


**18-345 – Fall 08**

**Lecture 21**

**Transport layer**

Peter Steenkiste


Reading: Sections 7.8, 8.4, 8.5



1

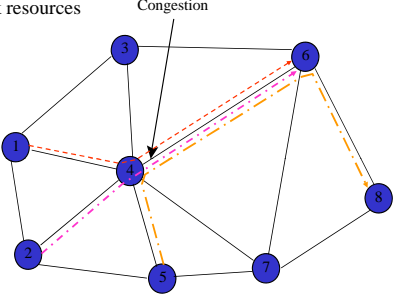

**Outline**

- Congestion control
- UDP
- TCP overview
- TCP flow and error control
- TCP congestion control

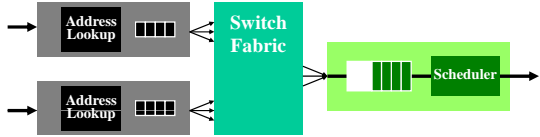


10-2


Congestion occurs when a surge of traffic overloads network resources

**Switch Behavior under Congestion**

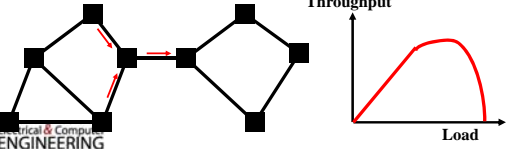



- Buffering protects flows from short-term congestion.
  - Buffer fills up when fill rate exceeds drain rate
  - Buffer drains when fill rate drops below drain rate
- Longer term overload will result in overflow of the buffer on the congested bottleneck link.
  - Packets are entering the buffer faster than they leave the buffer for extended period of time



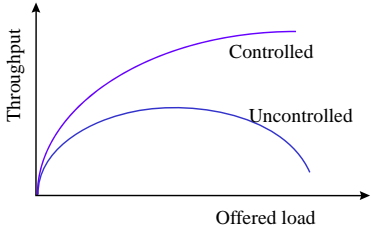

**Why is Congestion Bad?**

- Wasted bandwidth: retransmission of dropped packets.
- Poor user service : unpredictable delay, reduced throughput.
- Increased load can even result in lower network throughput.
  - Switched nets: heavy traffic -> long queues -> lost packets -> retransmits
  - Ethernet: high demand -> many collisions
  - compare with highways: too much traffic slows down throughput

**Ideal Outcome of Congestion Control**

- Resources used efficiently until available capacity is reached

## Congestion: Challenge

- Two views of the congestion problem
  - Network has certain amount of bandwidth
    - Sum of the transmission rates of all applications must be less than the aggregate network bandwidth
  - Network is a giant memory: router buffers + links
    - Number of packets each application has in the network must be smaller than the network memory size
- Challenge:
  - Keep network at a good operating point
  - “Fair” distribution of bandwidth across users
- Is this hard??

Electrical & Computer ENGINEERING

## Possible Solutions

- Redesign the network.
  - Add capacity to congested links
  - Very slow solution: takes days to months!
- Reroute traffic.
  - Alternate paths are not always available
  - Also too slow: takes 10s of seconds
  - In practice, most routing algorithms are traffic independent
    - Why?
- Adjust the load in the network.
  - What are the options?

Electrical & Computer ENGINEERING

## Approaches to Congestion Control

- Open loop: senders limit traffic without information about state of the network
  - Reservations and traffic pacing
  - Discussed in QoS lecture
- Closed loop: senders receive feedback and adjust transmit rate accordingly
  - How do they receive feedback?
  - How do they adjust rate?

Electrical & Computer ENGINEERING

## Closed-Loop Flow Control

- Congestion control
  - feedback information to regulate flow from sources into network
  - Based on buffer content, link utilization, etc.
  - Examples: TCP at transport layer; congestion control at ATM level
- End-to-end vs. Hop-by-hop
  - Delay in effecting control
- Implicit vs. Explicit Feedback
  - Source deduces congestion from observed behavior
  - Routers/switches generate messages alerting to congestion

Electrical & Computer ENGINEERING

## End-to-End vs. Hop-by-Hop Congestion Control

Electrical & Computer ENGINEERING


## The Big Picture: Closed Loop Flow Control

- How does the network apply back pressure?
  - Challenge: delay in feedback loop and vagueness of feedback
- How do the sources adapt?
  - Challenge: diverse sources and malicious users
- How does the switch distribute link bandwidth?
  - Challenge: treat sources fairly in a scalable way

