
TCP in Wireless Mobile Networks

3/24/2004

Based on <http://zoo.cs.yale.edu/classes/cs434/lectures/WirelessTCP.ppt>

Outline

- ❑ *Introduction to transport layer*
- ❑ Introduction to TCP (Internet) congestion control
- ❑ Congestion control in wireless networks

Transport Layer v.s. Network Layer

- ❑ Network layer: connection or no-connection.
- ❑ Connection-based: phone network
 - Before end-to-end transmission, a call has to be setup.
 - Resources are reserved for this session.
 - After transmission, the call is hung up. Resources are released.
 - Quality of service is guaranteed, e.g., delay.
 - Good for continuous data rate.

Transport Layer v.s. Network Layer

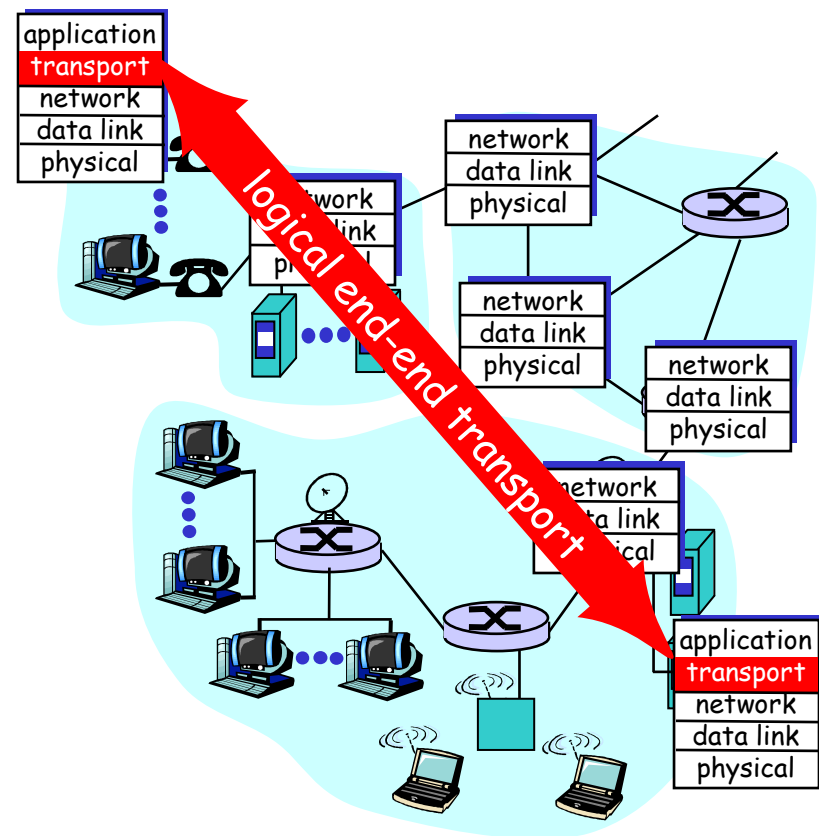
- ❑ Network layer: connection or no-connection.
- ❑ Connection-less: IP
 - No connection is established.
 - Packets are routed independently.
 - Packets may arrive out of order, lost, etc.
 - Good for busty traffic, e.g., web browsing.
 - Good for resource sharing.
 - More reliable under link/router failures. Packets are re-routed.

Transport Layer v.s. Network Layer

- ❑ Network layer, e.g., IP
 - Finding a route + delivering packets
- ❑ Obvious issues to be dealt with:
 - Packets may get lost.
 - Packets may arrive out-of-order.
 - Packets may be corrupted.
- ❑ Other issues:
 - Data rate? How fast should the source input packets to the network?
 - Congestion control.

Transport Layer vs. Network Layer

- ❑ Provide *logical communication* between app' processes
- ❑ Transport protocols run in end systems
- ❑ **Transport vs. network layer services:**
 - *network layer*
 - data transfer between end systems
 - *transport layer*
 - data transfer between processes
 - relies on, enhances, network layer services



Transport Layer Services and Protocols

- ❑ Transport services
 - multiplexing/demultiplexing
 - flow control
 - reliable data transfer
 - *congestion control*
- ❑ Transport protocols in the Internet
 - UDP
 - TCP

Outline

- ❑ Introduction to transport layer
 - *Introduction to TCP (Internet) congestion control*
- ❑ Congestion control in wireless networks

History

- ❑ TCP congestion control in mid-1980s
 - sliding window congestion protocol
 - use a sliding window to control the number of outstanding packets
 - fixed window size W

- ❑ Congestion collapse in the mid-1980s
 - UCB \leftrightarrow LBL throughput dropped by 1000X!

