Lecture 19 Overview

- Dealing with non-responsive traffic
  - Token bucket
  - Policing and shaping

- Scheduling
  - (Weighted) Fair Queuing
Non-responsive Senders

1 UDP (10 Mbps) and 31 TCPs sharing a 10 Mbps line

UDP (#1) - 10 Mbps
TCP (#2)
TCP (#32)

TCP (#2)
TCP (#32)

Bottleneck link (10 Mbps)
UDP vs. TCP

Throughput (Mbps)

Flow Number

CSE 123 – Lecture 19: Traffic Management
Token Bucket Basics

- Parameters
  - $r$ – average rate, i.e., rate at which tokens fill the bucket
  - $b$ – bucket depth (limits size of burst)
  - $R$ – maximum link capacity or peak rate (optional parameter)
- A bit can be transmitted only when a token is available
Traffic Policing

- Drop packets that don’t meet user profile
- Output limited to average of $r$ bps and bursts of $b$

![Diagram showing traffic policing process]

Packet input → Test if token

- Token: Proceed
- No token: Drop

$r$ bps

User Profile (token bucket)

Packet output
Traffic Shaping

- Shape packets according to user profile
- Output limited to average of $r$ bps and bursts of $b$

Queue, Drop on overflow
Packet input

Wait for token

User Profile (token bucket)

Packet output
Shaping Example

- $r = 100$ Kbps; $b = 3$ Kb; $R = 500$ Kbps

(a) $3$ Kb packet arrives at $T = 0$.

(b) $2.2$ Kb transmitted at $T = 2$ ms. 
   \[ b = 3$ Kb $- 1$ Kb $+ 2$ ms $* 100$ Kbps $= 2.2$ Kb \]

(c) $2.4$ Kb packet arrives at $T = 4$ ms.

(d) $T = 10$ ms: packet needs to wait until enough tokens are in the bucket.

(e) $0.6$ Kb transmitted at $T = 16$ ms.
Scheduling

● So far we’ve done flow-based traffic management
  ◆ Limit the rate of one flow regardless of the load in the network

● In general, need scheduling
  ◆ Dynamically allocate resources when multiple flows compete
  ◆ Give each “flow” (or traffic class) own queue (at least theoretically)

● Weighted fair queuing
  ◆ Proportional share scheduling
  ◆ Schedule round-robins among queues in proportion to some weight parameter
Our Previous Example

1 UDP (10 Mbps) and 31 TCPs sharing a 10 Mbps line
UDP vs. TCP w/FIFO
TCP vs. UDP w/Fair Queuing

![Graph showing throughput comparison between TCP and UDP flows with fair queuing.](image-url)

Throughput (Mbps) vs. Flow Number

Flow Number

1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24 26 28 30 32

TCP and UDP flows with fair queuing comparison.
(Weighted) Fair Queuing

Flow 1

Flow 2

Flow n

I/P

O/P
Maintain a queue for each flow
- What is a flow?

Implements **max-min fairness**: each flow receives \( \min(r_i, f) \), where
- \( r_i \) – flow arrival rate
- \( f \) – link fair rate (see next slide)

**Weighted Fair Queuing (WFQ)** – associate a weight with each flow to divvy bandwidth up non-equally
Fair Rate Computation

- If link congested, compute $f$ such that

\[ \sum_{i} \min(r_i, f) = C \]

\[ f = 4: \]
\[ \min(8, 4) = 4 \]
\[ \min(6, 4) = 4 \]
\[ \min(2, 4) = 2 \]
Another Example

- Associate a weight $w_i$ with each flow $i$
- If link congested, compute $f$ such that

$$\sum_i \min(r_i, f \times w_i) = C$$

\[
\begin{align*}
(w_1 = 3) & \quad 8 \\
(w_2 = 1) & \quad 6 \\
(w_3 = 1) & \quad 2 \\
\end{align*}
\]

\[
\begin{align*}
f = 2: \\
\min(8, 2 \times 3) & = 6 \\
\min(6, 2 \times 1) & = 2 \\
\min(2, 2 \times 1) & = 2
\end{align*}
\]

Flow $i$ is guaranteed to be allocated a rate $\geq w_i \times C / (\sum_k w_k)$

If $\sum_k w_k \leq C$, flow $i$ is guaranteed to be allocated a rate $\geq w_i$
For next time...

- HW 3 Due Friday
- Read P&D 6.3-4
- Keep going on Project 2