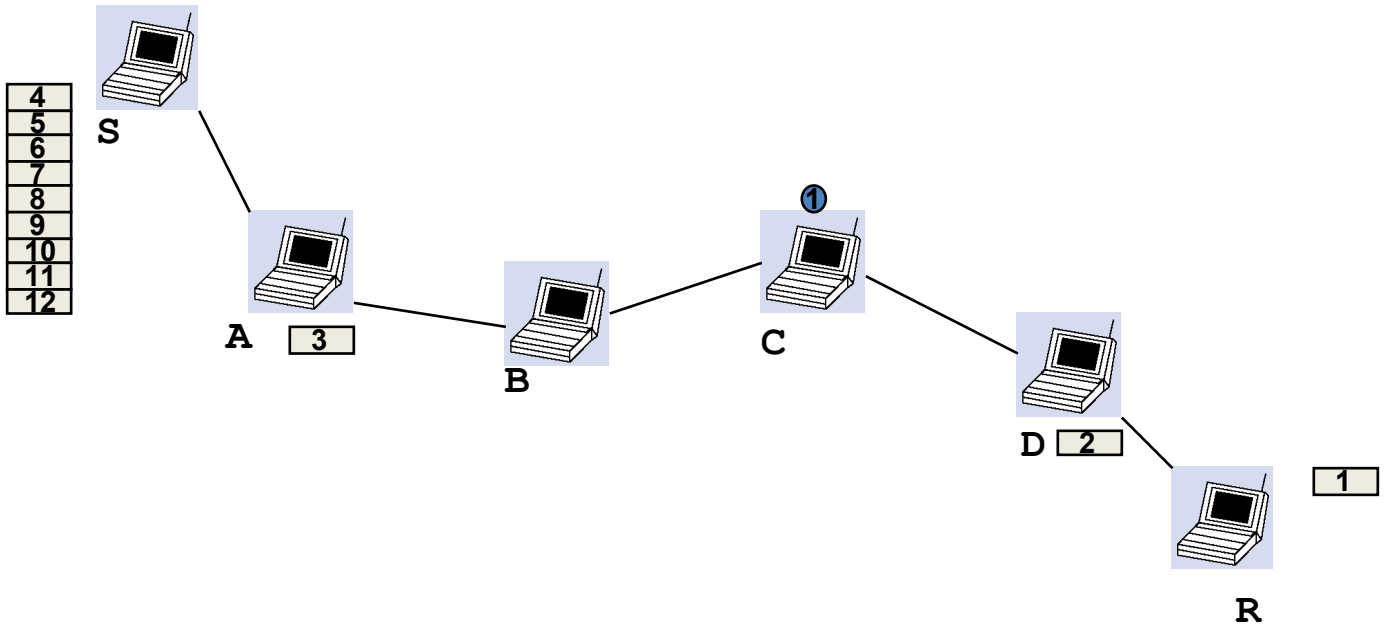
Multi-Hop Wireless Ad Hoc Networks



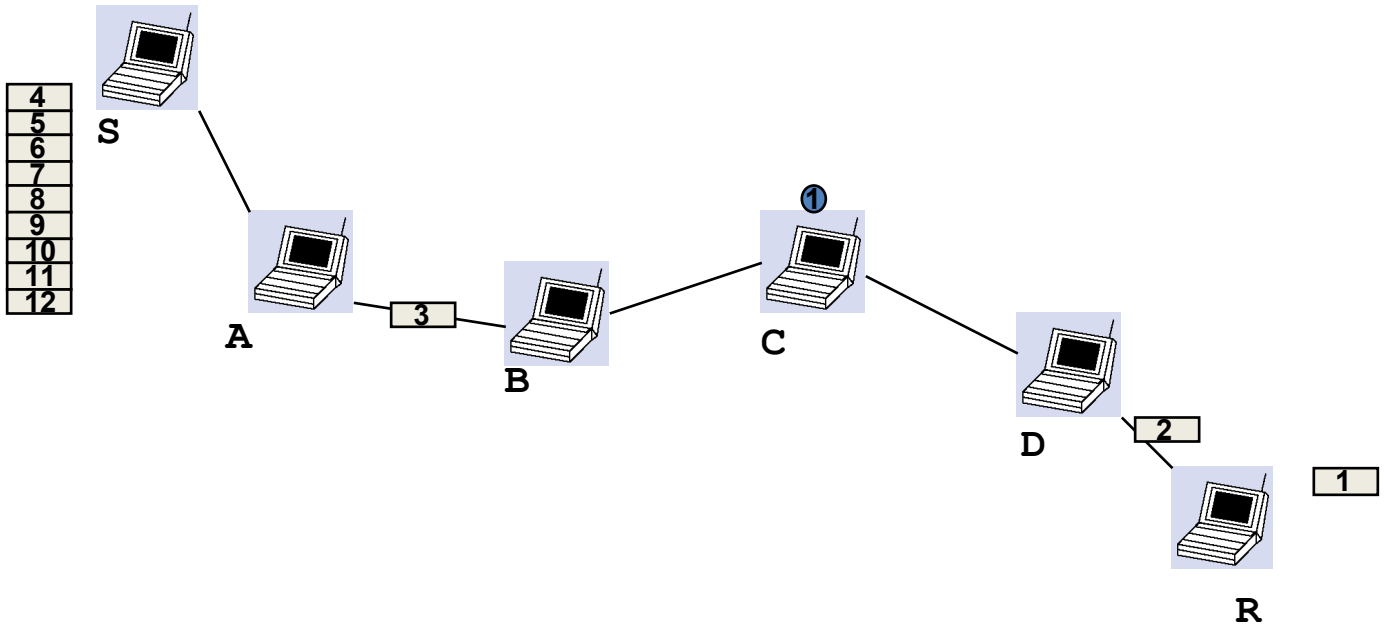
Multi-Hop Wireless Ad Hoc Networks



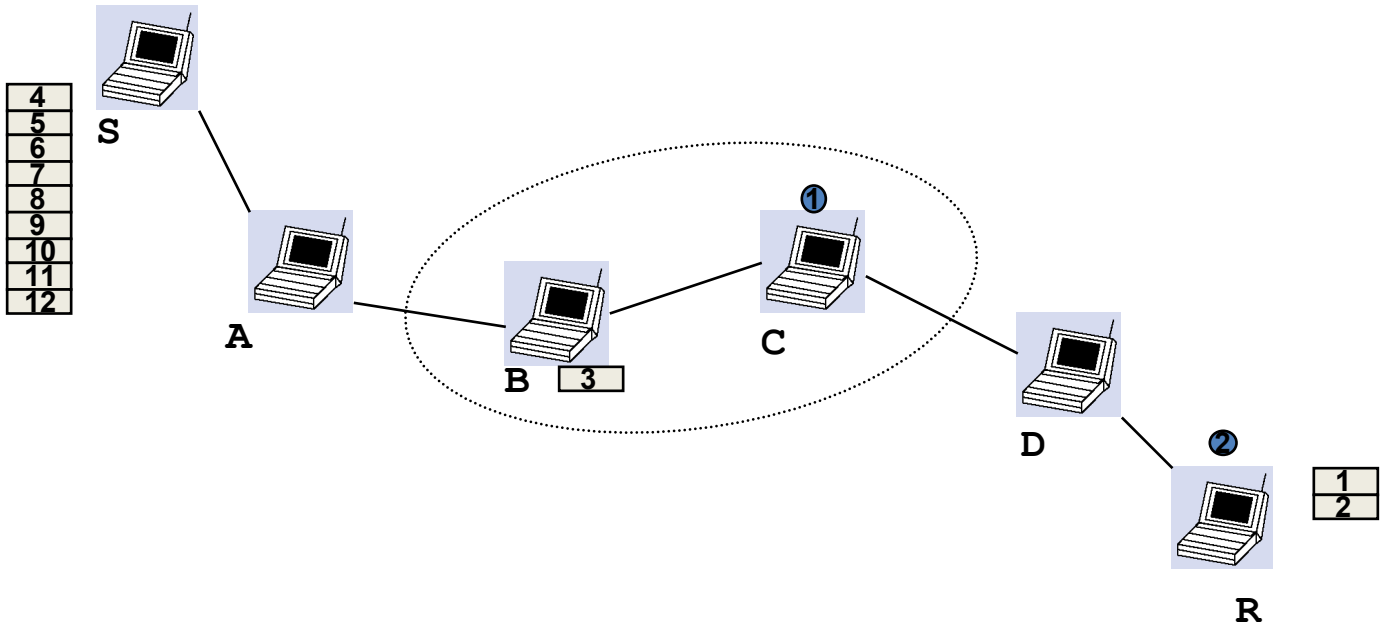
Multi-Hop Wireless Ad Hoc Networks



Multi-Hop Wireless Ad Hoc Networks



Multi-Hop Wireless Ad Hoc Networks



Key Requirements for Multi-hop Routing

- Find a reliable path
- Minimize # of hops

TCP Congestion Control

- Problem Definition
  - How much data should I pump into the network to ensure
    - Intermediate router queues not filling up
    - Fairness achieved among multiple TCP flows
- Why is this problem difficult?
  - TCP cannot have information about the network
  - Only TCP receiver can give some feedbacks

The Control Problem

- Two main components in TCP
  - Flow Control and Congestion Control
- Flow Control
  - If receiver's bucket filling up, pour less water
- Congestion Control
  - Don't pour too much if there are leaks in intermediate pipes
  - Regulate your flow based on how much is leaking out
    - Aggressive pouring → calls for retransmission of lost packets
    - Conservative pouring → lower e2e capacity
- Challenge: At what rate should you pour ?