Principles of Congestion Control

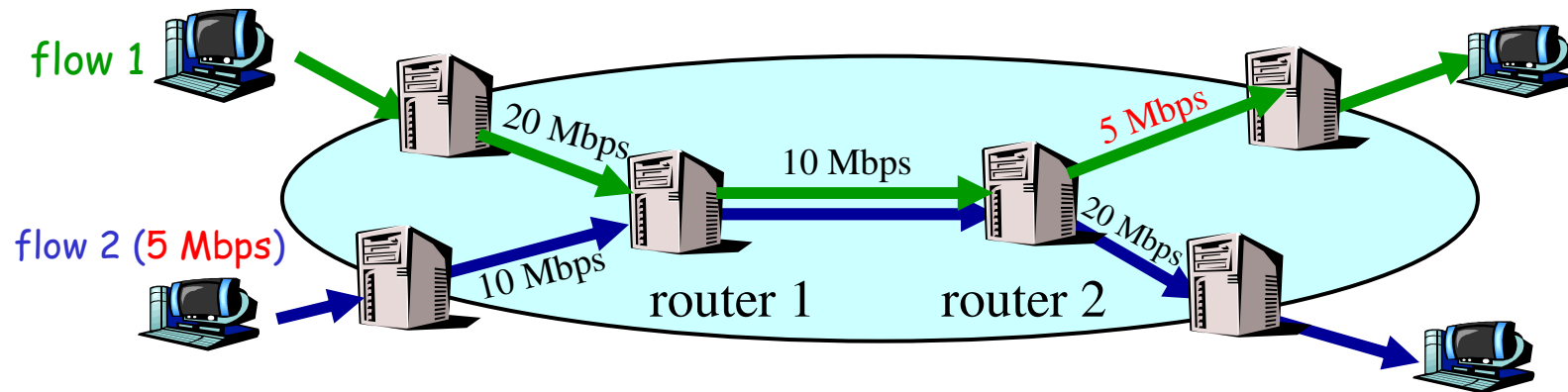
Big picture:

- ❑ How to determine a flow's sending rate?

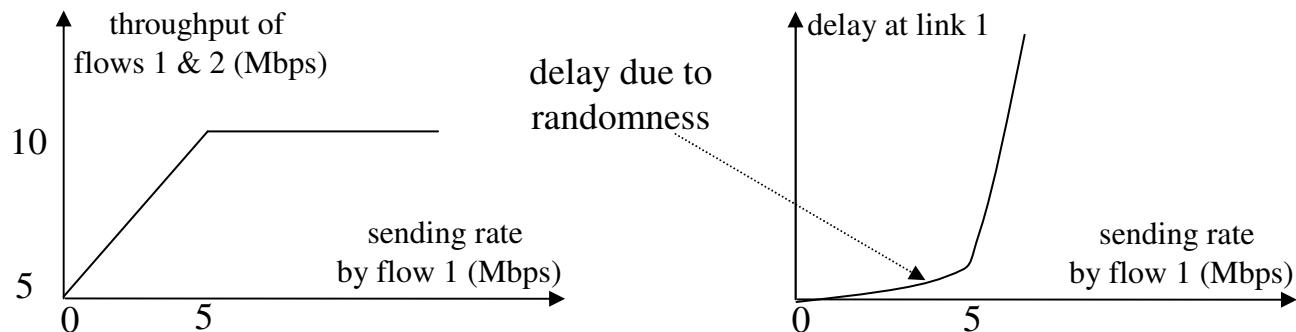
Congestion:

- ❑ informally: "too many sources sending too much data too fast for the *network* to handle"
- ❑ different from flow control!
- ❑ manifestations:
 - lost packets (buffer overflow at routers)
 - wasted bandwidth
 - long delays (queueing in router buffers)

