CSCE 463/612
Networks and Distributed Processing
Spring 2024

Transport Layer VII

Dmitri Loguinov
Texas A&M University

March 20, 2024

Chapter 3: Roadmap

3.1 Transport-layer services

3.2 Multiplexing and demultiplexing

3.3 Connectionless transport: UDP

3.4 Principles of reliable data transfer

3.5 Connection-oriented transport: TCP

- Segment structure
- Reliable data transfer
- Flow control
- Connection management

3.6 Principles of congestion control

3.7 TCP congestion control

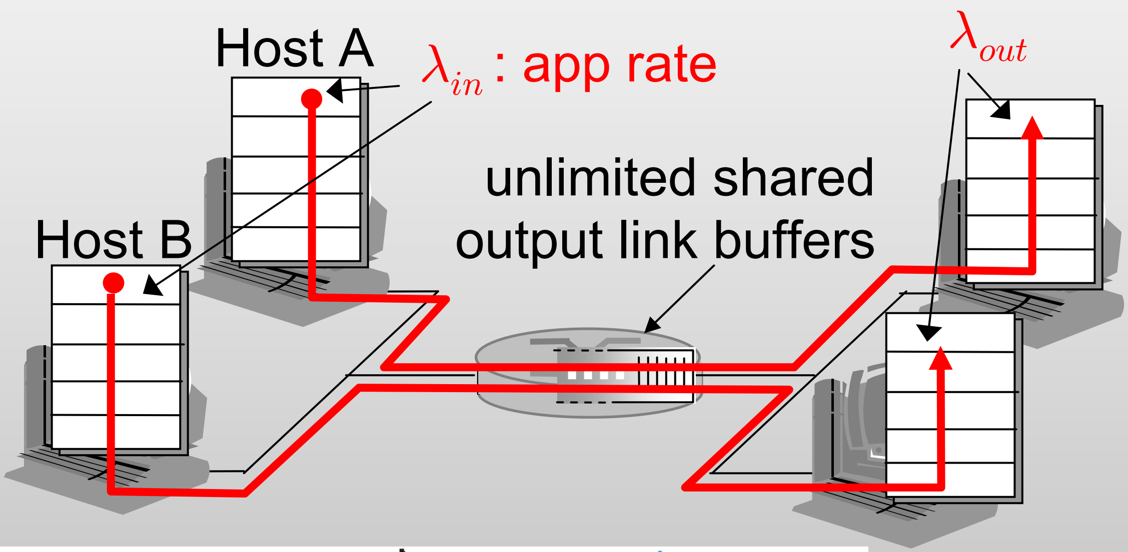
Principles of Congestion Control

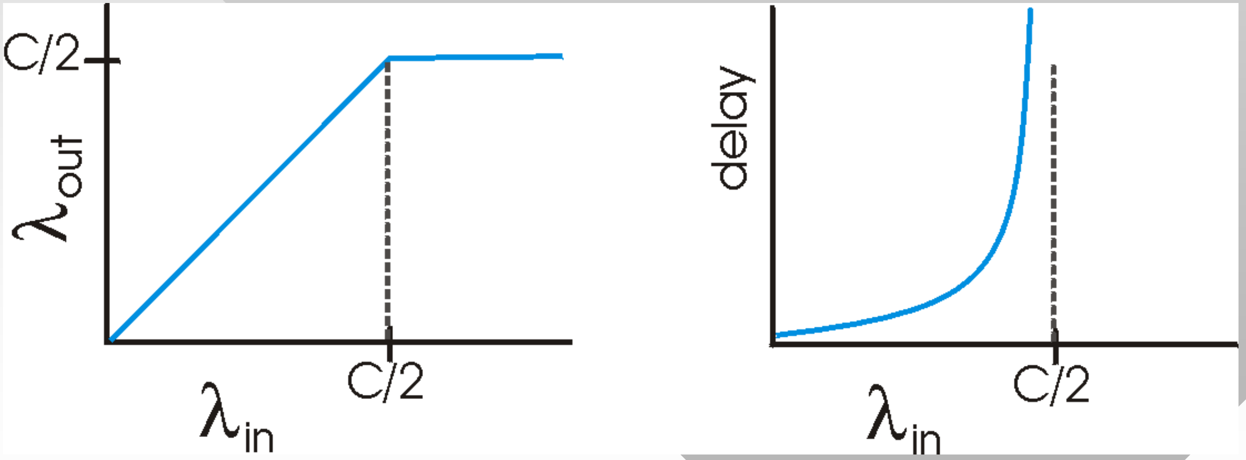
Congestion:

