Resource Pooling

A system exhibits complete resource pooling if it behaves as if there was a single pooled resource.

The Internet has many mechanisms for resource pooling, but they clash and cause problems. Can we do better?

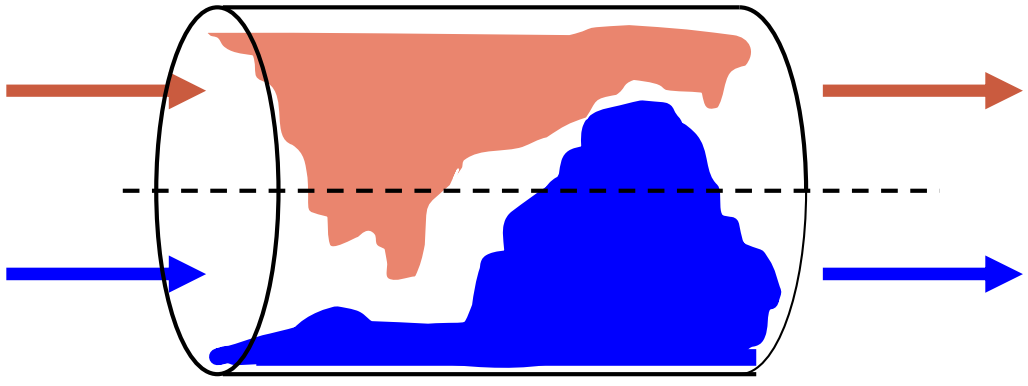
Damon Wischik, UCL

+

Mark Handley, UCL

Members of the Trilogy project

Packet switching is a type of resource pooling



- In the beginning there were circuit-switched networks
- A packet-switched link pools the capacity of several circuits
- A queue pools service capacity from one time period to the next

- Resource pooling allows better utilization when traffic is bursty
- In particular, it helps accommodate surges in traffic

We've got resource pooling in Internet routing

- Network operators do resource pooling across internal links
  - using OSPF/MPLS traffic engineering,
    or Valiant load balancing, or Dynamic Alternative Routing of telephone calls
  - to prevent overload / accommodate surges,
    to offer a service that is more reliable than any single component,
    to allow higher utilization
- Network operators do resource pooling across peering links
  - using BGP
  - to prevent overload / accommodate surges,
    to be reliable against failure of a neighbouring operator,
    to match revenues to costs

- But resource pooling via routing isn't great
  - it is too slow, and potentially unstable
  - traffic engineering with BGP leads to huge route tables

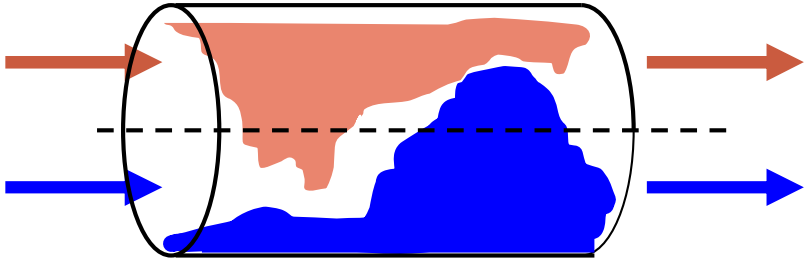
Now end-systems are also pooling resources

- Companies can multihome, i.e. connect to several ISPs
  - This pools reliability
  - It can also be used to pool capacity
- Content delivery networks like Google or Akamai balance traffic across datacenters
  - This pools the reliability of servers, datacenters, and ISPs
  - It also pools bandwidth
- BitTorrent
  - The main goal is to pool upload capacity
  - Capacity is pooled over space (many peers) and time (store and forward)
  - Chunking has the effect of pooling reliability across unreliable peers

