

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

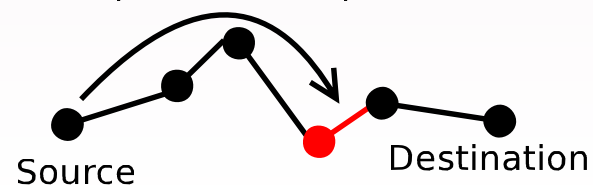
Target: Maintain **stability** and guarantee **fairness**

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

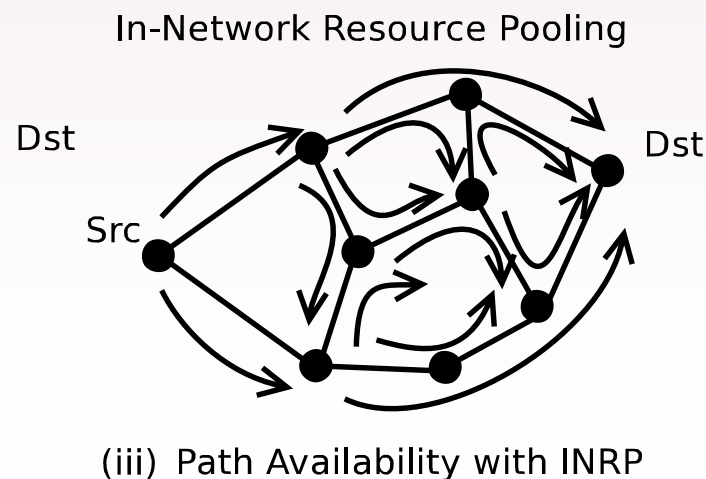
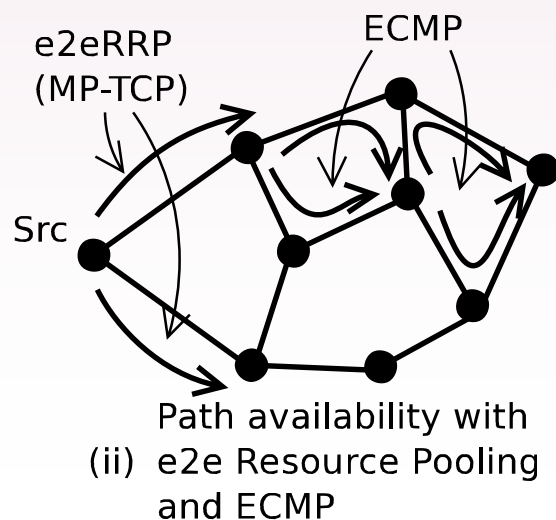
Source has to estimate
resource availability
x hops down the path



(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 sending traffic that is pushed in the network, and
 - instead of the “request-data” of TCP, in ICN we have a
- Result:
 - Based on requests, each router knows what to expect in terms of traffic.

Uncertainty #1 is minimised!

