A Comparison of Mechanisms for Improving TCP Performance over Wireless Links

Hari Balakrishnan Venkata N. Padmanabhan

Srinivasan Seshan Randy H. Katz

Daedalus Group,
Department of EECS,
University of California at Berkeley

Outline of Talk

- Motivation
- Description of protocols
- Experimental methodology
- Results and explanations
- Conclusions and future work

Problems with TCP over Wireless Links

- TCP: reliable byte-stream protocol with cumulative acknowledgments and retransmissions.
- Packet losses due to wireless bit-errors mistaken for congestion losses.
- Bulk losses cause coarse-granularity timeouts.
- Variable bandwidths and delays make transport protocol adaptation hard.
- Handoffs often cause packet loss and variable delays, resulting in coarse timeouts for connections.

Proposed Solutions

- Link-layer protocols
  - Locally optimized solutions.
  - Transport-aware link protocols.
- End-to-end protocols
  - Vanilla TCP (TCP Reno).
  - Selective acknowledgments based on SMART scheme.
  - Explicit Loss Notifications (ELN) to make sender aware of non-congestion losses.
- Split-connection protocols
  - Attempt isolation of source from wireless link by splitting TCP connection at base station.

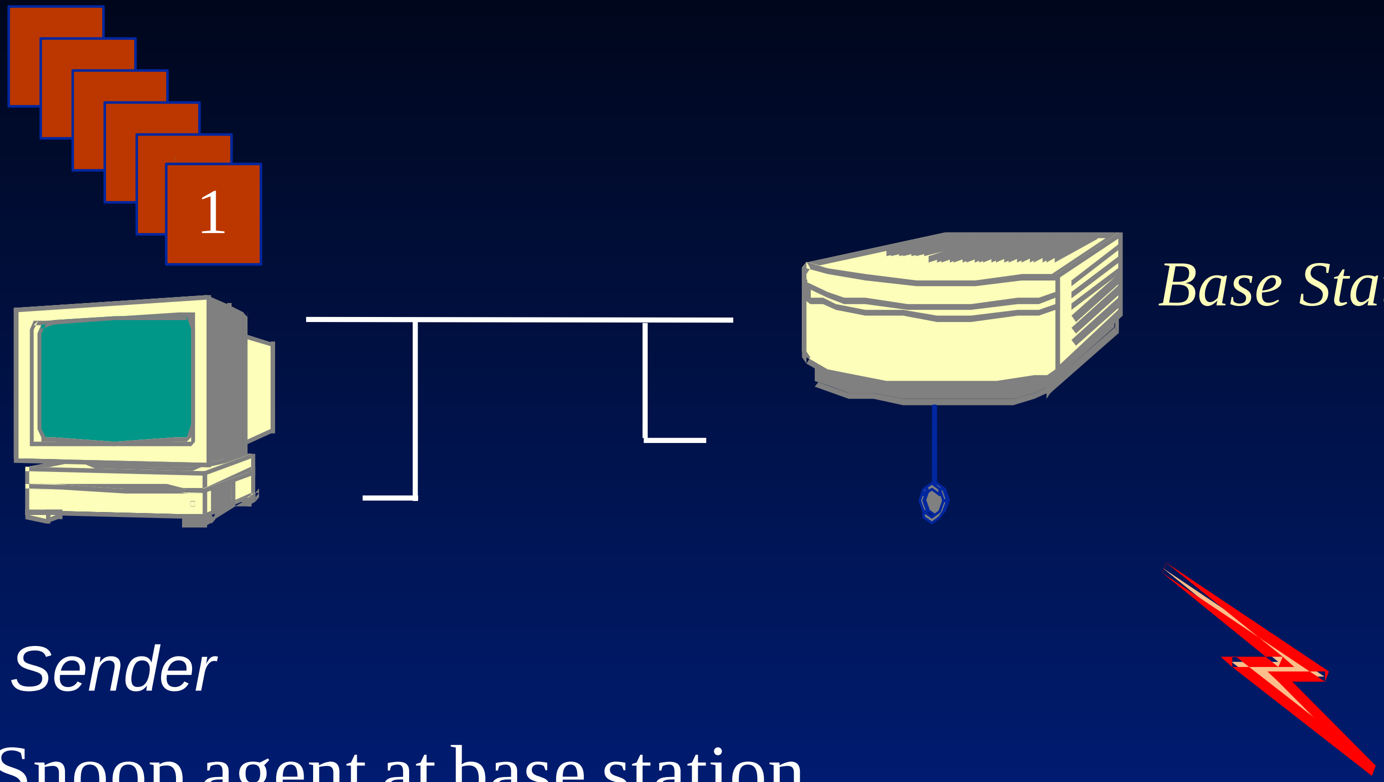
Objectives

- To evaluate and compare performance of protocols:
  - Best combination of mechanisms in each protocol class.
  - Importance of TCP-awareness for link-layer protocols.
  - Usefulness of selective acknowledgments and explicit loss notifications.
  - Effectiveness of split connections.
- Performance metrics:
  - *throughput:* number of bytes/transfer time (Mbps).
  - *goodput :* number of *useful* bytes/total number of bytes sent over link (%age).
- Context: bulk data transfer to a mobile host connected over wired links and one wireless hop.

Main Results

- A reliable link-layer protocol with some TCP-awareness provides very good performance.
- Splitting TCP connections is *not* essential for good performance; using unmodified TCP over wireless hop does not improve performance much.
- Selective acknowledgments and explicit loss notifications are very effective in recovering from wireless losses.

Protocols: TCP-Aware Link-layer



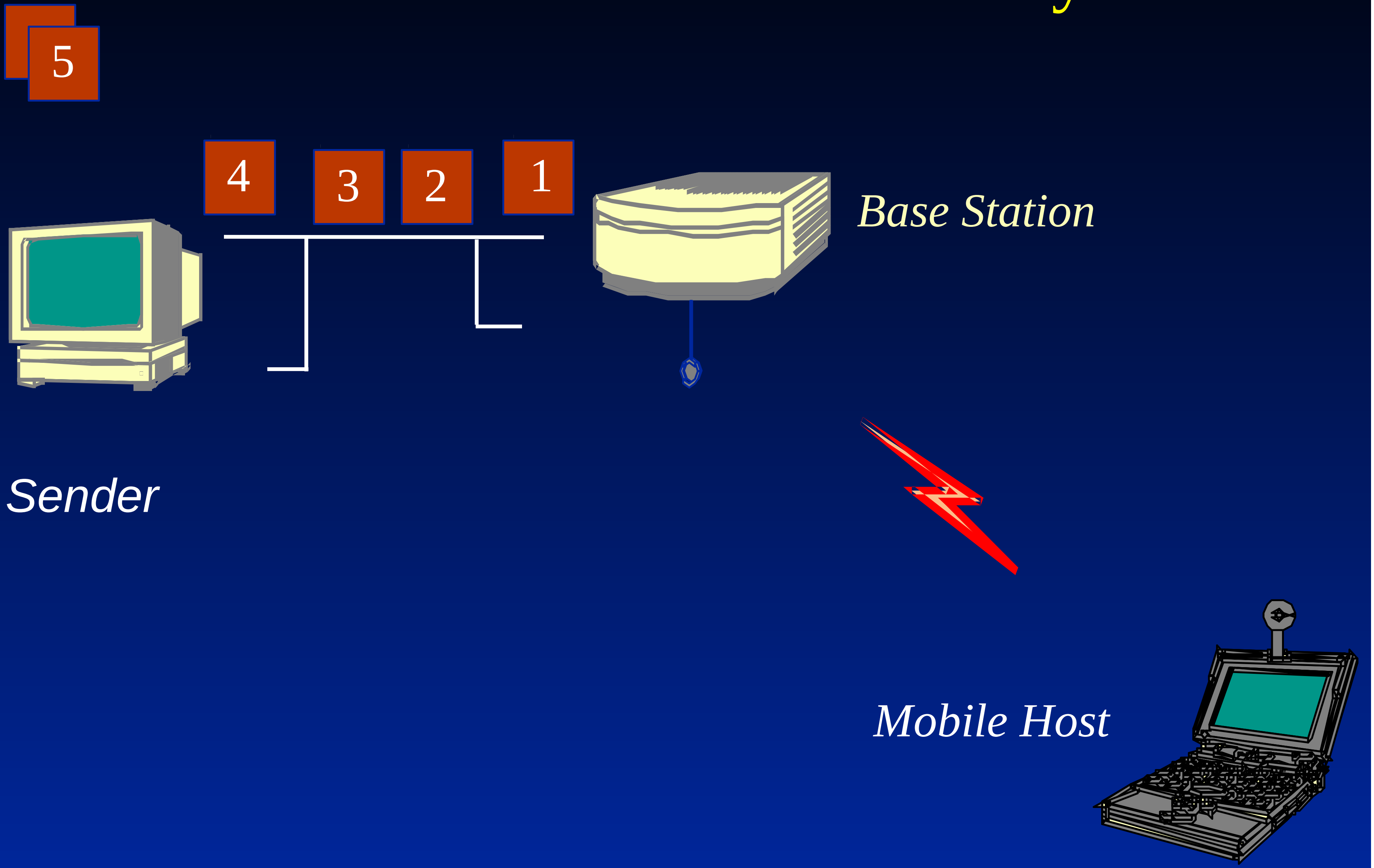
Snoop agent at base station.

Caches TCP segments going to mobile.

Local retransmissions by observing duplicate acks and timeouts.

No extra messaging for good performance.

