

# Resource Pooling across the Internet

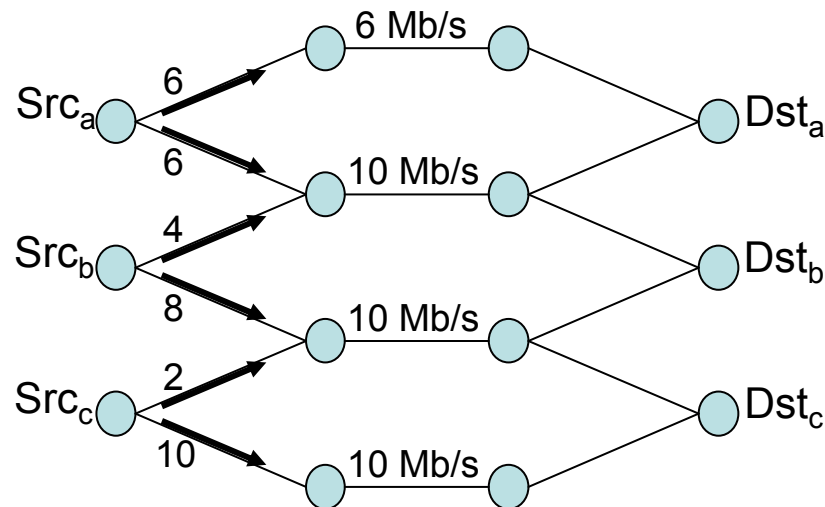
Mark Handley, UCL



# Resource Pooling

Make a network's resources behave like a single pooled resource.

- Aim is to increase reliability, flexibility and efficiency.
- Method is to build mechanisms for shifting load between the various parts of the network.



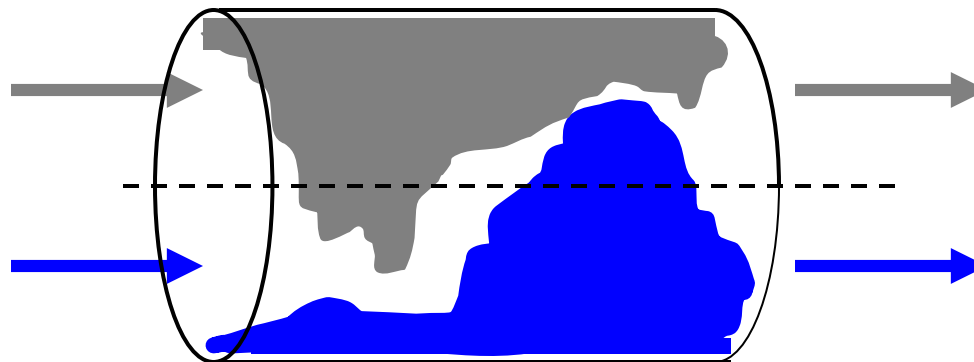
Everyone keeps reinventing resource pooling  
to solve their own local problems.



# Resource Pooling is not new...

Computer communication is bursty, so a virtual circuit-based model with rate allocations gives poor utilization.

- A packet-switched network pools the capacity of a single link.
  - Goal: high utilization
- Router queues pool capacity from one time interval to the next
  - Goal: high utilization, robustness to arrival patterns



# We're doing resource pooling in routing

- BGP traffic engineering:
  - Slow manual process to pool resources across peering links.
- OSPF/MPLS traffic engineering:
  - Slow mostly manual process to pool resources across internal ISP links.
- BT, AT&T (and others) dynamic alternative routing



# Recent resource pooling trends

- Multihoming
  - Primary goal: pool reliability.
  - Secondary goal: pool capacity
- Google, Akamai, content distribution networks
  - Pool reliability of servers, datacenters, ISPs.
  - Pool bandwidth.
- Bittorrent
  - Overall: Pool upstream capacity (over space and time)
  - Per-chunk: pool for reliability from unreliable servers.



