Solving the Congestion Problem using ICN Principles

Ioannis Psaras
EPSRC Fellow
University College London
i.psaras@ucl.ac.uk
http://www.ee.ucl.ac.uk/~uceeips/

ICNRG Meeting, Prague, July 2015

In-Network Resource Pooling

In-net caching from a different angle

ACM HotNets 2014

I. Psaras, L. Saino, G. Pavlou

"Revisiting Resource Pooling:
The case for In-Network Resource Sharing"

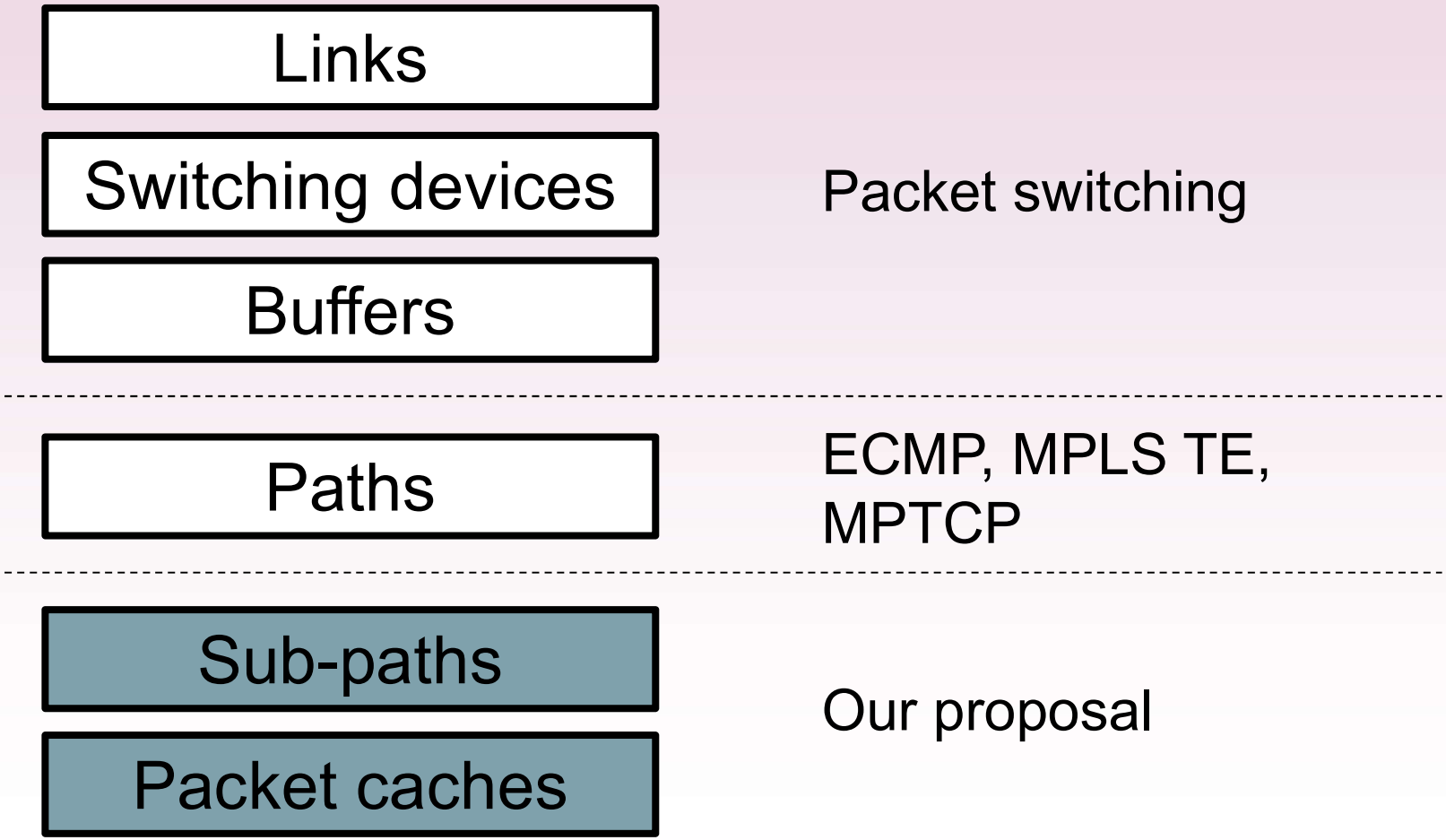
The Resource Pooling Principle

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

"networked resources behave as a pooled resource"