Protocols: TCP-Aware Link-layer



5

4

3

2

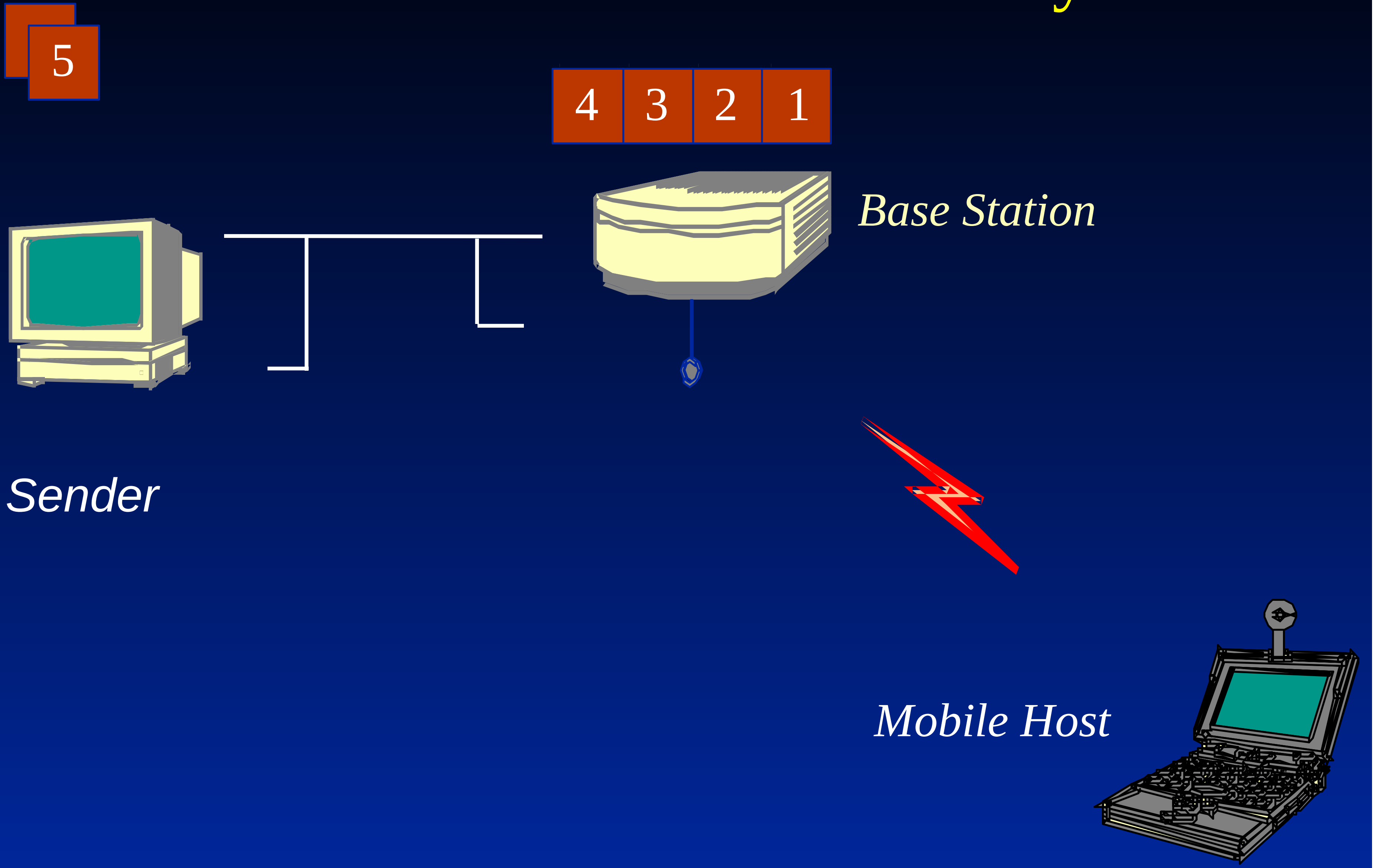
1

Base Station

Sender

Mobile Host

Protocols: TCP-Aware Link-layer



5

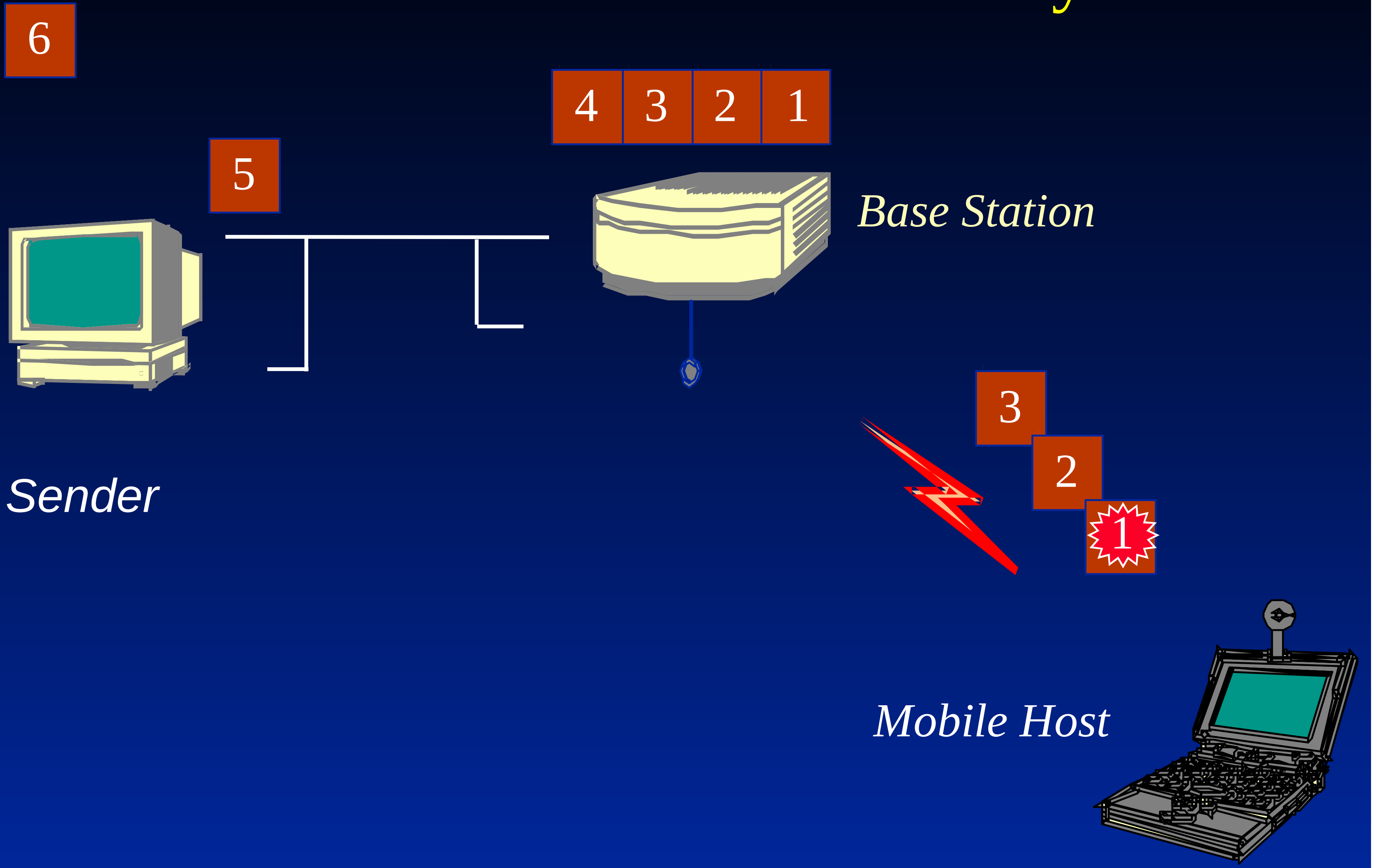
4 3 2 1

Base Station

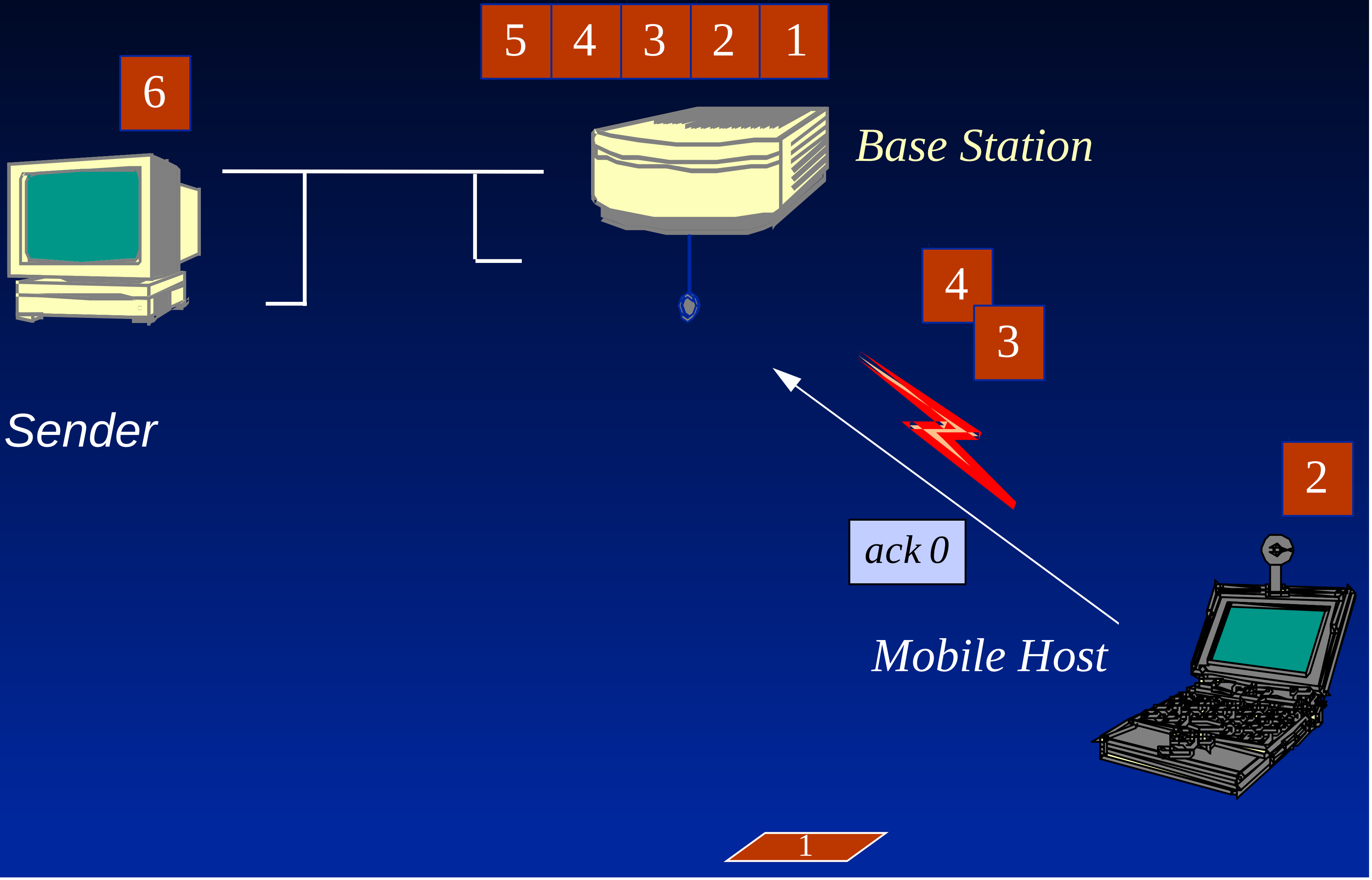
Sender

Mobile Host

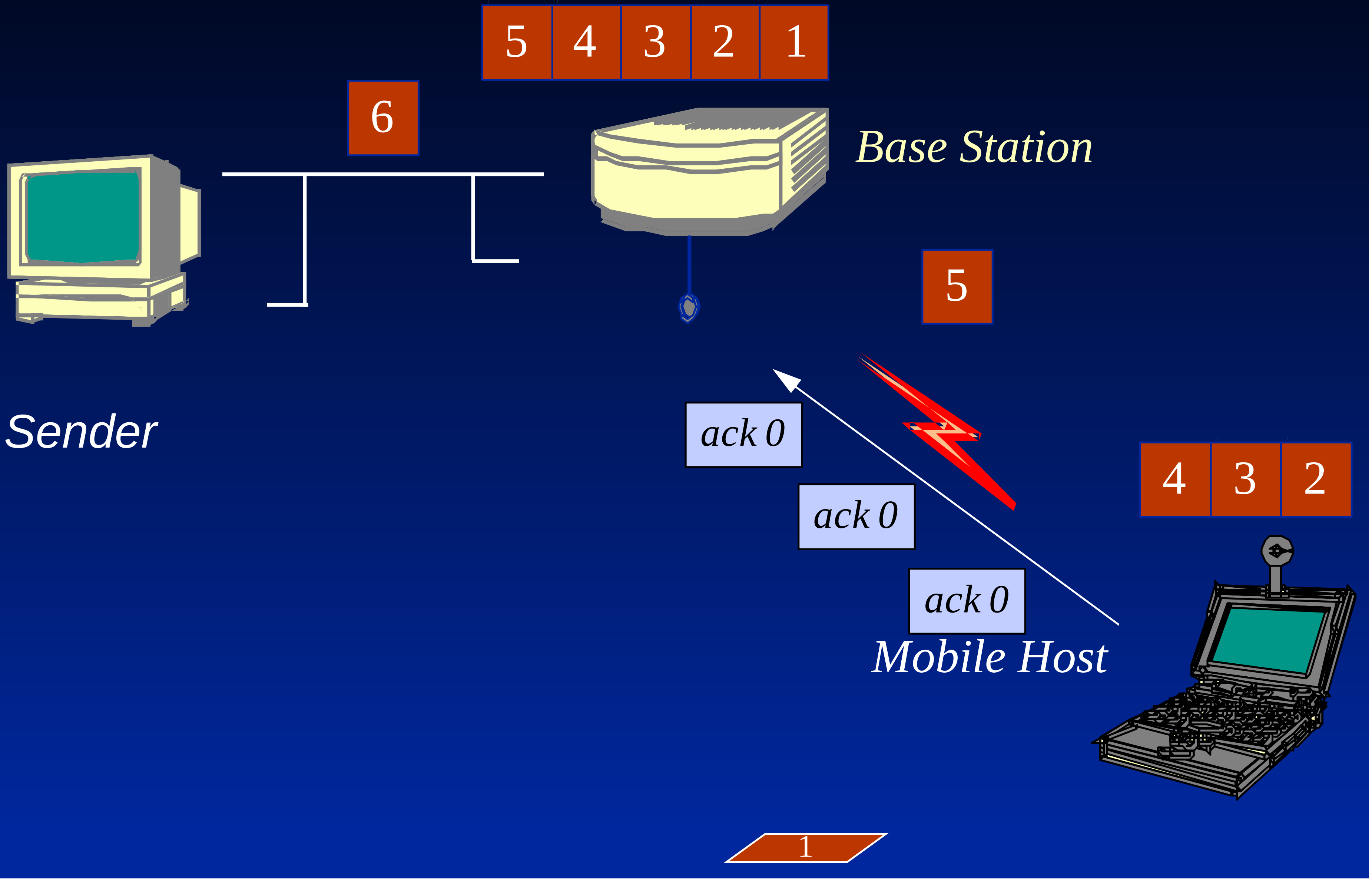
Protocols: TCP-Aware Link-layer



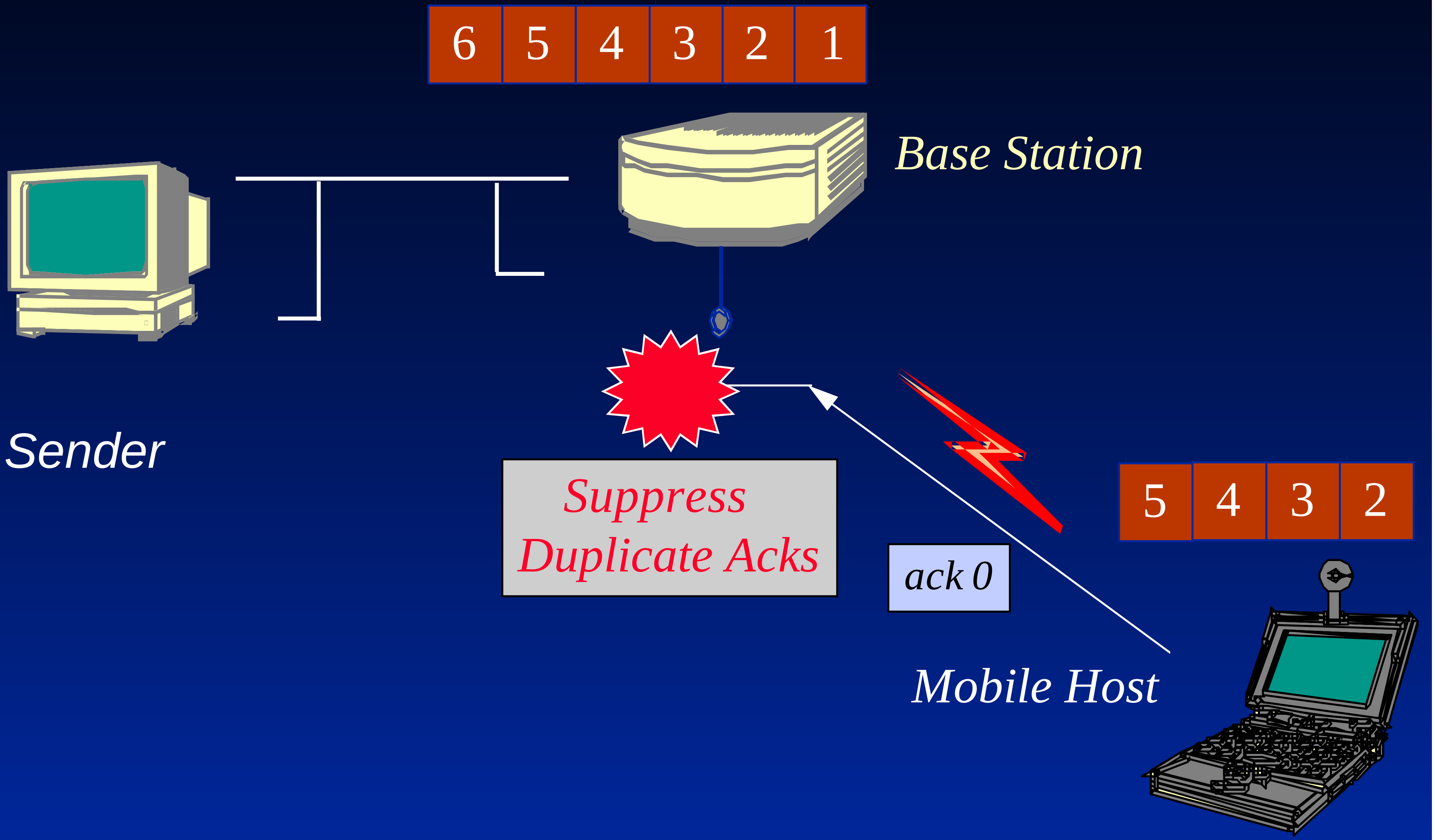
Protocols: TCP-Aware Link-layer