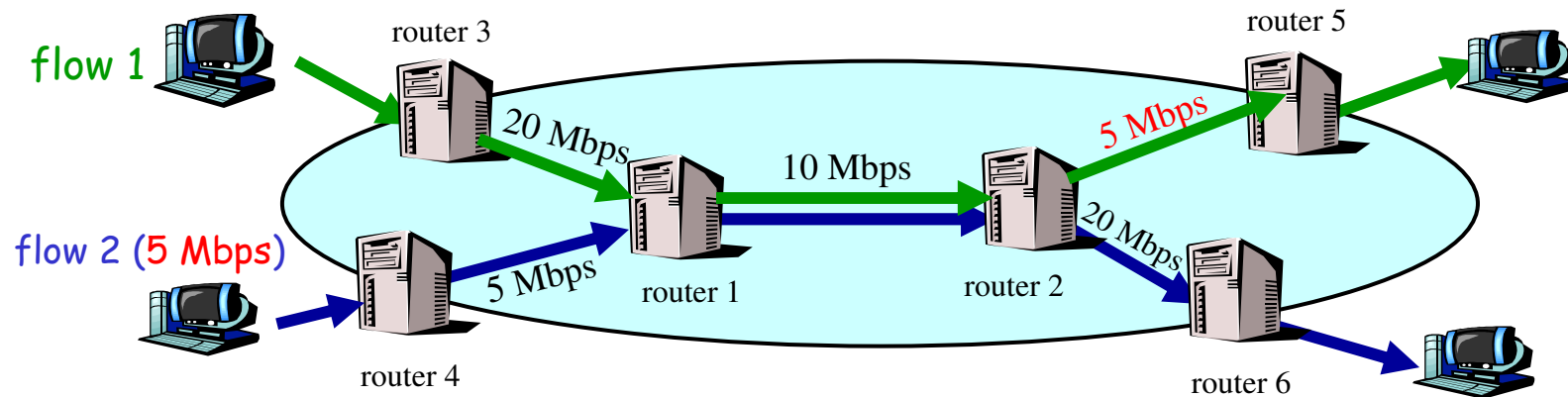
Cause/Cost of Congestion: Scenario 1



- ☐ Flow 2 has a fixed sending rate of 5 Mbps
- ☐ We vary the sending rate of flow 1 from 0 to 20 Mbps
- ☐ Assume
 - no retransmission
 - the link from router 1 to router 2 has **infinite** buffer
 - Performance metric: throughput (packets that go through in unit time)

