Fair Packet Queueing Algorithm

Yang Richard Yang

11/7/2001

Review

- IntServ
  - service model:
    - real-time service
    - best-effort service
    - link sharing
  - reference implementation
    - control path
      - RSVP, admission control
    - data path
      - classification, scheduling

- Diffserv
Packet Scheduling

- Decide when and what packets to send on output link
  - usually implemented at output interface

Seminal Paper: Three Issues

- Fair packet queueing algorithm
- Bandwidth and delay tradeoff
- Interactions between scheduling and congestion control
Outline

- Fair packet queueing algorithm
- Bandwidth and delay tradeoff
- Interactions between scheduling and congestion control

Objective: Fair Rate

- If link congested, compute $f$ such that
  \[ \sum_i \min(r_i, f) = C \]

\[
\begin{align*}
8 & \quad \text{min}(8, 4) = 4 \\
6 & \quad \text{min}(6, 4) = 4 \\
2 & \quad \text{min}(2, 4) = 2
\end{align*}
\]
Objective: Weighted Fair Rate Computation

- Associate a weight $w_i$ with each flow $i$
- If link congested, how to compute $f$?

Implementation: Fluid Approximation

- General Processor Share (GPS) is defined in an idealized fluid flow model
  - multiple queues can be serviced simultaneously
  - no non-preemption unit
- Real systems are packet systems
  - only one queue is served at a given time
  - packet transmission will not be preempted
- Goal
  - define packet algorithms that approximate the fluid system
  - maintain most of the important properties
**Generalized Processor Share**

- Red session has packets backlogged between time 0 and 10
- Other sessions have packets continuously backlogged

**Approximating GPS with WFQ**

- Fluid GPS system service order
- Weighted Fair Queueing (WFQ)
  - select the first packet that finishes in GPS
Implementation: Virtual Clock

- $V(t)$: virtual clock (or virtual round in this paper)
- If we serve $w_i$ bits per round for a backlogged flow with weight $w_i$, how fast will $V(t)$ progress:
  \[
  \frac{d}{dt} V = \frac{C}{\sum_{j \in N_{ac}(t)} W_j}
  \]
  - where
    - $N_{ac}(t)$ - number of flows backlogged

Virtual Time Implementation of Weighted Fair Queueing

\[
\begin{align*}
V(0) &= 0 \\
S_j^{k+1} &= F_j^k & \text{if session } j \text{ backlogged} \\
S_j^{k+1} &= \max(F_j^k, V(a_j^k)) & \text{if session } j \text{ un-backlogged} \\
F_j^{k+1} &= S_j^{k+1} + \frac{L_j^k}{W_j}
\end{align*}
\]

- $a_j^k$ - arrival time of packet $k$ of flow $j$
- $S_j^k$ - virtual starting time of packet $k$ of flow $j$
- $F_j^k$ - virtual finishing time of packet $k$ of flow $j$
- $L_j^k$ - length of packet $k$ of flow $j$
System Virtual Time in GPS

Virtual Start and Finish Times

- Utilize the time the packets would start $S^k_i$ and finish $F^k_j$ in a fluid system
  
  $F^k_j = S^k_i + \frac{L^k_j}{W_j}$
**Improvement**

- Fluid-Flow (GPS)
- WFQ (smallest finish time first)

**Outline**

- Fair packet queueing algorithm
  - Bandwidth and delay tradeoff
- Interactions between scheduling and congestion control
Reduce the Delay of Low Rate Flows

Define bid $B$:

$$B_j = \max(F_j, V(a_j^k) - \delta)$$

Alternative: Service Curve

[Cruz '95]

The QoS measures (delay, throughput, loss, cost) depend on offered traffic, and possibly other external processes.

A service model attempts to characterize the relationship between offered traffic, delivered traffic, and possibly other external processes.
Arrival and Departure Process

Network Element

$R_{in}$  $R_{out}$

$R_{in}(t)$ = arrival process
= amount of data arriving up to time $t$

$R_{out}(t)$ = departure process
= amount of data departing up to time $t$

Traffic Envelope
(Arrival Curve)

- Maximum amount of traffic that a flow can send during an interval of time $t$

$b(t) = Envelope$

slopemax average rate

"burstiness constraint"
Service Curve

- Assume a flow that is idle at time $s$ and it is backlogged during the interval $(s, t)$
- Service curve: the minimum service received by the flow during the interval $(s, t)$

Big Picture

- $R_{in}(t)$
- $R_{out}(t)$
- Service curve with slope $= C$
Delay and Buffer Bounds

- $S(t) = \text{service curve}$
- $E(t) = \text{Envelope}$
- Maximum delay
- Maximum buffer

Service Curve-based Earliest Deadline (SCED)

- Packet deadline - time at which the packet would be served assuming that the flow receives no more than its service curve
- Serve packets in the increasing order of their deadlines

- Properties
  - If sum of all service curves $\leq C* t$
  - All packets will meet their deadlines modulo the transmission time of the packet of maximum length, i.e., $L_{\text{max}} / C$
Linear Service Curves: Example

Video packets have to wait after FTP packets.

Non-Linear Service Curves: Example

Video packets transmitted as soon as they arrive.
Outline

- Fair packet queueing algorithm
- Bandwidth and delay tradeoff
  - Interactions between scheduling and congestion control

Discussion

- Will fair queueing solve the congestion control problem?
Mixed Queueing and CC

Buffer Size: 15

Results

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Queueing Policy</th>
<th>Generic FTP</th>
<th>JK FTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput (packets)</td>
<td>FQ</td>
<td>1162</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>FCFS</td>
<td>1102</td>
<td>566</td>
</tr>
<tr>
<td>Average Roundtrip Time</td>
<td>FQ</td>
<td>2.15</td>
<td>2.14</td>
</tr>
<tr>
<td></td>
<td>FCFS</td>
<td>2.11</td>
<td>2.07</td>
</tr>
<tr>
<td>Retransmitted Packets</td>
<td>FQ</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>FCFS</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Dropped Packets</td>
<td>FQ</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>FCFS</td>
<td>0</td>
<td>48</td>
</tr>
</tbody>
</table>