- Internet (among others): a network of resources
  - From bandwidth, computation and storage resources, to information/content and service resources
  - *Packet switching* enables pooling of link capacities and routers processing power
  - *Buffers* enable pooling of link capacity at adjacent time periods
  - *MPLS TE* and *ECMP* enable pooling of multiple paths

Pooled resources



The Resource Pooling Principle

We claim that:

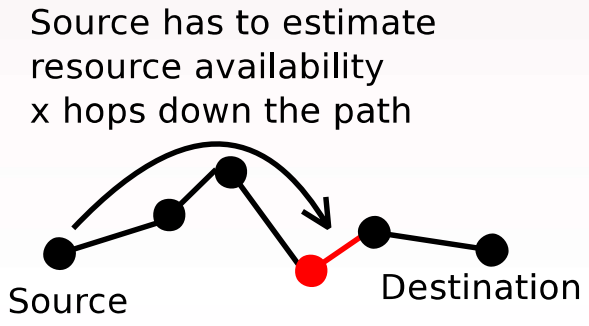
Pooling can be thought of as **a tool to manage uncertainty**.

- 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

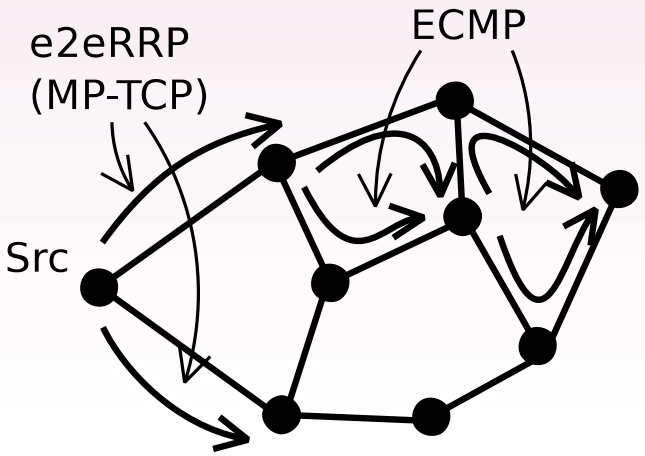
- TCP deals with uncertainty using the *"one-out one-in"* principle
- TCP effectively deals with uncertainty by (proactively) suppressing demand!
- TCP is moving traffic as fast as the path's slowest link
- End-points have to speculate on the resources available along the e2e path



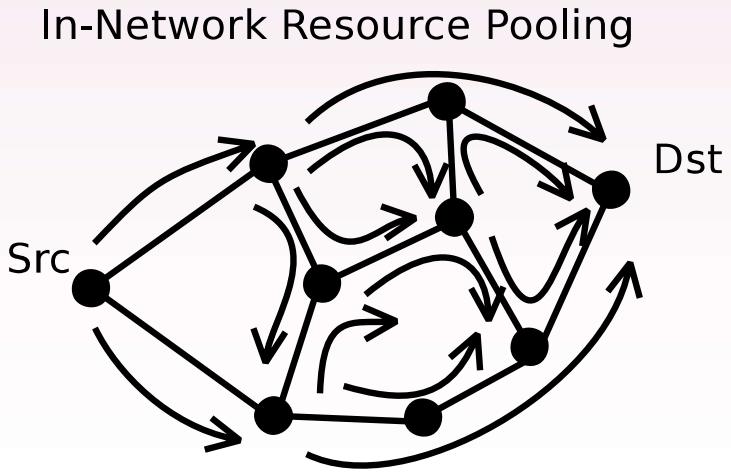
(i) e2e Resource Management

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*.