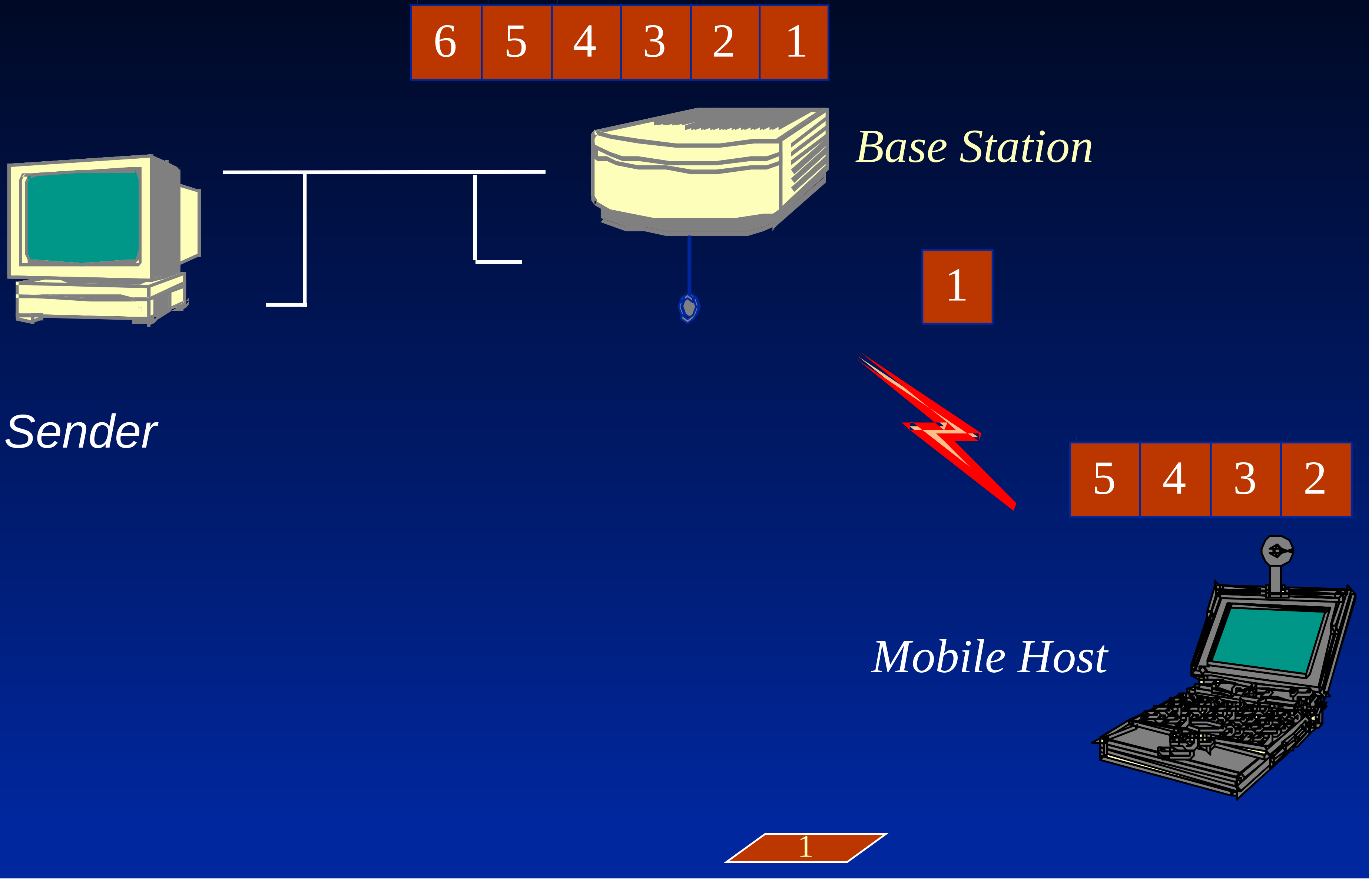
Protocols: TCP-Aware Link-layer



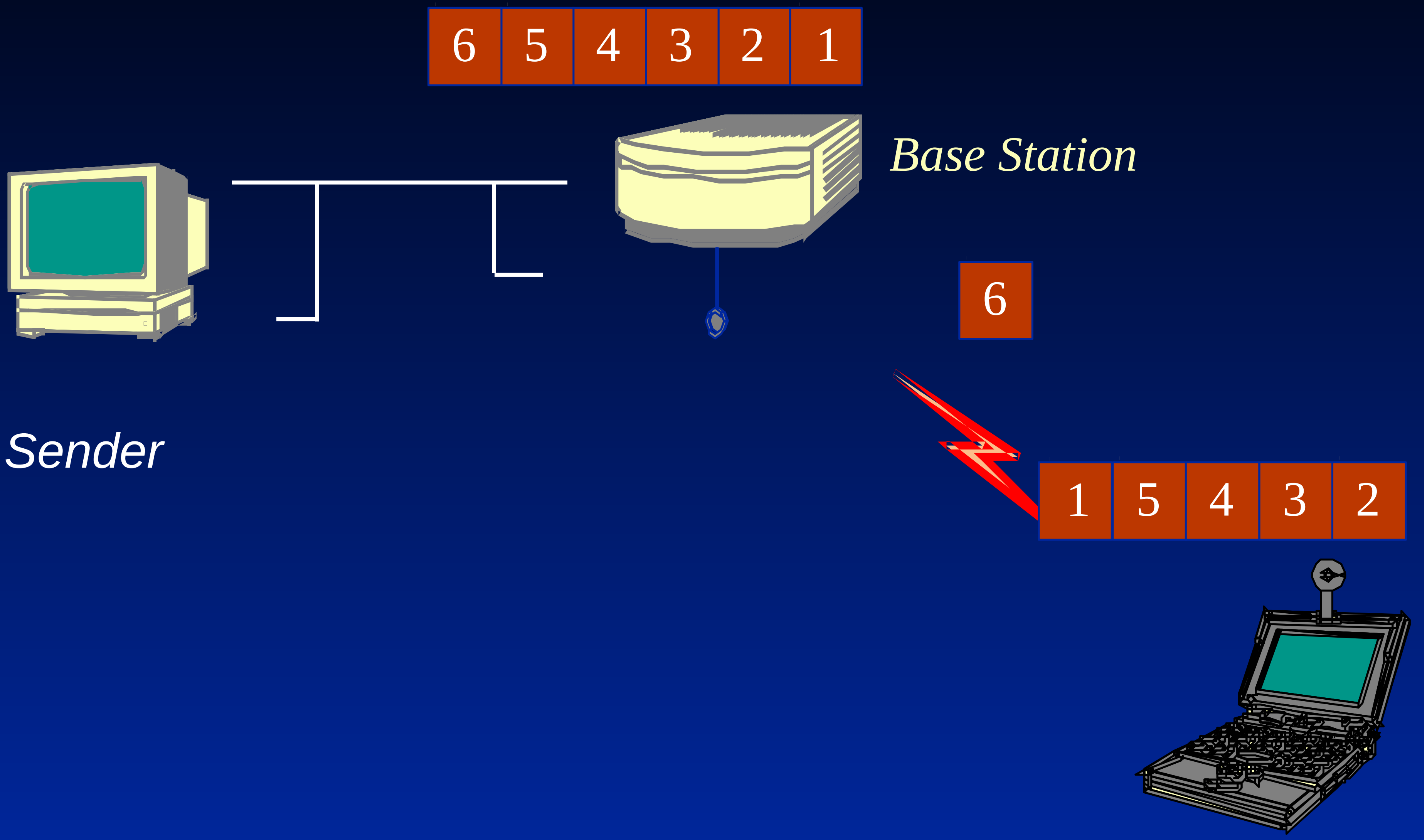
Protocols: TCP-Aware Link-layer



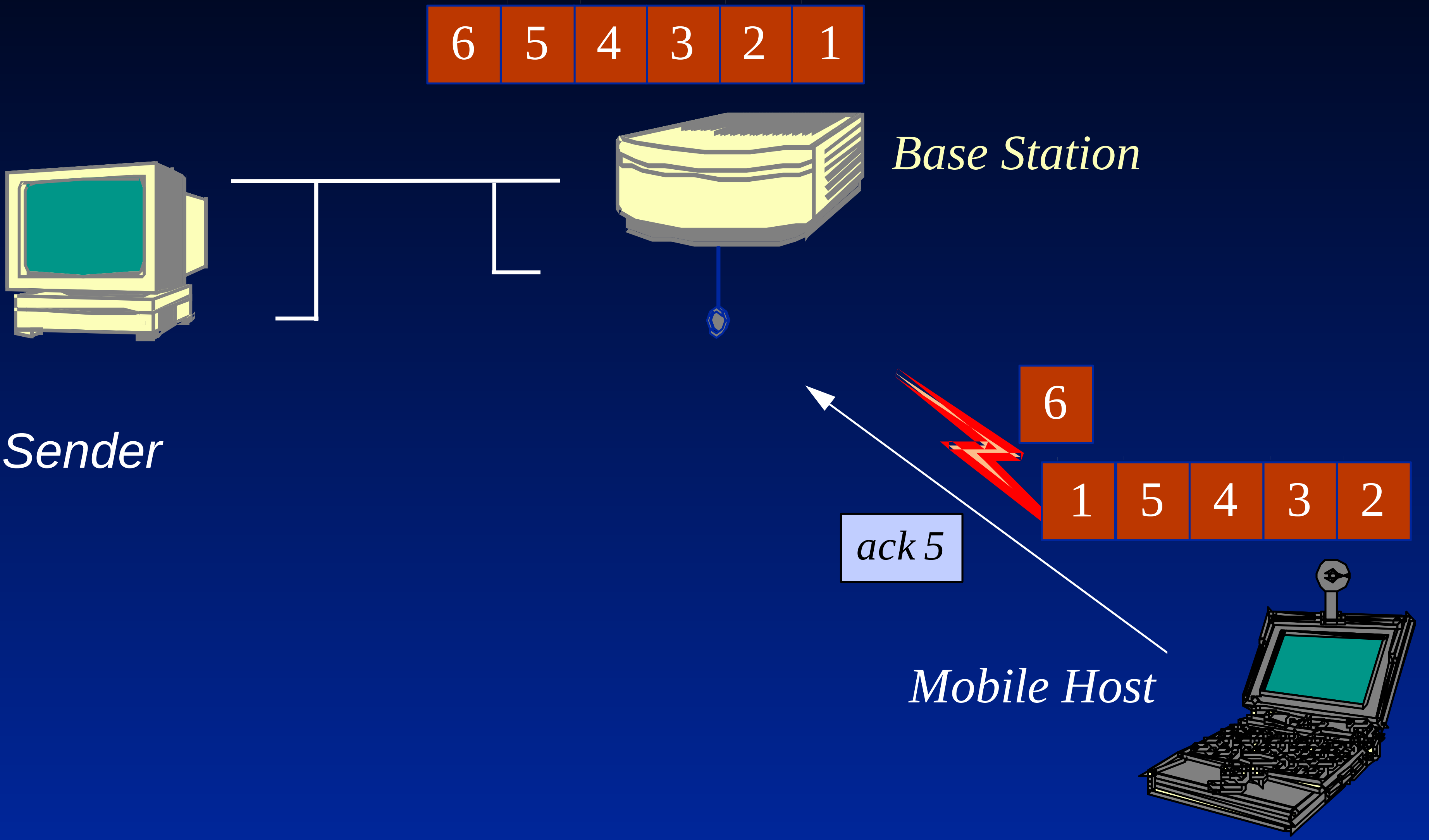
Protocols: TCP-Aware Link-layer



Protocols: TCP-Aware Link-layer



Protocols: TCP-Aware Link-layer



6 5 4 3 2 1

Base Station

Sender

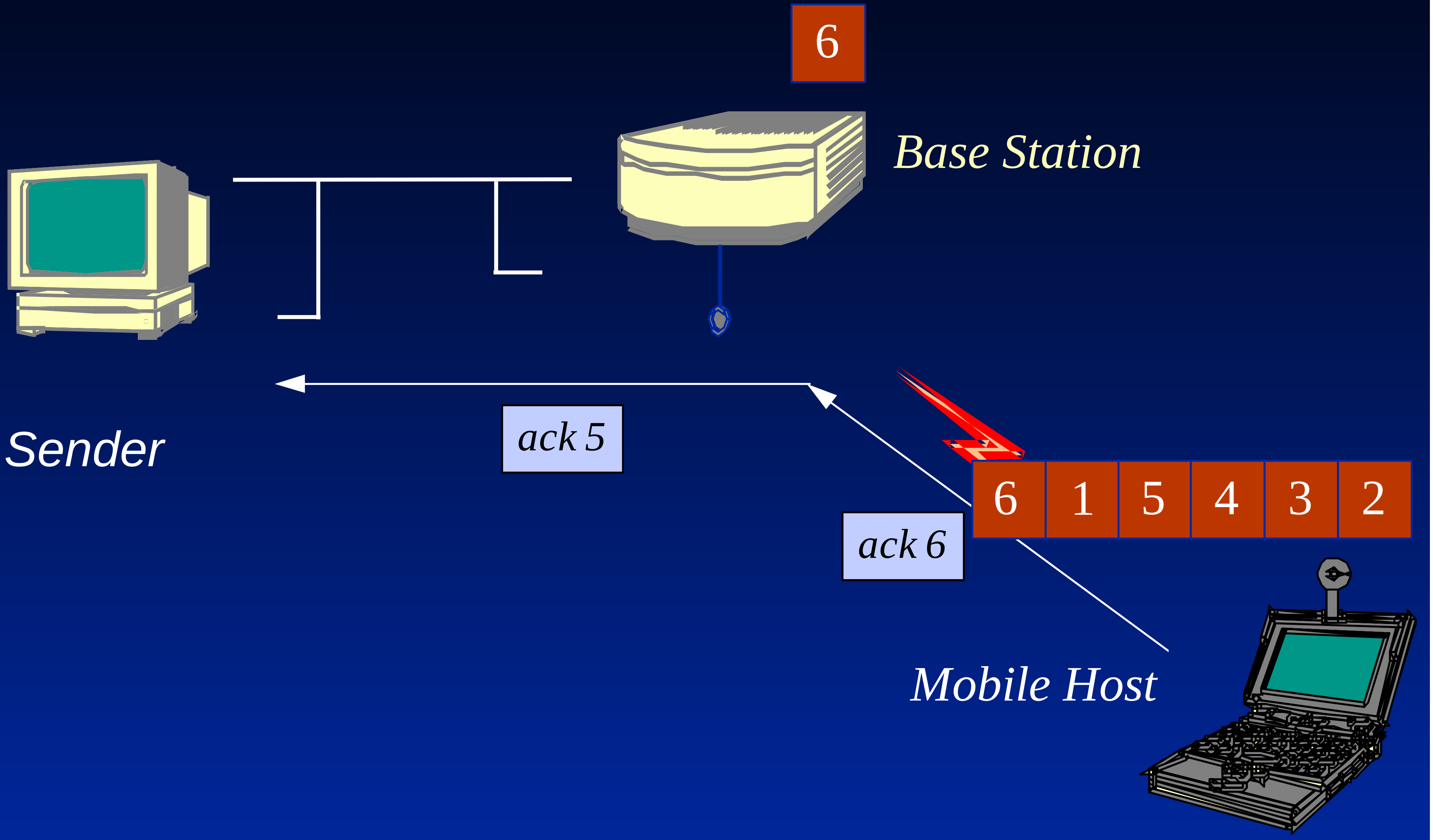
6

1 5 4 3 2

ack 5

Mobile Host

Protocols: TCP-Aware Link-layer