PART 5: TCP OVER WIRELESS

The TCP Protocol (in a nutshell)

- Sender transmits few packets, waits for ACK
  - Called slow start
- Receiver acknowledges all packet till seq #i by ACK i (optimizations possible)
  - ACK sent out only on receiving a packet
  - Can be Duplicate ACK if expected packet not received
- ACK reaches Sender → indicator of more capacity
  - T transmits larger burst of packets (self clocking) … so on
  - Burst size increased until packet drops (i.e., DupACK)
- When Sender gets DupACK or waits for longer than RTO
  - Assumes congestion → reduces burst size (congestion window)

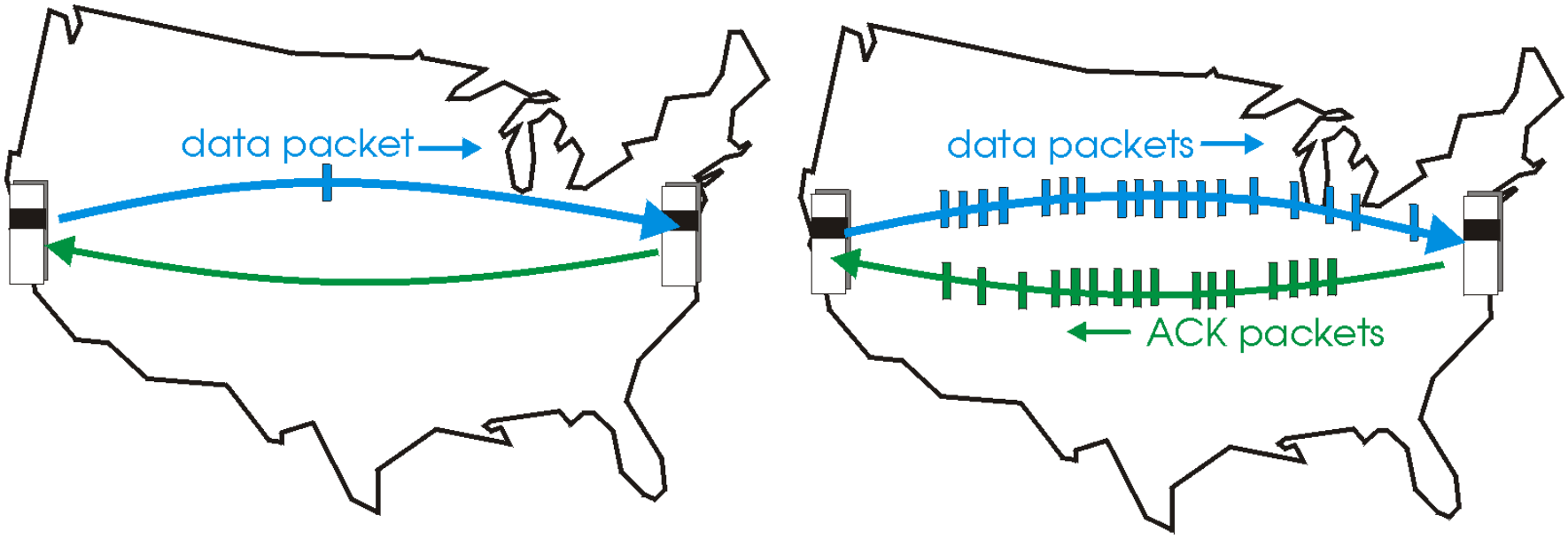
TCP Basics

- Cumulative acknowledgements
- Window based flow control
- Fast recovery from packet losses
- Congestion avoidance

TCP: Pipelined Protocol

Pipelining: sender allows multiple, "in-flight", yet-to-be-acknowledged pkts

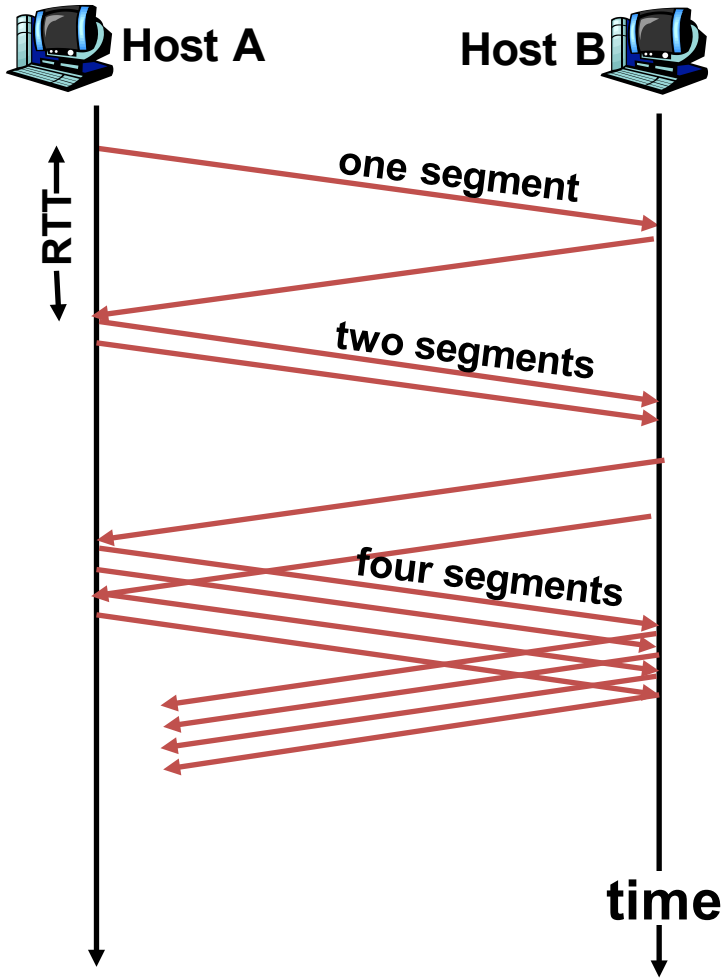
- range of sequence numbers must be increased
- buffering at sender and/or receiver