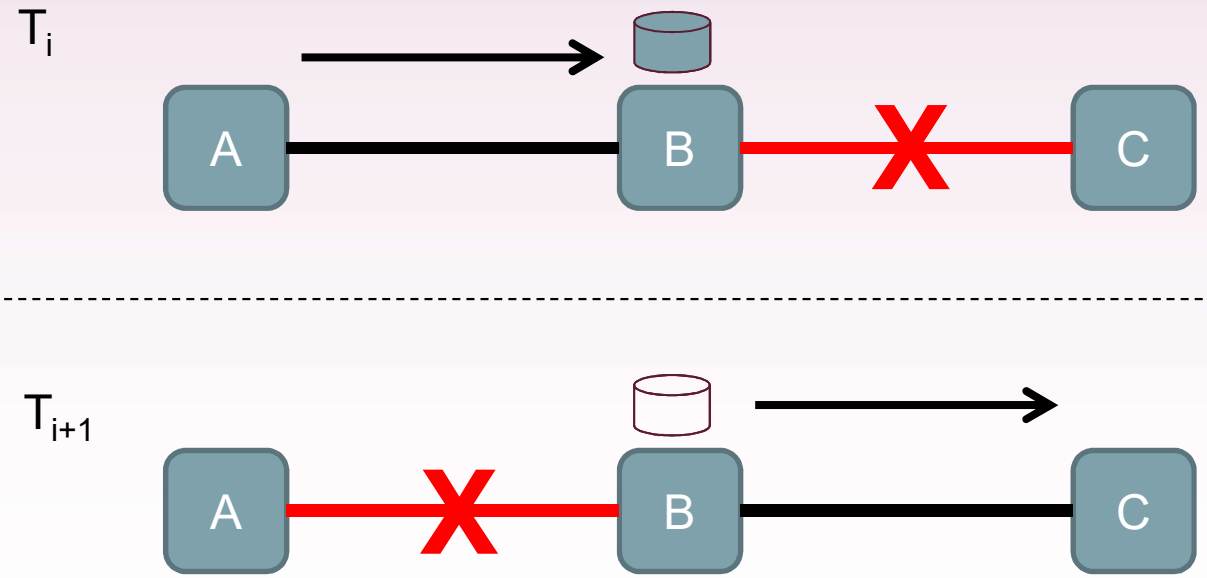
(ii) Path availability with e2e Resource Pooling and ECMP



(iii) Path Availability with INRP

Caches and resource pooling

- The presence of ubiquitous packet caches enables more efficient usage of resources by enabling pooling of sub-paths.



Eliminating Uncertainty
Information-Centric Networking

- Request and Data paths are symmetric
- Instead of the *"data-ACK"* model of TCP, in ICN we have a *"request-data"* model

Uncertainty #1 is minimised!

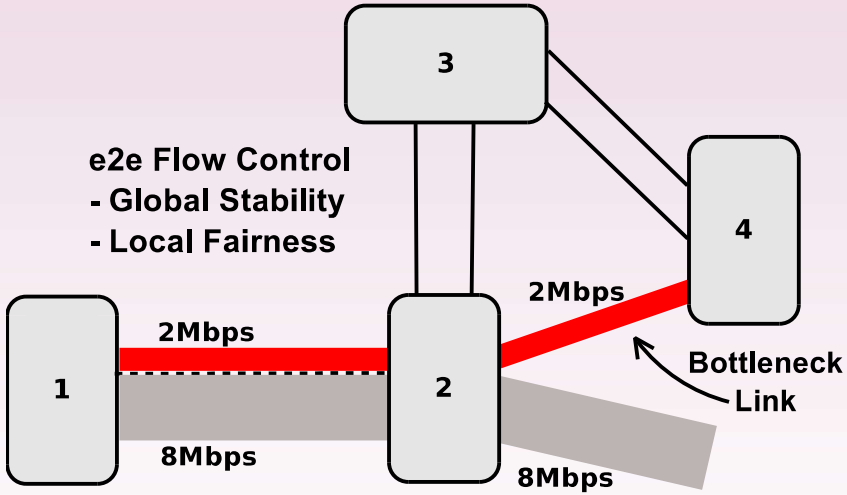
- Receivers (inst[illegible]nders) regulate the traffic that is pushed in the network
- Based on requests forwarded, each forwarding entity knows how much traffic to expect within one RTT.