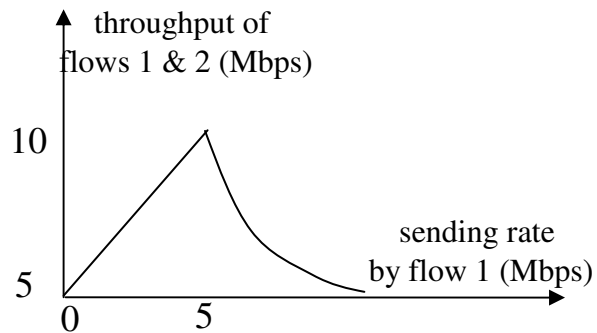


Cause/Cost of Congestion: Scenario 2



□ Assume

- no retransmission
- the link from router 1 to router 2 has **finite** buffer

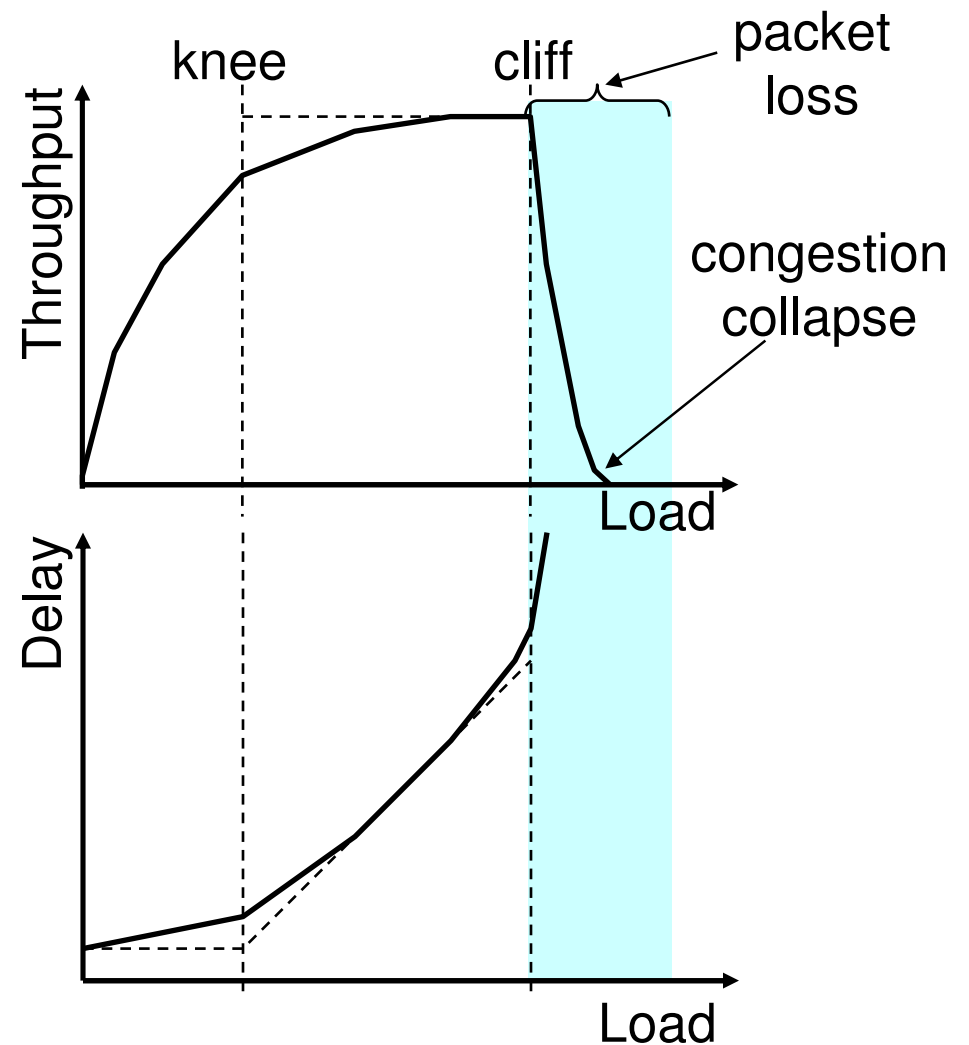


- when packet dropped at the link from router 2 to router 5, the upstream transmission from router 1 to router 2 used for that packet was wasted!

Summary: The Cost of Congestion

Cost

- ❑ High delay
- ❑ Packet loss
- ❑ Wasted upstream bandwidth when a pkt is discarded at downstream
- ❑ Wasted bandwidth due to retransmission (a pkt goes through a link multiple times)