- Informally: "too many sources sending too much data too fast for the *network* to handle"
- Different from flow control!
- Manifestations:
  - Lost packets (buffer overflows)
  - Delays (queueing in routers)
- Important networking problem

Causes/Costs of Congestion: Scenario 1

- Two senders, two receivers
- One router of capacity C, infinite buffers, no loss
- No retransmission

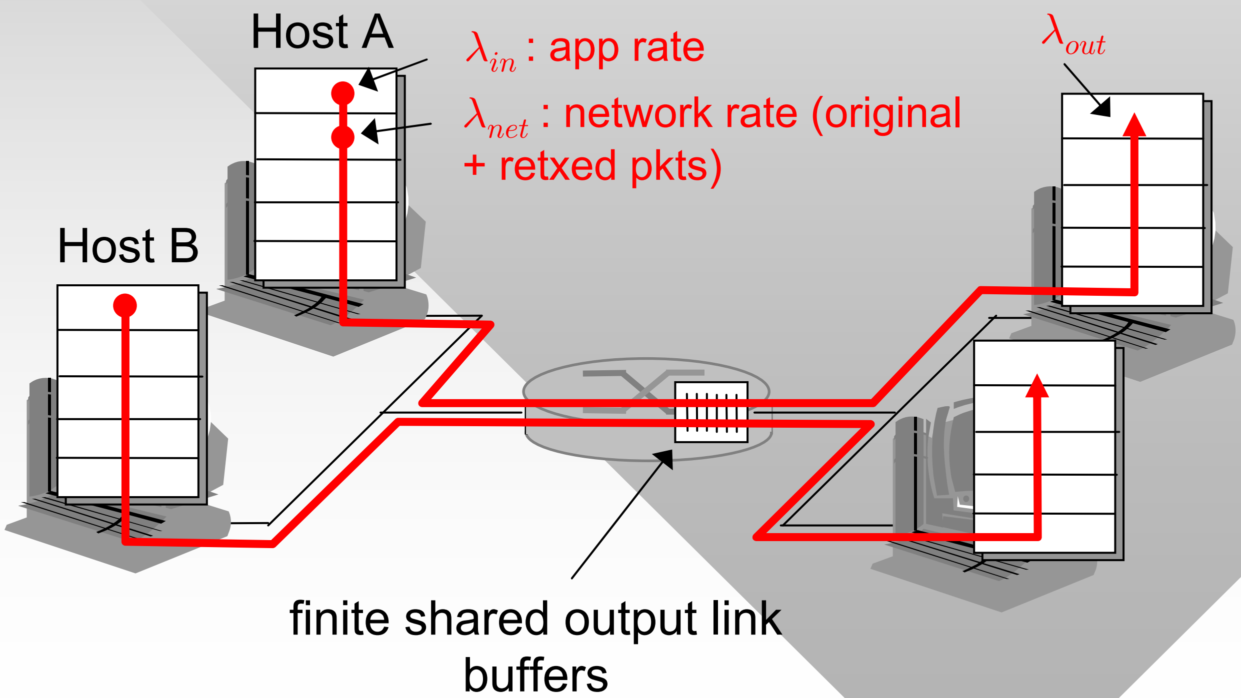




Cost 1: queuing delays in congested routers

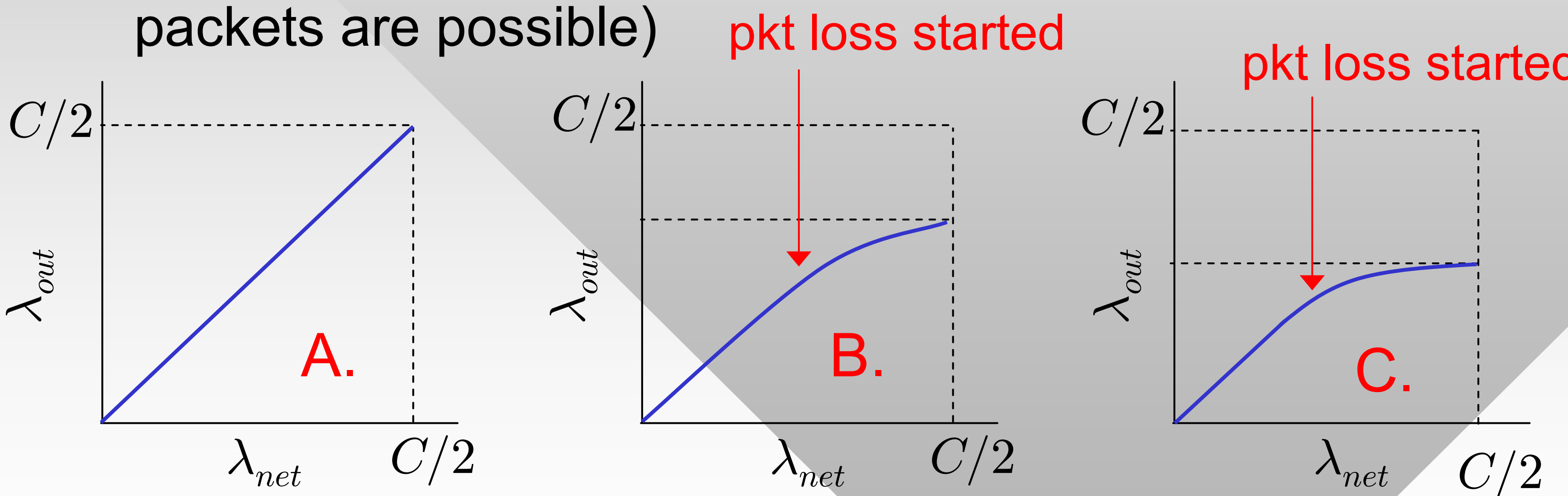
Causes/Costs of Congestion: Scenario 2

- One router, *finite* buffers (pkt loss is possible now)
- Sender retransmission of lost packet
- During congestion $2\lambda_{net} = 2(\lambda_{in} + \lambda_{retx}) = C$



Causes/Costs of Congestion: Scenario 2

- We call λ_{out} *goodput* and λ_{net} throughput
  - Case A: pkts never lost while $\lambda_{net} < C/2$ (not realistic)
  - Case B: pkts are lost when λ_{net} is "sufficiently large," but timeouts are perfectly accurate (not realistic either)
  - Case C: same as B, but timer is not perfect (duplicate packets are possible)