Eliminating Uncertainty
In-Network Caching

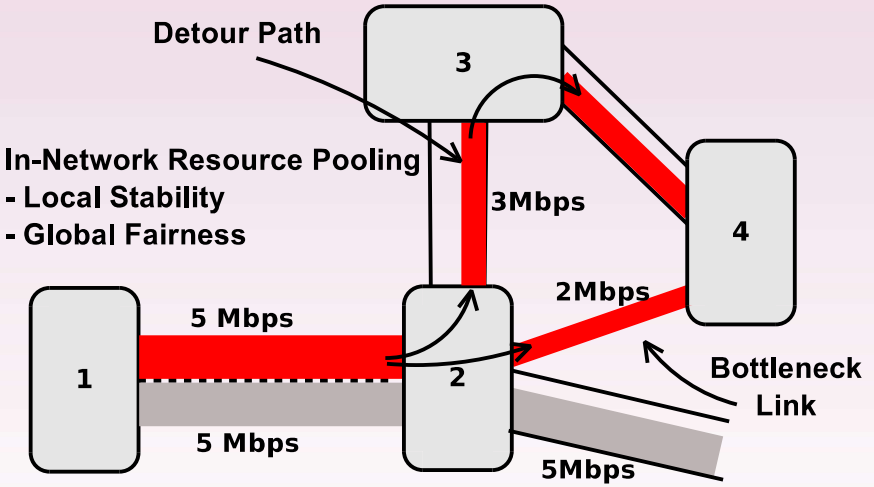
- Caching has been used for *resource optimisation*
  - Reduce delay, save on bandwidth etc.
- Overlay Caching:
  - Put caches in "strategic" places and redirect (HTTP) requests to those caches
- In-Network Caching:
  - Individually na… …ets/chunks allow for in-network sto…
  - Put caches i… …serve network-layer requests for named chunk… …hes on the path
- We use in-network caching for **temporary storage**