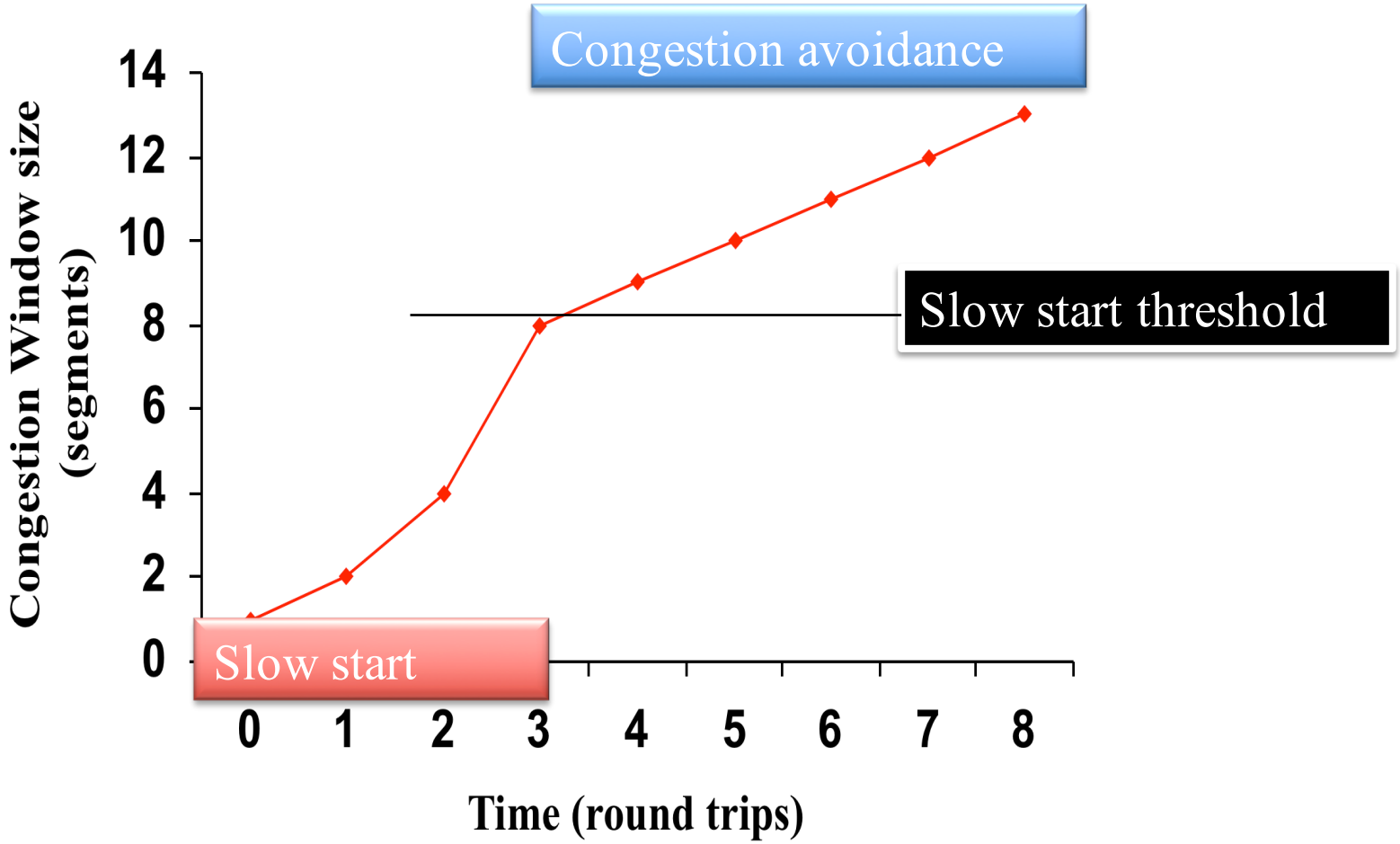


(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

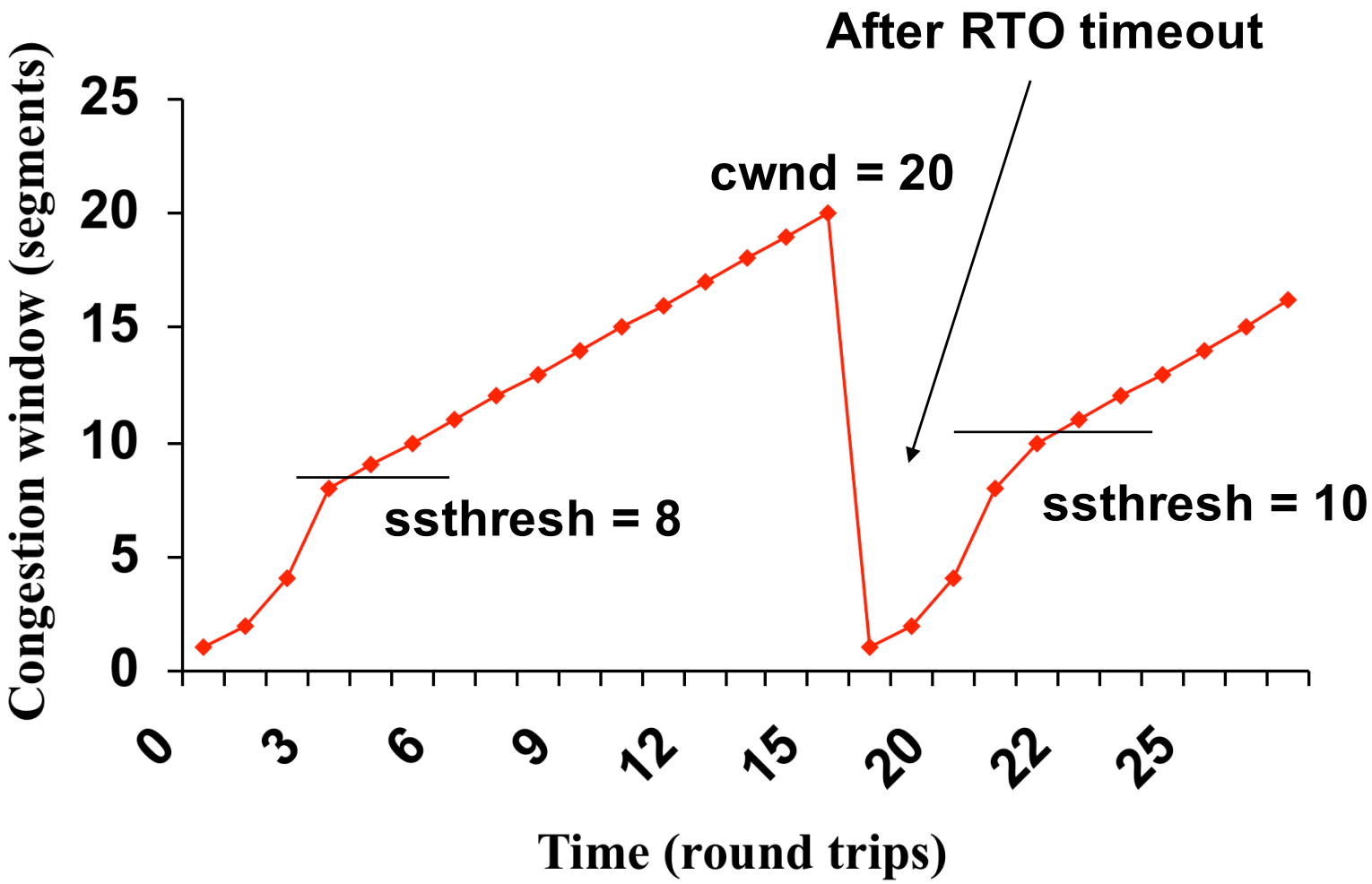
TCP Timeline





Example assumes that acks are not delayed

When waited for > RTO



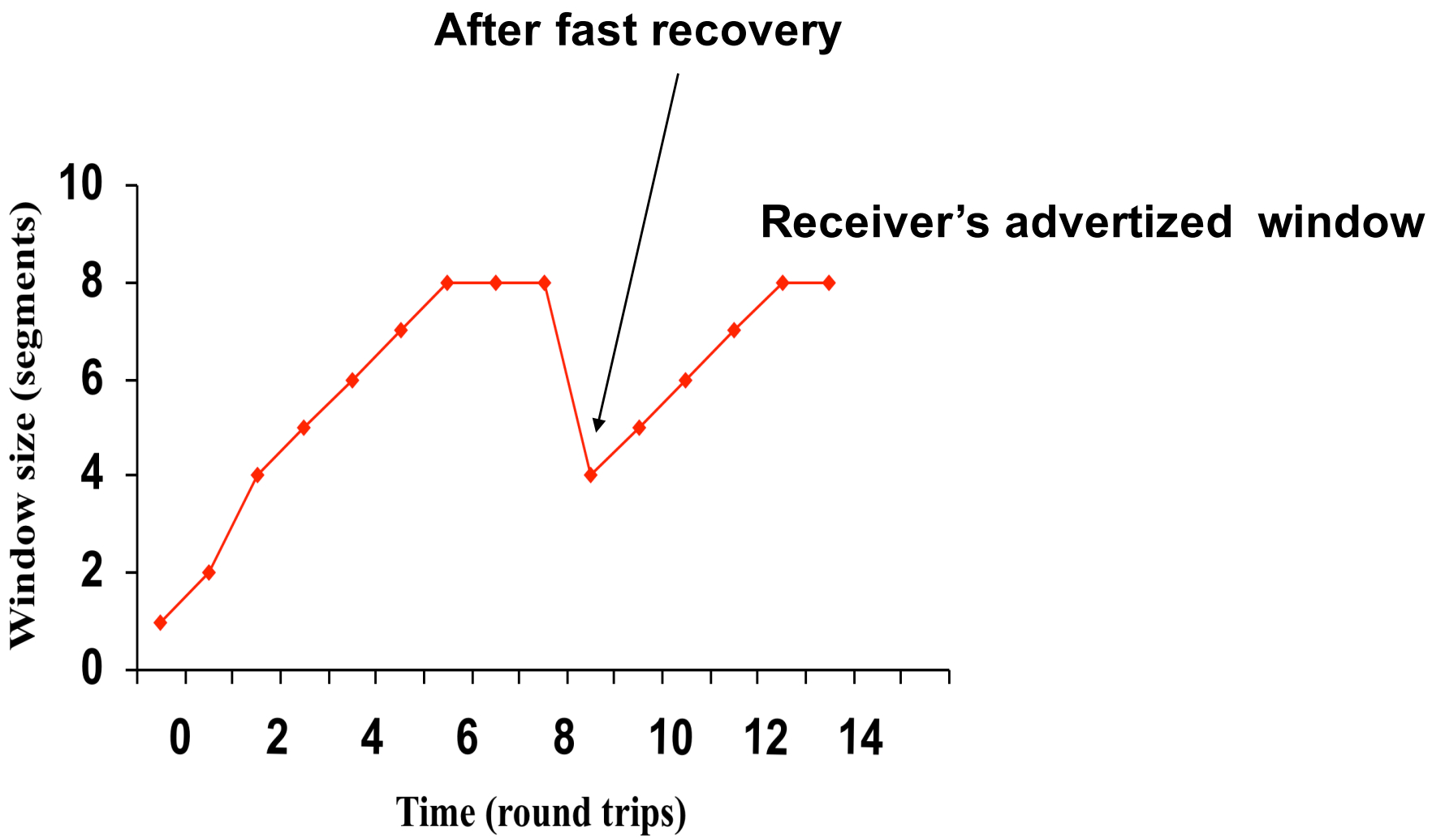
The TCP Protocol (in a nutshell)

