TCP Fairness

**Fairness goal:** if $K$ TCP sessions share same bottleneck link of bandwidth $R$, each should have average rate of $R/K$

Simple scenario: assume same MSS and RTT
Is TCP Fair?

Two competing sessions:

• Additive increase gives slope of 1, as throughput increases
• Multiplicative decrease drops throughput proportionally

Equal bandwidth share

Connection 1 throughput

Connection 2 throughput

Loss: decrease window by factor of 2
Congestion avoidance: additive increase

Loss: decrease window by factor of 2
Congestion avoidance: additive increase
More on Fairness

Fairness and UDP

• Multimedia apps often do not use TCP
  – do not want rate throttled by congestion control
• Instead use UDP:
  – pump audio/video at constant rate, tolerate packet loss
• Research area: TCP friendly unreliable transport

Fairness and parallel TCP connections

• nothing prevents app from opening parallel connections between 2 hosts.
• Web browsers do this
• Example: link of rate $R$ supporting 9 connections;
  – new app asks for 1 TCP, gets rate $R/10$
  – new app asks for 11 TCPs, gets $11R/20$
    (over half the bandwidth!)
Queuing Mechanisms

Random Early Detection (RED)
Explicit Congestion Notification (ECN)
Bursty Loss From Drop-Tail Queuing

• TCP depends on packet loss to detect congestion
  – In fact, TCP *drives* the network into packet loss
  – … by continuing to increase the sending rate
• Drop-tail queuing leads to *bursty* loss
  – When a link becomes congested…
  – … many arriving packets encounter a full queue
  – And, as a result, many flows divide sending rate in half
  – … and, many individual flows lose multiple packets
Slow Feedback from Drop Tail

- Feedback comes when buffer is completely full
  - ... even though the buffer has been filling for a while
- Plus, the filling buffer is increasing RTT
  - ... and the variance in the RTT
- Might be better to give early feedback
  - Get one or two flows to slow down, not all of them
  - Get these flows to slow down before it is too late
Random Early Detection (RED)

- Basic idea of RED
  - Router notices that the queue is getting backlogged
  - ... and randomly drops packets to signal congestion

- Packet drop probability
  - Drop probability increases as queue length increases
  - If buffer is below some level, don’t drop anything
  - ... otherwise, set drop probability as function of queue

![Graph showing Probability vs. Average Queue Length](image-url)
Properties of RED

• Drops packets before queue is full
  – In the hope of reducing the rates of some flows
• Drops packet in proportion to each flow’s rate
  – High-rate flows have more packets
  – … and, hence, a higher chance of being selected
• Drops are spaced out in time
  – Which should help desynchronize the TCP senders
• Tolerant of burstiness in the traffic
  – By basing the decisions on *average* queue length
Problems With RED

• Hard to get the tunable parameters just right
  – How early to start dropping packets?
  – What slope for the increase in drop probability?
  – What time scale for averaging the queue length?

• Sometimes RED helps but sometimes not
  – If the parameters aren’t set right, RED doesn’t help
  – And it is hard to know how to set the parameters

• RED is implemented in practice
  – But, often not used due to the challenges of tuning right

• Many variations
  – With cute names like “Blue” and “FRED”… 😊
Explicit Congestion Notification

• Early dropping of packets
  – Good: gives early feedback
  – Bad: has to drop the packet to give the feedback

• Explicit Congestion Notification
  – Router marks the packet with an ECN bit
  – … and sending host interprets as a sign of congestion

• Surmounting the challenges
  – Must be supported by the end hosts and the routers
  – Requires two bits in the IP header (one for the ECN mark, and one to indicate the ECN capability)
  – Solution: borrow two of the Type-Of-Service bits in the IPv4 packet header
<table>
<thead>
<tr>
<th>Event</th>
<th>State</th>
<th>TCP Sender Action</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK receipt for previously unACKed data</td>
<td>Slow Start (SS)</td>
<td>CongWin = CongWin + MSS, If (CongWin &gt; Threshold) set state to “Congestion Avoidance”</td>
<td>Resulting in a doubling of CongWin every RTT</td>
</tr>
<tr>
<td></td>
<td>Congestion Avoidance (CA)</td>
<td>CongWin = CongWin+MSS * (MSS/CongWin)</td>
<td>Additive increase, resulting in increase of CongWin by 1 MSS every RTT</td>
</tr>
<tr>
<td>Loss event detected by triple duplicate ACK</td>
<td>SS or CA</td>
<td>Threshold = CongWin/2, CongWin = Threshold, Set state to “Congestion Avoidance”</td>
<td>Fast recovery, implementing multiplicative decrease. CongWin will not drop below 1 MSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Threshold = CongWin/2, CongWin = 1 MSS, Set state to “Slow Start”</td>
<td>Enter slow start</td>
</tr>
<tr>
<td>Duplicate ACK</td>
<td>SS or CA</td>
<td>Increment duplicate ACK count for segment being ACKed</td>
<td>CongWin and Threshold not changed</td>
</tr>
<tr>
<td>Transmission Round</td>
<td>Congestion Window</td>
<td>Threshold</td>
<td>State</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Out</td>
<td>Time Out</td>
<td>Time Out</td>
<td>Time Out</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triple Duplicate ACK</td>
<td>Triple Duplicate ACK</td>
<td>Triple Duplicate ACK</td>
<td>Triple Duplicate ACK</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fill-In Event Type</td>
<td></td>
<td></td>
<td>Slow Start</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission Round</td>
<td>Congestion Window</td>
<td>Threshold</td>
<td>State</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>Slow Start</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>SS</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>Switch to CA</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>CA</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4</td>
<td>CA</td>
</tr>
<tr>
<td>Time Out</td>
<td>Time Out</td>
<td>Time Out</td>
<td>Time Out</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>SS</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3</td>
<td>SS</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>3</td>
<td>Switch to CA</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>Triple Duplicate ACK</td>
<td>Triple Duplicate ACK</td>
<td>Triple Duplicate ACK</td>
<td>Triple Duplicate ACK</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>3</td>
<td>CA</td>
</tr>
<tr>
<td>Time-Out</td>
<td>Time-Out</td>
<td>Time-Out</td>
<td>Slow Start</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>4</td>
<td>SS</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>4</td>
<td>SS</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>4</td>
<td>Switch to CA</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>4</td>
<td>CA</td>
</tr>
</tbody>
</table>
Fair Queuing

