Introduction to Computer Networks

In-Network Support for TCP

CS640

https://pages.cs.wisc.edu/~mgliu/CS640/F22/

Ming Liu
mgliu@cs.wisc.edu

Today

Last lecture

- How to improve the efficiency of TCP congestion control?

Today

- How to take advantage of in-network support for TCP efficiency improvement?

Announcements

- Lab4 is due 12/02/2022, 11:59 PM
- Lab5 is due 12/14/2022, 11:59 PM
- Final exam: Dec 17, 2022 5:05 PM – 7:05 PM @Engineering Hall 1800

Q: What is in-network support?

Resource Allocation

Dividing up resources among contending entities

Resource Allocation

Dividing up resources among contending entities

Resources
- Network bandwidth
- Router buffer space

Resource Allocation

Dividing up resources among contending entities

Resources
- Network bandwidth
- Router buffer space

Entity: granularity at which resource is allocated
- Default: “flow”
- Flow: corresponds to a connection 5-tuple (protocol number, srcIP, srcPort, dstIP, dstPort)

Congestion Control Revisited

Congestion control is an example of a resource allocation scheme

- It runs on end-hosts
- It runs in a distributed manner, no central coordination is required

Flows adjust their transmission rates by changing window size

- Rate = window size / RTT

Congestion Control Revisited

Congestion control is a host-based, feedback-based distributed resource allocation scheme.

Flows adjust their transmission rates by changing window size

- Rate = window size / RTT

Congestion Control Revisited

Congestion control is a host-based, feedback-based distributed resource allocation scheme.

↓

Congestion control can become a **router-assisted**, host-based, feedback-based distributed resource allocation scheme.

Q: What is in-network support?

A: Active Queue Management (AQM)

- An “active” router queue management to facilitate better flow behavior under resource contention

Q: Why does in-network support help?

The Importance of Queuing

A router must implement two queuing disciplines:

- #1: Scheduling discipline
- #2: Drop policy

Queuing allocates both bandwidth and buffer space

- Bandwidth: which packet to serve (transmit) next
- Buffer space: which packet to drop next (when required)

The Importance of Queuing

A router must implement two queuing disciplines:

- #1: Scheduling discipline
- #2: Drop policy

Queuing allocates both bandwidth and buffer space

- Bandwidth: which packet to serve (transmit) next
- Buffer space: which packet to drop next (when required)

Queuing is important for the quality of service.

The Naive (but widely-used) Approach

Scheduling Discipline: FIFO (first-in-first-out)

- Packets are dequeued based on the arrival order in regardless of the flow priority

Drop Policy: Drop-Tail

- Packets are dropped when queue is full regardless of the flow priority

Two Issues

#1: Lock-out Problem

- A few flows can easily monopolize the queue space
- Lack of traffic isolation

#2: Full Queues

- TCP adjust rates based on time out (packet loss)
- But one would always observe bursty loss

Q: Why does in-network support help?

A: Effectively use the router buffer when the network load is high

- Divide the buffer space equally among ongoing flows
- Notify the endhosts early to avoid bursty packet drops

Q: How does in-network support work?

A: Three examples:

- #1: Fair Queueing (FQ)
- #2: Random Early Detection (RED)
- #3: Explicit Congestion Notification (ECN)

#1: Fair Queueing

Goal: allocate resources "fairly"

- Keep individual (virtual) queue for each flow

Isolate ill-behaved users

- The router does not send explicit feedback to the endhost
- Endhosts still need end-to-end congestion control

Max-min Fairness

Allocate user with “small” demand what it wants, evenly divide unused resources to “big” users

Formally

- Resource allocated in terms of increasing demand
- No source gets a resource share larger than its demand
- Sources with unsatisfied demands get equal share of resource

Implementing Max-min Fairness

Generalized processor sharing

- Fluid fairness
- Bitwise round robin among all queues

Why not a simple round-robin?

- Variable packet length -> can get more service by sending bigger packets
- Unfair instantaneous service rate
- What if arrive just before/after the packet departs?