- DupACK not necessarily indicator of congestion
  - Can happen due to out of order (OOO) delivery of packets

- If 3 OOO pkts, then CW **need not** be cut drastically
  - The DupACK packet retransmitted
  - Continue with same pace of transmission as before (fast recovery)

- Receiver advertizes its receiver window in ACKs
  - If filling up, Sender reduces congestion window

Fast Recovery on 3 OOO DupACKs



Several flavors of TCP: combines options / optimizations

Reno, Vegas, Eifel, Westwood …

Overall TCP has worked well – proven on the internet

Then why study it again for wireless networks ?

wired vs. wireless networks

- Wired
  - Network congestion (at routers)
- Wireless:
  - Transmission errors (due to radio propagation; interference)
  - Contentions
  - Mobility

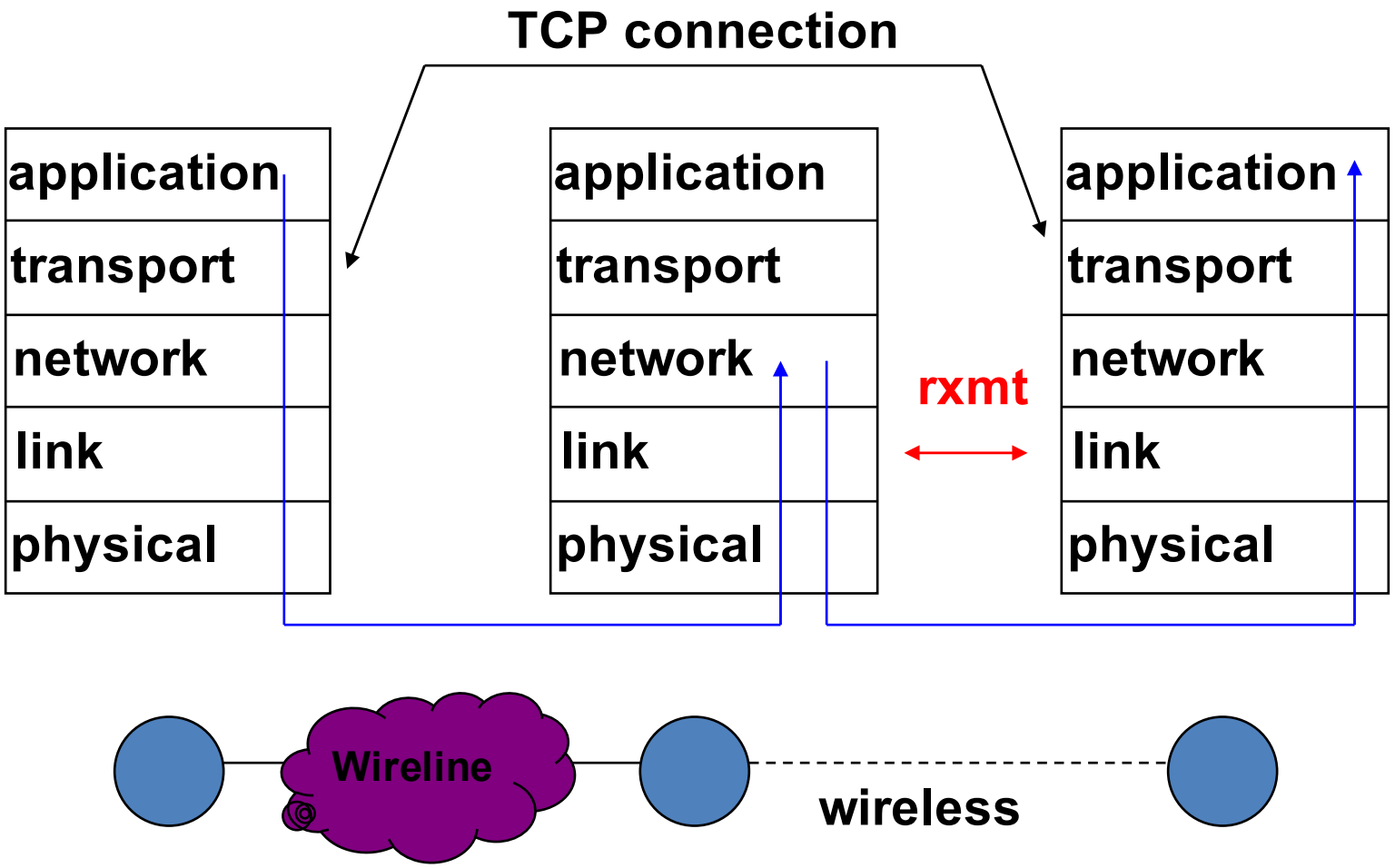
Renewed Challenge

- Key assumption in TCP
  - A packet loss is indicative of network congestion
  - Source needs to regulate flow by reducing CW
- Assumption closely true for wired networks
  - BER ~ 10^{-6}
- With wireless, errors due to fading, fluctuations
  - Need not reduce CW in response …
  - But, TCP is e2e → CANNOT see the network
  - Thus, TCP cannot classify the cause of loss → CHALLENGE

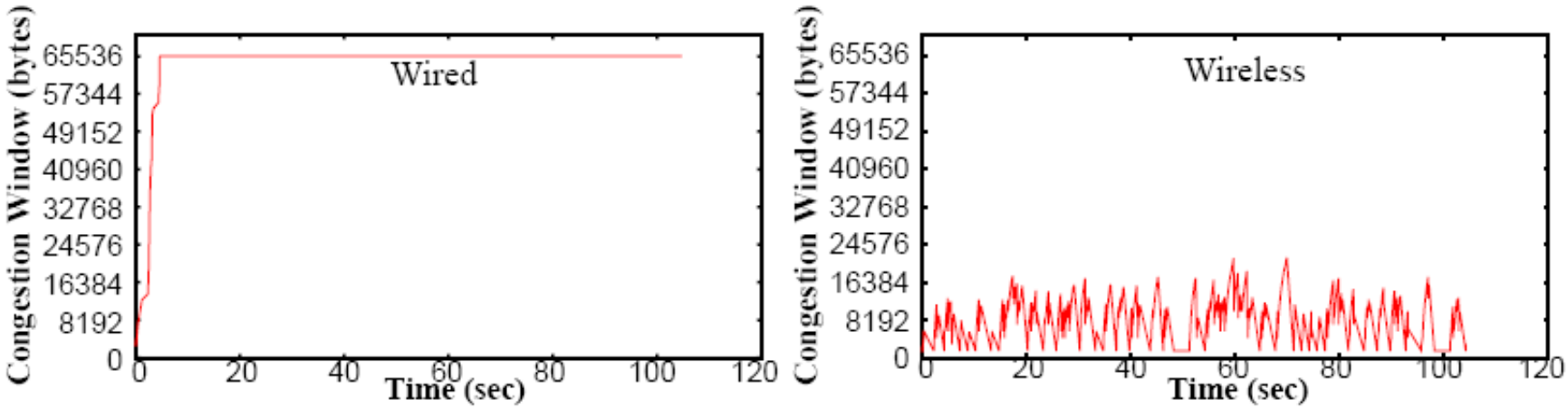
Running TCP on wireless links

- TCP interprets any packet loss as a sign of congestion
  - TCP sender reduces congestion window
- However, on wireless links, packet loss mostly occur due to random channel errors, cellular/WLAN handoffs
  - It is temporary loss, not due to congestion
- Reducing congestion window may be too conservative
  - Leads to poor throughput

The Problem Model



Impact of transmission errors on TCP



Ideal Behavior

- Ideal TCP behavior:
  - TCP sender should simply retransmit a packet lost due to transmission errors, without taking any congestion control actions
  - Fundamental question: How to distinguish loss due to congestion from loss due to wireless/mobility??
  - Very hard to do:
    - TCP is fundamentally end-to-end
    - Only know the packet is lost, but not why
- Ideal network behavior
  - Transmission errors should be hidden from the sender -- the errors should be recovered transparently and efficiently
- Proposed schemes attempt to approximate one of the above two ideals

Summary of Approaches

- **Group 1: Mask wireless loss from the TCP sender**
  - So that TCP sender will not reduce congestion window unnecessarily
- **Group 2: Explicitly notify the TCP sender about the cause of packet loss**
  - TCP sender will not reduce congestion window for wireless losses
- **Solutions implemented at**
  - TCP sender
  - TCP receiver
  - Intermediate node (wireless basestation, wifi access point)

(1) Mask wireless losses from TCP sender

- Link-level approaches
- Snoop TCP
- Split connection approach (Indirect-TCP)
- Assumptions:
  - Assuming wireless connection is just one hop
  - All wireless losses are NOT due to congestion
    - Assuming MAC layer can address contention, e.g. via CSMA/CA