• Some notation:
  \( R(t) \) = the number of rounds up to time \( t \)
  \( Nac(t) \) = the number of active flows at time \( t \)
  \( S(i,a) \) = the round packet \( i \) from flow starts
  \( F(i,a) \) = the round packet \( i \) from flow finishes

• In which round is a packet transmitted?

  \[
  S(i,a) = \text{Max}(R(\text{arrival}_\text{time}), F(i-1,a))
  \]

  \[
  F(i,a) = S(i,a) + P
  \]

• Can order packets by their finishing time \( F(i,a) \)
<table>
<thead>
<tr>
<th>Flow Number</th>
<th>Packet 4 (End of Queue)</th>
<th>Packet 3</th>
<th>Packet 2</th>
<th>Packet 1 (Front of Queue)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 1</td>
<td>100 bit packet</td>
<td>30 bit packet</td>
<td>20 bit packet</td>
<td>70 bit packet</td>
</tr>
<tr>
<td>Flow 2</td>
<td>20 bit packet</td>
<td>100 bit packet</td>
<td>20 bit packet</td>
<td>40 bit packet</td>
</tr>
<tr>
<td>Flow 3</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
</tr>
<tr>
<td>Flow 4</td>
<td>10 bit packet</td>
<td>80 bit packet</td>
<td>20 bit packet</td>
<td>100 bit packet</td>
</tr>
<tr>
<td>Flow Number</td>
<td>Packet 4 (End of Queue)</td>
<td>Packet 3</td>
<td>Packet 2</td>
<td>Packet 1 (Front of Queue)</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------</td>
<td>----------</td>
<td>----------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Flow 1</td>
<td>100 bit packet</td>
<td>30 bit packet</td>
<td>20 bit packet</td>
<td>70 bit packet</td>
</tr>
<tr>
<td>Flow 2</td>
<td>20 bit packet</td>
<td>100 bit packet</td>
<td>20 bit packet</td>
<td>40 bit packet</td>
</tr>
<tr>
<td>Flow 3</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
</tr>
<tr>
<td>Flow 4</td>
<td>10 bit packet</td>
<td>80 bit packet</td>
<td>20 bit packet</td>
<td>100 bit packet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow Number</th>
<th>Packet 4</th>
<th>Packet 3</th>
<th>Packet 2</th>
<th>Packet 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 1</td>
<td>S=120, F=220</td>
<td>S=90, F=120</td>
<td>S=70, F=90</td>
<td>S=0, F=70</td>
</tr>
<tr>
<td>Flow 2</td>
<td>S=160, F=180</td>
<td>S=60, F=160</td>
<td>S=40, F=60</td>
<td>S=0, F=40</td>
</tr>
<tr>
<td>Flow 3</td>
<td>S=120, F=160</td>
<td>S=80, F=120</td>
<td>S=40, F=80</td>
<td>S=0, F=40</td>
</tr>
<tr>
<td>Flow 4</td>
<td>S=200, F=210</td>
<td>S=120, F=200</td>
<td>S=100, F=120</td>
<td>S=0, F=100</td>
</tr>
<tr>
<td>Flow Number</td>
<td>Packet 4 (End of Queue)</td>
<td>Packet 3</td>
<td>Packet 2</td>
<td>Packet 1 (Front of Queue)</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------</td>
<td>----------</td>
<td>----------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Flow 1</td>
<td>100 bit packet</td>
<td>30 bit packet</td>
<td>20 bit packet</td>
<td>70 bit packet</td>
</tr>
<tr>
<td>Flow 2</td>
<td>20 bit packet</td>
<td>100 bit packet</td>
<td>20 bit packet</td>
<td>40 bit packet</td>
</tr>
<tr>
<td>Flow 3</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
<td>40 bit packet</td>
</tr>
<tr>
<td>Flow 4</td>
<td>10 bit packet</td>
<td>80 bit packet</td>
<td>20 bit packet</td>
<td>100 bit packet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow Number</th>
<th>Packet 4</th>
<th>Packet 3</th>
<th>Packet 2</th>
<th>Packet 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 1</td>
<td>S=120, F=220</td>
<td>S=90, F=120</td>
<td>S=70, F=90</td>
<td>S=0, F=70</td>
</tr>
<tr>
<td>Flow 2</td>
<td>S=160, F=180</td>
<td>S=60, F=160</td>
<td>S=40, F=60</td>
<td>S=0, F=40</td>
</tr>
<tr>
<td>Flow 3</td>
<td>S=120, F=160</td>
<td>S=80, F=120</td>
<td>S=40, F=80</td>
<td>S=0, F=40</td>
</tr>
<tr>
<td>Flow 4</td>
<td>S=200, F=210</td>
<td>S=120, F=200</td>
<td>S=100, F=120</td>
<td>S=0, F=100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow Number</th>
<th>Packet 4</th>
<th>Packet 3</th>
<th>Packet 2</th>
<th>Packet 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 1</td>
<td>770</td>
<td>360</td>
<td>230</td>
<td>170</td>
</tr>
<tr>
<td>Flow 2</td>
<td>580</td>
<td>520</td>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td>Flow 3</td>
<td>560</td>
<td>400</td>
<td>210</td>
<td>80</td>
</tr>
<tr>
<td>Flow 4</td>
<td>670</td>
<td>660</td>
<td>420</td>
<td>330</td>
</tr>
</tbody>
</table>
BGP Decision Process