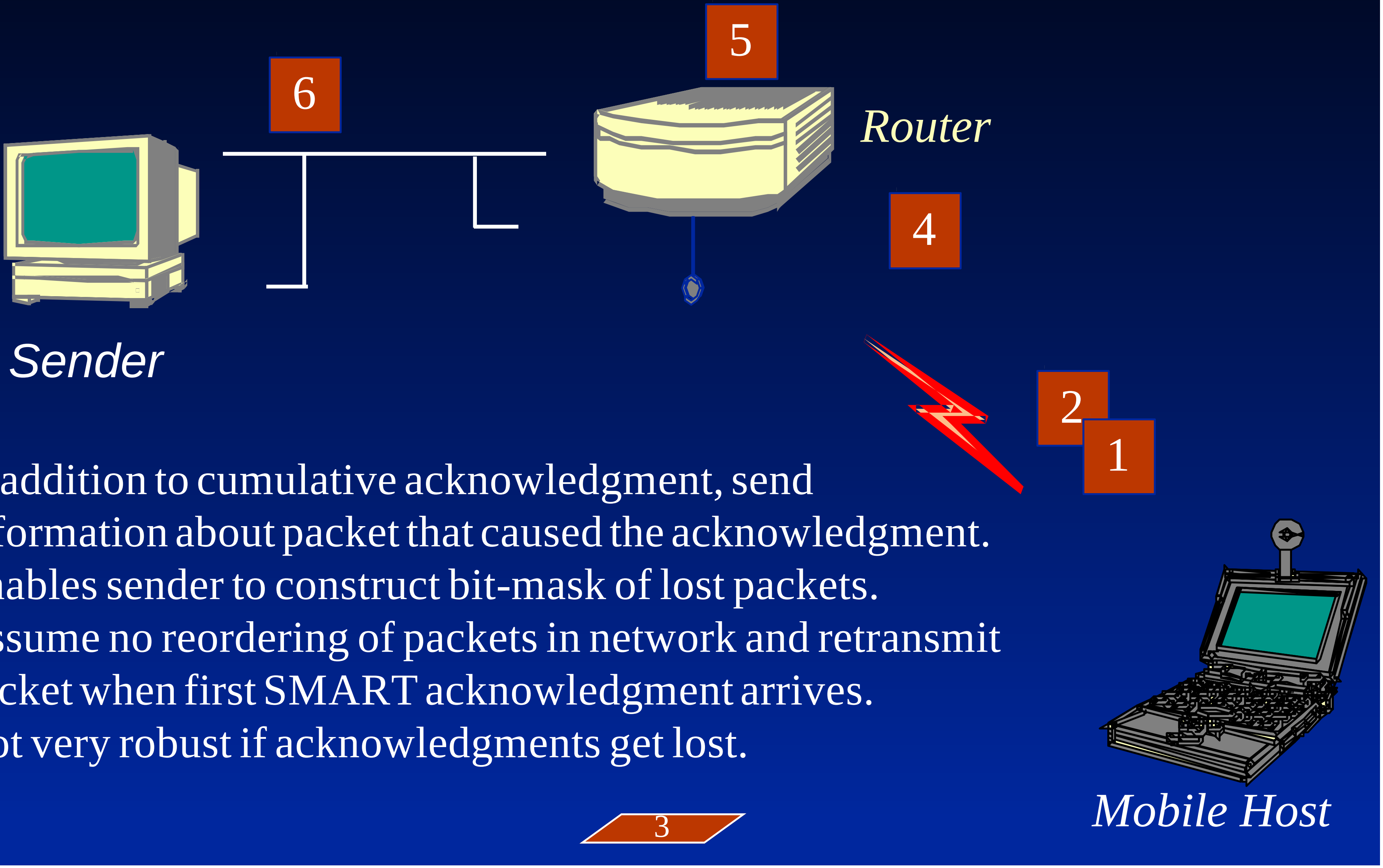
Protocols: Link-layer

- Link-layer protocol (LL) same as TCP-aware link layer, but without suppressing duplicate acknowledgments.
- LL retransmits packets on observing TCP duplicate acknowledgments and on last-hop timeouts.
- No extra protocol messaging.
- Out-of-order packet delivery.
- LL-SACK: uses selective acknowledgments over wireless link.
- LL-OPT: TCP-aware and also uses SACKs.