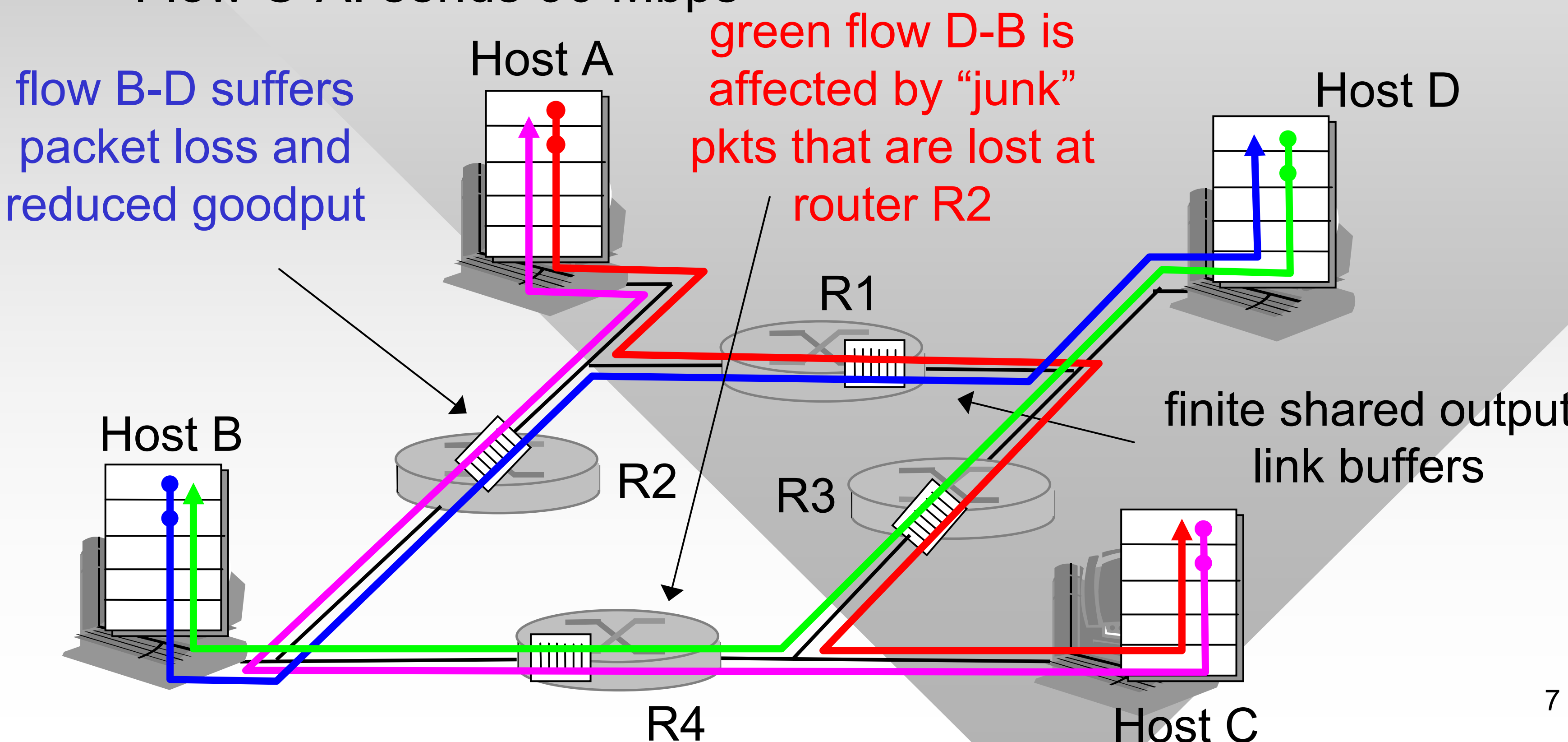


Cost 2: retransmission of lost packets and premature timeouts increase network load, reduce *flow's own* goodput

Causes/Costs of Congestion: Scenario 3

- Multihop case
  - Timeout/retransmit
  - R2 = 50 Mbps, R1 = R3 = R4 = 100 Mbps
  - Flow C-A: sends 90 Mbps

Cost 3: congestion causes goodput reduction for *other* flows



Approaches Towards Congestion Control

Two broad approaches towards congestion control:

End-to-end:

- No explicit feedback from network
- Congestion *inferred* by end-systems from observed loss/delay
  - Approach taken by TCP (relies on loss)

Network-assisted:

- Routers provide feedback to end systems
  - Single bit indicating congestion (DECbit, TCP/IP ECN)
  - Two bits (ATM)
  - Explicit rate senders should send at (ATM)