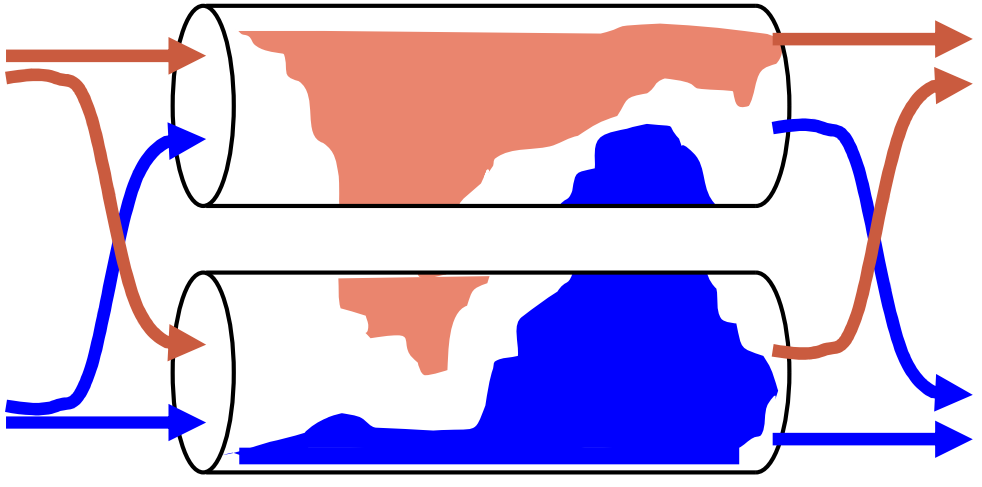
- These mechanisms clash with current network routing mechanisms
  - Mutihoming leads to huge BGP tables
  - BitTorrent causes trouble for traffic engineering
  - BitTorrent and content delivery networks reveal that ISP pricing is sub-optimal

The Trilogy project is building a multipath transport layer

- We should rely on multipath-capable end-system congestion control to shift traffic away from congested links
- This will give resilience, robustness to surges in traffic, and better utilization
- It will relieve stress on the routing system
- It can achieve better resource pooling than a single network operator can on its own
- End-systems do it anyway; we might as well harness it!