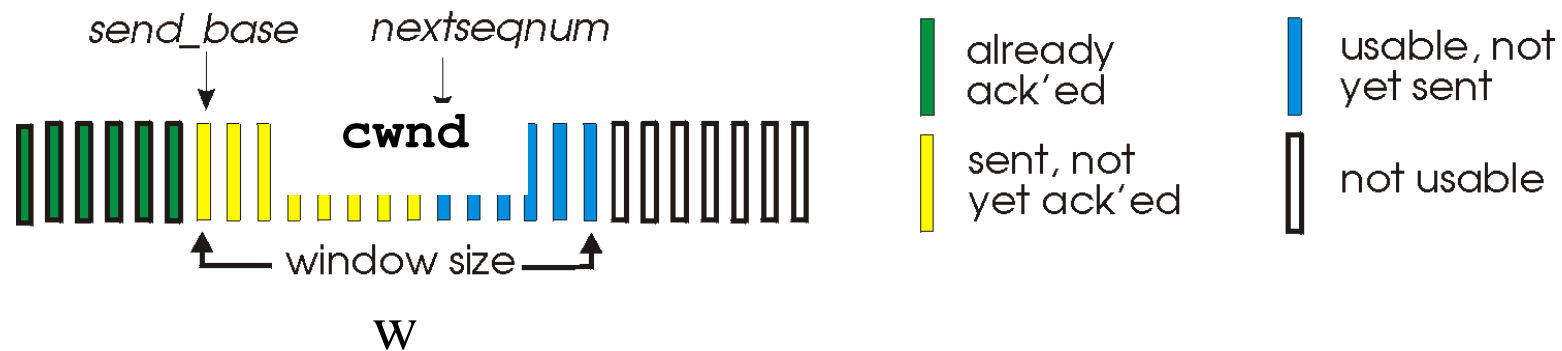


The Desired Properties of a Congestion Avoidance Scheme

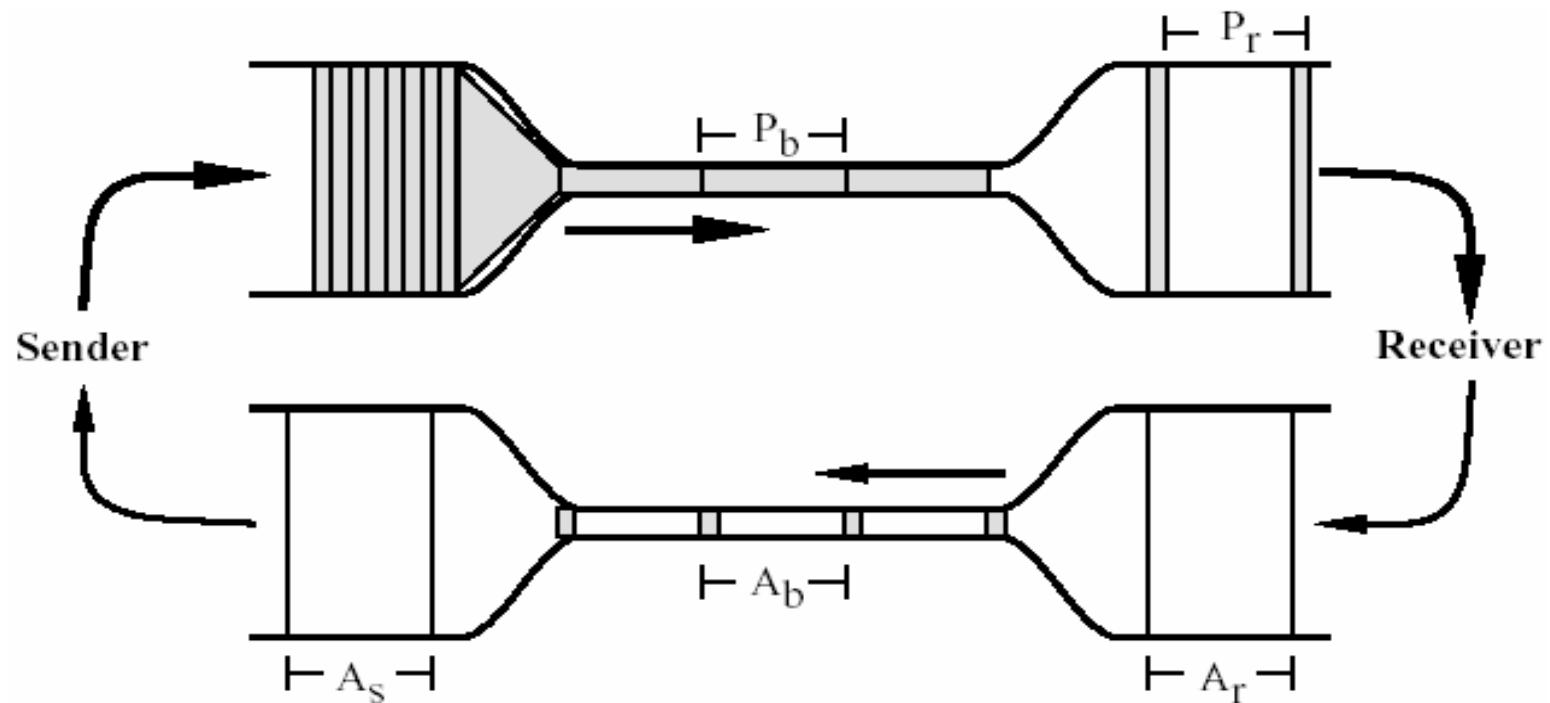
- ❑ Efficiency (fully utilization)
- ❑ Fairness (resource sharing)
- ❑ Distributedness (no central knowledge to achieve scalability)
- ❑ Convergence (fast convergence after disturbance, low oscillation)

TCP Congestion Control

- ❑ End-to-end, window-based congestion control
- ❑ Transmission rate limited by congestion window size, `cwnd`, over segments:



Window-based Scheme



Window-based Congestion Control is Self Clocking!

TCP Congestion Control

- ❑ Ideally, at equilibrium, we want to set the window size (approximately) to the product of available bandwidth (for this flow) and round-trip delay
- ❑ However,
 - we don't know these parameters at the beginning of a flow
 - further, the available bandwidth and round-trip are changing, because of competing flows

TCP Congestion Control: Basic Structure

- ❑ Many versions of TCP
 - TCP/Tahoe: this is a less optimized version
 - TCP/Reno: most OSs today implement TCP/Reno
 - TCP/Vegas: not currently used
- ❑ Two “phases”
 - slow-start
 - congestion avoidance
- ❑ Important variables:
 - cwnd: congestion window size
 - ssthresh: threshold between the slow-start phase and the congestion avoidance phase

TCP Slow-start

Initially:

`cwnd = 1;`

`ssthresh = infinite (64K);`

For each newly ACKed segment:

