Congestion Control: Concepts

Topics

- Introduction, objectives, and issues
- Buffer management and queuing disciplines
- Taxonomy
- Window-based flow control
- Rate-based flow control
Network Congestion

- Suppose that the externally offered load is larger than what can be carried by the network, even with optimal routing -- this leads to network congestion
  - Queue lengths increase
  - Delays increase
  - Buffers fill and packets must be discarded
  - Acknowledgment waits time-out
  - Packets are retransmitted, further increasing load

- At times, it is necessary to restrict the accepted traffic to avoid network congestion

Congestion Control

- Congestion control attempts to limit the accepted traffic to reduce network congestion

- Delay within the network is reduced, but congestion may be pushed to high layers
Flow Control Versus Congestion Control

- The two end-points of a connection may have greatly different speeds
  - Sender can transmit packets faster than they can be processed at the receiver
  - Buffers fill at receiver and packets are discarded
  - Leads to unnecessary retransmissions, lost data
- The sender must restrict its transmissions to match the speed of the receiver
  - For example, receiver in TCP indicates a window size
- This is flow control versus congestion control

Ways to Impose Congestion Control (1)

- Call blocking
  - A “session” is blocked from entering the network
  - Traditionally used in circuit-switched networks
    - “Busy signal” in circuit-switched voice networks
  - Becoming important in ATM
    - CBR, VBR, or ABR session requests a connection with guaranteed bandwidth and/or maximum latency -- quality of service (QoS)
    - Network will accept or block the connection
    - Network could offer a “degraded” connection that may be acceptable to the application
Ways to Impose Congestion Control (2)

- Packet (cell) discarding
  - Packets are discarded when buffers are full or are becoming full
  - Discarding can be selective
    - Packets from low priority sessions
    - Packets from sessions exceeding their allotted bandwidth
    - Packets that are likely to be discarded elsewhere
- Recovery for discarded packets
  - Retransmitted or lost (no value if delayed, e.g. video)
  - Need for high and low priority (ATM CLP bit)

Ways to Impose Congestion Control (3)

- Packet blocking
  - Individual packets can be prevented from entering the network
    - Buffer for later transmission (high priority)
    - Discard (low priority)
  - It is better to prevent a packet from entering the network than to have it use resources yet be discarded elsewhere
  - Blocking should adhere to quality of service agreed to upon connection establishment
Ways to Impose Congestion Control (4)

- Packet scheduling
  - Nodes in the network may selectively expedite or delay packet transmission
  - Selection criteria
    - Delay low priority traffic in favor of high priority traffic
    - Use a round-robin schedule to ensure fairness
    - Delay packets that will use congested resources elsewhere in the network

Imposing Congestion Control

- The *means* of congestion control are relatively easy to implement
- The difficulty is determining *when* and to *what degree* to invoke flow control
Objectives of Congestion Control

- Two main objectives of congestion control
  - Delay ... Achieve a balance between rejecting traffic (which decreases throughput) and maintaining low delay
  - Fairness ... Maintain fairness to all “sessions” in maintaining quality of service

- Secondary objective of congestion control
  - Buffer overflow ... Prevent buffer overflow that can lead to deadlock or extreme throughput degradation
    - Related to delay and fairness

---

Delay Objective (1)

- Different objectives for high and low priority traffic
  - High priority traffic: Congestion control should attempt to provide negotiated quality of service (low delay)
  - Low priority traffic: Congestion control should reject traffic to avoid disastrous traffic jams

- Rejecting traffic prevents retransmissions due to ...
  - Discarded packets within the network
  - Slow acknowledgment
**Delay Objective (2)**

- Rejecting traffic shifts delay to a higher layer -- it does not necessarily reduce latency as seen by a user
  - Total delay can be reduced by adding capacity
  - Routing may be adjusted to better utilize capacity

---

**Delay Objective (3)**

- Ideal congestion control would keep accepted load less than $\gamma^*$
- Unfortunately, it is impossible to accurately determine throughput and delay in a dynamic and distributed manner

Excessive delay

- Buffer overflow
- Rejected packets
- Slow acks
There is a relationship between throughput and delay.