1.1 Link Level Mechanisms

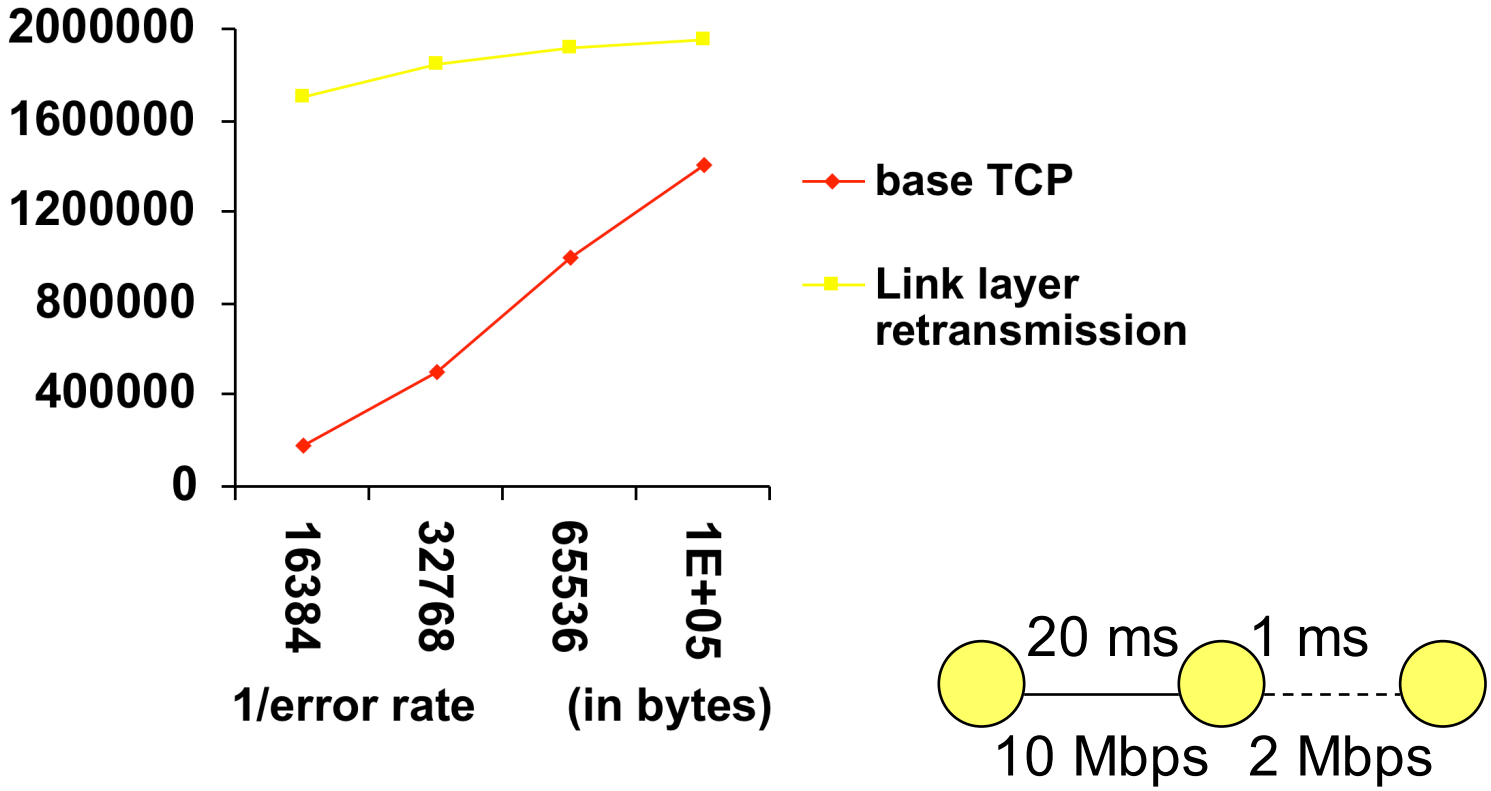
Link Layer Schemes

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

Link Layer Mechanisms

- Forward Error Correction (FEC)
  - Correct small number of errors
  - Correctable errors hidden from the TCP sender
  - FEC incurs overhead even when errors do not occur
- Packet retransmissions
  - Link level retransmission schemes retransmit a packet at the link layer, if errors are detected
  - Retransmission overhead incurred only if errors occur
- Use both

Link Layer Retransmissions [Vaidya99]



2 Mbps wireless duplex link with 1 ms delay
Exponential error model
No congestion losses

Issues

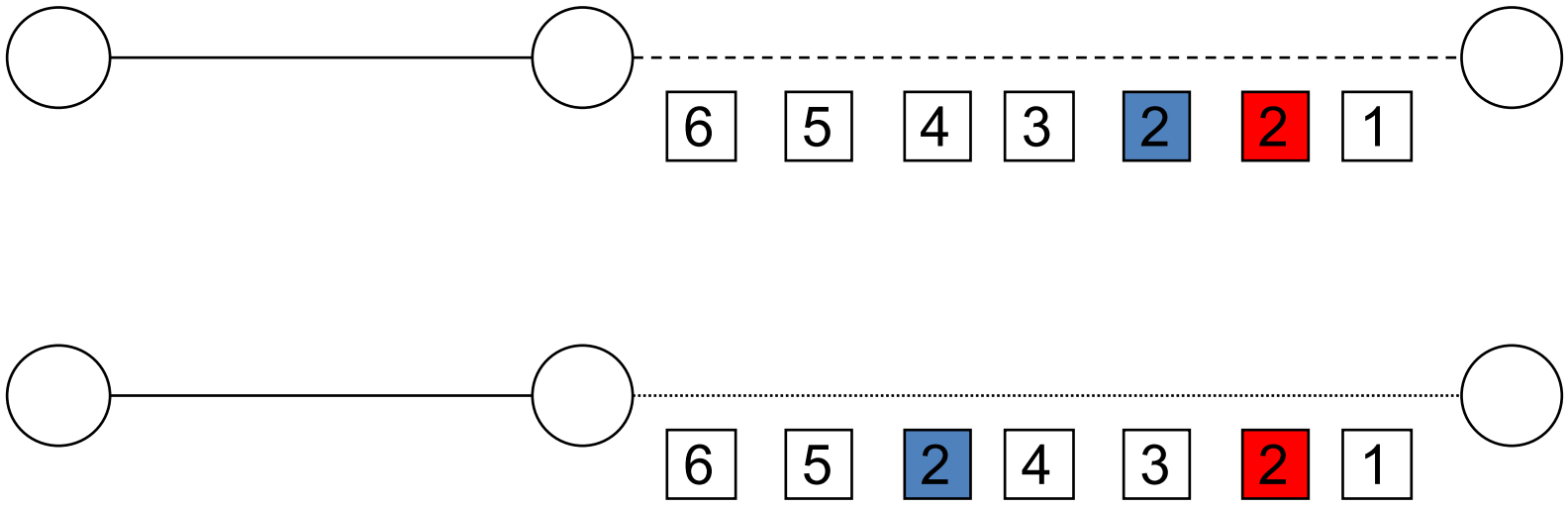
- How many times to retransmit at the link level before giving up?
  - Finite bound -- semi-reliable link layer
  - No bound -- reliable link layer
- What triggers link level retransmissions?
  - Link layer timeout mechanism
  - Link level acks (negative acks, dupacks, …)
  - Other mechanisms (e.g., 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

Issues (cont.)

