CSE 550: Systems for all

Au 2022

Ratul Mahajan
Transport protocol challenges

Flow control
  • Do not send faster than the receiver’s consumption capacity

Congestion control
  • Do not send faster than the networks delivery capacity
Congestion control challenges

Network capacity is unknown
  • Individual links can vary from less than 100 Kbps to more than 100 Gbps

Network capacity varies with time
  • E.g., distance from cellular base stations or WiFi access points

Network capacity is shared
  • Don’t know how many other users are there and their needs
What if you send too slow?
What if you send too fast?
Congestion Collapse in the 1980s

• Early TCP used fixed size window (e.g., 8 packets)
  • Initially fine for reliability
• But something happened as the network grew
  • Links stayed busy but transfer rates fell by orders of magnitude!
Nature of Congestion

• Routers/switches have internal buffering
Nature of Congestion (2)

• Simplified view of per port output queues
  • Typically FIFO (First In First Out), discard when full
Nature of Congestion (3)

Queues help by absorbing bursts when input > output rate

But if input > output rate persistently, queue will overflow
   This is congestion

Congestion is a function of the traffic patterns – can occur even if every link has the same capacity
Effects of Congestion

- Capacity
- Goodput (packets/sec)
- Offered load (packets/sec)
- Delay (seconds)
- Onset of congestion
- Desired response
- Congestion collapse
Effects of Congestion (2)

As offered load rises, congestion occurs as queues begin to fill:
  Delay and loss rise sharply with load
  Throughput < load (due to loss)
  Goodput << throughput (due to spurious retransmissions)

None of the above is good!
  Want network performance just before congestion
## Congestion control design space: Congestion indicator

Several possible signals, with different pros/cons

<table>
<thead>
<tr>
<th>Signal</th>
<th>Example Protocol</th>
<th>Pros / Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet loss</td>
<td>TCP</td>
<td>Hard to get wrong</td>
</tr>
<tr>
<td></td>
<td>Cubic TCP (Linux)</td>
<td>Hear about congestion late</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other events can cause loss</td>
</tr>
<tr>
<td>Packet delay</td>
<td>BBR (Google)</td>
<td>Hear about congestion early</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Need to infer congestion</td>
</tr>
<tr>
<td>Router indication</td>
<td>ECN, DCTCP</td>
<td>Hear about congestion early</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires router support</td>
</tr>
</tbody>
</table>
Congestion control design space: Control mechanism

Two main controls, but hybrids also possible

<table>
<thead>
<tr>
<th>Control</th>
<th>Example Protocol</th>
<th>Pros/Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window based</td>
<td>TCP</td>
<td>Self-clocking</td>
</tr>
<tr>
<td></td>
<td>Cubic TCP (Linux)</td>
<td>Can be Jittery</td>
</tr>
<tr>
<td></td>
<td>DCTCP</td>
<td></td>
</tr>
<tr>
<td>Rate based</td>
<td>BBR (Google)</td>
<td>Smoother</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tricky estimation</td>
</tr>
</tbody>
</table>
Over to Oliver and Wenxuan