Electrical & Computer ENGINEERING


## Feedback Mechanisms: Explicit Feedback

- The network provides the sender with explicit information on network conditions.
  - Explicit information reduces the chance of error
  - Can be sent as separate packet or can be piggybacked on data packets
- Always requires support on switch.
  - Calculate the feedback and generate the signal
  - Switch overhead and bandwidth use
- Many flavors of explicit feedback exist.
  - forward versus backward congestion indication
  - binary versus multivalued
- Requires some degree of homogeneity.
  - Switches have to generate consistent feedback
  - End-points have to interpret feedback consistently




## Feedback Mechanisms: Implicit Feedback

- Sender observes quality of connection to guesstimate congestion status
  - Example: dropped packets indicate congestion
    - usually true
  - More sophisticated solutions, e.g. packet pair
- Does not require explicit network support
  - Router has no choice - what else will it do?
  - No additional bandwidth or additional router complexity
  - But: interpretation often requires some knowledge of the internals of the network
- Interpreting the information may be difficult
  - Why was the packet dropped?



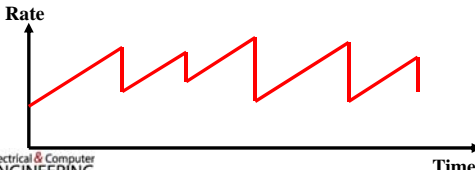

## End-Point Behavior

- Hosts should reduce transmission rate when they receive a congestion indication
  - That much is obvious: important for network stability
- Hosts are allowed to increase rate if there is no congestion (– use excess capacity).
  - Also obvious at least in the case of explicit, multi-valued feedback
- But feedback typically only gives information about the presence of congestion
  - Hard to distinguish between an ideal rate and a low rate
  - Solution is probing: periodically increase rate to see whether more bandwidth is available




## Adaptation with Binary Feedback

- Network stability requires multiplicative decreases and additive increases in the transmission rate (AIMD).
  - Instance of simple linear control  $W_{i+1} = a + b \times W_i$
- Congestion is a dangerous condition
  - > back off quickly
- Unused bandwidth is undesirable but not dangerous
  - > probe carefully

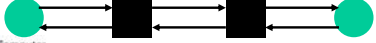

## End-Point Behavior: Discussion

- Response has to balance congestion and performance considerations.
  - Large backoff can reduce performance unnecessarily
  - Slow backoff may not eliminate congestion
- End-point response determines stability of the network.
  - Congestion collapse: throughput drops to zero when hosts do not back off
  - Rule: multiplicative slowdown and incremental speedup
- Congestion adaptation must be robust in a heterogeneous environment.
  - Switches and hosts will all have slightly different behavior
  - Some applications may want to adapt to congestion in specialized ways



## Example: DEC bit

- Switches set an explicit congestion bit in the packet header if the queue size is larger than one.
  - Receiver collects the information and forwards it to the sender
- Senders slow down if the bit is set in more than 50% of the packets in a window.
  - multiplicative slow down
  - stepwise increase if bit is not set for certain period of time
- Behavior is very similar to TCP, except that it has explicit feedback.

### Example: ATM Rate-based Flow Control

- Provide explicit “per flow” feedback.
  - Host tells the network how fast it is sending
  - Switches calculate how fast the host should be sending and include this information in explicit flow control packets that are sent periodically
- Feedback can be binary or explicit.
  - binary: source slows down or speeds up
  - explicit: switch specifies the rate

Electrical & Computer ENGINEERING

### Example: TCP

- Implicit feedback based on packet drops
  - FIFO scheduling with shared buffer pool
- TCP interprets packet drops as signs of congestion.
  - Multiplicative back off when there are packet drops
  - This is an assumption: packet drops are not a sign of congestion in all networks
    - e.g. wireless networks
- Periodically probes the network to check whether more bandwidth has become available.
  - Results in linear speed up

Electrical & Computer ENGINEERING

### Tradeoffs in Congestion Control

- Explicit schemes that isolate traffic flows seem preferable, but require more support inside the networks.
  - per-flow buffering, weighted fair queuing, policing
  - scalability???
  - determining nature of the explicit feedback in heterogeneous environment
- Diversity in networks makes TCP approach a good solution.
  - Dropping packets is universally a natural response to congestion
  - But many open issues: how to isolate poorly behaved sources, diversity in TCP implementations, ...