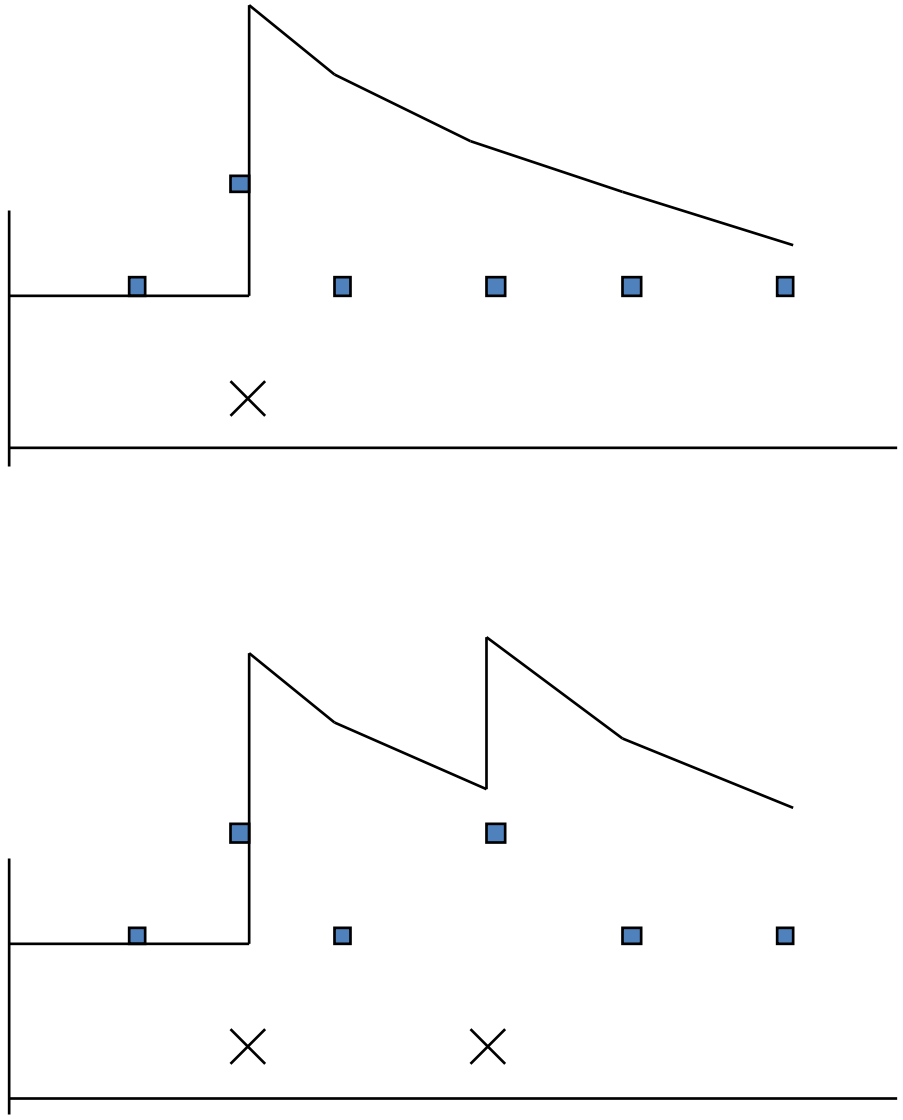
- Should the link layer deliver packets as they arrive, or deliver them in-order?
  - Link layer may need to buffer packets and reorder if necessary so as to deliver packets in-order

Link Level Retransmissions In-order delivery

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



Retransmission Time Out (RTO) Variations



Link Layer Schemes: Summary

When is a reliable link layer beneficial to TCP performance?

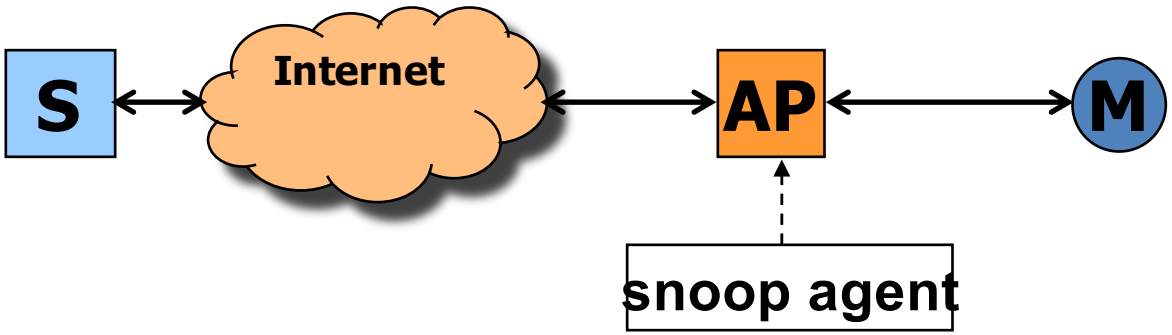
- if it provides almost in-order delivery

and

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

1.2 TCP-Aware Link Layer -- SNOOP

Snoop



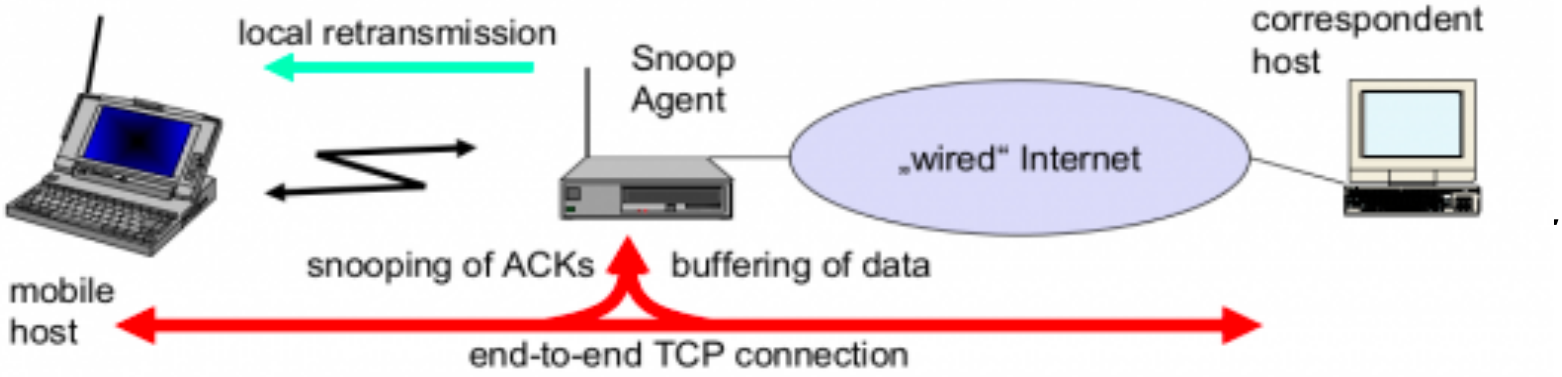
- Hide wireless-related losses from sender by doing local retransmission on wireless link
- Example: Snoop TCP
  - Snoop agent at AP maintains cache of TCP segments not yet ACK'd
  - packet losses indicated by timeouts or duplicate ACKs from mobile triggers AP to retransmit lost packet and suppress duplicate ACKs

Snoop Protocol

- Hides wireless losses from the sender
- Requires modification to **only BS/AP** (network-centric approach)

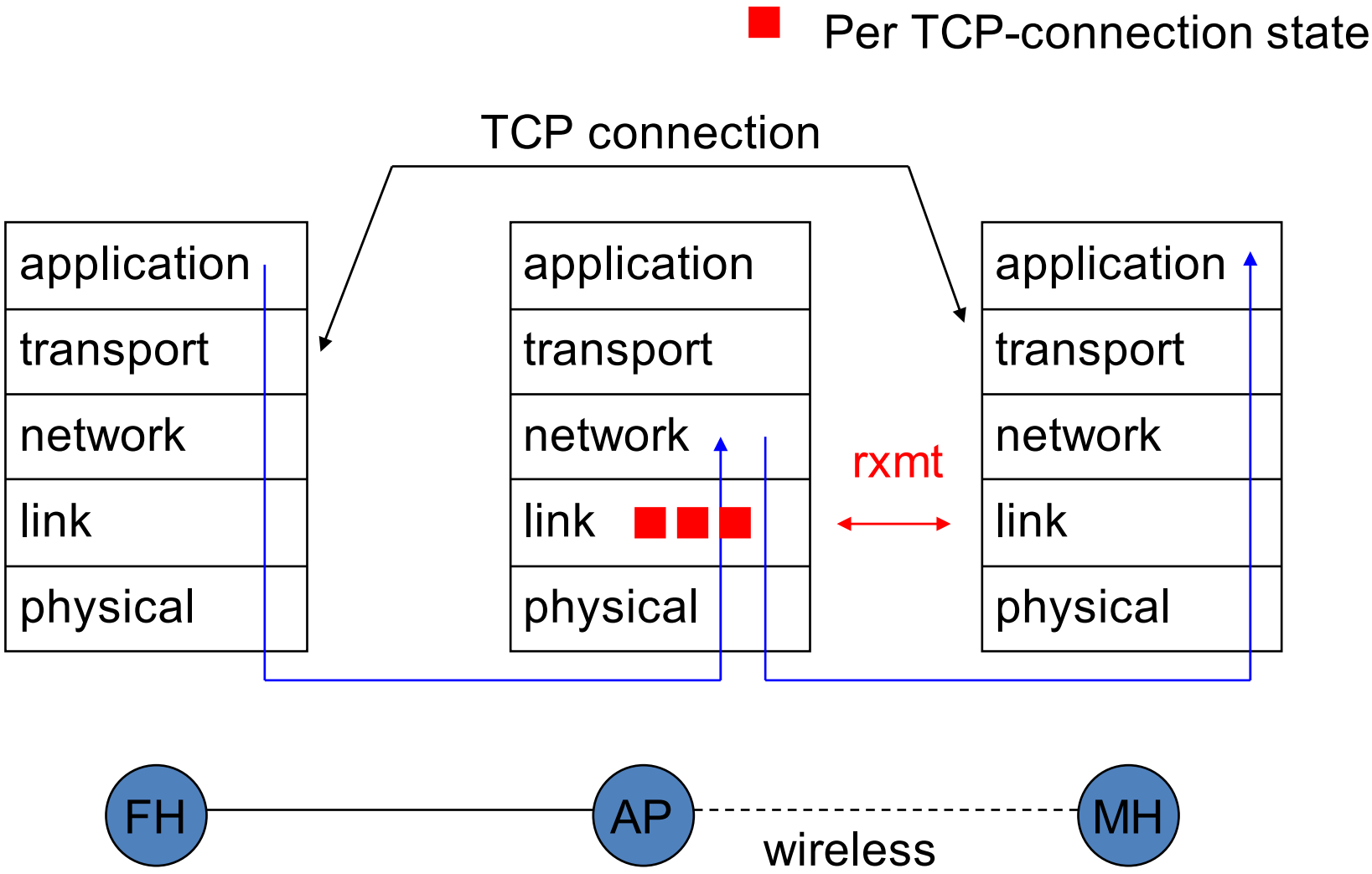
- Modify BS/AP to snoop on all TCP packets
  - AP→ MH: AP buffers all data to MH until it receives ACK from the MH, AP detects packet loss via dupacks or time-out, and retransmits packet
  - MH → AP: AP detects packet loss using sequence numbers and answers directly to MH