Power is a measure of the efficiency of the congestion control scheme:

\[ \text{Power} = \frac{\text{Throughput}^\alpha}{\text{Delay}}, \quad 0 < \alpha < 1 \]

- Maximized at knee of curve with \( \alpha = 1 \)

When traffic is rejected, congestion control should do so “fairly.”

Fairness applies within a priority or service class:
- High priority traffic normally accepted in lieu of low priority traffic

Fairness must reach a compromise between:
- Providing equal access to resources for sessions
- Throttling sessions that cause congestion
**Fairness versus Throughput: Example**

- Assume all session generate the same amount of traffic
- Throughput maximized at $n$ if Session 1 through $n$ given full capacity, Session 0 given 0 capacity
- Fairness maximized at $(n+2)/2$ if all sessions given one half capacity

![Diagram of network with sessions and links]

**Buffer Overflow Objective**

- The objective of reducing buffer overflow is related to the delay and fairness objectives
  - Rejected packets due to buffer overflow increase retransmissions which then waste capacity and increase delay
  - Buffer resources should be shared fairly by competing sessions
- Buffer overflow can also lead to deadlock
Deadlock Due to Buffer Overflow

- Deadlock can occur due to buffer overflow
- Simple example:

\[ \begin{array}{cc}
A & B \\
\text{---} & \text{---} \\
\end{array} \]

- \( A \) has a packet for \( B \), but \( B \) cannot accept the packet since its buffer is full
- \( B \) has a packet for \( A \), but \( A \) cannot accept the packet since its buffer is also full
- Situation cannot change unless either \( A \) or \( B \) discards a packet
- May also occur with more complex cycles

Buffer Management (1)

- A buffer management scheme is needed
  - To ensure fairness
  - To avoid deadlock
- One scheme is to divide buffer space into \( N \) pools (\( N \) is number of nodes in network in the “ideal” case)
  - Space is allocated from pool \( k \) for packets that have traversed \( k \) links
  - Packets are rejected if there is no space in their pool
  - Provably prevents deadlock if there are \( N \) pools and no looping (discard if \( N \) hops)
Buffer Management (2)

- Different pools (classes) used at source and destination

Node A

| class N-1 | class N-1 |
| class k+1 | class k+1 |
| class k   | class k   |
| class 1   | class 1   |
| class 0   | class 0   |

Node B

- Which packet to send next (order of service)?
- In the event of congestion, which packet(s) to discard?

Queuing Disciplines (1)

- Issues
  - Which packet to send next (order of service)?
  - In the event of congestion, which packet(s) to discard?

- First-In First-Out (FIFO) or First-Come First-Served (FCFS)
  - Typically couples order of service with discard strategy -- but can be separated
  - Number of queues
    - One queue, all traffic at same priority
    - Multiple queues, each for a different priority of traffic
Queuing Disciplines (2)

- **Fair Queueing (FQ)**
  - Can handle a badly behaved flow
  - Separate queue for each flow
  - To avoid “hogging” by flows with long packets, must operate as much as possible like a “bit-by-bit” round robin scheme

---

Taxonomy (1)

- Router Centric
- Rate-Based
- Feedback-Based
- Reservation-Based
- Window-Based
- Host Centric
Router-centric versus host-centric: Where is the main functionality of congestion control?
- Hosts -- at the edge of the network
- Routers -- inside the network

Reservation-based versus feedback-based
- Hosts can reserve capacity
  - Network accepts or rejects (blocks)
- Network can provide feedback to hosts
  - Explicit
  - Implicit

Window-based versus rate-based
- Window-based: hosts are controlled by limiting their send window size
  - Messages update the size of the window
- Rate-based: hosts transmit at some rate
### Another Taxonomy