Summary:

## Motivations for Resource Pooling

- Robustness
- Increased capacity or utilization



# Currently two main resource pooling mechanisms:

- Routing-based traffic engineering.
  - Either slow, or potentially unstable.
  - There are many examples where no network-based flow-based mechanism can achieve pooling.
- Application-based load-balancing between multiple servers.
  - Pretty effective, but strong tussle with what the network operators are doing.



# The requirements have changed

- Need a more robust Internet than we can get from simply making better components.
  - Traditional routing can't solve this in a scalable way.
- Applications are becoming more demanding:
  - VoIP, TV, Games.
- Most of the end-systems will be mobile, with multiple radios that can be used simultaneously.



# So what might work?

- Multihoming, via multiple addresses.
- Mobility, via adding and removing addresses, so upper layers can see and adapt to this.

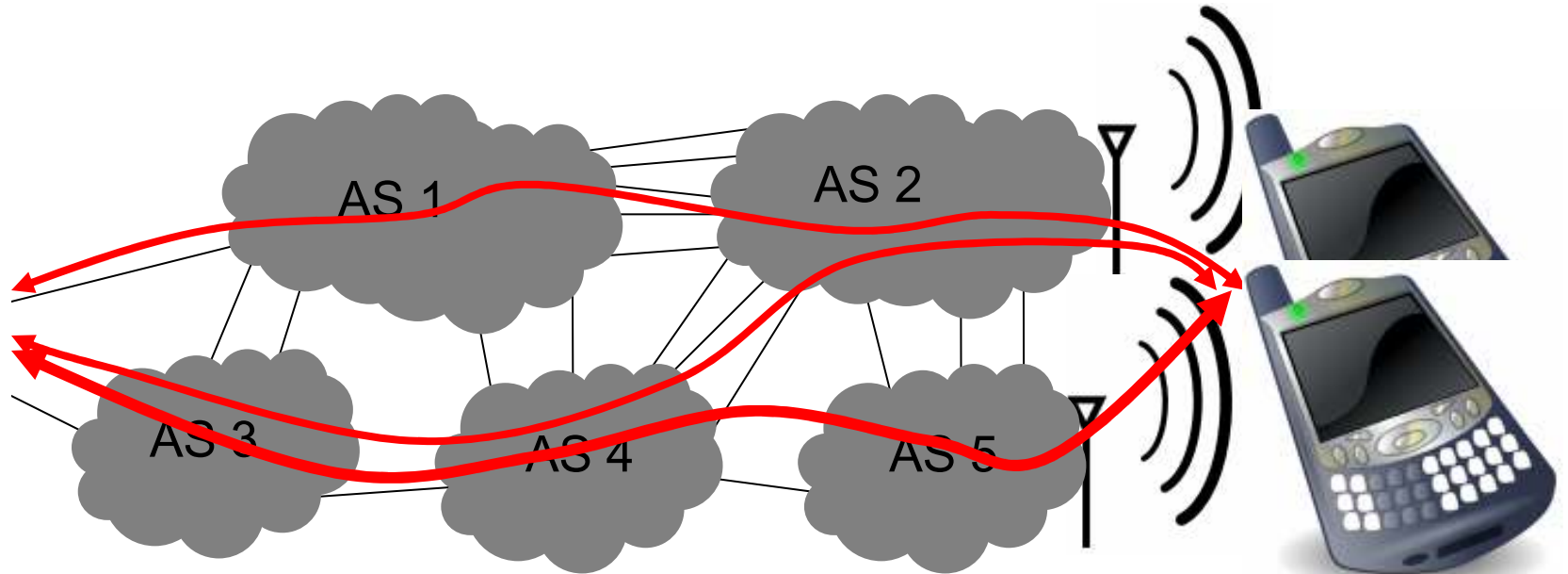
Then use these to do:

- Multipath.
  - Use multiple paths simultaneously for each transfer.
  - Only real way to get robustness is redundancy.