Uncertainty #2 (temporarily) accommodated

Stability & Fairness



Global Stability
Local Fairness



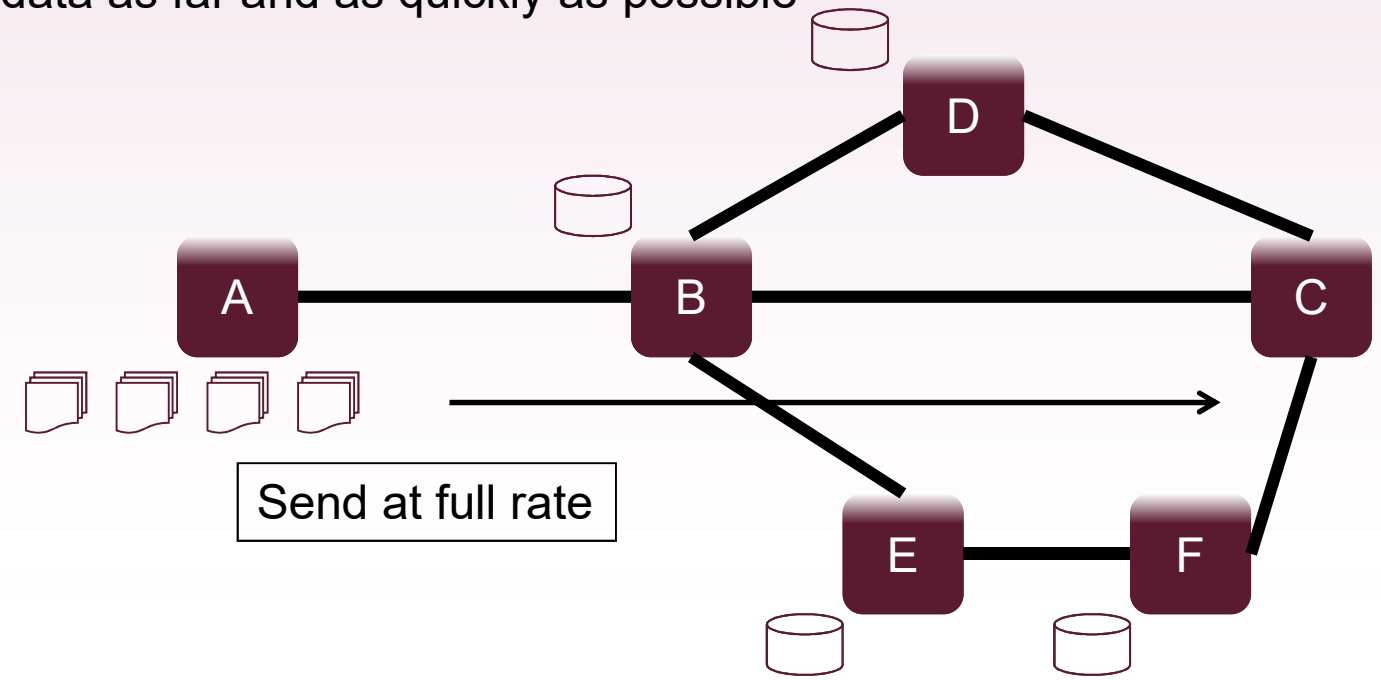
Local Stability
Global Fairness

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

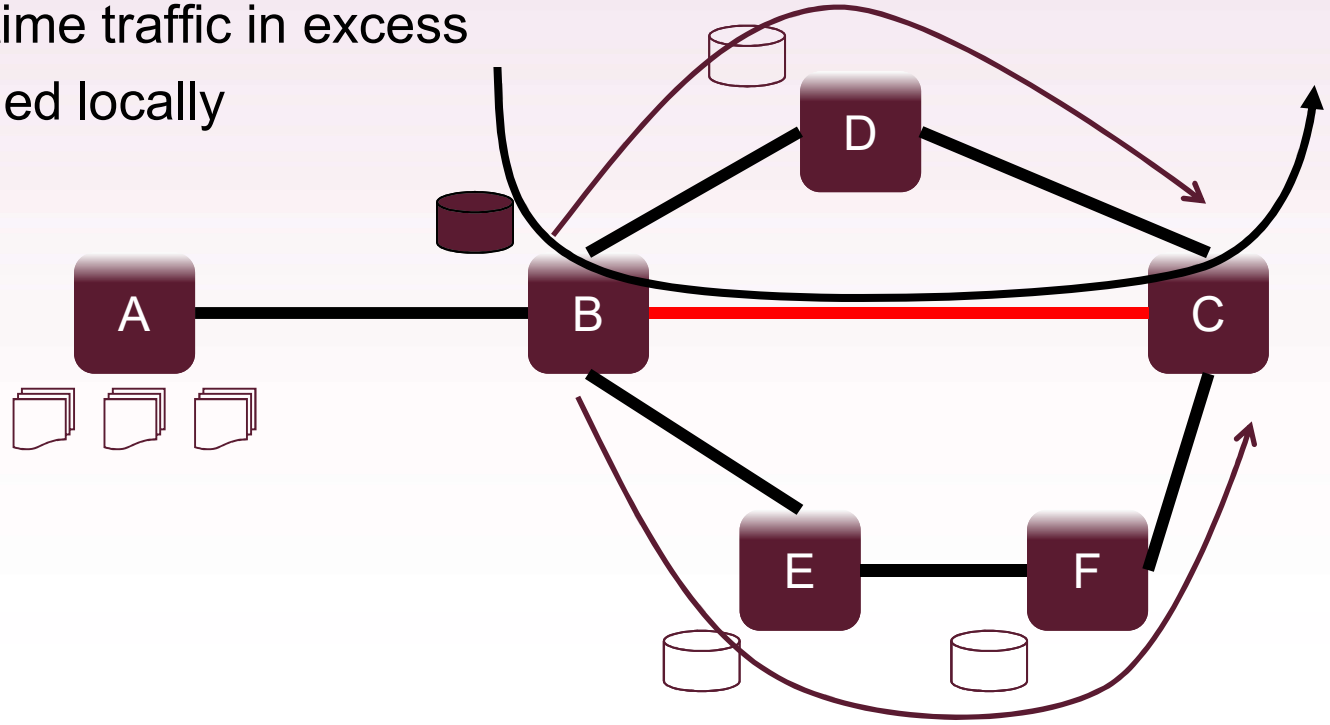
- **Push-data phase** – Open-Loop System
  - Processor-sharing, RCP-like transmission
  - Open loop system – senders send even more than what they have received requests for
    - Push data as far and as quickly as possible