<table>
<thead>
<tr>
<th>Congestion control schemes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Open loop control</td>
<td></td>
</tr>
<tr>
<td>Source control</td>
<td></td>
</tr>
<tr>
<td>Destination control</td>
<td></td>
</tr>
<tr>
<td>Closed loop control</td>
<td></td>
</tr>
<tr>
<td>Implicit feedback (global)</td>
<td></td>
</tr>
<tr>
<td>Explicit feedback (global)</td>
<td></td>
</tr>
<tr>
<td>Persistent (global)</td>
<td></td>
</tr>
<tr>
<td>Responsive (global)</td>
<td></td>
</tr>
<tr>
<td>Global</td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td></td>
</tr>
</tbody>
</table>


### Window-Based Congestion Control (1)

- Window-based schemes widely used for flow control and congestion control

- A session between transmitter A and receiver B is window flow controlled if there is some finite upper limit on the number of data units sent by A but not known to have been received (and processed) by B
  - Upper bound is known as the window size or the window
  - Data units can be, for example, messages, packets, bytes
Window-Based Congestion Control (2)

- **Classification**
  - If window size remains fixed for duration of the virtual circuit:
    - Open loop with source control
  - If window size can be modified during the session by feedback from the destination:
    - Closed loop with implicit feedback (controlled by implicit status)
    - Closed loop with local explicit feedback (explicitly set by destination)

- With dynamic windows, this is a form of credit-based flow control or congestion control

Window-Based Congestion Control (3)

- B notifies A that it has processed a data unit by sending a special message to A
  - Known as a permit or acknowledgment
  - Permit allows A to send another data unit to B
  - Permits are sent as
    - Special messages
    - Piggybacked on reverse direction packets

\[ A \xrightarrow{\text{data units}} B \xleftarrow{\text{permits}} \]
Window-Based Congestion Control (4)

- Window-based flow control is often combined with error control
  - Permits are also acknowledgments for ARQ
  - TCP uses combined error and flow control
- Delayed return of permits will throttle the sender
  - Permits (or packet triggering permit) may be delayed due to congestion in the network, so sender is slowed down to reduce congestion
  - Receiver may intentionally delay permits to throttle sender, typically to prevent buffer overflow

Window-Based Congestion Control (5)

- Window-based flow and congestion control strategies
  - End-to-end
  - Node-by-node

[Diagram of end-to-end and node-by-node flow control]
End-to-End Windows

- End-to-end flow control operates at the entry and exit point of the subnet
- Window size is set at $\alpha W$ ($\alpha$ and $W$ positive integers)
- After $\alpha$ packets (or other data units) are received, a permit is returned to the sender
  - At most $W$ permits may be outstanding

Operation of End-to-End Windows (1)

- Assume $\alpha = 1$
- Packets are usually assigned an identifier
  - Identifier $k$, $0 \leq k < m$, where $m \geq W+1$
  - Permit indicates next expected $k$
- A sliding window protocol (go-back $n$ ARQ) is used
- Each packet contains
  - A sequence number (identifier $k$)
  - A permit for the reverse direction
- If next expected is $k$, sender may send up to packet $k + W - 1$ (modulo $m$)
Operation of End-to-End Windows (2)

Let
- \( X \) = transmit time of a packet (assume fixed)
- \( W \) = window size (in packets)
- \( d \) = round trip delay (transmit time, round-trip propagation delay, permit delay)

\[
X = \text{transmit time of a packet (assume fixed)}
\]
\[
W = \text{window size (in packets)}
\]
\[
d = \text{round trip delay (transmit time, round-trip propagation delay, permit delay)}
\]

If \( d \leq WX \), permit will return within time needed to transmit a full window of packets
- No delay due to flow control (flow control inactive)
- Throughput, \( \gamma \):
  \[
  \gamma = \frac{1}{X} \text{ (packet/sec)}
  \]
Operation of End-to-End Windows (4)

- If \( d > WX \), sender must stop transmission and wait for a permit
  - Flow control is active
  - Throughput, \( \gamma \):
    \[ \gamma = \frac{W}{d} \text{ (packet/sec)} \]

\[ WX \quad d \quad 1/X \quad W/d \]

\[ W = 4 \]