ATM = Asynchronous Transfer Mode

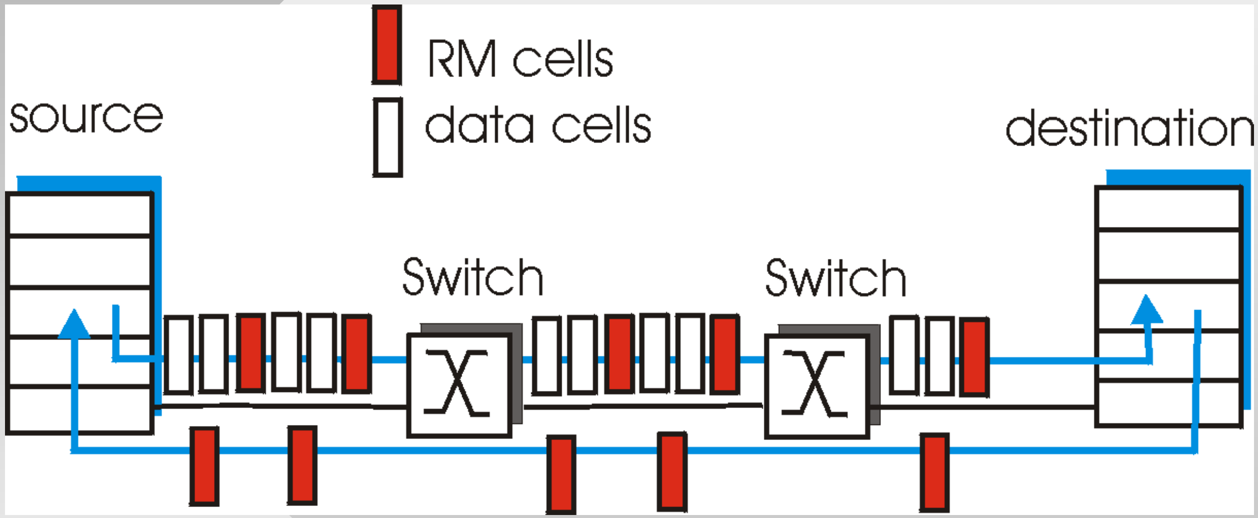
Case Study: ATM ABR Congestion Control

- For network-assisted protocols, the logic can be binary:
  - Path underloaded, increase rate
  - Path congested, reduce rate
- It can also be ternary
  - Increase, decrease, hold steady
  - ATM ABR (Available Bit Rate) profile

RM (resource management) packets (cells):

- Sent by sender, interspersed with data cells
- Bits in RM cell set by switches/routers
  - NI bit: no increase in rate (impending congestion)
  - CI bit: reduce rate (congestion in progress)
- RM cells returned to sender by receiver, with bits intact

Case Study: ATM ABR Congestion Control



- Additional approach is to use a two-byte ER (explicit rate) field in RM cell
  - Congested switch may lower ER value
  - Senders obtain the maximum supported rate on their path
- Issues with network-assisted congestion control?

Chapter 3: Roadmap

3.1 Transport-layer services
3.2 Multiplexing and demultiplexing
3.3 Connectionless transport: UDP
3.4 Principles of reliable data transfer
3.5 Connection-oriented transport: TCP
- Segment structure
- Reliable data transfer
- Flow control
- Connection management

3.6 Principles of congestion control
3.7 TCP congestion control

TCP Congestion Control

- TCP congestion control has a variety of algorithms developed over the years
  - TCP Tahoe (1988), TCP Reno (1990), TCP SACK (1992)
  - TCP Vegas (1994), TCP New Reno (1996)
  - High-Speed TCP (2002), Scalable TCP (2002)
  - FAST TCP (2004), TCP Illinois (2006)
- Many others: H-TCP, CUBIC TCP, L-TCP, TCP Westwood, TCP Veno (Vegas + Reno), TCP Africa
- Linux: BIC TCP (2004), CUBIC TCP (2008)
- Vista and later: Compound TCP (2005)
  - Server 2019 switched to CUBIC
- Google: BBR (2016)