Revisiting Resource Pooling
The Case for In-Network Resource Sharing

Ioannis Psaras, Lorenzo Saino, George Pavlou
University College London

[i.psaras, l.saino, g.pavlou]@ucl.ac.uk

Personal Webpage: http://www.ee.ucl.ac.uk/~uceeips/
EPSRC COMIT Project Webpage: http://www.ee.ucl.ac.uk/comit-project/
The Resource Pooling Principle

“Pooling of customer demands, along with pooling of the resources used to fill those demands”

“networked resources behave as a pooled resource”

• Pooling can be thought of as a tool to manage uncertainty.

• Internet (among others): a network of resources
  – From bandwidth, computation and storage resources, to information/content and service resources

• Uncertainty in the Internet (among others):
  1. Senders overloading the network with traffic
  2. Excessive demand for bandwidth over some particular link/area

Target: Maintain stability and guarantee fairness
Current State of Affairs
The Long Long Discussion on TCP

- TCP deals with uncertainty using the "one-out one-in" principle.
- TCP effectively deals with uncertainty by (proactively) suppressing demand!
- TCP is moving traffic as fast as the path’s slowest link.
- End-points have to speculate on the resources available along the e2e path.

Source has to estimate resource availability x hops down the path.

(i) e2e Resource Management.
Vision

1. Push traffic *as far in the path and as fast* as possible
2. Once in front of the bottleneck, *store traffic temporarily* in custodian nodes/routers and deal with congestion locally
3. Exploit all available (sub-)paths making decisions on a *hop-by-hop manner.*
Eliminating Uncertainty
Information-Centric Networking (ICN)

• We assume a generic ICN environment, where:
  – Packets (or chunks) are explicitly named
  – Clients send network-layer requests for named-/addressable-
    packets (or chunks) – similar to HTTP-GET, but for every packet
• Effectively..
  – clients (instead of senders) regulate the traffic that is pushed in the
    network, and
  – instead of the “data-ACK” model of TCP, in ICN we have a
    “request-data”.
• Result:
  – Based on requests, each router knows what to expect in terms of
    traffic.
Eliminating Uncertainty
In-Network Caching

• Caching has been used for *resource optimisation*
  – Reduce delay, save on bandwidth etc.

• Overlay Caching:
  – Put caches in “strategic” places and redirect (HTTP) requests to those caches

• In-Network Caching:
  – Individually named packets/chunks allow for in-network storage
  – Put caches in every router and serve network-layer requests for named chunks on the path

• We use in-network caching for *temporary storage*
Stability & Fairness

- **Global Stability**
- **Local Fairness**

- **e2e Flow Control**
  - Global Stability
  - Local Fairness

- **Local Stability**
  - **Global Fairness**

Global Stability
Local Fairness

Local Stability
Global Fairness
3-Phase Operation

• **Push-data phase** – Open-Loop System
  – Processor-sharing, RCP-like transmission
  – Open loop system – senders send even more than what they have received requests for
    • Push data as far and as quickly as possible

• **Cache & Detour phase**
  – Every router monitors incoming *Requests*
  – When demand is expected to exceed supply, the local router tries to find alternative paths to detour
  – In the meantime traffic in excess (if any) is cached locally

• **Backpressure phase** – Closed-Loop System
  – If alternative paths do not exist or are equally congested:
    • Pace Requests
    • Send notification upstream to slow down and enter closed-loop transmission
3-Phase Operation

- **Push-data phase** – Open-Loop System
  - Processor-sharing, RCP-like transmission
  - Open loop system – senders send even more than what they have received requests for
    - Push data as far and as quickly as possible
3-Phase Operation

- **Cache & Detour phase**
  - Every router monitors incoming *Requests*
  - When demand is expected to exceed supply, the local router tries to find alternative paths to detour
  - In the meantime traffic in excess (if any) is cached locally
3-Phase Operation

- **Backpressure phase** – Closed-Loop System
  - If alternative paths do not exist or are equally congested:
    - Pace Requests
    - Send notification upstream to slow down and enter closed-loop transmission
Data on detour availability

<table>
<thead>
<tr>
<th>ISP</th>
<th>1 hop</th>
<th>2 hops</th>
<th>3+ hops</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exodus (US)</td>
<td>49.77%</td>
<td>35.48%</td>
<td>6.68%</td>
<td>8.06%</td>
</tr>
<tr>
<td>VSNL (IN)</td>
<td>25.00%</td>
<td>33.33%</td>
<td>0.00%</td>
<td>41.67%</td>
</tr>
<tr>
<td>Level 3</td>
<td>92.22%</td>
<td>6.55%</td>
<td>0.68%</td>
<td>0.55%</td>
</tr>
<tr>
<td>Sprint (US)</td>
<td>56.66%</td>
<td>37.08%</td>
<td>1.81%</td>
<td>4.45%</td>
</tr>
<tr>
<td>AT&amp;T (US)</td>
<td>34.84%</td>
<td>61.69%</td>
<td>0.72%</td>
<td>2.74%</td>
</tr>
<tr>
<td>EBONE (EU)</td>
<td>50.66%</td>
<td>36.22%</td>
<td>6.30%</td>
<td>6.82%</td>
</tr>
<tr>
<td>Telstra (AUS)</td>
<td>70.05%</td>
<td>10.42%</td>
<td>1.06%</td>
<td>18.47%</td>
</tr>
<tr>
<td>Tiscali (EU)</td>
<td>24.50%</td>
<td>39.85%</td>
<td>10.15%</td>
<td>25.50%</td>
</tr>
<tr>
<td>Verio (US)</td>
<td>71.50%</td>
<td>17.09%</td>
<td>1.74%</td>
<td>9.68%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>52.80%</td>
<td>30.86%</td>
<td>3.24%</td>
<td>13.10%</td>
</tr>
</tbody>
</table>
Some (very initial) Results
Summary, Open Issues and Things We Don’t (Yet) Know

• Information-Centric Networks:
  – Lots of attention lately
  – Requires investment and effort
  – Worth doing, but need to get the full set of advantages

• There is an opportunity to deal with congestion control at the network layer

• Open Issues:
  – How do you know detour paths are not congested
  – How will this co-exist with traditional TCP flows?
  – Out of order delivery
  – Flows swapping between original and detour paths
Questions?

We are looking for a talented postdoc to join our team @ UCL!

Thanks!

Ioannis Psaras
i.psaras@ucl.ac.uk