HPF: A Transport Protocol For Supporting Heterogeneous Packet Flows in the Internet

Jia-Ru Li, Sungwon Ha, Vaduvur Bharghavan
University of Illinois at Urbana-Champaign

IEEE INFOCOM ‘99

Presented by Hui-Wen Kan   2006/10/11
Outline

• Introduction
• Architecture and design
  – Application Framing Sub-Layer
  – WRTF Sub-Layer
• Performance measurements
• Conclusions
Introduction

• Current Internet transport protocols do not support the concept of *heterogeneous flows*
  – heterogeneous packet flow means different packets in the same flow will have different QoS requirements

• HPF: a transport protocol for supporting heterogeneous packet flows
Introduction (cont.)

• Four key features of HPF:
  – Support for reliable and unreliable packets with different priority and timing requirements
  
  – Support for application-level framing, and for applications to specify the priority, reliability and timing requirements of each frame
Introduction (cont.)

- Use application-specified priorities as hints for routers to preferentially drop low-priority packets during congestion

- Decouple the congestion control and reliability mechanisms
Architecture and design

• Allow applications to specify blocks of data called *frames*, and to provide frame-specific policies for reliability, priority and timing

• All the packets belonging to the same frame have the same policy parameters

• A frame is read or written as a single unit by the application
Architecture and design (cont.)

HPF has three logical sub-layers.

Application
- read/write frames or packets
- specify priority, reliability, delay
- send/receive frames/packets

Application Framing Sub-Layer

WRTF Sub-Layer

Congestion Control Sub-Layer

Network Layer
- optional: priority-based packet drop

HPF

send/receive packets
Architecture and design (cont.)

• Application Framing Sub-Layer
  – segment frames into packets
  – reassemble packets into frames
  – convert frame policies to packet policies
  – provide HPF API
• Each packet has a sequence number and the following parameters:
  – reliability field (1 bit)
  – priority field (n bits)
  – delay field (16 bits)
  – frame field (1 bit)
Architecture and design (cont.)

• Window, Reliability, Timing, and Flow Control (WRTF) Sub-Layer
  – deadline tagging
  – retransmissions
  – flow control
  – connection establishment and teardown
Architecture and design (cont.)

• Three types of packets
  – reliable packets
    • retransmitted until it has been acknowledged
  – unreliable delay-bounded packets
    • retransmitted until it has either been acknowledged or its deadline has expired
  – unreliable best-effort packets
    • never retransmitted
Architecture and design (cont.)

• Retransmissions
  – reliable and unreliable best effort packets: simple
  – unreliable delay-bounded packet:
    • the packet which can be retransmitted and acknowledged before deadline is retained in the retransmission queue

\[ \frac{b}{r} + T < s \]

\( b: \) packet size  
\( r: \) transmission rate  
\( T: \) RTT  
\( s: \) deadline - time \( t \)
Architecture and design (cont.)

- Flow Control and Window Advancement
  - sender can control the receiver window by parameters in the packet
  - Each packet has three parameters
    - s: sequence number
    - h: previous reliable packet sequence number
    - w: move the receiver window up to w, if this packet is received
Architecture and design (cont.)

Fig. 2. Illustration of the window advancement algorithm in HPF.
Performance measurements

The experimental testbed configuration used for the performance tests.
Performance measurements (cont.)

<table>
<thead>
<tr>
<th>Percent</th>
<th>TCP (Sec)</th>
<th>HPF (Sec)</th>
<th>HPF bytes dropped</th>
<th>TCP/HPF speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>13.806</td>
<td>13.810</td>
<td>0%</td>
<td>≈ 1</td>
</tr>
<tr>
<td>5%</td>
<td>24.56</td>
<td>13.37</td>
<td>4.5%</td>
<td>1.84</td>
</tr>
<tr>
<td>10%</td>
<td>53.26</td>
<td>13.73</td>
<td>8.4%</td>
<td>3.88</td>
</tr>
<tr>
<td>20%</td>
<td>149.91</td>
<td>17.33</td>
<td>16.8%</td>
<td>8.65</td>
</tr>
</tbody>
</table>

Table I. The speedup of HPF vs TCP with different percentages of random drop.
Table II. The performance of HPF vs TCP for various High:Low priority ratios with multiple concurrent streams.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>High:Low Ration</th>
<th>Packets Dropped</th>
<th>Time improvement vs TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>all high</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>HPF</td>
<td>50:50</td>
<td>4.9710%</td>
<td>9.9421%</td>
</tr>
<tr>
<td></td>
<td>33:66</td>
<td>7.6258%</td>
<td>16.8454%</td>
</tr>
<tr>
<td></td>
<td>20:80</td>
<td>12.5544%</td>
<td>17.7908%</td>
</tr>
<tr>
<td></td>
<td>10:90</td>
<td>16.9267%</td>
<td>34.7568%</td>
</tr>
<tr>
<td></td>
<td>5:95</td>
<td>23.7603%</td>
<td>41.5078%</td>
</tr>
</tbody>
</table>

Performance measurements (cont.)
Performance measurements (cont.)

Table III. The performance of HPF vs TCP for various High:Low priority ratios over the internet.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>High:Low Ratio</th>
<th>Packets Dropped</th>
<th>Improvement vs TCP(time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>all high</td>
<td>0%</td>
<td>—</td>
</tr>
<tr>
<td>HPF</td>
<td>66:33</td>
<td>0.71%</td>
<td>31.77%</td>
</tr>
<tr>
<td></td>
<td>50:50</td>
<td>2.86%</td>
<td>51.60%</td>
</tr>
<tr>
<td></td>
<td>33:66</td>
<td>4.29%</td>
<td>64.76%</td>
</tr>
<tr>
<td></td>
<td>10:90</td>
<td>5.71%</td>
<td>69.08%</td>
</tr>
<tr>
<td></td>
<td>5:95</td>
<td>7.86%</td>
<td>69.93%</td>
</tr>
<tr>
<td></td>
<td>0:100</td>
<td>7.86%</td>
<td>75.07%</td>
</tr>
</tbody>
</table>
Conclusions

• Support heterogeneous packet flows

• Support application-specified priority, reliability and timing requirements in each frame