Snoop Protocol

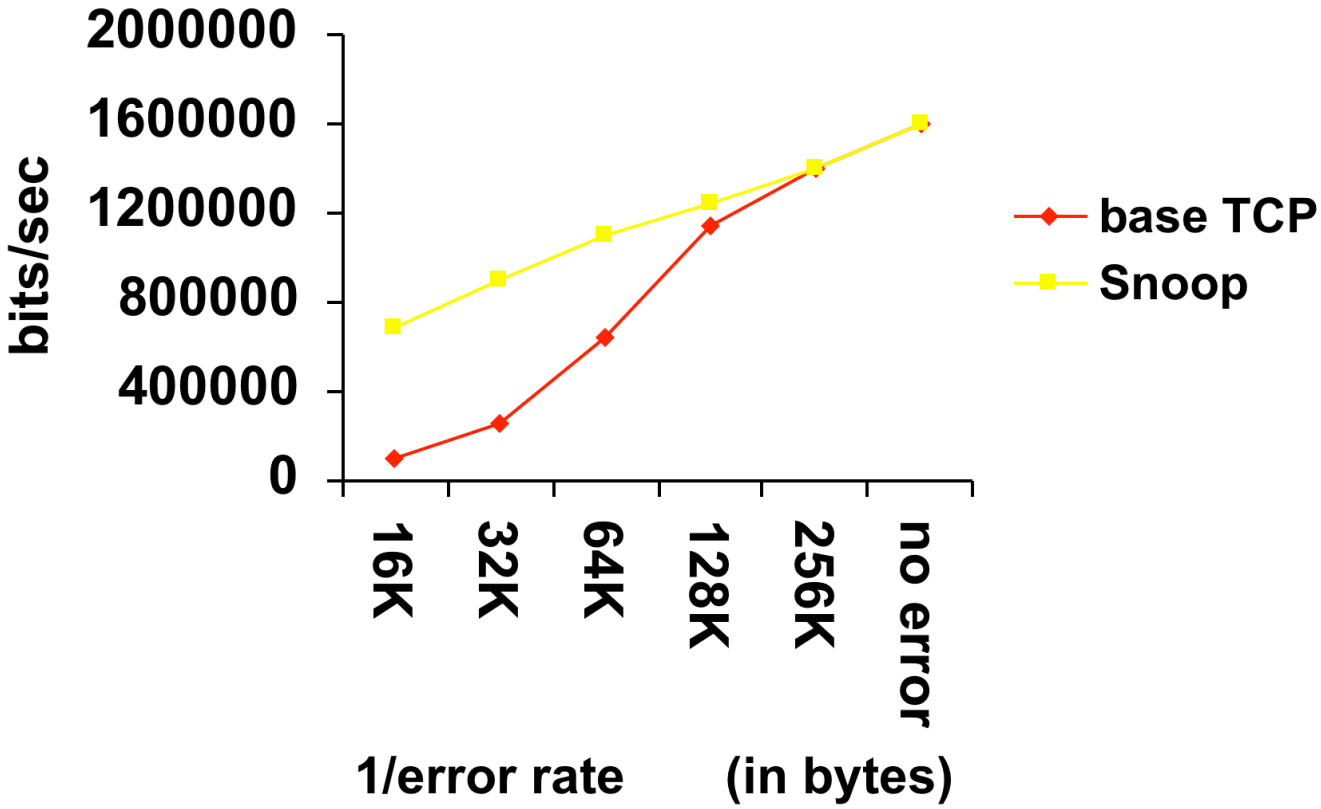


- Modify BS/AP to snoop on all TCP packets
  - AP→ MH: AP buffers all data to MH until it receives ACK from the MH, AP detects packet loss via dupacks or time-out, and retransmits packet
  - MH → AP: AP detects packet loss using sequence numbers and answers directly to MH

Snoop Protocol



Snoop Results



2 Mbps Wireless link

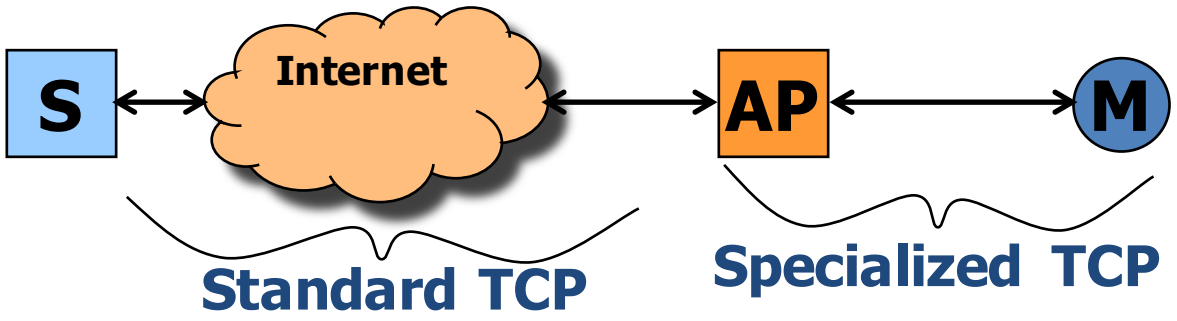
Snoop summary

- Hides wireless losses from the sender
- Requires modification to only BS (network-centric approach)
- **Advantages:**
  - Can work without modification on MH.
  - Preserves end-to-end semantics.

- **Disadvantages:**
  - Does not work with encrypted TCP headers.
  - Does not work for asymmetric routes.

1.3 Split Connection Approach

Split Connection Approaches



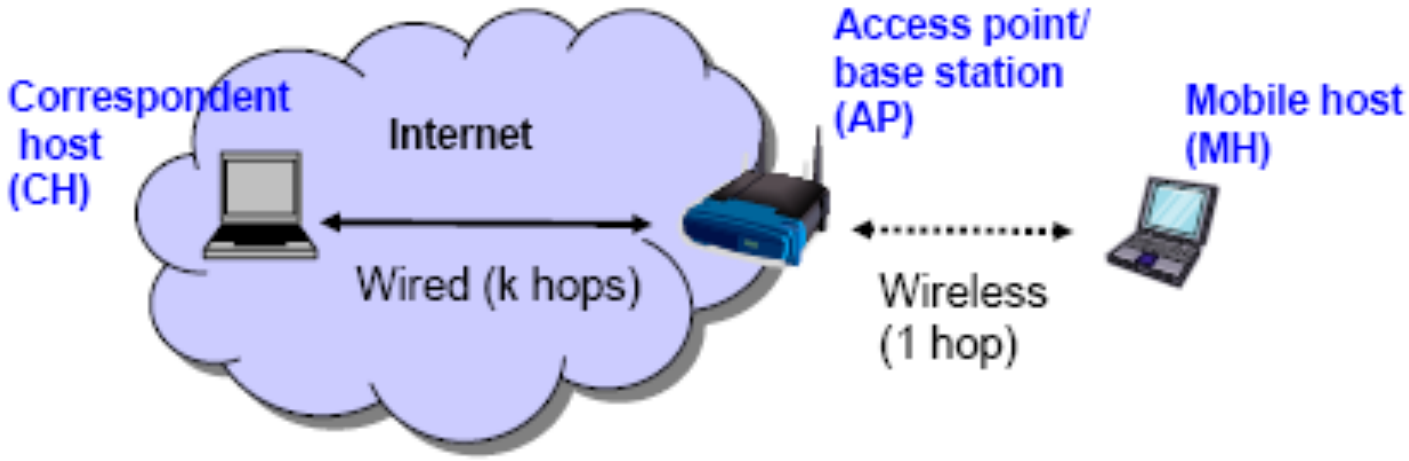
- Divide the TCP connection into a wired part and a wireless part
  - AP acts as a proxy endpoint for both connections
- Transport layer protocol customized for wireless can be used on the wireless link
- AP sends ACKs to S and is responsible for delivering data to MH
- during handoffs, state is transferred between APs
- Examples: Indirect TCP (I-TCP), Mobile TCP (M-TCP)

