Horde: Separating Network Striping Policy from Mechanism

ACM MobiSys 05’
Asfandyar Qureshi and John Guttag
MIT Computer Science and AI Lab

Presented by Hsien-Tai Chiu
02/15/06
Outline

- Introduction
- Horde Architecture
- Policy and Scheduling
- Experimental Evaluation
- Conclusion
Problem 1:

- WWAN are neither stable nor homogeneous

<table>
<thead>
<tr>
<th></th>
<th>Mean ($\mu$)</th>
<th>Sdev ((\sigma))</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPRS upload bandwidth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(stationary)</td>
<td>25kbps</td>
<td>1</td>
</tr>
<tr>
<td>(moving)</td>
<td>19kbps</td>
<td>5</td>
</tr>
<tr>
<td>CDMA upload bandwidth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(stationary)</td>
<td>130kbps</td>
<td>5</td>
</tr>
<tr>
<td>(moving)</td>
<td>120kbps</td>
<td>22</td>
</tr>
<tr>
<td>GPRS small packet RTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(stationary)</td>
<td>560ms</td>
<td>100</td>
</tr>
<tr>
<td>(moving)</td>
<td>760ms</td>
<td>460</td>
</tr>
<tr>
<td>CDMA small packet RTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(stationary)</td>
<td>460ms</td>
<td>90</td>
</tr>
<tr>
<td>(moving)</td>
<td>470ms</td>
<td>120</td>
</tr>
<tr>
<td>768-byte packet RTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(CDMA)</td>
<td>810ms</td>
<td></td>
</tr>
<tr>
<td>(GPRS)</td>
<td>920ms</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Summary of WWAN QoS characteristics.
Introduction(3/6)

- Problem 2:
  - Different requirement
    - Real-time video streams care about latency
    - Bulk-data transfer don’t care about latency
    - Interactive audio streams care about loss
Introduction(4/6)

- Horde’s Approach
  - Problem 1 Solution:
    - Network channel manager
  - Problem 2 Solution:
    - Allows an application to modulate network QoS by exporting Objective
Figure 2: Packet round-trip-time distributions with different schedulers,

The two schedulers were run over the same packet traces, collected from three existing WWAN channels (one CDMA and two GPRS channels).

Stream 1: EKG data  Stream 2: Video (latency sensitive)
Stream 3: additional data  Stream 4: Audio (latency sensitive)
Horde is useful when dealing with:
- Heterogeneous Data Streams
- Heterogeneous/Time-varying Network Channels
- Bandwidth-Limited Application
- Single Application
Horde Architecture (1/7)

- Application Interface
  - Connection-based model.
  - Application Stream $\leftrightarrow ADU \leftrightarrow$ Horde
  - Sends *throttle* event to application when network bandwidth changes
  - Inject *objectives* into middleware to modulate the network QoS for those streams
Horde Architecture (2/7)

- Internal structure

Interface with application

Code

Stripping layer

Deal with network channels

Congestion control

Packet transmission probe
Horde Architecture (3/7)

- Network Channel Managers
  - Handle Network I/O
  - Maintains a predictive model for the QoS
  - Performs congestion control
 Horde Architecture (4/7)

- Stream Flow Control
  - Uses a simple adaptive min-max fairness policy to allocate available bandwidth among streams
  - Available bandwidth changes, \textit{throttle} delivered to application.
    Simple stream can ignore, adaptive stream will change their sending rates up or down.
Horde Architecture (5/7)

- Transmission Slots
  - Interface between channel manager and packet scheduler
  - The capability to transmit data on a channel at a time, along with the expected QoS
  - Scheduler work base on Slot’s information
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelID</td>
<td>Parent network channel for this slot.</td>
<td></td>
</tr>
<tr>
<td>sequence</td>
<td>The sequence number for this slot on its parent channel.</td>
<td></td>
</tr>
<tr>
<td>lossProbability</td>
<td>The estimated loss probability for a packet transmitted in this slot.</td>
<td>( loss \in [0, 1] )</td>
</tr>
<tr>
<td>expectedRTT</td>
<td>Expected time between the transmission of data in this slot and the reception of an acknowledgment.</td>
<td>( milliseconds \geq 0 )</td>
</tr>
<tr>
<td>lossCorrelation(\text{other})</td>
<td>Compares two transmission slots to see if packet losses in the two slots are expected to correlate. This is an estimate for ( P(\text{both lost }</td>
<td>\text{ either one lost}) ).</td>
</tr>
<tr>
<td>cost(x)</td>
<td>Cost of transmitting (x) bytes in this slot.</td>
<td>( cost(x) \geq 0 )</td>
</tr>
<tr>
<td>maximumSize</td>
<td>The maximum number of bytes that can be transmitted in this slot.</td>
<td>( size &gt; 0 )</td>
</tr>
</tbody>
</table>