Protocols: End-to-End

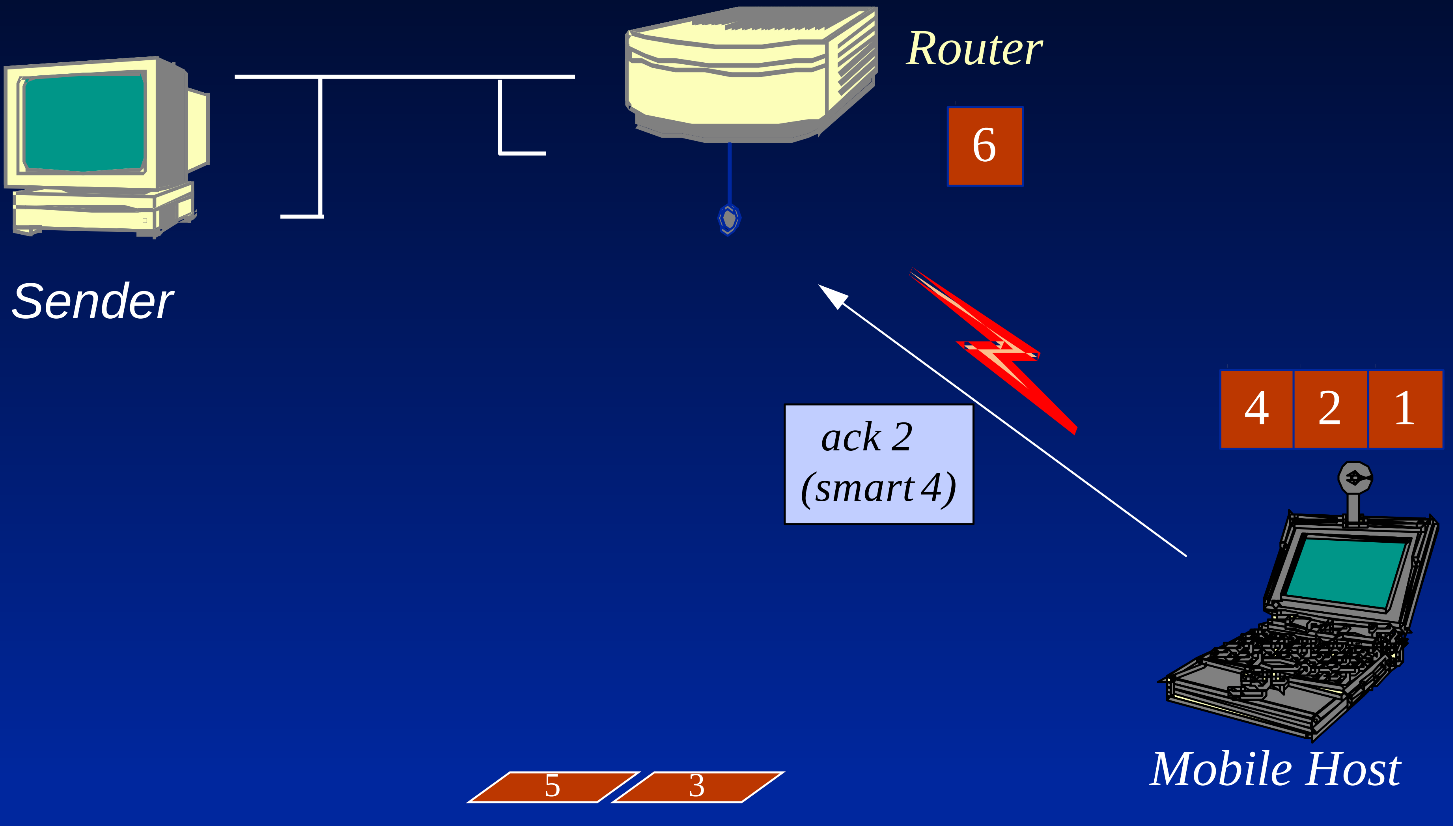
- TCP Reno: cumulative acknowledgments and coarse-granularity timeouts.
- Reno+SACK: SMART-based selective acknowledgments.
- Reno+ELN: Explicit Loss Notifications (ELN) generated at receiver when wireless losses occur and propagated to sender.

SMART-based Selective Acknowledgments

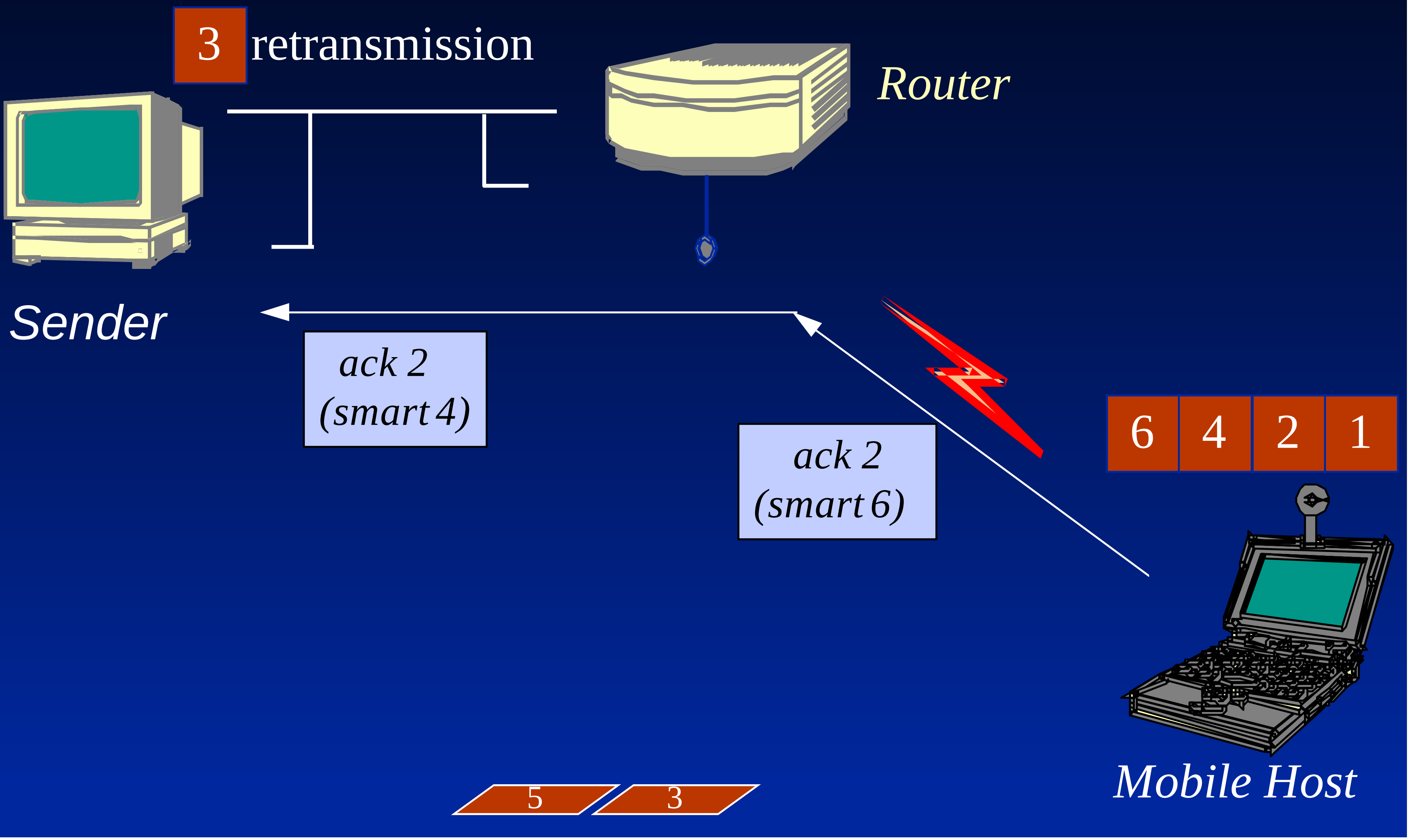


In addition to cumulative acknowledgment, send information about packet that caused the acknowledgment.
Enables sender to construct bit-mask of lost packets.
Assume no reordering of packets in network and retransmit packet when first SMART acknowledgment arrives.
Not very robust if acknowledgments get lost.