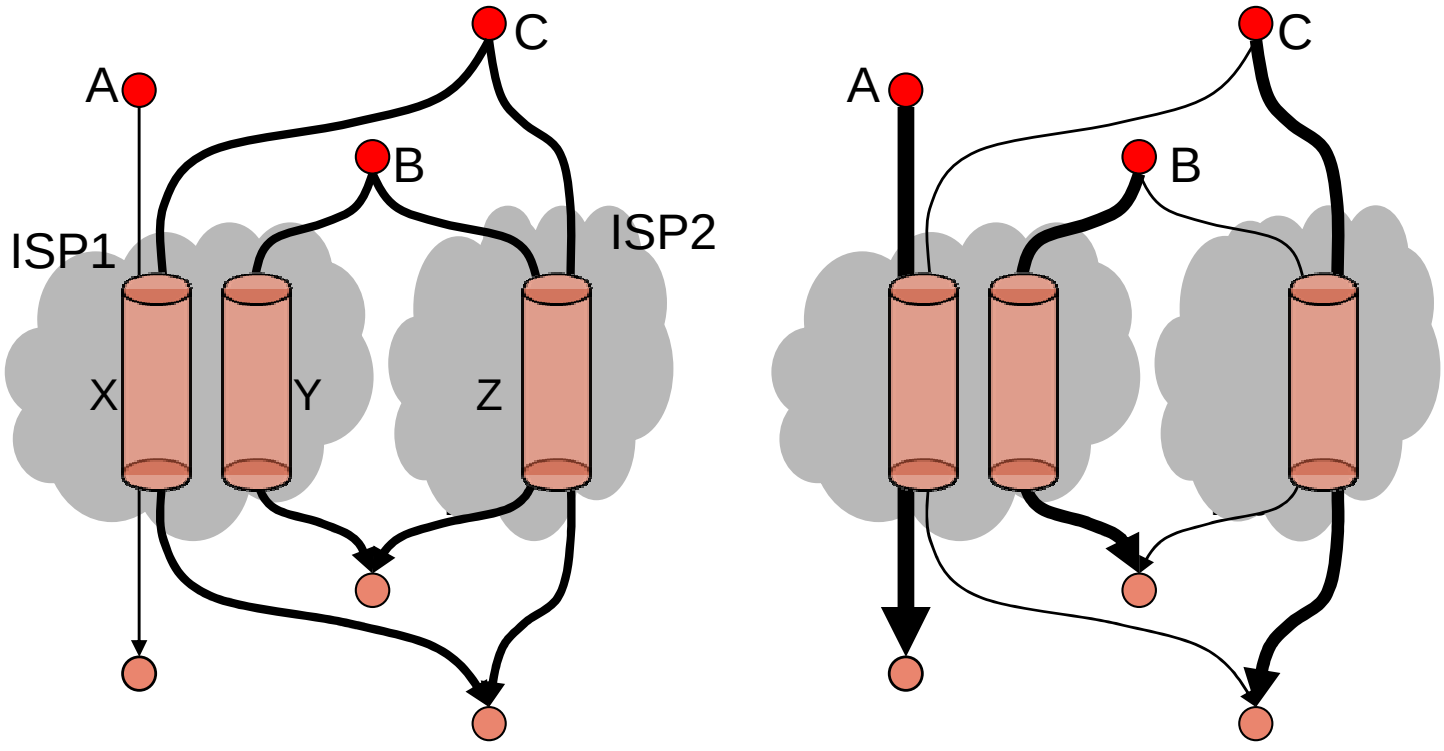


Packet-switched resource pooling



Multipath resource pooling

End-systems can optimize globally (often ISPs cannot)



If there is a surge of traffic on route A, then ISP1 on its own cannot shift traffic from X to Y

Trilogy questions

- [RESOURCE CONTROL]
  How should we design a multipath version of TCP?

- [ROUTING]
  How much routing support is needed from the network, to get enough path diversity?

- [ECONOMICS]
  What will be the impact on traffic management and charging?

How should we design a multipath congestion control?

A simple modification of TCP will make it seek the least congested paths.

A simple adaptation of TCP:

increase w_r by $\frac{1}{w}$ per ACK on path r

decrease w_r by $\frac{w}{2}$ per drop on path r

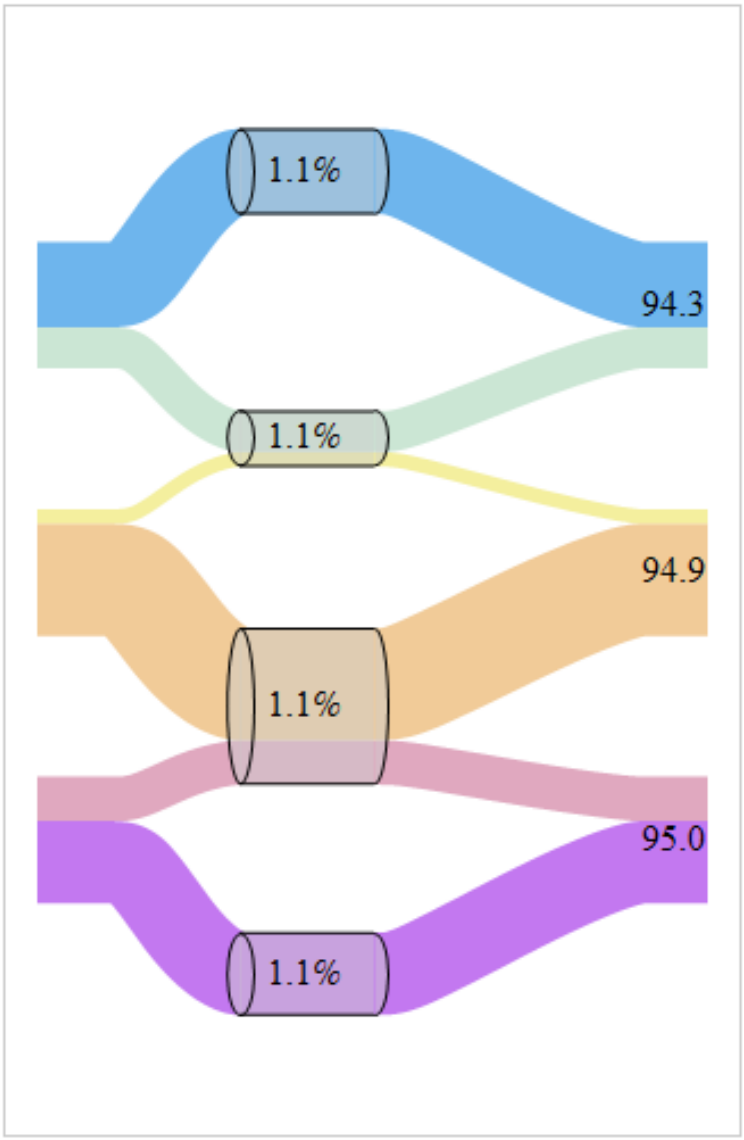
w_r = window on path r

$w = \sum_{r \in s} w_r$ = total window for flow s

Then $$\frac{dw_r}{dt} = \frac{w_r}{RTT_r}\left(\frac{1}{w} - p_r\frac{w}{2}\right)^{+[w_r=0]}$$