3-Phase Operation

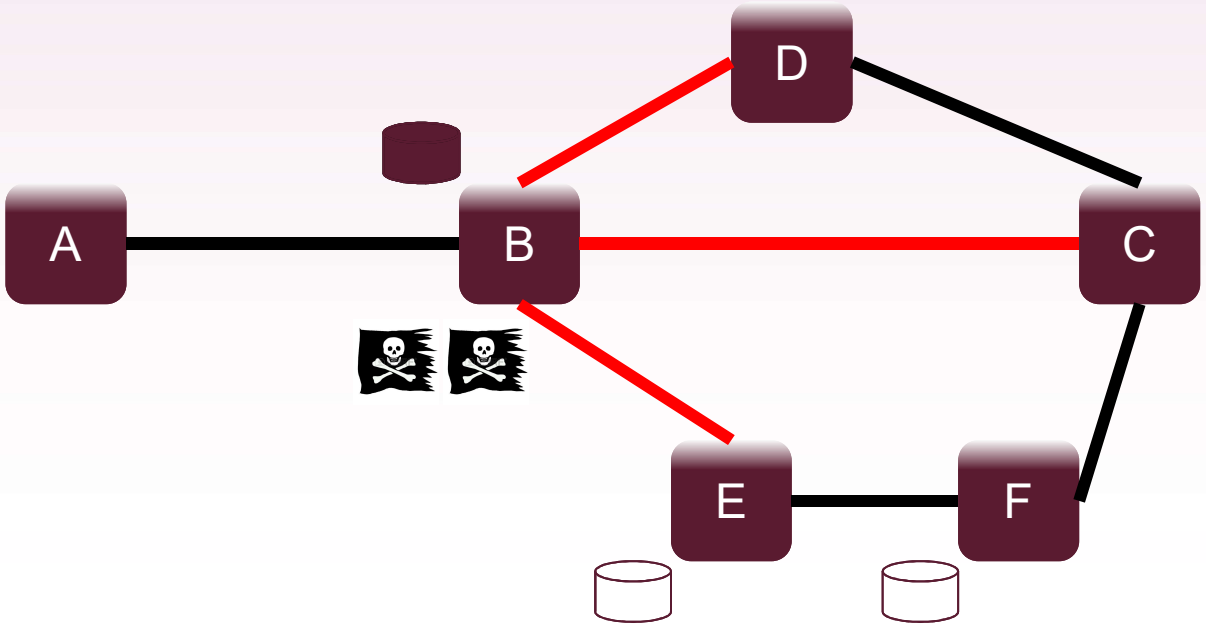
- **Cache & Detour phase**
  - Every router monitors incoming *Requests*
  - When demand is expected to exceed supply, the local router tries to find alternative paths to detour
  - In the meantime traffic in excess