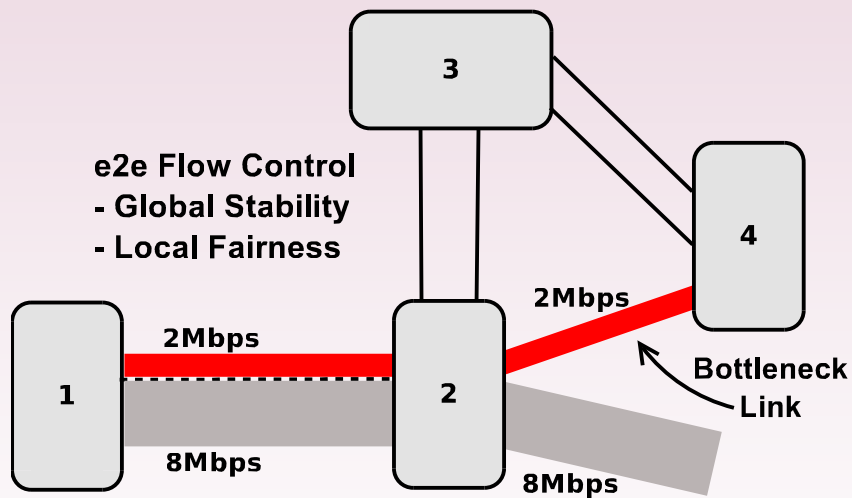
Eliminating Uncertainty

In-Network Caching

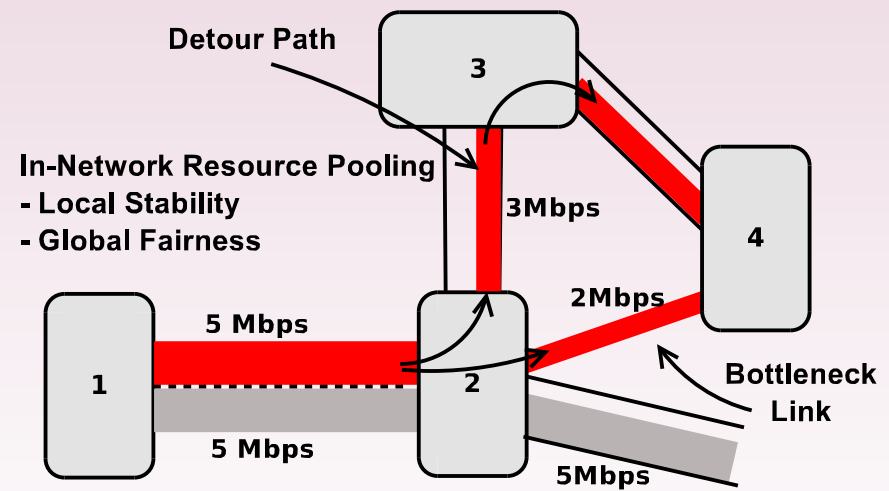
- Caching has been used for *resource optimisation*
 - Reduce delay, save on bandwidth etc.
- Overlay Caching:
 - Put caches in “strategic” places and redirect (HTTP) requests to those caches
- In-Network Caching:
 - Individually named objects/chunks allow for in-network storage
 - Put caches in strategic places to serve network-layer requests for named chunks at various points on the path
- We use in-network caching for **temporary storage**