Bit-by-bit Round Robin

Single flow: clock ticks when a bit is transmitted. For packet i:

- Pi = length, Ai = arrival time, Si = begin transmit time, Fi = finish transmit time
- Fi = Si + Pi = max (F(i-1), Ai) + Pi

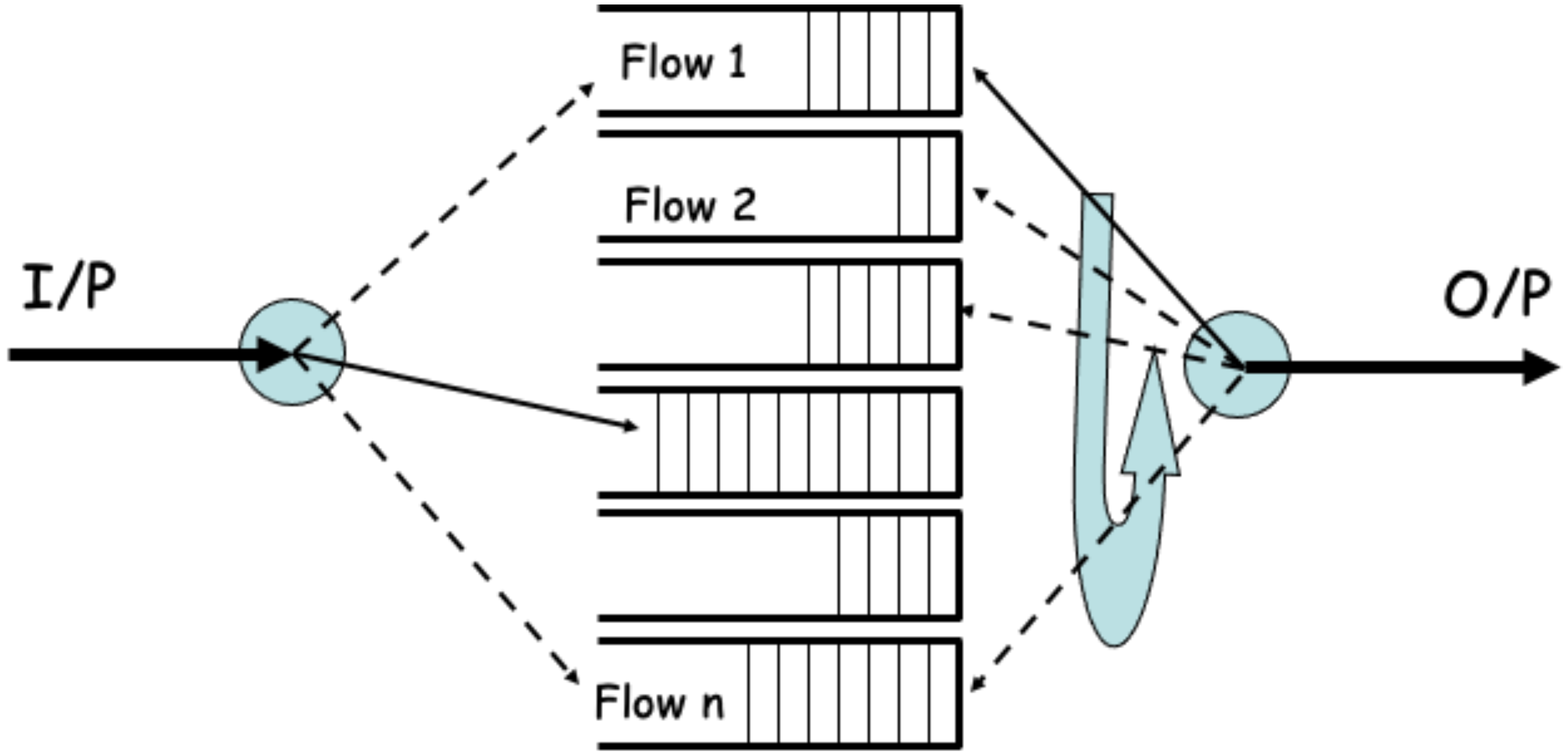
Multiple flows: clock ticks when a bit from all active flows is transmitted —> round number

- Can calculate Fi for each packet if the number of flows is known at all times

Fair Queuing Mechanism

Mapping the bit-by-bit round-robin schedule onto packet transmission

Transmit packet with the lowest Fi at any given time



#2: Random Early Detection (RED)