  (if any) is cached locally



3-Phase Operation

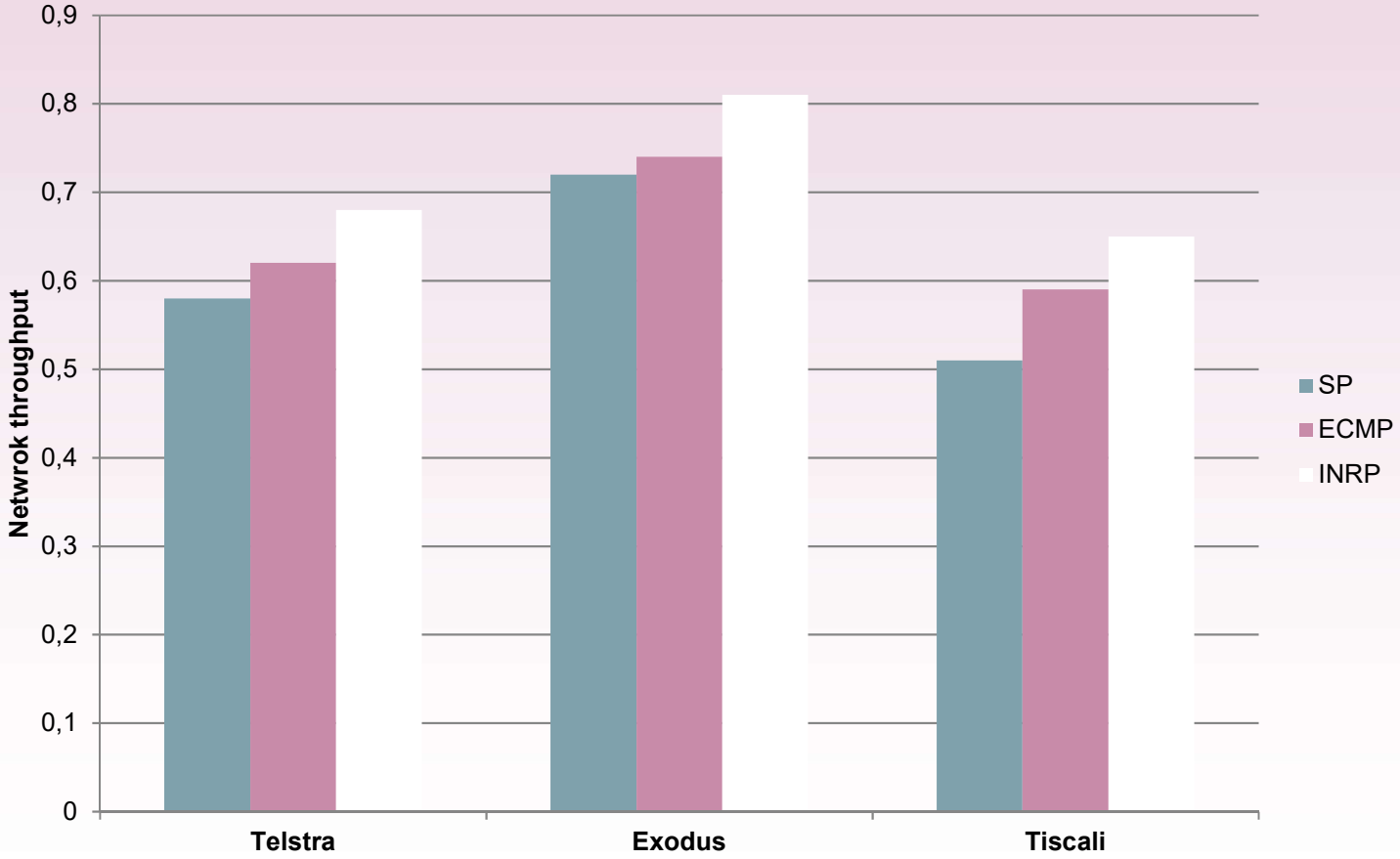
- **Backpressure phase** – Closed-Loop System
  - If alternative paths do not exist or are equally congested:
    - Pace Requests
    - Send notification upstream to slow down and enter closed-loop transmission



Data on detour availability

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:
  - 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?

Thanks!

Ioannis Psaras
i.psaras@ucl.ac.uk
http://www.ee.ucl.ac.uk/~uceeips/