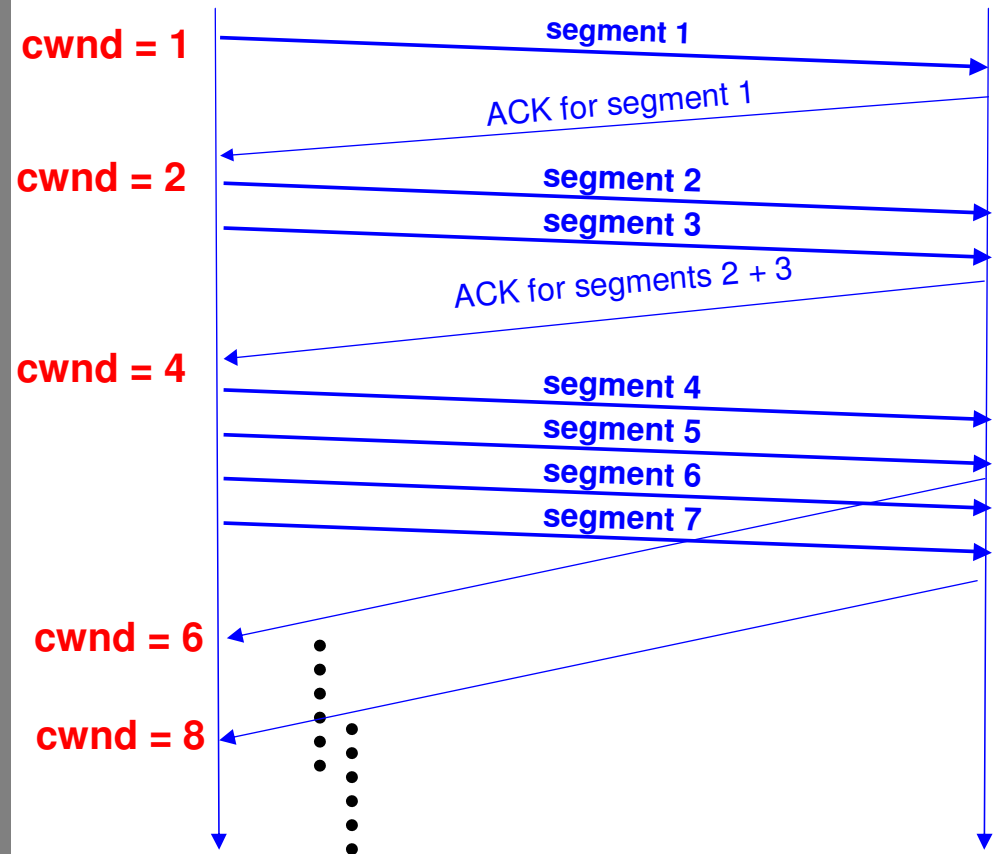
`if (cwnd < ssthresh)`

`/* slow start*/`

`cwnd = cwnd + 1;`

Upon detecting loss:

`/*slowstart stops*/`



TCP Congestion Avoidance: Reno

Initially:

 cwnd = 1;

 ssthresh = infinite (64K);

For each newly ACKed segment:

 if (cwnd < ssthresh)

 /* slow start */

 cwnd = cwnd + 1;

 else

 /* congestion avoidance; cwnd increases by 1 per RTT */

 cwnd += 1;

Triple-duplicate ACKs (TCP/Reno infers congestion by duplicate ACKs):

 /* multiplicative decrease */

 cwnd = ssthresh = cwnd/2;

Timeout:

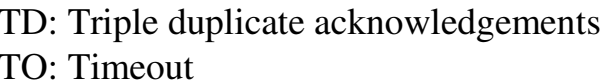
 ssthresh = cwnd/2;

 cwnd = 1;

TCP slow start

□ TCP slow-start algorithm

- sender calculates a congestion window for a receiver
- start with a congestion window size equal to one segment
- exponential increase of the congestion window up to the congestion threshold, then linear increase
- missing acknowledgement causes the reduction of the congestion threshold to one half of the current congestion window
- congestion window starts again with one segment



Question: Does TCP fully utilize bandwidth?

TCP

□ TCP congestion control

- packet loss in fixed networks typically due to (temporary) overload situations
- router have to discard packets as soon as the buffers are full
- TCP recognizes congestion only indirect via missing acknowledgements, retransmissions unwise, they would only contribute to the congestion and make it even worse
- slow-start algorithm as reaction

Discussion

- ❑ Why is TCP congestion control less effective in mobile wireless networks?
- ❑ How to improve the performance of TCP?

TCP in wireless mobile networks

- ❑ TCP assumes congestion if packets are dropped
 - typically wrong in wireless networks, here we often have packet loss due to *transmission errors*
 - furthermore, *mobility* itself can cause packet loss, if e.g. a mobile node roams from one access point (e.g. foreign agent in Mobile IP) to another while there are still packets in transit to the wrong access point and forwarding is not possible.