Operation of End-to-End Windows (5)

\[ \gamma = \min \left( \frac{1}{X}, \frac{W}{d} \right) \]

\[ W/d \quad WX \quad \text{round-trip delay, } d \]
Limitations of End-to-End Windows

- Minimum throughput for a session cannot be guaranteed
  - Unsuitable for video and voice sessions
- May fail to control packet delay
- May fail to enforce fairness
- Trade-off in setting window size
  - Small window size needed to limit congestion in the subnet
  - Large window size needed to allow full speed transmission and maximum throughput under light to moderate loading

High Data Rate Networks (1)

- Efficient window size depends on bandwidth-delay product
  - Data rate -- transmitter should not have to wait
  - Propagation delay -- time to return permit
- $2BT_p$ is the number of bytes in transit for a permit return
High Data Rate Networks (2)

- Consider a $B = 1$ Gbps link between New York and California using TCP/IP
  - Propagation delay is about $T_P = 30$ ms
  - $2BT_P = 7.5$ Mbytes (7,500 packets at 1000 bytes per packet)
- Need $W \geq 7,500,000$ bytes for efficient operation
  - TCP provides “credits” on a per byte basis
  - Sequence number modulus $m \geq W + 1$
  - A 24-bit window size field would work
    - $2^{24} = 16,777,216$
  - Current TCP window field is 16 bits
    - $2^{16} = 65,536$ (much too small!)

High Data Rate Networks (3)

- What would happen using current maximum TCP window size for the NY-to-CA link?

\[ W_X \approx 0.5 \text{ ms} \]
\[ d - W_X \approx 60 \text{ ms} \]

Utilization is about 0.9%!
A better solution would be to “pace” transmissions at some appropriate rate -- basis for *rate-based flow control*

- Throughput is the rate
- Time not wasted waiting for permits to return

It is often desirable to dynamically adjust window size

- Keep $W$ large under light to moderate loading to maximize throughput
- Reduce $W$ when congestion occurs

Feedback is needed from the point of congestion
Dynamic Window Size Adjustment (2)

- Special *choke packets* can be sent from congested nodes to appropriate source nodes
  - Source reduces window when choke received
  - Source gradually attempts to increase window size after a time-out
  - Strategy is usually *ad hoc* and based on trial and error experiments

Dynamic Window Size Adjustment (3)

- Source can sense delay in returned permits (increase in d) or excessive retransmissions
  - Such conditions are assumed to be caused by congestion
  - Source reduces window size
  - Source attempts to increase window after time-out
- Bits in packet header can be set by intermediate nodes to indicate congestion
  - System Network Architecture (SNA) uses two bits that can be set by intermediate nodes
    - Moderately congested
    - Badly congested
Node-by-Node Windows (1)

- Node-by-node windows may be used in virtual circuit networks
  - There is a window for each virtual circuit (an end-to-end window)
  - There is a separate window for each pair of adjacent nodes along the path of the circuit

- Used, for example in
  - Digital Network Architecture (DNA)
  - TYMNET

Node-by-Node Windows (2)

- Node-by-node path is just one link long, so window size of $W = 2$ (or 3) is reasonable
  - High data rate links require larger window

- Each node has $W$ buffer slots for each transmitter
  - Buffer slot is freed when:
    - Packet leaves subnet at local node, or
    - Packet relayed to next node
  - Permit sent when buffer is freed

\[ \text{to transport layer} \]

\[ \text{to other nodes} \]
**Back-Pressure**

- Each transmitting node will be forced to slow its rate if window (of size $W$) fills.
- Windows will progressively be filled from a congested link back to the source.
  - This process is known as *backpressure*.

![Diagram of Back-Pressure](image)

**Distribution of Packets (1)**

- Assume that an end-to-end flow control scheme and a node-by-node flow control scheme have similar aggregate window capacity for an n-link connection.
  - End-to-end: size is $nW$
  - Node-by-node: size is $W$

- Distribution of packets due to a congested link.
  - End-to-end: $nW$ packets at congested link
  - Node-by-node: $W$ packets at each node.
Even distribution of packets in node-by-node congestion control can reduce buffer requirements.