Breaking the End-to-End Connection

- End-to-end TCP connection is broken into one connection on the wired part of route and one over wireless part of the route
- A single TCP connection split into two TCP connections
  - if wireless link is not last on route, then more than two TCP connections may be needed
- Split connection results in independent flow control for the two parts
  - Flow/error control protocols, packet size, time-outs, may be different for each part

Example: Indirect tcp

- Segment the TCP connection into two parts
  - Wired part
    - No changes to the TCP protocol for hosts in this part
  - Wireless part
    - Modify the transport protocol that is tuned for the wireless connections



Good & Bad

- Good:
  - No changes to the fixed network
  - Transmission errors on the wireless link do not propagate into the fixed network
    - Local recovery from errors
  - Custom (optimized) transport protocol for wireless connection
- Bad
  - No more end-to-end semantics
  - AP needs large buffer space
  - AP must maintain per-TCP connection state

(2) End-to-end Approaches

- Modify TCP so that sender can respond differently to non-congestion losses
- Adv: TCP can be optimized for wireless
- Disadv: Requires modifications to TCP
- Examples:
  - Selective Acknowledgements (TCP-SACK)
  - Explicit Loss Notification (ELN)

2.1 Explicit Notification

Explicit Notification Schemes
General Philosophy

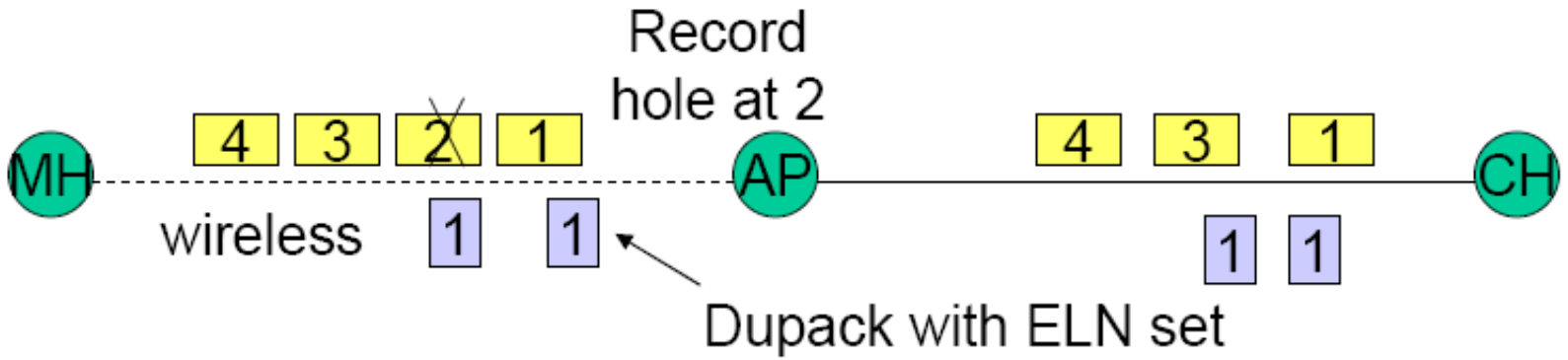
- Approximate Ideal TCP behavior:
  - Ideally, the TCP sender should simply retransmit a packet lost due to transmission errors, without taking any congestion control actions
- A wireless node somehow determines that packets are lost due to errors and informs the sender using an explicit notification
- Sender, on receiving the notification, does not reduce congestion window, but retransmits lost packet

Explicit notification

- Send notification to the TCP sender about wireless packet loss.
- Upon notification, TCP sender retransmits packet, but does not reduce congestion window.
  - Motivated by the Explicit Congestion Notification (ECN) Approach [Floyd-94].
- Many design options: Who sends notification? How? How notification is interpreted at sender?

An Example

- Assume MH is the TCP sender.
- AP keeps track of holes in the packet sequence received from the sender
- When a dupack is received from the receiver (CH), AP compares the dupack sequence number with the recorded holes
  - if there is a match, an ELN bit is set in the dupack
- •When sender (MH) receives dupack with ELN set, it retransmits packet, but does not reduce congestion window.



2.2 Selective Acknowledgement (TCP SACK)

- Motivation: with cumulative ACKs, sender might not have enough information to recover from multiple packet losses within one window
  - In wireless, clusters (bursts) of losses are likely to occur

- TCP SACK allows multiple packet losses to be indicated by a single acknowledgement
  - lists last segment up to which all packets have been received (as in a normal ACK)
  - can list up to 3 blocks of data that have been received
  - with this information, sender should be able to retransmit packets without generating timeout

Example

- Prior to selective acknowledgment, if TCP lost packets 4 and 7 out of an 8-packet window, TCP would receive acknowledgment of only packets 1, 2, and 3. Packets 4 through 8 would have to be resent.

- With selective acknowledgment, TCP receives acknowledgment of packets 1, 2, 3, 5, 6, and 8. Only packets 4 and 7 have to be resent.

TCP SACK Option format

	Type	Length
Left edge of 1st block		
Right edge of 1st block		
Left edge of 2nd block		
Right edge of 2nd block		
Left edge of 3rd block		
Right edge of 3rd block		

i.e., packets received
are LeftEdge – RightEdge – 1;
lost packet = RightEdge

TCP SACK

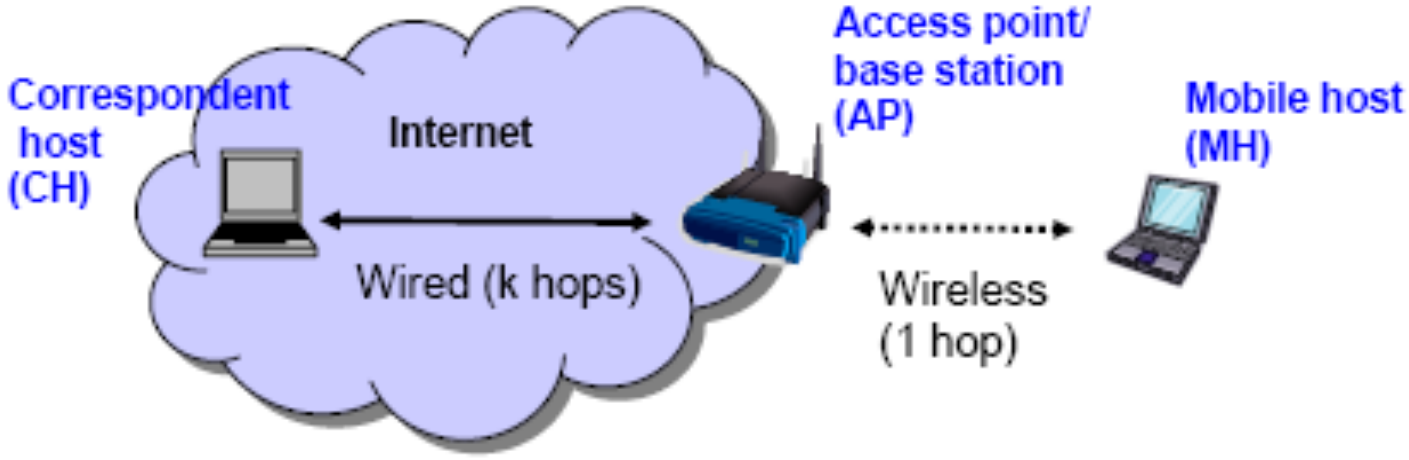
- Improve performance in the event that multiple packets are lost from one TCP window of data.

- Without this feature, TCP uses cumulative acknowledgments.
  - TCP sender could learn about only one lost packet per round trip time.
  - An aggressive sender could choose to retransmit packets early, but such retransmitted segments might have already been successfully received.

- With this feature, the receiving TCP host returns selective acknowledgment packets to the sender, informing the sender of data that has been received.
  - In other words, the receiver can acknowledge packets received out of order.
  - The sender can then retransmit only the missing data segments (instead of everything since the first missing packet).

Summary

- Need extra knowledge on wireless side to detect loss due to wireless/mobility effects that is unconnected to congestion.
  - MH and/or AP may know such information.
- Approaches modify TCP on MH or introduce a support protocol on AP (or both).
  - Doing anything on AP contradicts end-to-end principle.



Summary of Approaches

- **Group 1: Mask wireless loss from the TCP sender**
  - So that TCP sender will not reduce congestion window unnecessarily
- **Group 2: Explicitly notify the TCP sender about the cause of packet loss**
  - TCP sender will not reduce congestion window for wireless losses
- **Solutions implemented at**
  - TCP sender
  - TCP receiver
  - Intermediate node (wireless basestation, wifi access point)

Summary

- Transport layer in wireless networks can be very complex due to additional causes of packet loss
  - Need to ensure TCP works well since its so prevalent

- A variety of approaches for improving performance
  - Advantages of some:
    - Isolating retransmissions to the wireless link
    - Not modifying the sender
  - Disadvantages of some:
    - Breaking the end-to-end design of the Internet
    - Making handoffs difficult

TCP Over Wireless Ad-hoc Networks