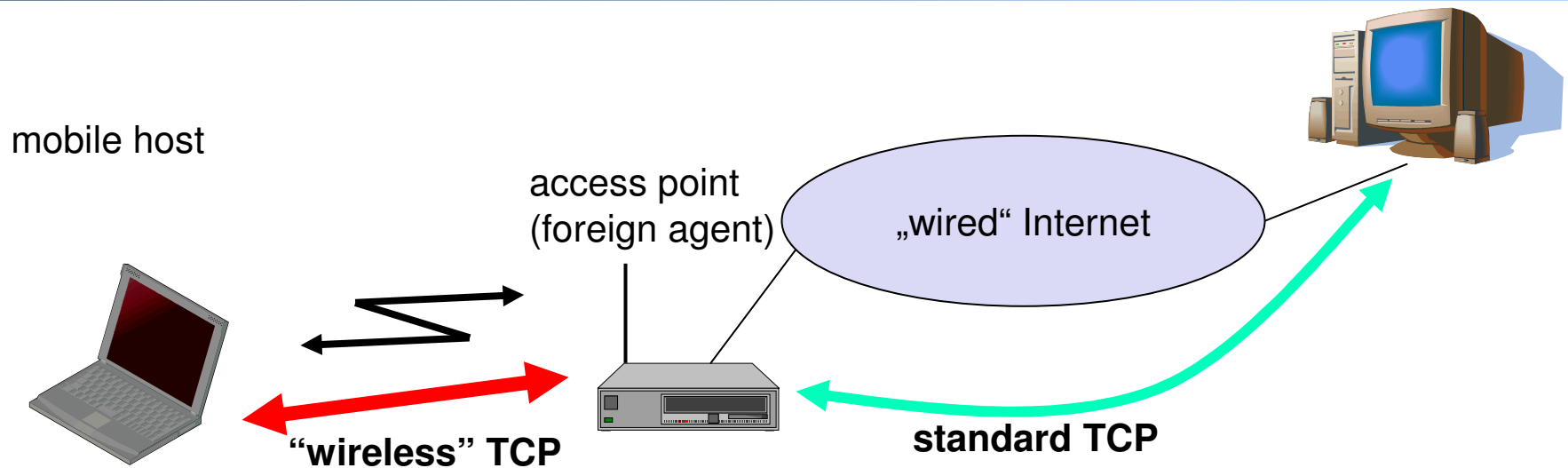
TCP in wireless mobile networks

- ❑ The performance of an unchanged TCP degrades severely
 - however, TCP cannot be changed fundamentally due to the large base of installation in the fixed network, TCP for mobility has to remain compatible
 - the basic TCP mechanisms keep the whole Internet together.

Outline

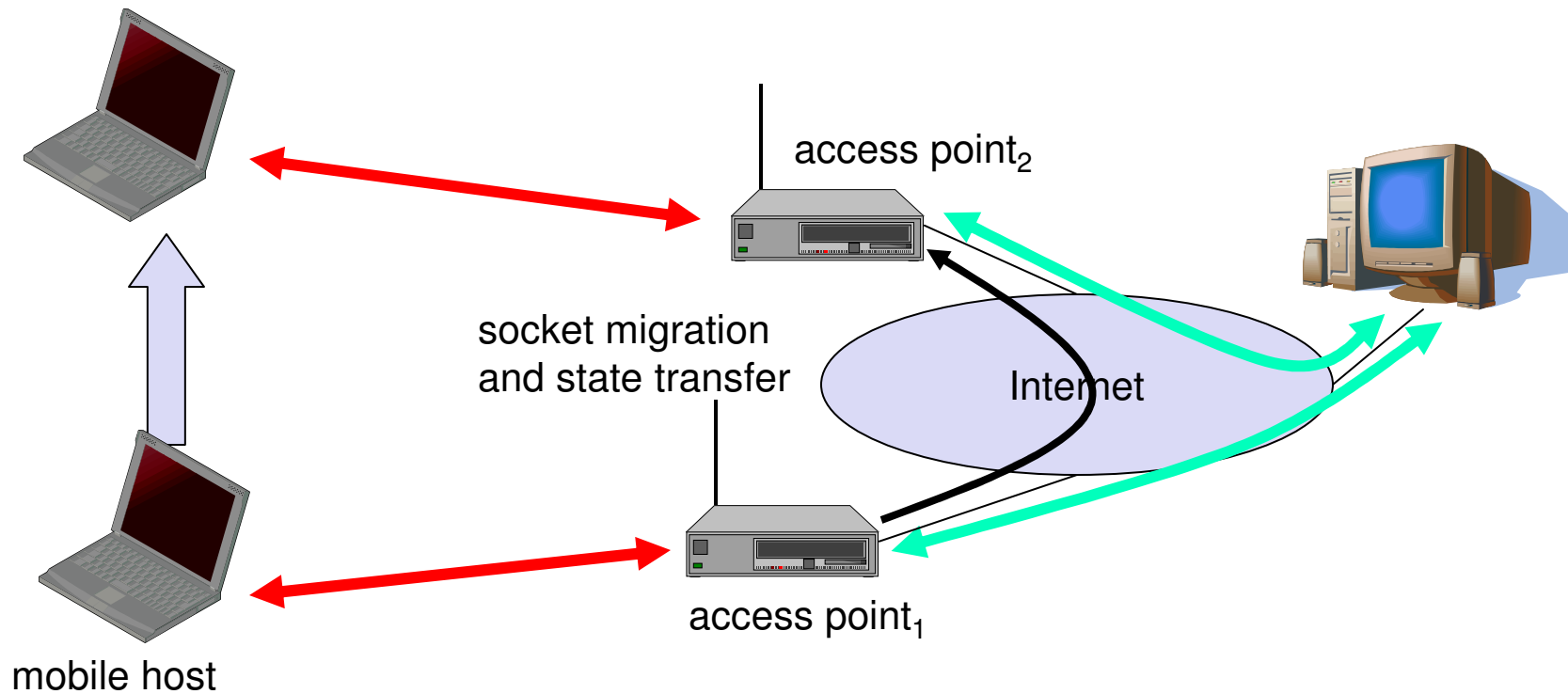
- ❑ Introduction to transport protocol
- ❑ Introduction to TCP (Internet) congestion control
 - *Congestion control in wireless networks*
 - *Indirect-TCP*