This uses only those paths with lowest drop probability p_{min}; in equilibrium $w = \sqrt{2}/p_{min}$

- [Stability of end-to-end algorithms for joint routing and rate control, Kelly and Voice, 2005] do this properly



coupled | TCP/2

ϵ=lability 0.0

C_1 63 pkt/s

C_2 41 pkt/s

C_3 116 pkt/s

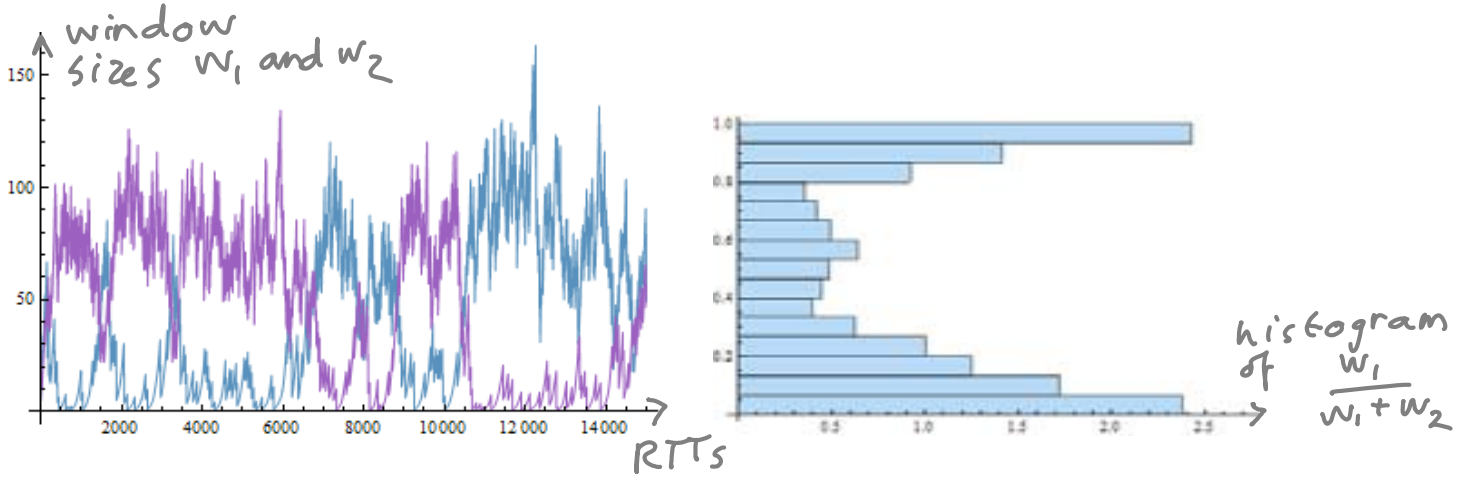
C_4 61 pkt/s

Multipath TCP is indifferent between flows with equal drop probability. And it flaps.

- Multipath TCP tends to equalize packet drop probabilities in the network
- When packet drop probabilities are equal, multipath TCP tends to flip from one path to another. The flips are random; if there are many flows then the aggregate will fluctuate smoothly.
- Is this a problem?
  Also, how best to deal with application-limited flows? timeouts? new paths?

$$w_r(t + RTT) = w_r(t) + \begin{cases} w_r/w & \text{with prob. } (1-p)^{w_r} \\ -\frac{w}{2} & \text{with prob. } 1-(1-p)^{w_r} \end{cases}$$

w_r = window for subflow r $\quad$ w = total window $\quad$ p = pkt drop probability



How should we design congestion control for BitTorrent?