Even distribution of packets also alleviates fairness problem:
- Window sizes are the same at each link, so all connections treated fairly.
- Problem still exists if a node has incoming links with different windows contending for the same outgoing link.
Distribution of Packets (4)

- Problem when a node has incoming links with different window sizes contending for the same outgoing link.

\[\text{satellite link} \quad \text{big } W \quad \text{terrestrial link} \quad \text{small } W\]

Problems in High Data Rate Networks

- Window sizes must be very large due to large propagation delay-bandwidth product.

- Window-based congestion control does not regulate end-to-end delays.
  - Unsuitable for applications requiring low latency like video or voice.

- Window-based flow and congestion control often merged with error control (ARQ).
  - Often goals are conflicting, for example
    - Small window size good for error control.
    - Large window size needed for flow control.
Rate-Based Congestion Control (1)

- Rate-based congestion control is gaining importance due to high data rate networks and applications
  - Each session allocated guaranteed data rate
  - Allocated rate based on application needs

- Input session rates set based on ...
  - Delay-throughput trade-off
    - High rates provide best throughput
    - Rates that are too high lead to excessive delay
  - Fairness
    - Rates must be fairly allocated across all sessions

Rate-Based Congestion Control (2)

- Suppose that a session is allocated \( r \) packets/second

- Strict implementation
  - Allow 1 packet every \( 1/r \) seconds
  - Performance is like time-division multiplexing (TDM) -- poor performance for bursty traffic

- More reasonable scheme
  - Allow \( W \) packets every \( W/r \) seconds
  - Allows for bursts of up to \( W \) packets
Time Window Congestion Control (1)

- Time window congestion control is similar to window-based flow control except that permits are generated on a timed basis at the source rather than being returned by the destination.
- Allocation $W$ given to each session
  - Count $x$ is unused part of window, $0 \leq x \leq W$
  - Source can transmit as long as $x \geq 0$

Time Window Congestion Control (2)

- Operation ...
  - Count $x$ is decremented each time a packet is sent
  - Count $x$ is incremented $W/r$ seconds after transmit
**Time Window Flow Control** (3)

\[
x = \begin{array}{cccccc}
3 & 2 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[x\text{ is value before transmission begins}\]

\[W = 3\]

---

**Leaky Bucket Scheme**

- Leaky bucket scheme does a better job of controlling burstiness
- Count is incremented by 1 every \(1/r\) seconds, up to a maximum of \(W\)
- Leaky bucket analogy

- Packets waiting for a permit
- Packets with permits awaiting transmission
- One permit added every \(1/r\) seconds
- Holds only \(W\) permits
### Leaky Bucket Parameter $W$ (1)

- The bucket size $W$ is an important parameter in determining the effectiveness of the leaky bucket scheme.
- Small $W$ delays bursty traffic
  - $W = 1$ yields inefficient time-division multiplexing (TDM) behavior.
- Large $W$ can lead to downstream congestion and buffer overflow.

### Leaky Bucket Parameter $W$ (2)

- $W$ can be dynamically adjusted
  - Node can “choke” upstream sources with a special message.
  - Prediction needed if propagation delays are large.
You should now be able to ... (1)

- Describe the need for flow control and congestion control
- Define the difference between flow control and congestion control
- Define the objectives of flow control and congestion control
- Describe the difference between FIFO queuing and fair queuing
- Describe the difference between an end-to-end and node-by-node sliding window flow control scheme

You should now be able to ... (2)

- Describe the operation the time window congestion control scheme
- Describe the operation the leaky bucket congestion control scheme
- Quantitatively evaluate the performance of sliding window, time window, and leaky bucket congestion control schemes

Next ... Examples
- Rate-based control in ATM’s ABR service
- TCP’s congestion control scheme