# So what might work?

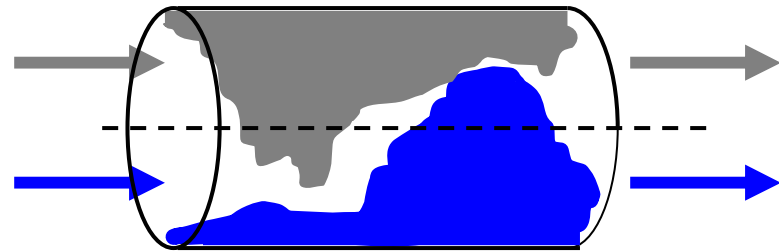
- Multipath-capable transport layers.
  - Use multiple subflows within each connection.
  - Congestion control the subflows, not the connection.



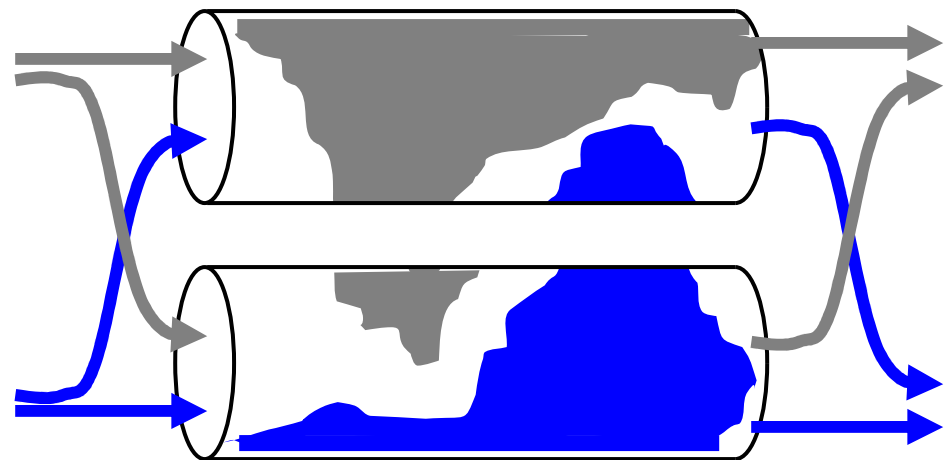
# Multipath transport

- Multipath transport allows multiple links to be treated as a single pooled resource.
- Traffic moves away from congested links.
- Larger bursts can be accommodated.