Electrical & Computer ENGINEERING

### Transport layer

- UDP
- TCP
  - Stream Data Transfer
  - Reliability
  - Flow Control

Electrical & Computer ENGINEERING

### Transport Layer

- Concern: Transfer of information between source & destination processes
- Independent of underlying physical networks
- Key to layering

Electrical & Computer ENGINEERING

### “TCP/IP” Protocol Suite

Electrical & Computer ENGINEERING

## IP Packet Format

← 32 bits →

Vers	HLLen	Type of Serv.	Total Length		
Ident. (Seq. #)		flags	fragment offset		
time-to-live	protocol	header checksum			
Source IP address					
Destination IP address					
Options					padding
Data					

← TCP or UDP →

- Protocol version allows coexistence of different protocol generations

Electrical & Computer ENGINEERING 10-25

## Ports

Each process is identified indirectly through a *port*

- Port: a 16-bit number that identifies higher-layer protocol or process to which message is to be delivered
- Socket or transport address: (protocol, IP address, port)
- Connection: (protocol, source IP address, source port, destination IP address, destination port)
- *Multiplexing*: multiple connections can use simultaneously use a port

Electrical & Computer ENGINEERING 10-26

## Internet Transport Layer Protocols

TCP	UDP
IP	
Network Interface	

- User Datagram Protocol (UDP) provides transport level datagram service
- UDP provides additional address information to identify user/port/application at the hosts
- 16-bit source & destination port numbers are used

Electrical & Computer ENGINEERING 10-27

## UDP Format

← 32 bits →

Source Port	Destination Port
UDP Msg Length	Checksum
Data	

- Port numbers identify application within host
- Msg Length = # bytes in datagram
- Checksum: optional!
- Very minimal!

Electrical & Computer ENGINEERING 10-28

## Using UDP

- Non-standard protocols can be implemented on top of UDP.
  - Non-standard = non-TCP in practice
  - use the port addressing provided by UDP
  - implement their own reliability, flow control, ordering, congestion control
- Examples:
  - remote procedure calls
  - multimedia
  - distributed computing communication libraries
  - look at some examples later

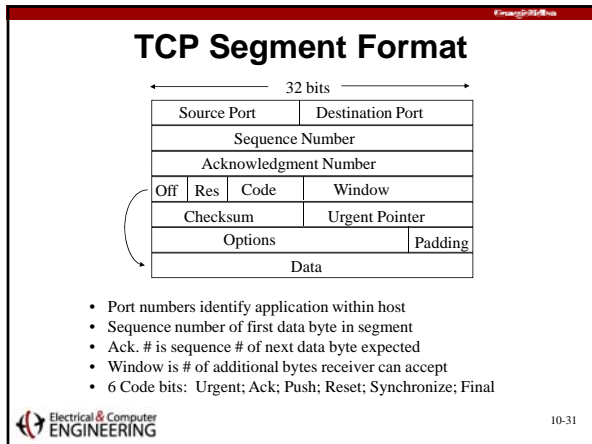
Electrical & Computer ENGINEERING

## TCP Overview

- Two-way reliable byte stream protocol.
  - “Feature rich”
- Connection establishment and tear down.
- Reliable data transfer.
- End-end flow control.
- Congestion control.
- Performance optimizations.
- Many extensions.

Source Port	Dest. Port
Sequence Number	
Acknowledgment	
HL/Flags	Window
D. Checksum	Urgent Pointer
Options..	

Electrical & Computer ENGINEERING



- ## High-Level TCP Characteristics
- Connection-oriented reliable byte-stream protocol.
    - Used for file transfers, telnet, web access, ....
  - Two way connections.
    - control information for one direction piggy-backed on data flow in other direction
    - header fields fall in three classes: general, forward flow, opposite flow
  - Protocol has evolved over time and will continue to do so.
    - Nearly impossible to change the header
    - Uses options to add information to the header
    - Change the processing at the two endpoints
    - Backward compatibility is what makes it TCP
- Electrical & Computer ENGINEERING