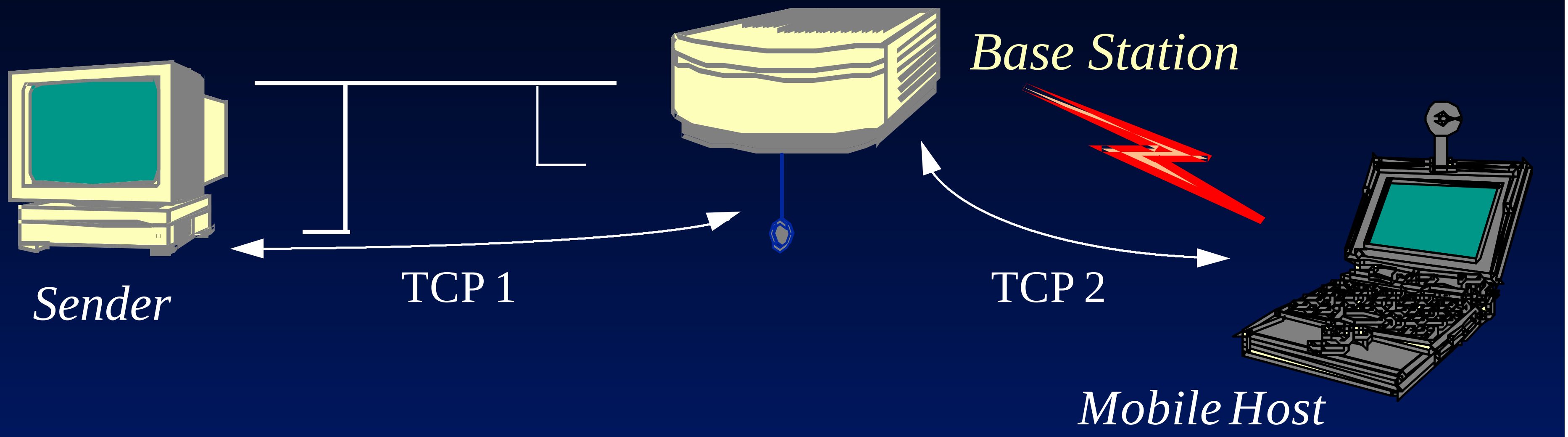
SMART-based Selective Acknowledgments



SMART-based Selective Acknowledgments

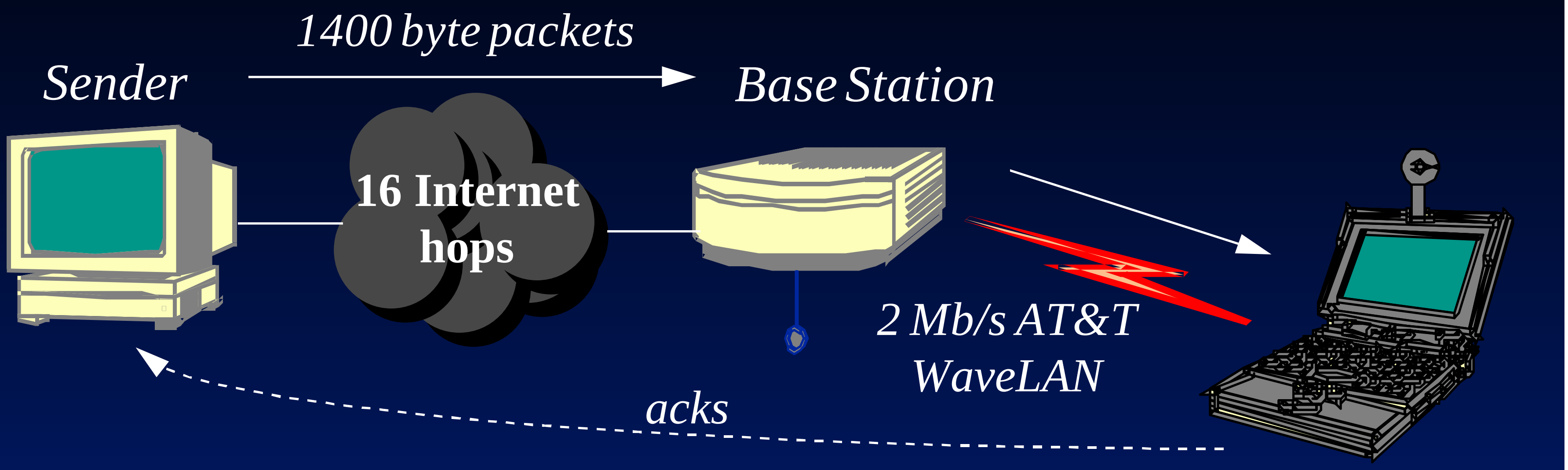


Description of Protocols: Split-Connections



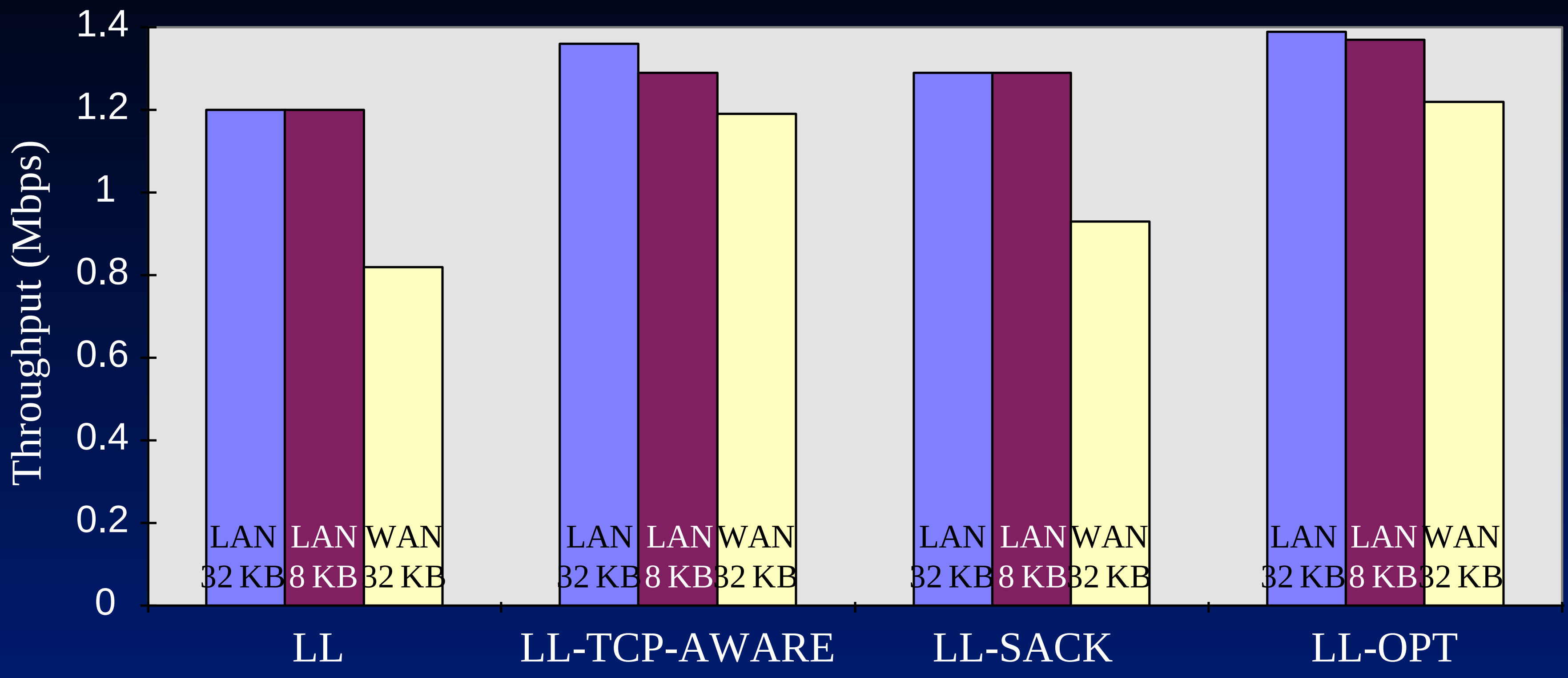
- Isolates sender from wireless link by splitting TCP connection at base station.
- Violates end-to-end semantics of TCP acknowledgments.
- Hard state at base station complicates handoffs and increases handoff latencies.
- SPLIT-SACK: Use selective acknowledgments over wireless connection.

Experimental Methodology



- LAN experiments with source on same 10 Mb/s Ethernet as base station.
- WAN experiments between IBM (NY) and UCB in the absence of congestion.
- Poisson-distributed bi-directional bit-errors (1 every 64 KB).
- Instrumented kernel to record timeouts, retransmissions, changes in congestion window, etc.

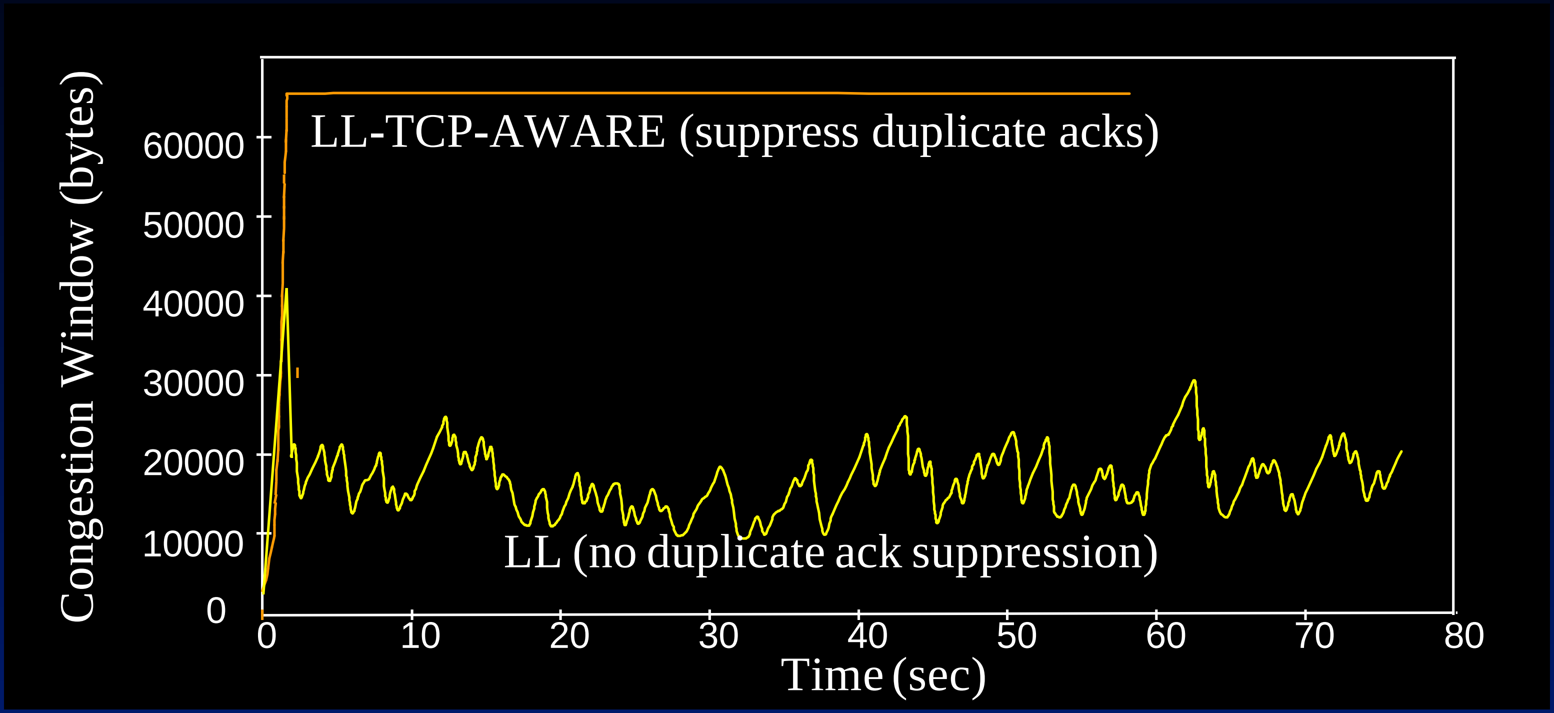
Experimental Results: Link-layer



- LAN performance almost the same for all LL protocols.
- Simple link-layer reliable protocols could adversely impact TCP performance.

Transport-aware link protocols perform well over lossy links.

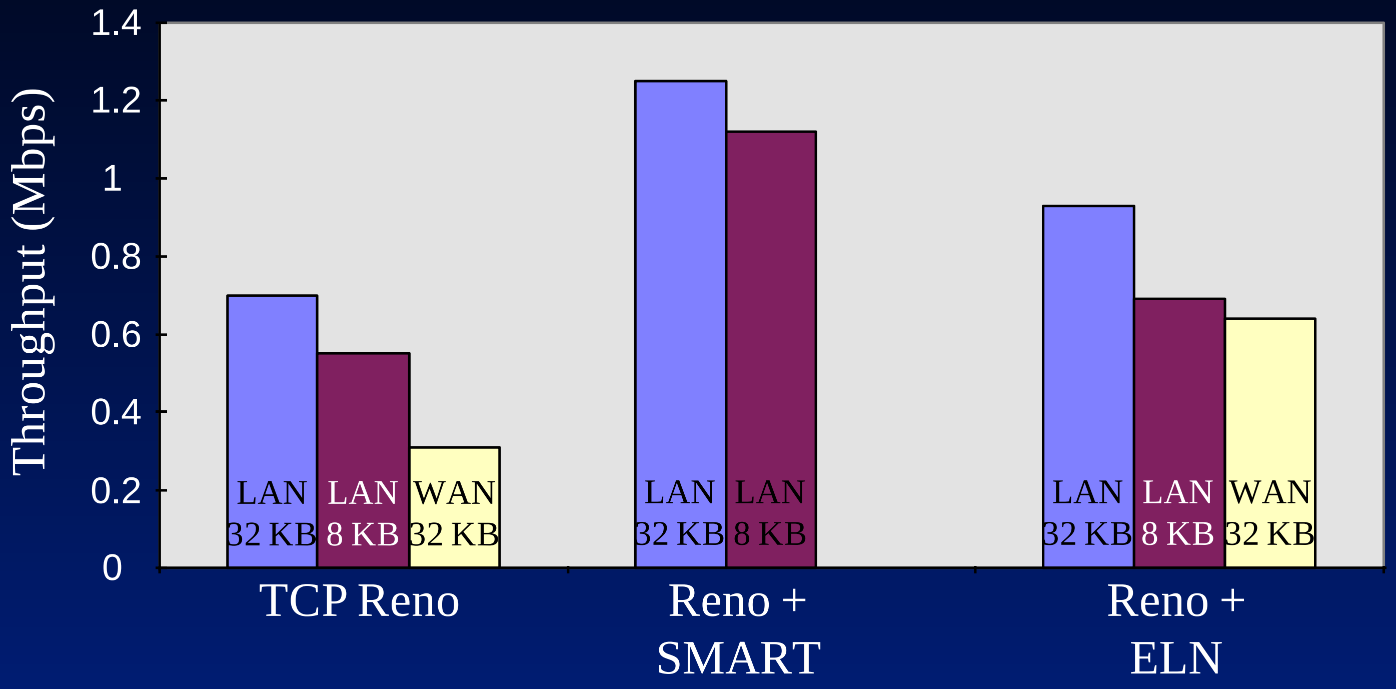
Benefits of TCP-Awareness



- 30% improvement for LL-TCP-AWARE: congestion window fluctuates rapidly for LL (no coarse timeouts occur).
- Connection bandwidth-delay product more than 20-25 KB.