Indirect-TCP



- ❑ Split a TCP connection at the foreign agent into 2 TCP connections
 - hosts in the fixed part of the network do not notice the characteristics of the wireless part
 - no changes to the TCP protocol for hosts connected to the wired Internet, millions of computers use (variants of) this protocol
 - optimized TCP protocol for mobile hosts

I-TCP Socket and State Migration



Advantages of I-TCP

- ❑ No changes in the fixed network necessary, no changes for the hosts (TCP protocol) necessary, all current optimizations to TCP still work
- ❑ Simple to control, mobile TCP is used only for one hop between, e.g., a foreign agent and mobile host
 - transmission errors on the wireless link do not propagate into the fixed network
 - therefore, a very fast retransmission of packets is possible, the short delay on the mobile hop is known

Disadvantages of I-TCP

❑ Loss of end-to-end semantics

- an acknowledgement to a sender no longer means that a receiver really has received a packet --- foreign agents might crash

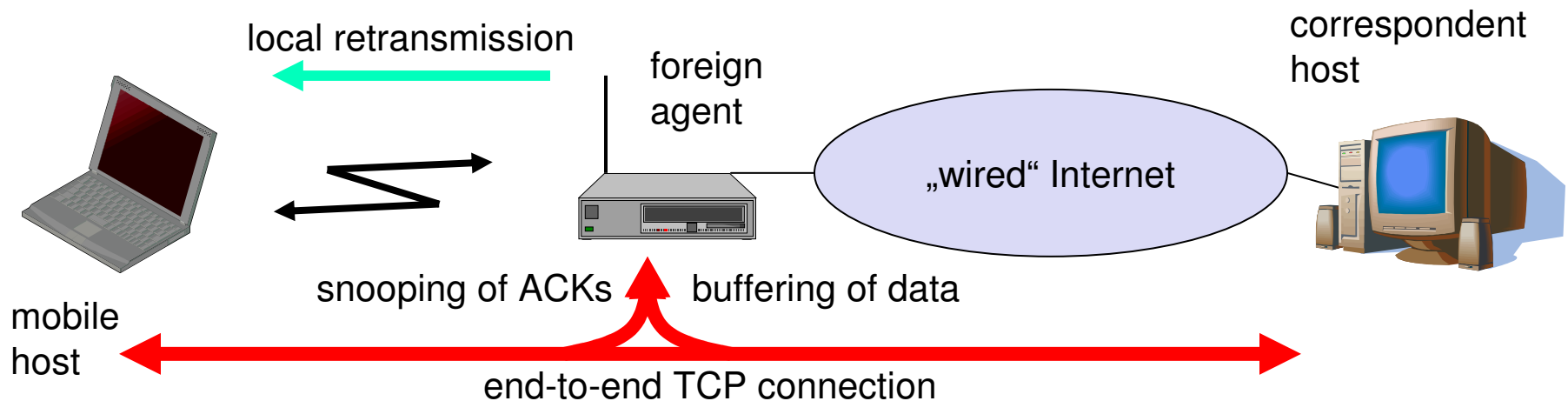
❑ Higher latency possible

- due to buffering of data within the foreign agent and forwarding to a new foreign agent

Outline

- ❑ Introduction to transport protocol
- ❑ Introduction to TCP (Internet) congestion control
- *Congestion control in wireless networks*
 - Indirect-TCP
 - *Snoop-TCP*

Snoop TCP



- ❑ **Transparent** extension of TCP within the foreign agent
 - changes of TCP only within the foreign agent
 - buffering of packets sent to the mobile host
 - lost packets on the wireless link (both directions!) will be retransmitted immediately by the mobile host or foreign agent, respectively (so called "local" retransmission)
 - the foreign agent therefore "snoops" the packet flow and recognizes acknowledgements in both directions, it also filters ACKs

Summary

- Network layer v.s. transport layer
- TCP for Internet
- TCP for wireless mobile networks
- Indirect TCP
- Snoop TCP