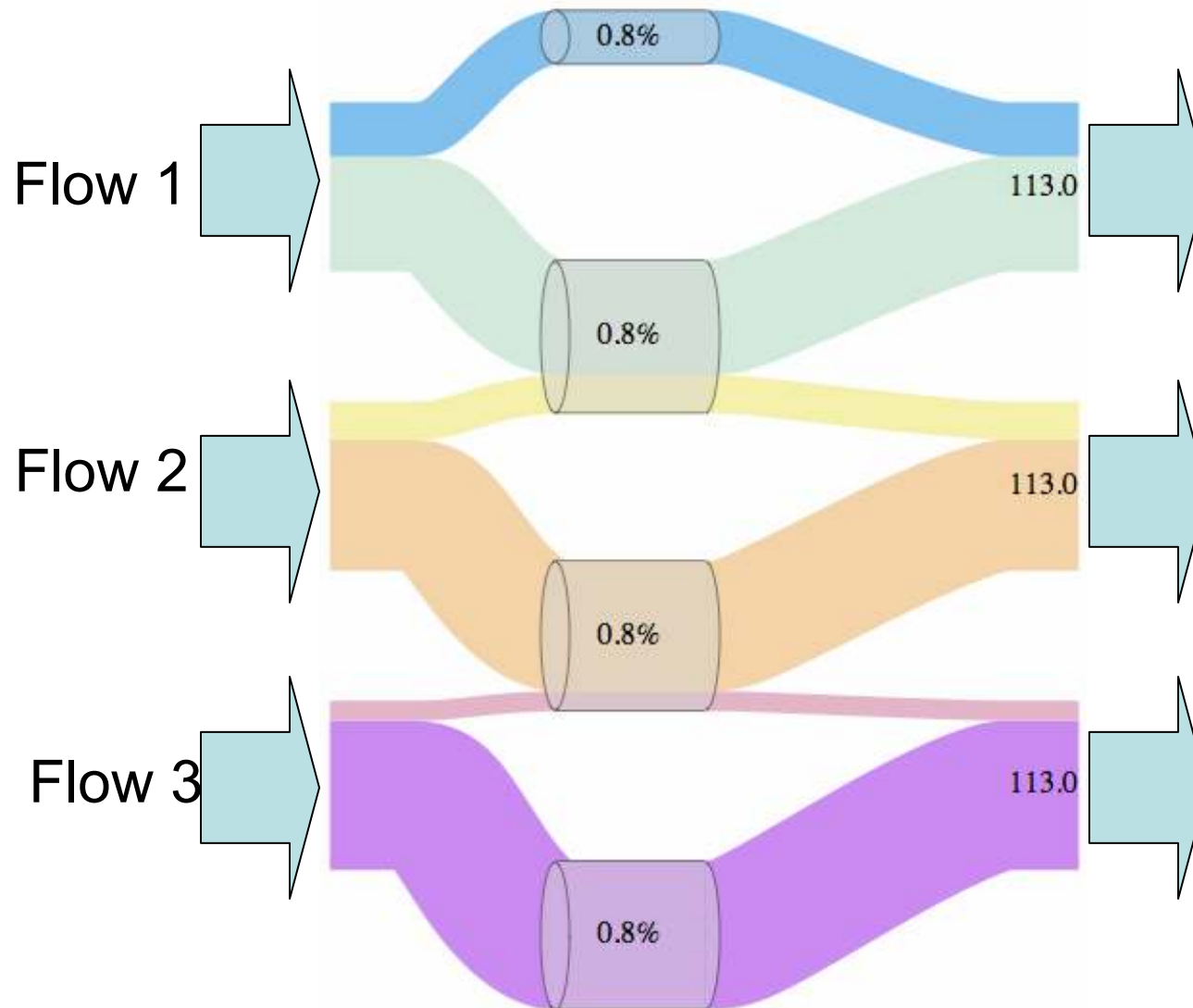
ARPAnet resource pooling:



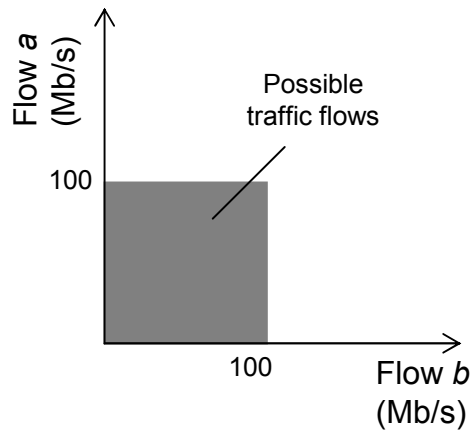
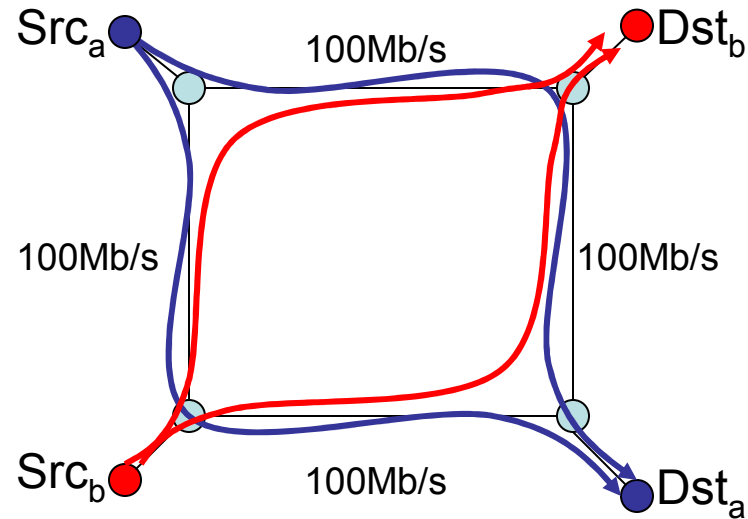
Multipath resource pooling:



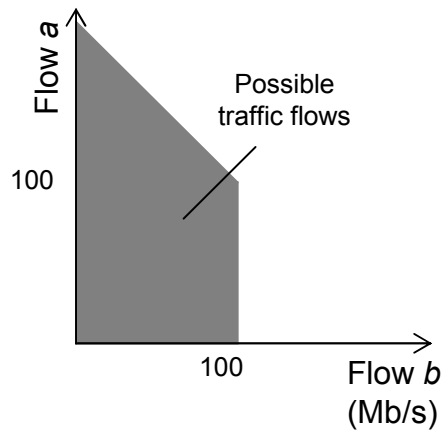
# Traffic moves away from congestion



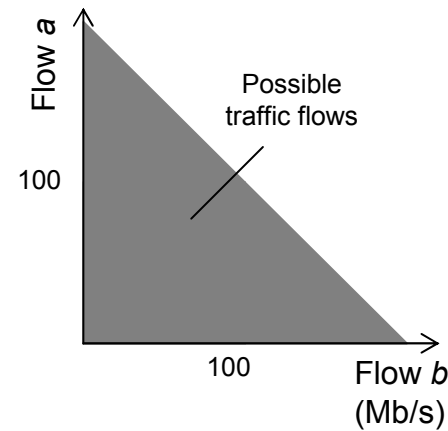
# Resource pooling allows a wider range of traffic matrices



No multi-path flows



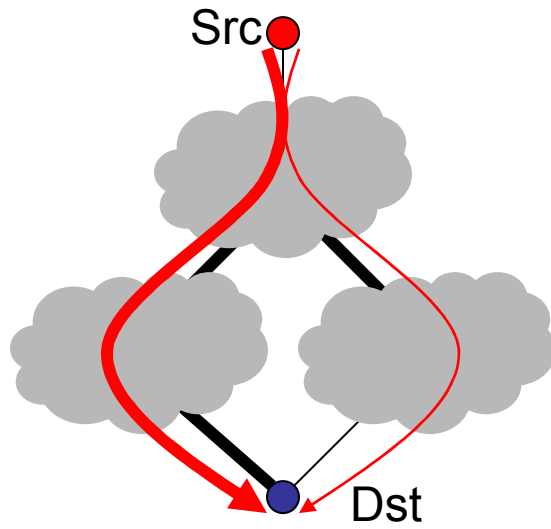
Only flow *a* is multi-path.