Uncertainty #2
(temporarily)
accommodated

Stability & Fairness



Global Stability
Local Fairness



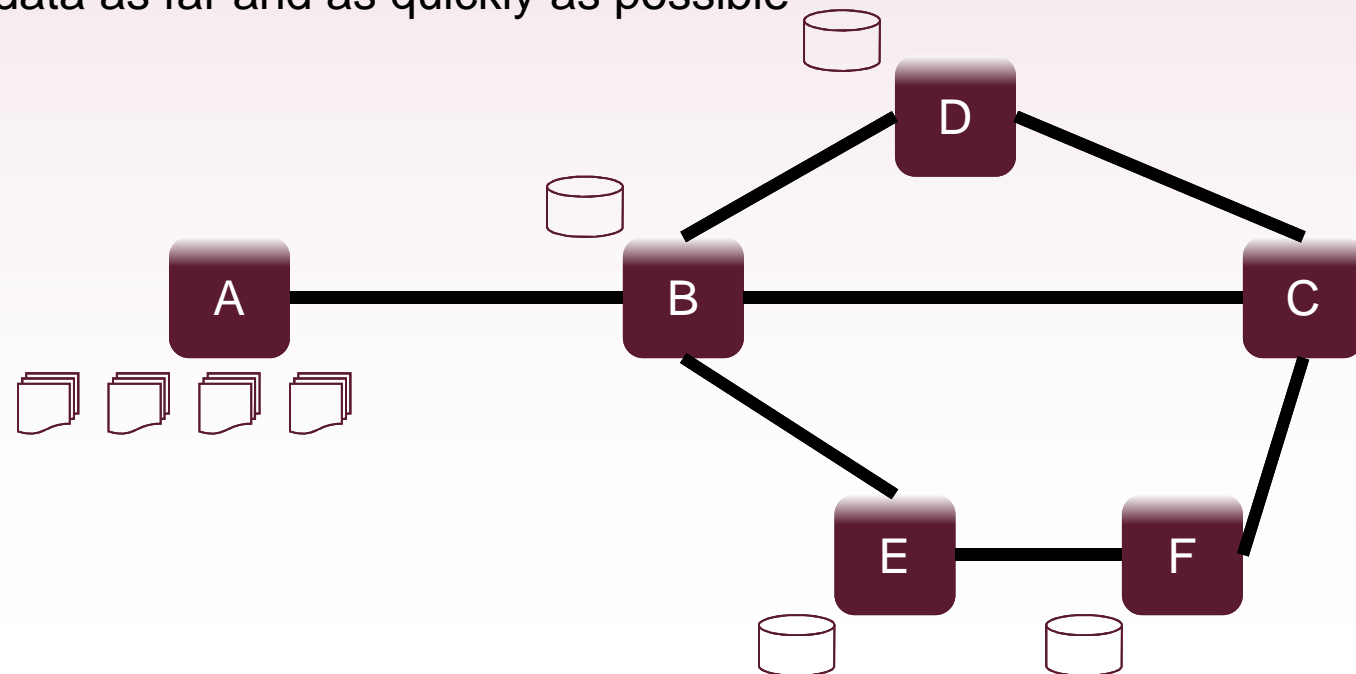
Local Stability
Global Fairness

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

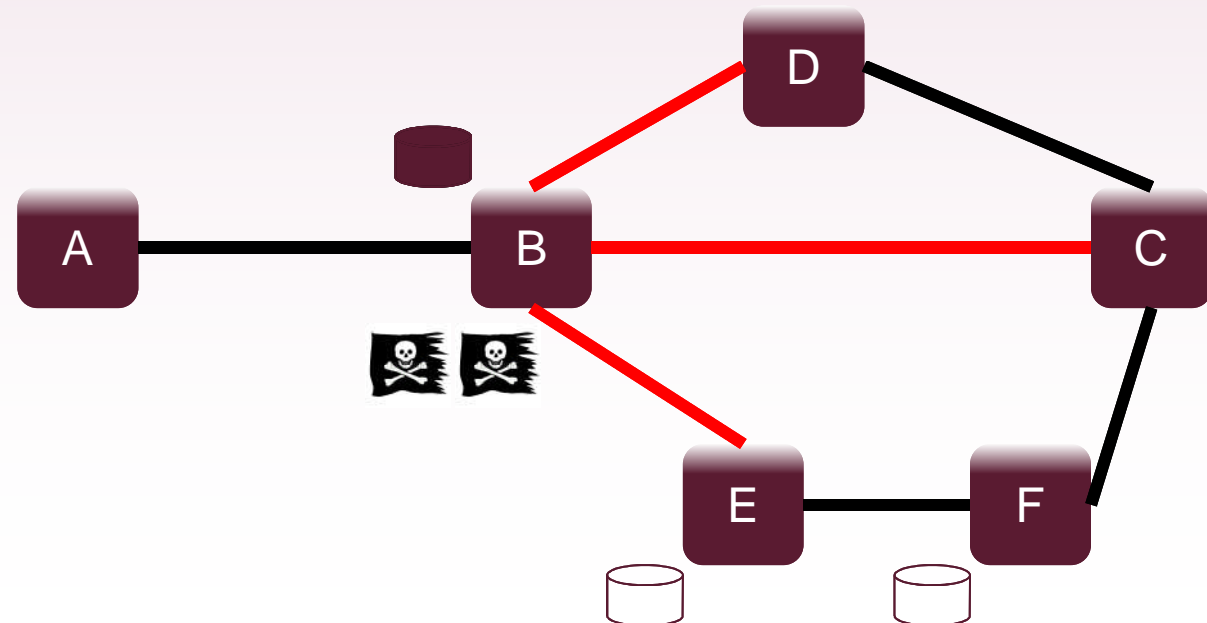
3-Phase Operation

- **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

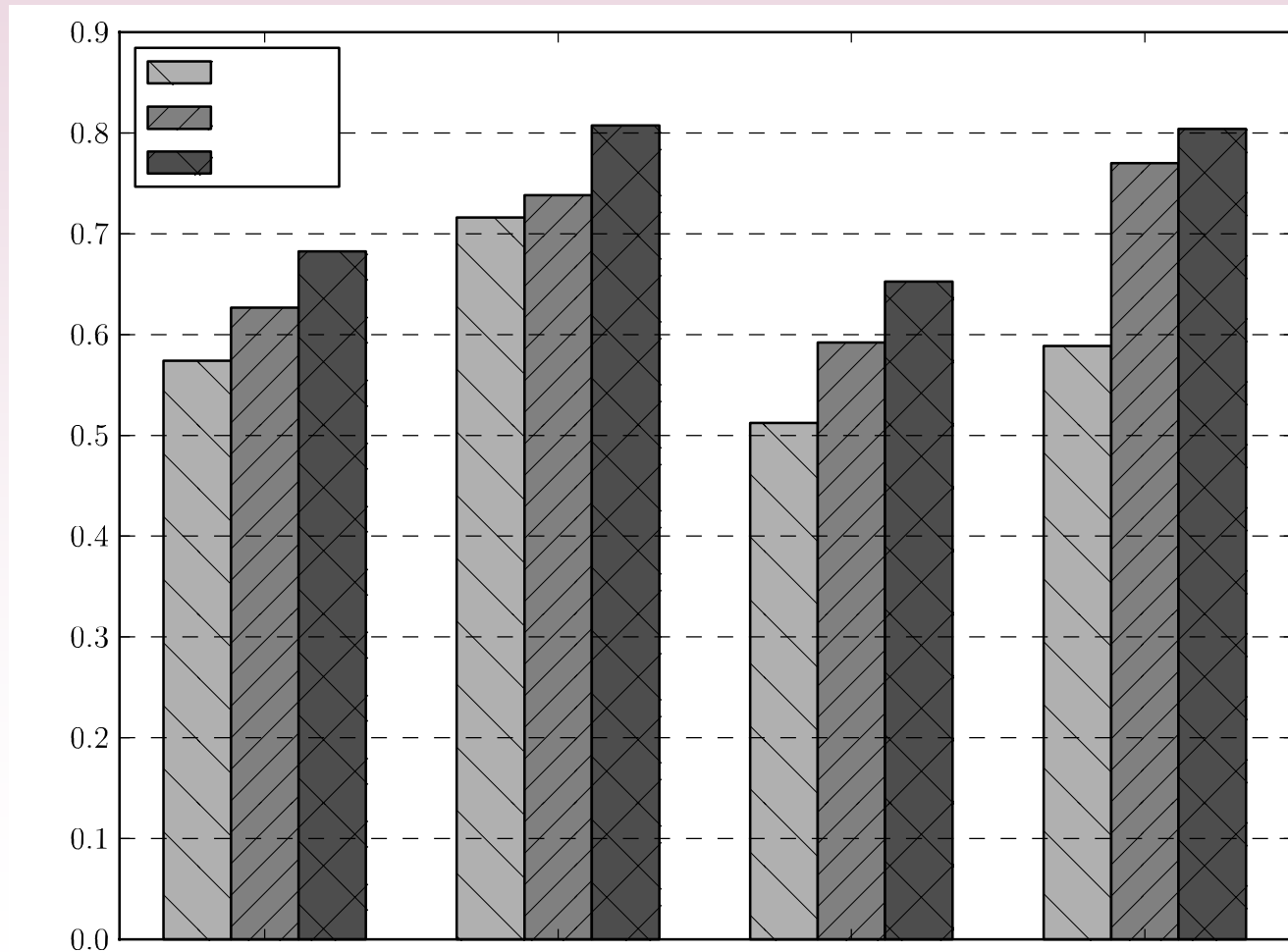
- **Backpressure phase** – Closed-Loop System
 - If alternative paths do not exist or are equally congested:
 - Pace Requests
 - Send notification upstream to slow down and enter closed-loop transmission



Data on detour availability

ISP	1 hop	2 hops	3+ hops	N/A
Exodus (US)	49.77%	35.48%	6.68%	8.06%
VSNL (IN)	25.00%	33.33%	0.00%	41.67%
Level 3	92.22%	6.55%	0.68%	0.55%
Sprint (US)	56.66%	37.08%	1.81%	4.45%
AT&T (US)	34.84%	61.69%	0.72%	2.74%
EBONE (EU)	50.66%	36.22%	6.30%	6.82%
Telstra (AUS)	70.05%	10.42%	1.06%	18.47%
Tiscali (EU)	24.50%	39.85%	10.15%	25.50%
Verio (US)	71.50%	17.09%	1.74%	9.68%
Average	52.80%	30.86%	3.24%	13.10%

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