Figure 4: The main components of Horde’s transmission slot (txSlot) abstraction.
Horde Architecture (7/7)

- Transmission Capabilities
  (from ncManager congestion control logic)
- Latency: weighted moving average of previously seen RTT
- Loss: average(total lost / total delivered)
- Correlated Loss: a loss in one slot is expected to correlate with loss in the other
- Transmission Costs: money
Policy and Scheduling (1/6)

- Application Utility

\[ \text{utility}_{app}(\tau) \sim f_{app}(\text{history}_{tx}(\tau), \text{history}_{rx}(\tau)) \]

- \((\text{history}_{tx})\) : packet transmission history

- \((\text{history}_{rx})\) : ack reception history

- Used by the scheduler rank its scheduling choices
‘Optimal’ Scheduling

\[
\text{maximize } \left[ \sum_{\forall \text{stream}} \left( \text{utility}_{\text{stream}}(\tau) \right) \right]
\]

- Impractical
Objective Driven Scheduling

- Objectives drives the packet scheduler towards schedules that provide high utility to applications
- Objectives can be injected and removed dynamically.
- Objectives describe value relationships between ADU’s and txSlot’s
Policy and Scheduling (4/6)

- Objective Specification Language

```objective
objective {
    context {
        adu:foo { (stream_id == "videol") &&
                    (frame_type == "I") }
        adu:bar { (stream_id == "videol") &&
                    (frame_type != "I") }
    }
    goal { prob(foo::lost?) < prob(bar::lost?) }
    utility { foo { 100 } }
}
```

```objective
objective {
    context {
        stream:foo { stream_id == "audiol" }
    }
    goal { foo::latency_ave < 1000 }
    utility { foo { 100 } }
}
```

Figure 6: An objective expressing the policy that the average latency on a stream should be less than one second.

Figure 5: An objective expressing the policy that, for stream videol, txSlot’s carrying I-frames should have lower loss probabilities than slots for other frames.
Policy and Scheduling (5/6)

- Scheduler Implementation

```c
// tx_schedule
create_random_schedule(slots, adus) {
  tx_schedule Random;

  // randomly reorder
  random_shuffle(slots);
  random_shuffle(adus);

  // produce schedule:
  for (int i = 0; i < adus.size(); i++)
    random.assign_slot(slots[i], adus[i]);

  return random;
}

tx_schedule random_walk_scheduler() {
  // collect all slots w/ look-ahead
  slots = collect_slots(LOOKAHEAD_MSECS);
  // collect as many adus as slots
  adus = collect_adus(slots.size());

  // do a random walk
  int max_util = MIN_INTEGER;
  tx_schedule best_schedule = NULL;
  for (int i = 0; i < WALK_LENGTH; i++) {
    // get a random schedule
    tx_schedule sched =
      create_random_schedule(slots, adus);

    // evaluate all objectives over
    // this schedule and get utility
    int util = evaluate_objectives(sched);

    // is this best schedule?
    if (util > max_util) {
      max_util = util;
      best_schedule = sched;
    }
  }
  return best_schedule;
}
```
Policy and Scheduling (6/6)

- Scheduling with Phantom Slots
  - Phantom Slot: predicted by a channel manager
  - Look-ahead logic
  - Boosts the accuracy of our policy driven scheduler
Experimental Evaluation (1/5)

Setup

- A laptop connect
  - A CDMA 2000 1xRTT (PCMCIA-based modem)
  - Two GSM/GPRS interfaces
    (blue-tooth link to cell-phones)
- Sending as many 768 byte packets as it allow
- From stationary laptop to a MIT ethernet host
- AIMD congestion control
Figure 10: Throughput provided by Horde in a stationary experiment, using three colocated interfaces.
Figure 11: Observed WWAN packet RTT distributions.
Experimental Evaluation (4/5)

Figure 12: Measured round-trip-time distributions with different schedulers, striping over three existing WWANs (one CDMA and two GPRS). The graphs show the median and the upper and lower quartile packet latencies for the streams.
Figure 14: Impact of look-ahead on accuracy.
More work is needed on the specification language and schedulers.