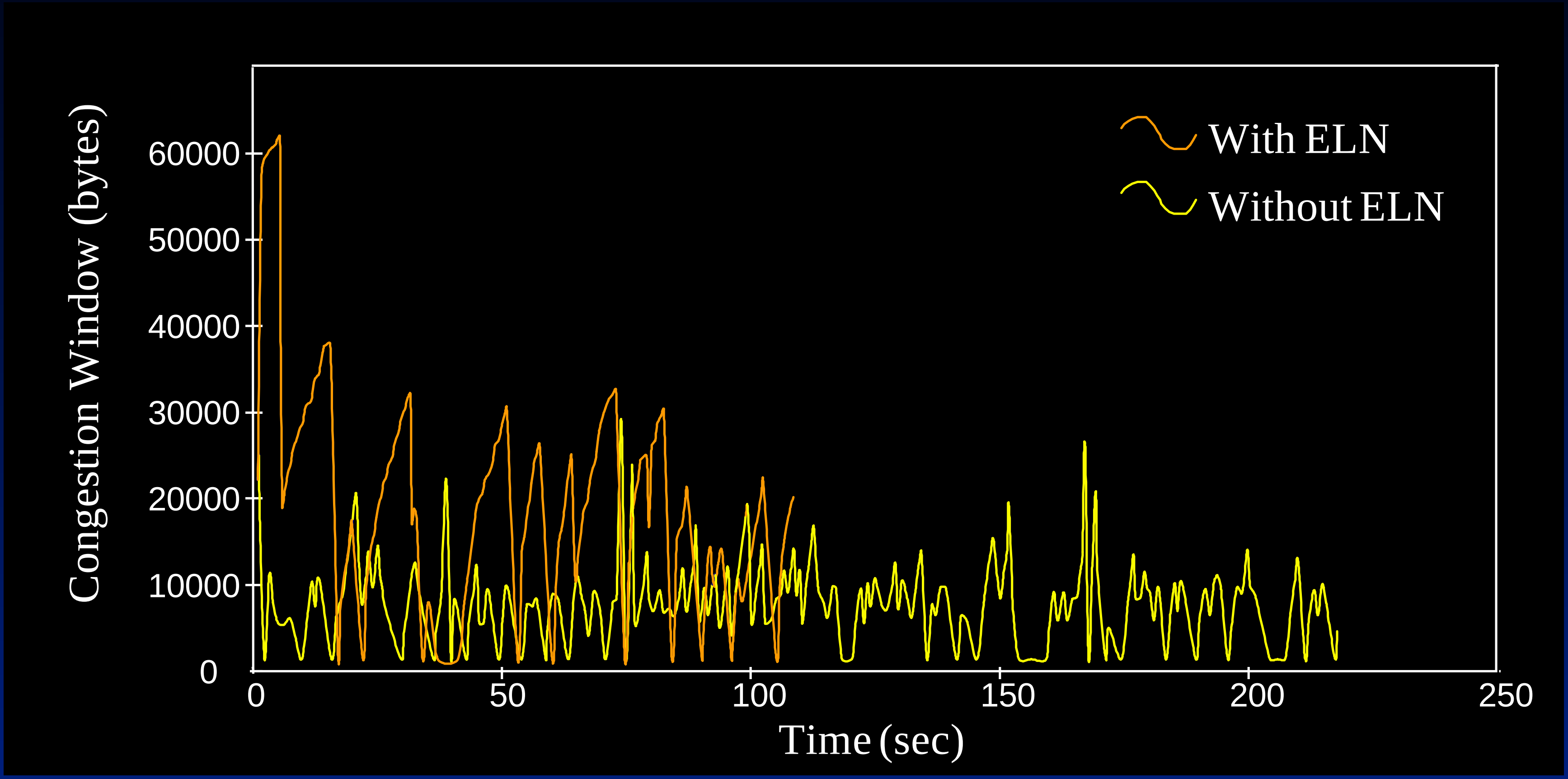
Suppressing duplicate acknowledgments and TCP-awareness leads to better utilization of link bandwidth and performance

Experimental Results: End-to-End



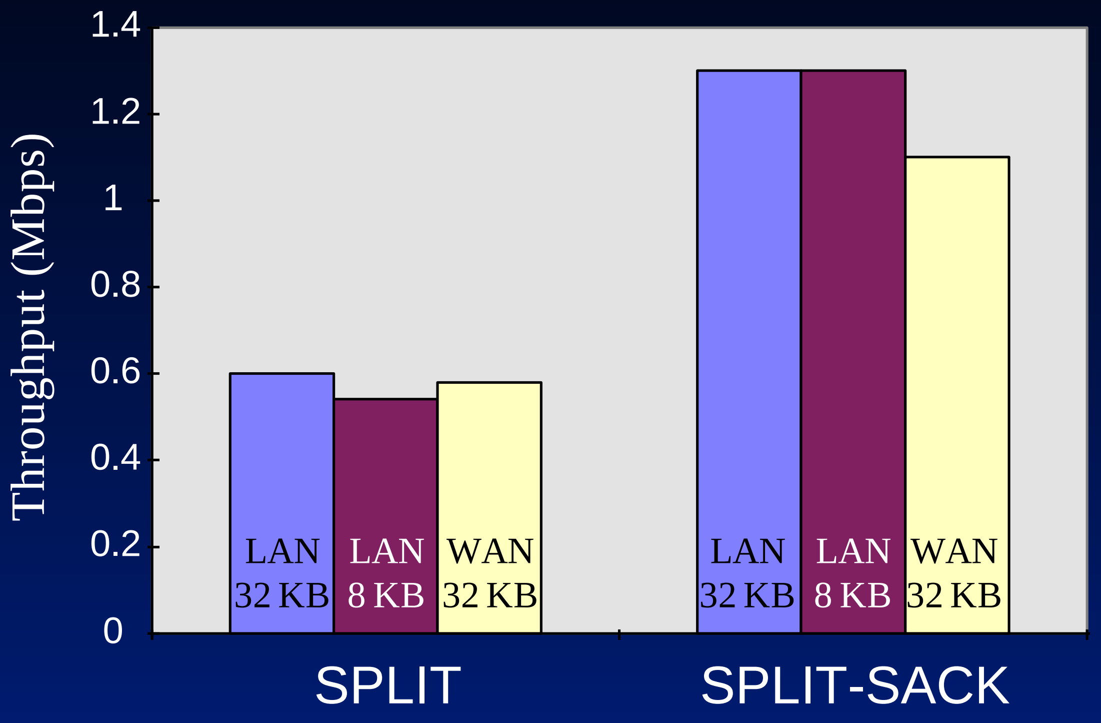
- Coarse timeouts impair throughput (50% of optimal in LAN, 25% in WAN); goodputs always 97.5%.
- Selective Acknowledgments and Explicit Loss Notifications significantly improve performance.

Benefits of ELN



Congestion window does not vary as rapidly with ELN, leading to a 100% improvement in throughput (using a simple message)

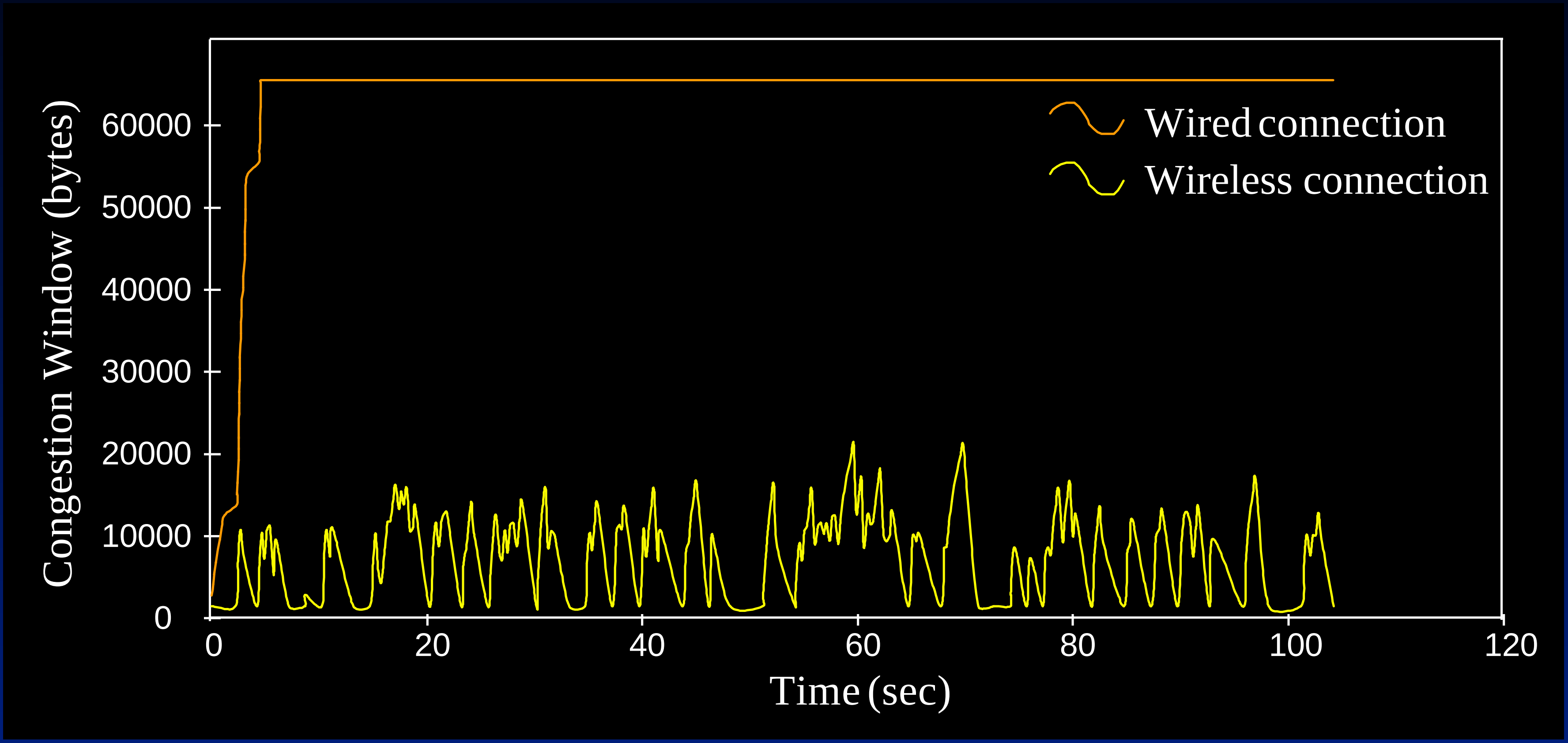
Experimental Results: Split-Connections



- SPLIT-SACK significantly better than SPLIT alone.
- Performance of SPLIT-SACK 5-10% less than LL-TCP-AWARE.