Consider the rate allocation problem

$$\max \sum_d U_d(\log r_d) - \sum_s V_s \log t_s - \sum_j L_j(z_j)$$

z_j = load on link j

over $x_{sd} \geq 0$

x_{sd}: xmit rate from s to d

r_d = total receive rate at destination d

t_s = total xmit rate at source s

This suggests a flow control algorithm

$$\frac{dx_{sd}}{dt} = \kappa\left(U_d - V_s \frac{r_d}{t_s} - \frac{mark_{sd}}{x_{sd}\, r_d}\right)$$

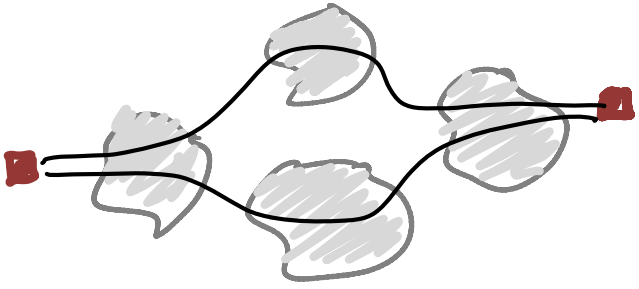
$\frac{r_d}{t_s}$: a tit for tat term

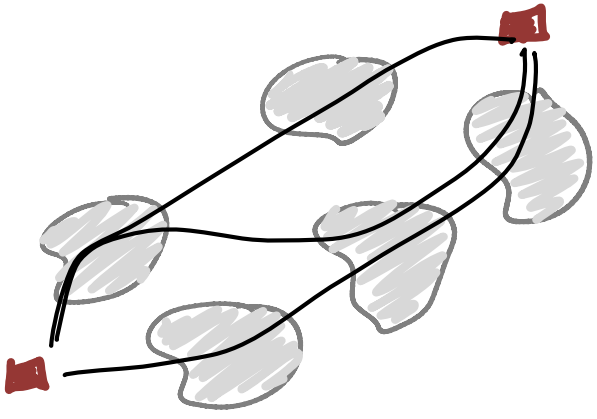
$\frac{mark_{sd}}{x_{sd}\, r_d}$: a network congestion term

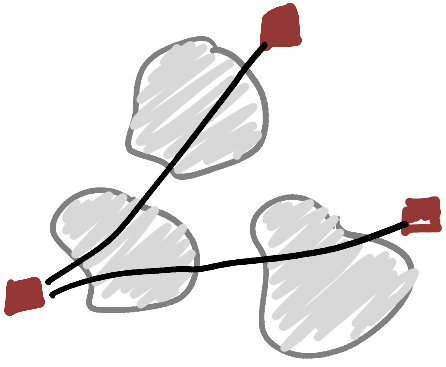
- With the right congestion signals, BitTorrent could be effective at Internet traffic management.

How much routing support is needed from the network, to get resource pooling?

What are some possibilities for multipath routing support?







- The end-system could set a path selector (e.g. 2 bits in the IP header), and multipath-aware routers could use this to choose one of several available routes
- End-points could be multi-homed, and expose multiple IP addresses; routing works as it does now
- P2P end-systems can choose where to download from

Resource pooling is not just a fancy name for load balancing.

- Load balancing means
  'the load on all resources is roughly equal'

- Complete resource pooling means
  'the system behaves as if there was one resource'

in what respect?

of what capacity?

Resource pooling makes predictions about behaviour / response.

I am interested in the behaviour: 'allocation of flow rates to long-lived flows'

- Kelly+Voice's Multipath TCP allocates transmission rates so as to solve an optimization problem

Allocate rates to subflows, so as to

$$\text{maximize} \quad \sum_{\text{all users } s} \left(\text{user } s\text{'s utility with the total rate he gets} \right) - \sum_{\text{all links } j} \left(\text{link penalty function for the load on link } j \right)$$

- I want to know: is this equivalent to solving a single-resource optimization problem?

$$\text{maximize} \quad \sum_{\text{all users } s} \left(\text{user } s\text{'s utility with the total rate he gets} \right) - \text{Penalty function} \left(\text{total load in the network} \right)$$

- If so, there is complete resource pooling.

