Computer Networks
ICS 651

- router intervention for congestion
- Internet Explicit Congestion Notification
- FIFO queueing
- fair queueing
- TCP review
- UDP
- Stream Control Transmission Protocol (SCTP)
- Real Time Protocol (RTP) and RTP Control Protocol (RTCP)

Router Intervention to Avoid or React to Congestion

- Random Early Discard -- 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) field are used to indicate (a) whether congestion notification is requested (ECT), and (b) whether the packet experienced congestion (CE).
- **TCP** uses two new bits: ECE (ECN-Echo, to report that a packet was received with the CE bit set), and CWR (Congestion Window Reduced), to indicate that the ECE bit was received.
- compatible with hosts and routers that don't do ECN

typical usage of ECN

- senders can set ECT
- routers can change ECT to CE to record that congestion was experienced, perhaps instead of dropping a packet
- transport layer is informed of CE, sends an ECE
- receiver of ECE reduces congestion window, sends CWR

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 Internet, with TCP congestion control
- if all but one sender do congestion control, and one 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
- "good enough" (TCP Reno) may be preferable to "better" (TCP Vegas)

UDP

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

```
 0      7 8     15 16    23 24    31
+--------+--------+--------+--------+
|     Source      |   Destination   |
|      Port       |      Port       |
+--------+--------+--------+--------+
|                 |                 |
|     Length      |    Checksum     |
+--------+--------+--------+--------+
|
|          data octets ...
+---------------- ...
```

- RFC 768

our network so far

- TCP and IP (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, which are limited in speed (about 100,000 b/s max), distance (typically within a building), and number of hosts (at most two hosts per serial line)

our network so far

- TCP and IP (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, which are limited in speed (about 100,000 b/s max), distance (typically within a building), and number of hosts (at most two hosts per serial line)

Additional Internet Transport-Layer Protocols

- Stream Control Transmission Protocol (SCTP)
- Real Time Protocol (RTP) and RTP Control Protocol (RTCP)
- Real Time Streaming Protocol (RTSP)

Stream Control Transmission Protocol

- all the functions of TCP, and much more
- each connection carries any number of independent, reliable streams of packets
- a transmission sequence number and also a stream sequence number
- each IP packet can carry multiple independent chunks
- fragmentation of application packets
- end-system may be identified by multiple IP addresses
- congestion control done on the basis of IP address pairs
- flow control similar to TCP
- some support for security
- selective acknowledgements are not really piggybacked (except that multiple chunks may be in the same packet)
- checksum is 32-bit CRC
- RFC 4960
- in-class discussion: what problems does SCTP solve that TCP doesn't?

SCTP format

- packet structure

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Common Header                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Chunk #1                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           ...                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Chunk #n                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- common header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Source Port Number        |     Destination Port Number   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Verification Tag                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Checksum                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SCTP chunk format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Chunk Type  | Chunk  Flags  |        Chunk Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                          Chunk Value                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- chunk types: data (0), init and init ACK (2), SACK (3), heartbeat and ack (4, 5), abort (6), shutdown and ack (7, 8), error (9), cookie and ack (10, 11), ECNE and CWR (12, 13), shutdown complete (14)

SCTP data chunk format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 0    | Reserved|U|B|E|    Length                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              TSN                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Stream Identifier S      |   Stream Sequence Number n    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Payload Protocol Identifier                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                 User Data (seq n of Stream S)                 /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- U means unsequenced (i.e., ignore the stream sequence number), B and E mark the beginning and end fragments of a user message

Real-Time Protocol, RTP

- for time-sensitive data where loss is acceptable
- defined as two protocols, RTP and RTP Control Protocol or RTCP (RFC 3550)
- RTCP provides quality of service and congestion feedback
- an RTP profile is defined for video and audio (RFC 3551)
- typically layered on top of UDP, with an even port number for RTP and the next higher port number for RTCP

- 0 1 2 3
- 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
- +-

Real-Time Protocol, RTP

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- P records whether the data is padded
- X records whether there is a header extension
- CC is the number of contributing source identifiers
- M is for application purposes in marking within the stream
- PT is profile-dependent, identifies media type
- sequence number is for application purposes, may not be delivered in sequence
- timestamp identifies the time of the sample
- synchronization source is a randomly chosen ID for each source of timestamps
- contributing sources are the SSRC IDs of the sources that have contributed to this data

Additional Real Time Protocols

- Real Time Streaming Protocol controls one or more sources of real time data, e.g. RTP streams (RFC 2326)
- Session Initiation Protocol provides additional session-level primitives, (RFC 3261)

Real-Time Audio and Video transmission

- original digital telephone network designed to carry 8,000 voice samples per second per channel: synchronous network
- compressed audio and video still need timely delivery, but do not have bandwidth guarantees
- audio and video may be multicast for conferencing
- data loss and data delay have the same effect
- for pre-recorded data, if maximum delay is known, can pre-stream and buffer a corresponding number of frames (Video On Demand, VoD)
- actual bandwidth, delay, and jitter experienced on stream is the Quality of Service, QoS
- a QoS may also be requested as part of resource reservation
- if audio stream is given higher priority,
  - can tolerate higher data loss, and
  - temporary loss of video stream is less disruptive than temporary loss of audio stream
- video and audio streams must be re-synchronized at receiver