Computer Networks
ICS 651

- router intervention to address congestion
- Internet Explicit Congestion Notification
- FIFO queueing
- fair queueing
- UDP

Router Intervention to Avoid or React to Congestion

- Random Early Discard (RED) -- causes TCP Reno to back off
- information feed-forward -- the receiver must then return congestion information to the sender (see Internet ECN, below)
- information feedback -- requires route back to sender, does not work in Internet (except source quench ICMP, which is deprecated)
- communication time from router to sender may be insufficient if sender is sending lots of stuff. Also, stability issues -- all senders could increase their sending rate at the same time
- credits: can only send as much as we have in the "bank", automatically (but not immediately) replenished
  - similar to a window

Internet Explicit Congestion Notification

- ECN, explicit congestion notification, RFC 3168.
- in ECN, two of the bits of the IP Type of Service (ToS)/Differentiated Service(DS)/IPv6 Traffic Class(TC) field are used to indicate (a) whether congestion notification is requested (ECT), and (b) whether the packet experienced congestion (CE).
  - 00 means no ECN-capable Transport (no ECT)
  - 10/ECT(0) or 01/ECT(1) means ECT, no congestion experienced – 10 is the default
  - ECN-capable TCP sets ECT 10 in data packets (not ack packets)
  - 11 means Congestion Experienced, CE – a router can set this instead of dropping a packet
- TCP uses two new bits:
  - ECE, ECN-Echo, for the receiver to report to the senderthat a packet was received with CE set
  - CWR (Congestion Window Reduced, bit before ECE), to indicate that the ECE bit was received and the congestion window was reduced as it would have been in response to a packet loss
  - client sets ECE and CWR in the SYN packet, server responds with ECE flag in SYN+ACK
- compatible with hosts and routers that don't do ECN
- linux: 1 in /proc/sys/net/ipv4/tcp_ecn
  - default value 2 means respond to ECN requests, but don't initiate them

typical usage of ECN

- senders can set ECT
  - ECT(0) or ECT(1)
- routers can change ECT to CE to record that congestion was experienced
  - usually instead of dropping the packet
- transport layer at receiver is informed of CE, sends an ECE
- receiver of ECE (sender of data) reduces congestion window, sends CWR
  - the reduction in the congestion window should be the same as if a packet loss was detected
  - receiver stops sending ECE once it receives CWR
  - all this should happen at most once per RTT (once per window)

FIFO queueing

- each packet is placed at the end of the queue
- packets that take the same route are never reordered
- delay is proportional to queue size
- works reasonably well in the Internet, with TCP congestion control
- if all senders but one do congestion control, and one sender does not, the one that doesn't (IP telephony, multicasting) might grab much of the bandwidth

Fairness

- "everyone" gets the same treatment
- hard to do in a distributed system:
- local fairness (every flow gets the same treatment on this router) discriminates against flows that cross more routers (parking garage problem)
- global fairness requires global co-ordination, so local fairness is often the best we are willing to do

Fair Queueing

- one FIFO queue for each flow
- packets are taken in round-robin order from each queue that has them
- problem: large packets counted the same as small packets
- logically, we want to send one bit from each flow in round-robin order

Fair Queueing with different size packets

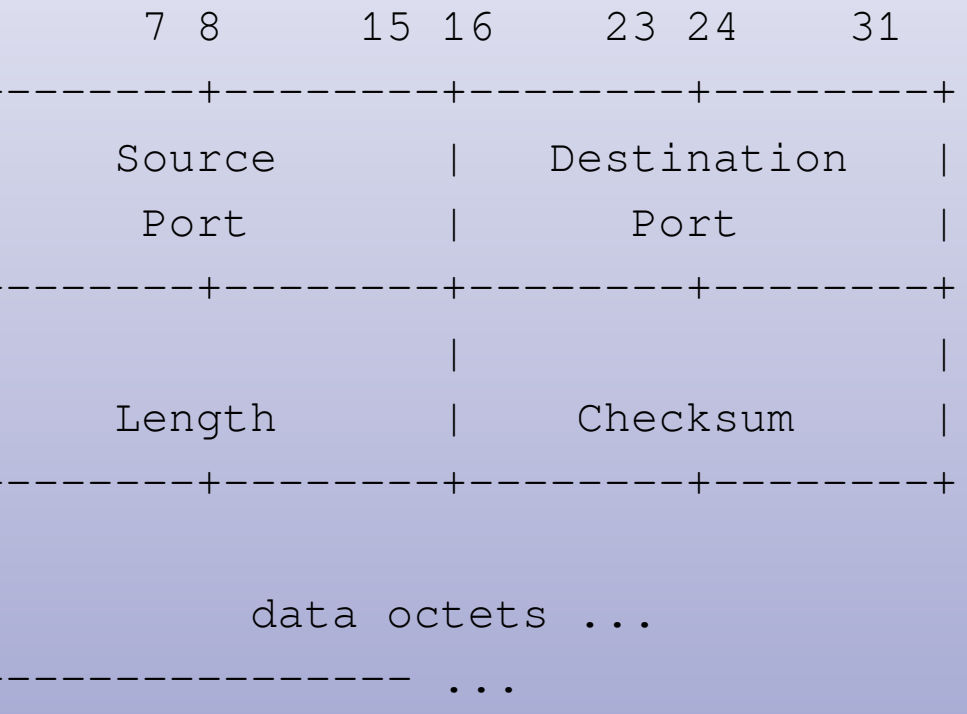
- the virtual clock ticks once for each bit sent from each of the queues
- so if there are more active queues, that means the virtual clock advances more slowly
- the virtual finish time for a packet is its start "time" plus the size of the packet
- the virtual start time of a packet is the largest of:
  - the finish time for the previous packet in the queue (a computed quantity), or
  - the actual virtual arrival time of the packet
- to be fair, select and transmit the packet with the lowest virtual finish time

TCP review

- state management: connection setup and teardown, Transmission Control Block (TCB)
- reliable transmission via sequence numbers, acknowledgements
- flow control to avoid overwhelming receiver:
- hard to obtain both reliability and performance
- acknowledgements are not acknowledged, but crucial information is carried in the acknowledgements (e.g. the window size)
- congestion control to avoid overwhelming network (or to slow down when we do) requires adaptive timer
- congestion control relies on Explicit Congestion Notification or on packets being lost
- congestion control has evolved in recent years

UDP

- User Datagram Protocol
- IP + ports + (optional in IPv4) checksum



- RFC 768

our network so far

- IP, TCP, and UDP
- reliable byte-stream and packet transmission among applications
- only application so far: DNS
- only Data Link layer so far: SLIP
- only Physical layer so far: serial lines
  - limited in speed, typically less than 1Mbit/s
  - limited in distance, typically only within a building
    - or over a radio link
  - limited in number of hosts, at most two hosts per serial line
  - error prone