Splitting the connection is not essential for good performance

Split-Connection Congestion Window

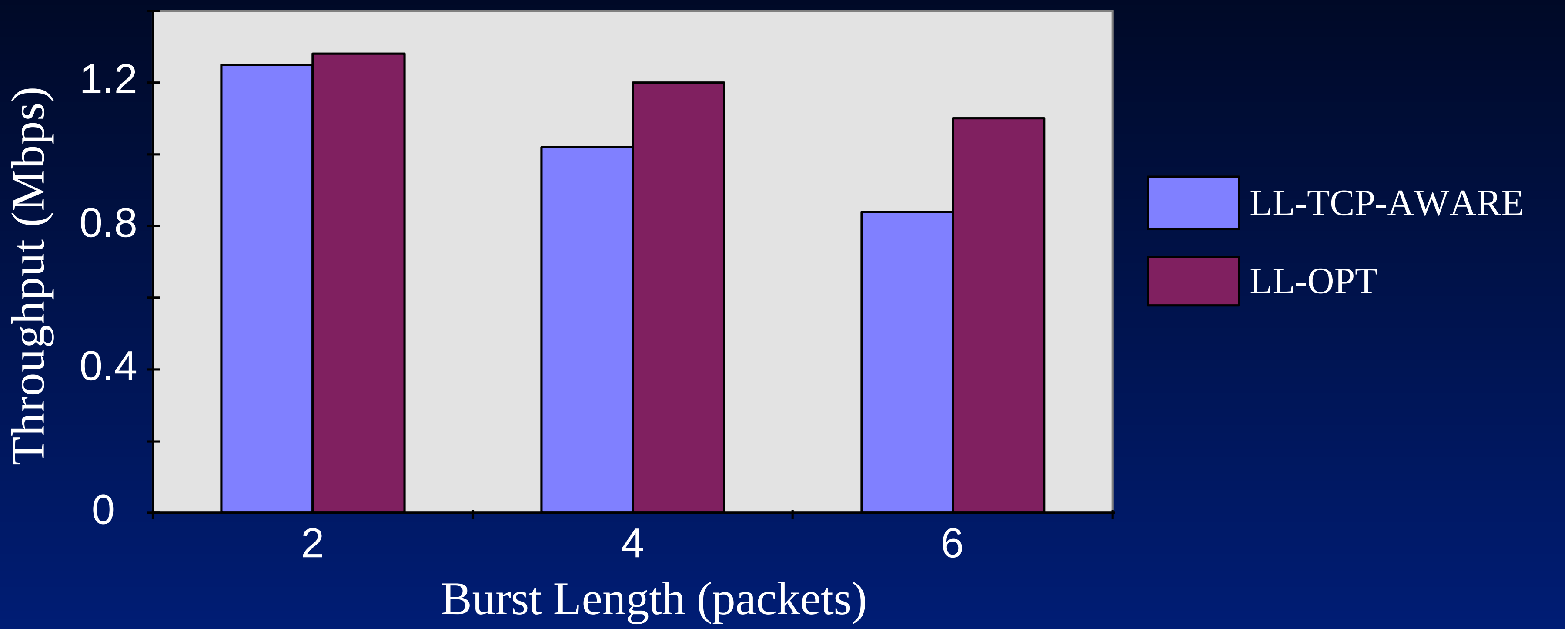


Wired connection does not shrink congestion window

but

wireless connection times out often, causing sender to stall

Burst Losses



While LL-TCP-AWARE can recover from small amounts of burst loss, LL-OPT uses SACKs to perform better loss recovery

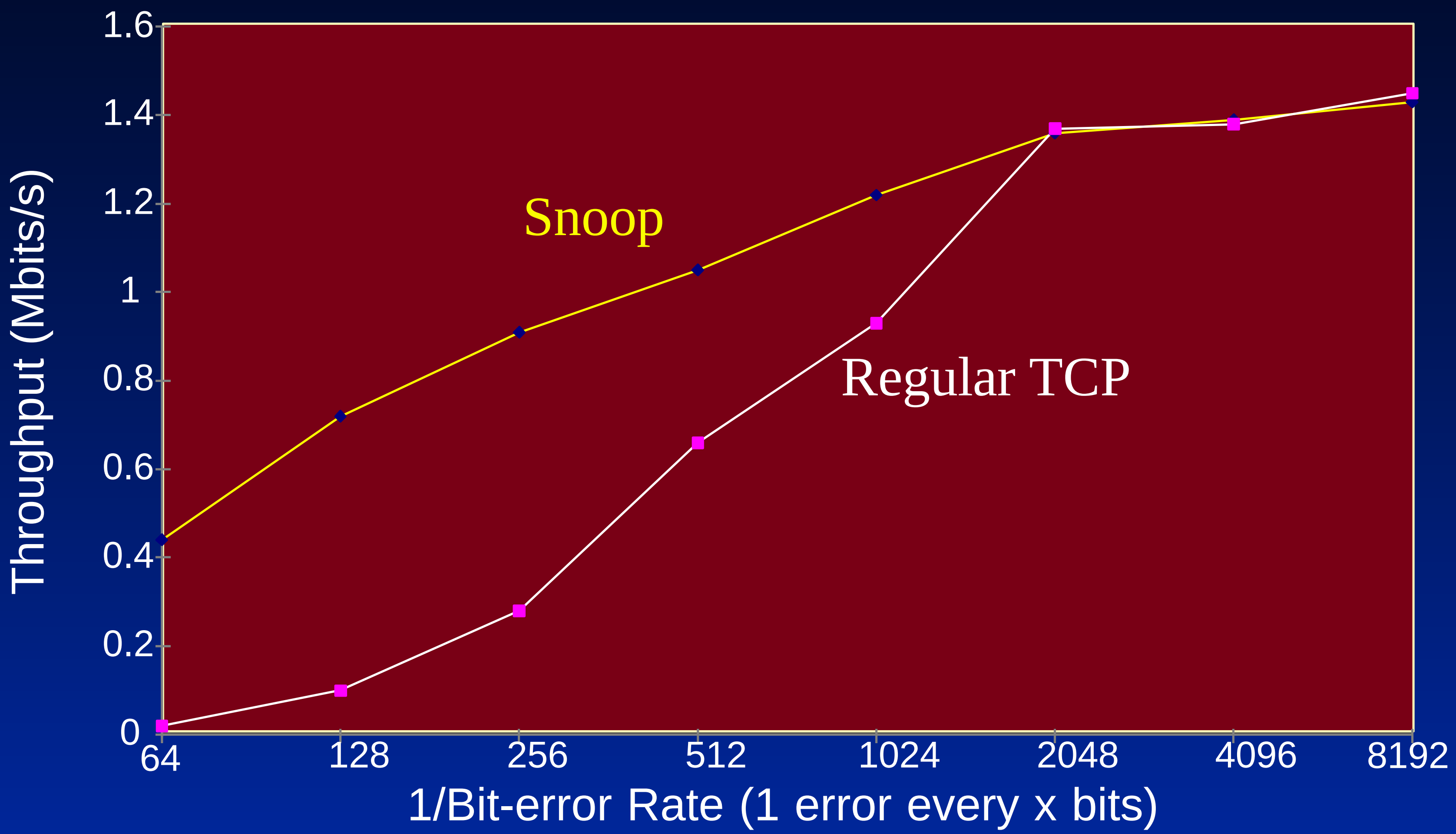
Conclusions

- A reliable link-layer protocol with some TCP-awareness provides very good performance.
- Splitting TCP connections is *not* essential for good performance; using unmodified TCP over wireless hop does not improve performance much.
- Selective acknowledgments and explicit loss notifications are very effective in recovering from wireless losses.

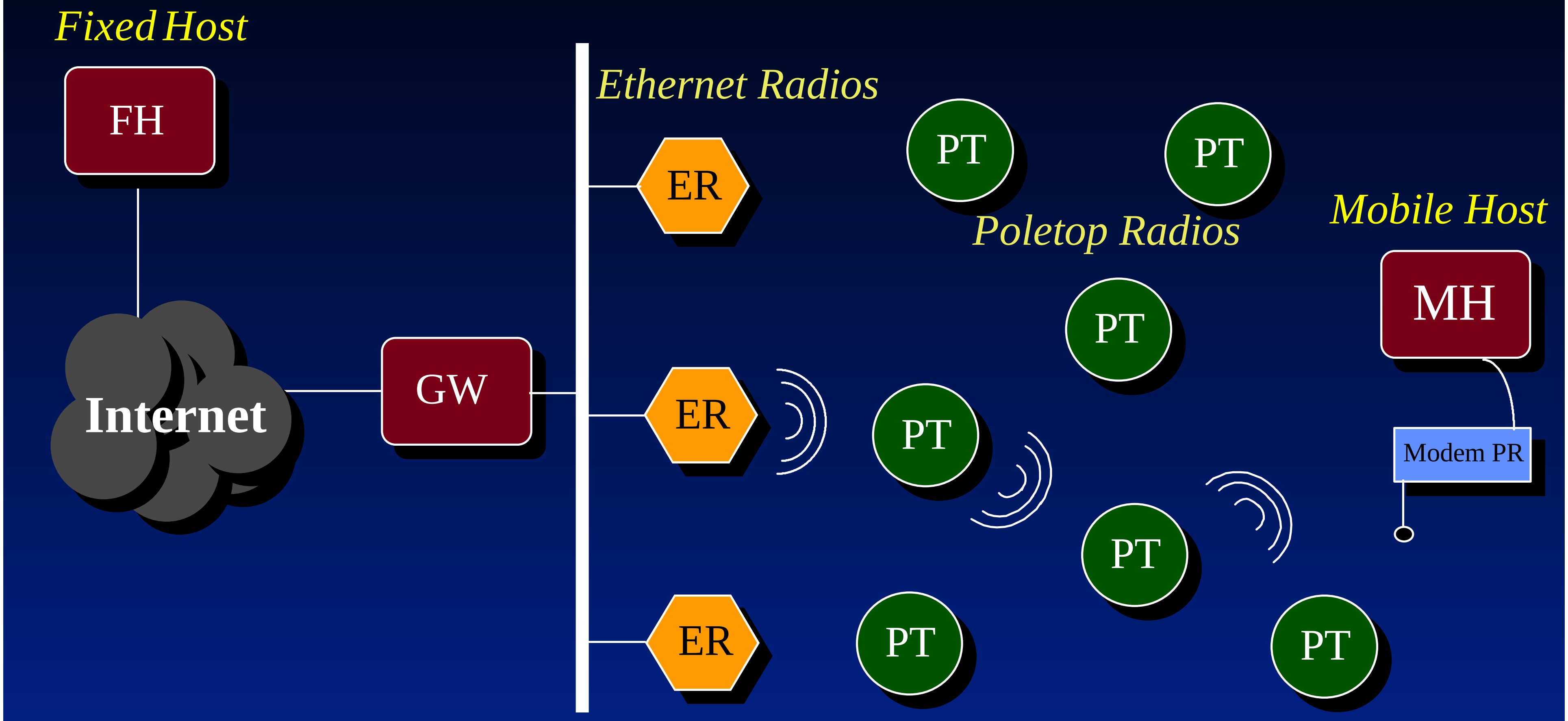
Future Work

- Evaluate performance of IETF SACK proposal, especially over wireless and satellite networks.
- Performance and protocol improvements for multi-hop packet radio networks (large variations).
- Improved reliable transport protocols for asymmetric (and possibly lossy) connections.
- More sophisticated link-layer protocols.

Snoop Performance Improvement



Large Round-Trip Variations



Large round-trip time variations due to variable latencies and contention