I am interested in the behaviour: 'allocation of flow rates to long-lived flows'

- Kelly+Voice's Multipath TCP allocates transmission rates so as to solve an optimization problem

Allocate rate x_r to subflow r so as to maximize user-utility of total flow rates, minus network cost of congestion

$$\max_{y}\left[\max_{x}\left\{\sum_{s} U_s(y_s) - \sum_{j} L_j\left(\sum_{r:j\in r} x_r\right)\right\} \text{ where } y_s = \sum_{r\in s} x_r\right]$$

$U_s(y_s)$: utility of user s when he gets y_s

$\sum_{r:j\in r} x_r$: total load on link j

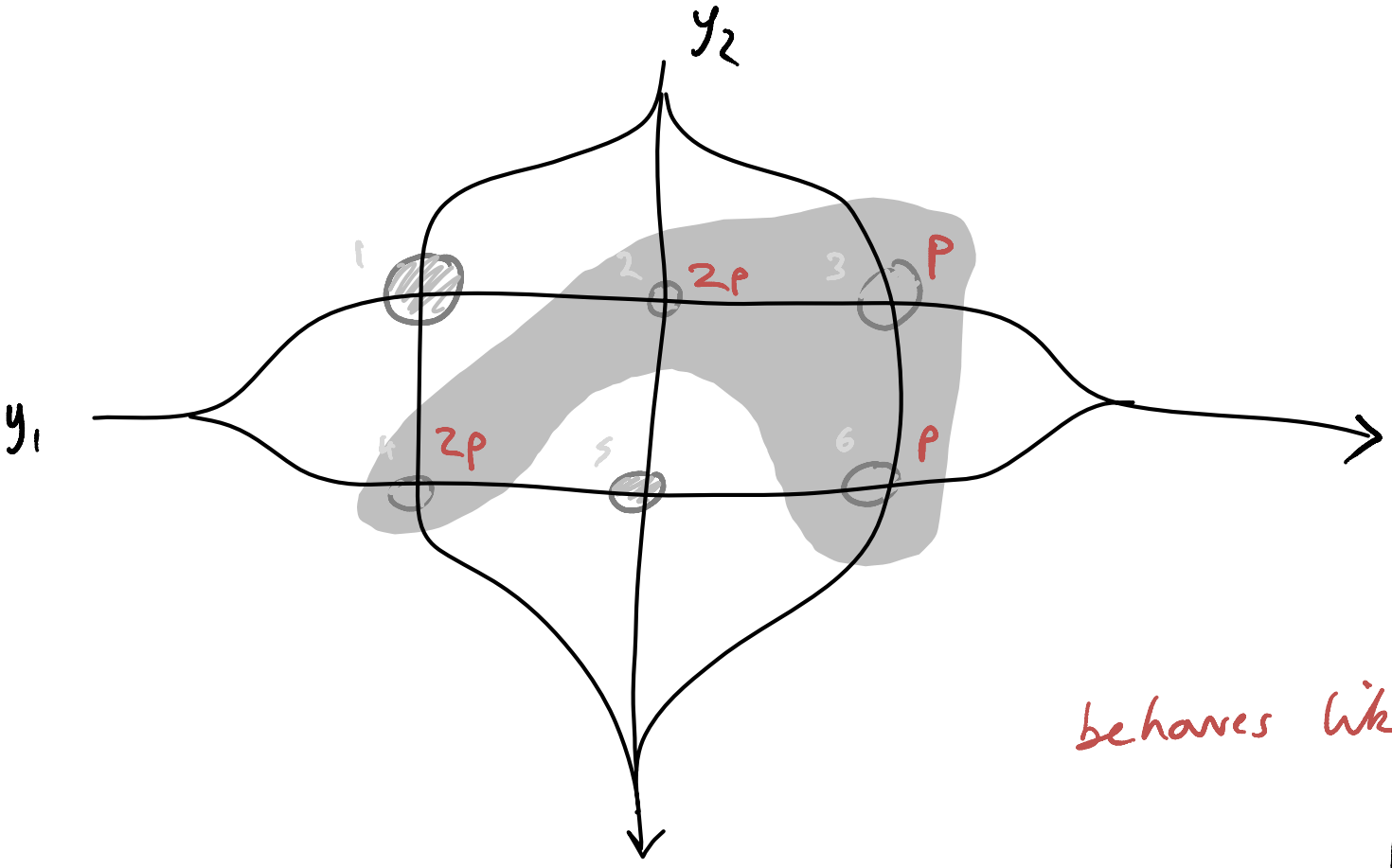
$y_s = \sum_{r\in s} x_r$: total flow rate given to user s

- I want to know: is this equivalent to solving a single-resource optimization problem?