Key idea: detect incipient congestion

Assume hosts respond to lost packets

- Compliant congestion control

RED Algorithm

Maintain a running average of queue length

Case 1: if avg <= min_threshold, do nothing

- Low queueing, send packets through

Case 2: if avg >= max_threshold, drop packet

- Protection from misbehaving sources

Case 3: if min_threshold < avg < max_threshold, calculate probability P and drop arriving packet with P

- Notify sources of incipient congestion

RED More

Compute the average queue length

- AvgLen = (1 - weight) * AvgLen + weight * SampleLen, 0 < weight < 1 (usually 0.002)
- SampleLen is queue length each time a packet arrives (same as the EWMA for RTT)

Compute probability P

- TempP = MaxP * (AvgLen - min_threshold) / (max_threshold - min_threshold)
- P = TempP / (1 - count * tempP)
- Count = number of newly arriving packets while AvgLen has been between two thresholds (P increases with count)

RED More

Compute the average queue length

- AvgLen = (1 - weight) * AvgLen + weight * SampleLen, 0 < weight < 1 (usually 0.002)
- SampleLen is queue length each time a packet arrives (same as the EWMA for RTT)

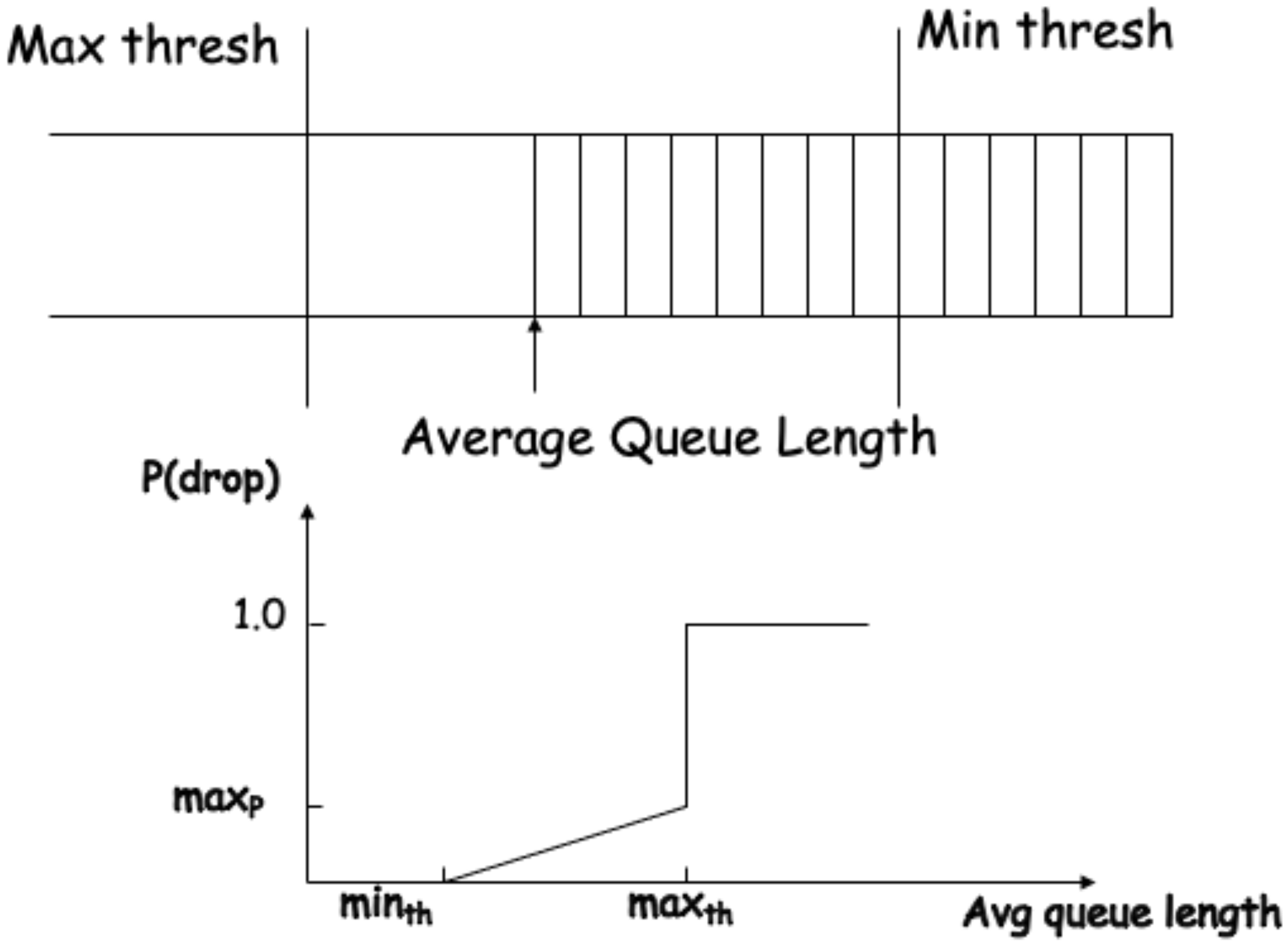
Compute probability P

Drop probability is a function of both AvgLen and how long it has been since the last drop

- TempP tracks how many newly arriving packets have been queued while AvgLen is between thresholds
- Count is the number of packets since the last drop
- This prevents clusters of drops

RED Operation

RED is good at keeping avg. queue size steady



#3: Explicit Congestion Notification (ECN)

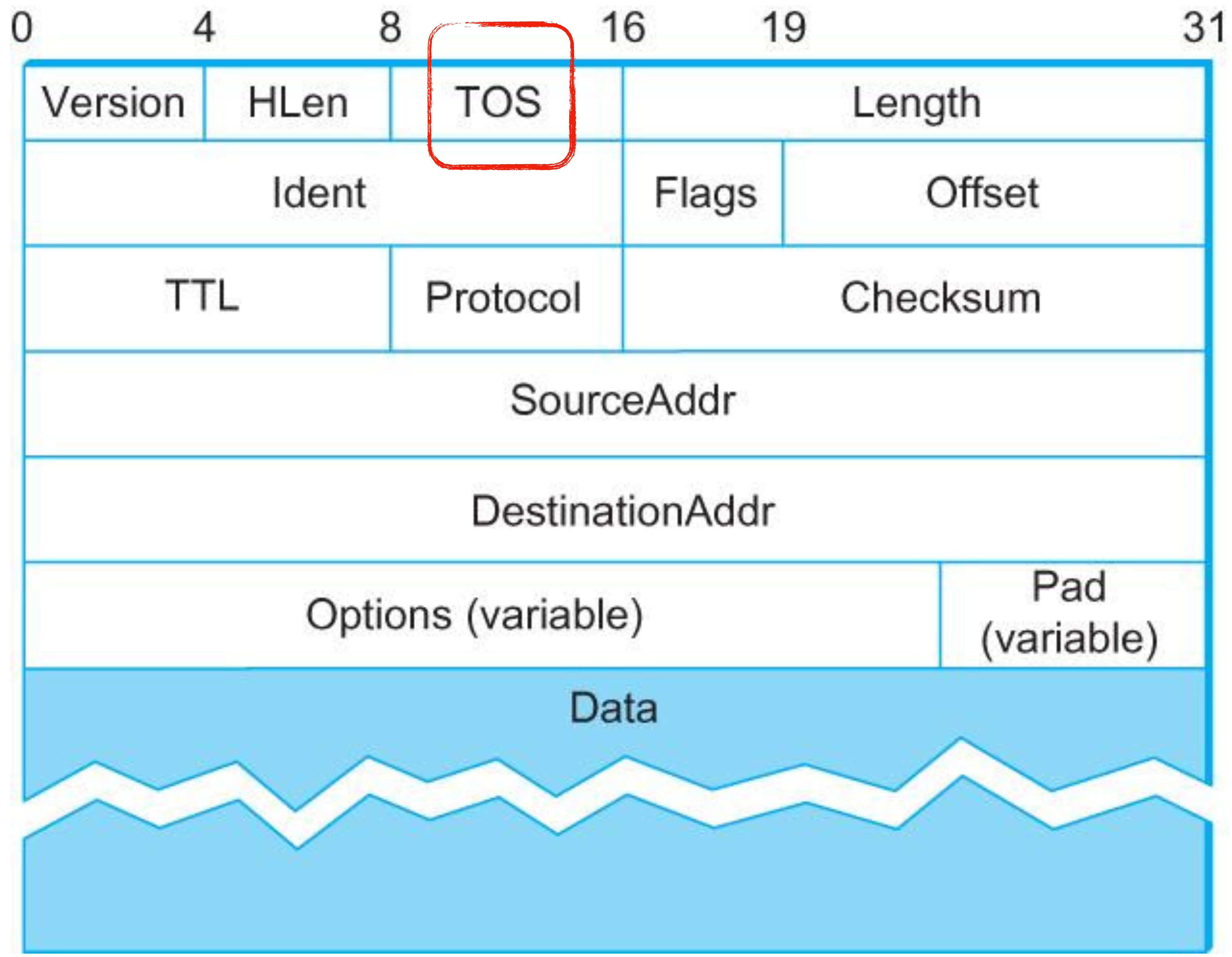
ECN allows end-to-end notification of network congestion without dropping packets

