

# INF3190 Group lecture

## Lecture #10

Jan Anders Bremer

UiO - IFI

05.04.2013

# Contact information

Phone: 91125994

Mail: [janabr@ifi.uio.no](mailto:janabr@ifi.uio.no)

IRC: [janabr at irc.ifi.uio.no](irc://irc.ifi.uio.no) (currently in [#ping.uio.no](irc://irc.ifi.uio.no/#ping.uio.no) and [#cyb](irc://irc.ifi.uio.no/#cyb), but you can always `/msg me`)

Feel free to send questions, suggestions and feedback.

- Congestion control
- Remote procedure call (RPC)

- Sending-rate exceeds the (available) transmission speed of the bottleneck of a route (note: the bottleneck may be a group of parallel connections)
- Devices (even end-points!) receiving more than they are able to handle.

- When congestion arises do the following (one or both):
  - Stop ACKing packets (as if they were dropped!)
  - Send congestion-messages
- The sender will respond by reducing the sending-rate. See the typical "TCP-saw".

# Congestion Control (CC) in networks

- Connection-less network: end-nodes must do the CC
- Connection-oriented network: may avoid congestion (and thus CC) by using line switching
- The internet is a connection-less network.

# Back pressure and soft state

- Backpressure routing - route packets using congestion gradients, continuous update
- Soft state - buffer packets in router if unable to send

# Congestion control in the OSI model

- In theory: I2-I4
- Actual: I4, with local negotiations on I2.



- Most common: FIFO - first in, first out
- Other: Fair queuing - round robin (everyone gets assigned bandwidth)

# Alternate CC mechanisms

- ECN - explicit congestion notification (bit in IP header)
- RED - random early detection, drop a random packet when congestion is imminent

- Slow start - exponential increase in sender window before a threshold, then increase linearly.
- Fast retransmit - when identical ACKs are received (usually: 3), retransmit the next segment as it is most likely lost
- Fast recovery - reduce window to slow-start threshold when congestion arises (rather than a lower value)

- Reaching max sender window: infinite bandwidth, 10 ms RTT, MSS is 2 KB, 24 KB recv-window
- $2 \times 2 * * 4 = 32$  - After 4 RTTs (40 ms), the recv-window size will be reached.
- TCP - 18 KB window, 2 KB MSS, calculate sender-window after timeout and 4 successful retransmissions (RTTs).
- 2, 4, 8, 10
- Fast recovery: 10 (rounded), 12, 14, 16