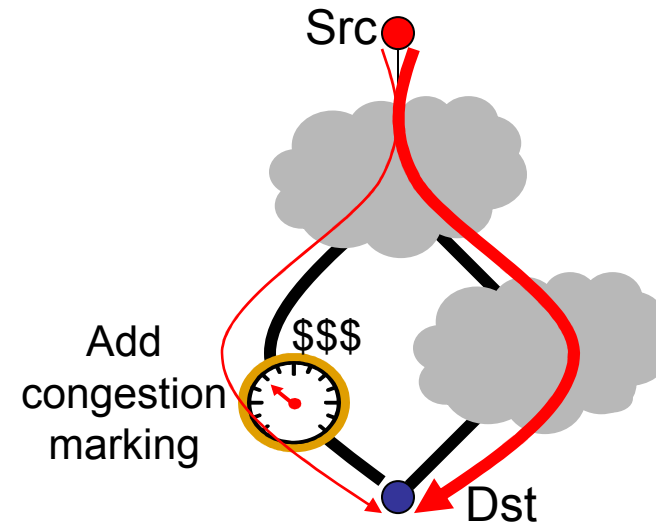
Both flows are multi-path



# Multipath Traffic Engineering

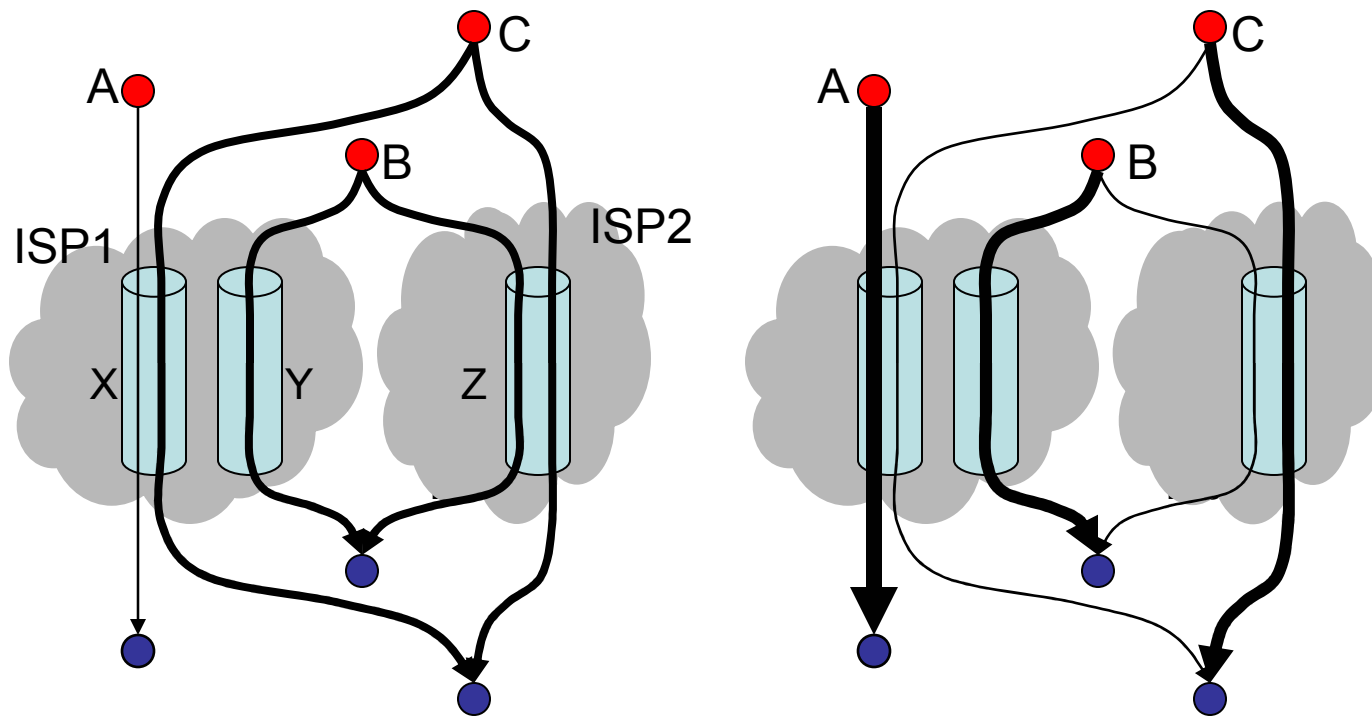


- Balancing across dissimilar speed links



- Balancing across dissimilar cost links

# End-systems can optimize globally (often ISPs cannot)



# Existing Multipath Transport

We already have it: BitTorrent.

Providing traffic engineering for free to ISPs who don't want that sort of traffic engineering :-)

If flows were accountable for congestion, BitTorrent would be optimizing for cost.

The problem for ISPs is that it reveals their pricing model is somewhat suboptimal.



# Robustness at an Affordable Price

- What if all flows looked a bit like BitTorrent?
  - Fetch from the best place right now.
- Can we build an extremely robust and cost effective network for billions of mobile hosts based on multipath transport and multi-server services?
  - Must build in controls to allow networks to tune traffic.



# Multipath Transport Design Space

## Multipath TCP

Add multi-path capability to the Transmission Control Protocol

## Multi-server HTTP

Allow browsers to fetch simultaneously from multiple mirror servers.

## P2P interactions with ISPs



# Impact

- Robustness
  - To link failures before routing can react
  - To ISP issues
  - To unexpected traffic patterns
- Seamless mobility (*really* use multiple radios)
- Multihoming
  - Link sharing (use my DSL and my neighbour's Cable (via Wifi) simultaneously).
- Reactive ends give the middle control to move traffic around.