- **On the sending path**: an ECN-aware router may set a mark in the IP header instead of dropping a packet in order to signal impending congestion when the queue gets full
- **On the receiving path**: the receiver echoes the congestion indication to the sender so that it can reduces the transmission rate

ECN in IP

Two least significant bits of the traffic class field

- 00 –> Non ECN-Capable Transport
- 10 –> ECN Capable Transport
- 01 –> ECN Capable Transport
- 11 –> Congestion Encountered, CE



Terminology

1. Host
2. NIC
3. Multi-port I/O bridge
4. Protocol
5. RTT
6. Packet
7. Header
8. Payload
9. BDP
10. Baud rate
11. Frame/Framing
12. Parity bit
13. Checksum
14. Ethernet
15. MAC
16. (L2) Switch
17. Broadcast
18. Acknowledgement
19. Timeout
20. Datagram
21. TTL
22. MTU
23. Best effort
24. (L3) Router
25. Subnet mask
26. CIDR
27. Converge
28. Count-to-infinity
29. Line card
30. Network processor
31. Gateway
32. Private network
33. IPv6
34. Multicast
35. IGMP
36. SDN
37. (Transport) port
38. Pseudo header
39. SYN/ACK
40. Incarnation
41. Flow
42. SYN flood
43. TCP Segment
44. Window
45. Advertised Window
46. Effective Window
47. TCP Reno
48. Duplicated ACK
49. Congestion Window
50. Congestion Threshold
51. Selective Acknowledgment
52. Active Queue Management (AQM)

Principle

1. Layering
2. Minimal States
3. Hierarchy
4. Mechanism/policy separation

Technique

1. NRZ Encoding
2. NRZI Encoding
3. Manchester Encoding
4. 4B/5B Encoding
5. Byte Stuffing
6. Byte Counting
7. Bit Stuffing
8. 2-D Parity
9. CRC
10. MAC Learning
11. Store-and-Forward
12. Cut-through
13. Spanning Tree
14. CSMA/CD
15. Stop-and-Wait
16. Sliding Window
17. Fragmentation and Reassembly
18. Path MTU discovery
19. DHCP
20. Subnetting
21. Supernetting
22. Longest prefix match
23. Distance vector routing (RIP)
24. Link state routing (OSPF)
25. Boarder gateway protocol (BGP)
26. Network address translation (NAT)
27. User Datagram Protocol (UDP)
28. Transmission Control Protocol (TCP)
29. Three-way Handshake
30. TCP state transition
31. EWMA
32. Sliding window
33. Flow control
34. AIMD
35. Slow start
36. Fast retransmit
37. Fast recovery
38. Nagle's algorithm
39. Karn/Partridge algorithm
40. TCP Vegas
41. Bit-by-bit Round Robin
42. Fair Queueing (FQ)
43. Random Early Detection (RED)
44. Explicit Congestion Notification (ECN)

Summary

Today's takeaways

#1: Active Queue Management (AQM) improves the TCP operation efficiency

#2: Fair queueing, RED, and ECN are three typical in-network support cases

Next lecture

- Infrastructure Services