- Calculate degree of preference for each route in Adj-RIB-In as follows (apply following steps until one route is left):
  - select route with highest LOCAL-PREF
  - select route with shortest AS-PATH
  - apply MED (if routes learned from same neighbor)
  - select route with smallest NEXT-HOP cost
More BGP Decision Process

- select route learned from E-BGP peer with lowest BGP ID
- select route from I-BGP neighbor with lowest BGP ID

- Install selected route in Loc-RIB
- Selectively disseminate routes to peers, update Adj-RIB-Out
- Done
You are AS1 with two links A & B to AS2. Suppose AS2 advertises 12/24 on both links. How to force all traffic to AS2’s prefix 12/24 through link A?

At a:
LP=10 for 12/24:b

At c:
LP=5 for 12/24:d
Multi-exit Discriminator (MED) Path Attribute

- Used when two AS’s connect to each other in more than one place
- Metric expresses degree of preference
- AS2 includes MEDs with prefixes sent to AS1 over links A, B
- AS1 uses these to select appropriate link when sending to prefix PFX
BGP Routing

• Route to 10.0.0.0/8 From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8 From RTR 2.3.4.5
  AS 65222,
  LocalPref = 120
  AS Path 65222
  Med = 60
BGP Routing

• Route to 10.0.0.0/8 From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8 From RTR 2.3.4.5
  AS 65222,
  LocalPref = 120  Higher Local Pref!
  AS Path 65222
  Med = 60
BGP Routing

• Route to 10.0.0.0/8 From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8 From RTR 2.3.4.5
  AS 65222,
  LocalPref = 100
  AS Path 65222
  Med = 60
BGP Routing

• Route to 10.0.0.0/8  From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8  From RTR 2.3.4.5
  AS 65222,
  LocalPref = 100
  AS Path 65222 Shorter AS Path!
  Med = 60
BGP Routing

• Route to 10.0.0.0/8  From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8  From RTR 2.3.4.5
  AS 65222,
  LocalPref = 100
  AS Path 65222, 65444, 65333
  Med = 60
BGP Routing

• Route to 10.0.0.0/8 From RTR 1.2.3.4
  AS 65111, Lowest Router ID
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8 From RTR 2.3.4.5
  AS 65222,
  LocalPref = 100
  AS Path 65222, 65444, 65333
  Med = 60
BGP Routing

• Route to 10.0.0.0/8 From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8 From RTR 2.3.4.5
  AS 65111,
  LocalPref = 100
  AS Path 65222, 65444, 65333
  Med = 60
BGP Routing

• Route to 10.0.0.0/8 From RTR 1.2.3.4
  AS 65111,
  LocalPref = 100
  AS Path 65111,65111,65333
  Med = 70

• Route to 10.0.0.0/8 From RTR 2.3.4.5
  AS 65111,
  LocalPref = 100
  AS Path 65222, 65444, 65333
  Med = 60

Lowest MED Value From Same AS
What’s Next

- Read Chapter 1, 2, 3, 4.1-4.3, 5.1-5.2, 6.1-6.4
- Next Lecture Topics from Chapter 9
  - Applications
- Homework
  - Due Thursday in lecture
- Project 3
  - Posted on the course website