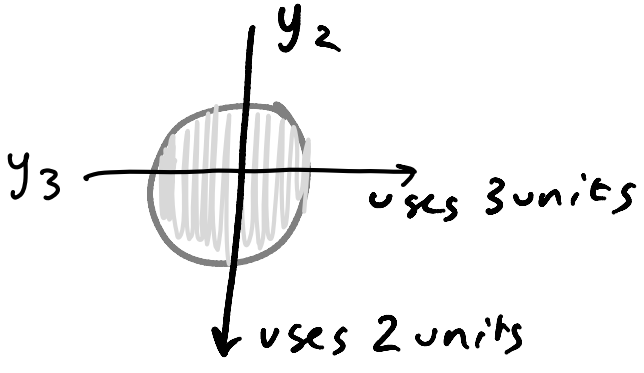
$$\max_{y} \sum_{\text{flows } s} U_s(y_s) - L\left(\sum_{s} w_s y_s\right)$$

- If so, there is complete resource pooling.

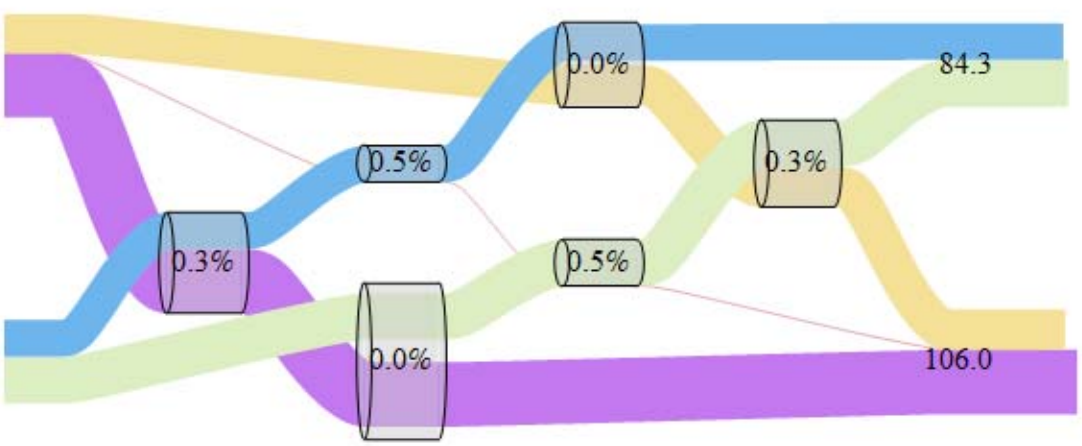
The network splits into one or more resource pools, depending on traffic and link capacities

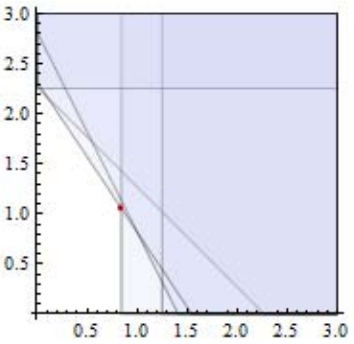


behaves like



[See Laws, *Resource pooling in queueing networks with dynamic routing*, 1992, Example 3.2. He finds generalized cut constraints of this form for a version of this problem with fixed capacity constraints and demands.]





coupled | TCP/2

ϵ=lability 0.0

C_1 102 pkt/s

C_2 37 pkt/s

C_3 86 pkt/s

C_4 158 pkt/s

C_5 47 pkt/s

C_6 88 pkt/s

RTTa 131 ms

RTTb 176 ms

How to find the resource pools

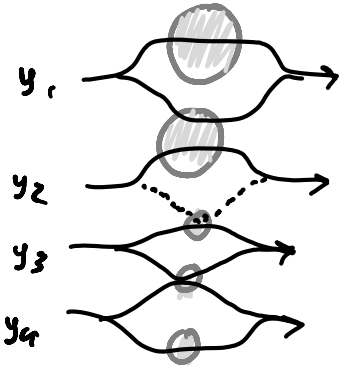
1. Create a multipath routing
2. Measure the total rate y_s for each flow s and the drop probabilities p_j at each link j
   - One can find these by running the fluid model of mTCP
3. Find subsets of resources where y and p satisfy a certain relationship; these are the resource pools

$$\sum_s y_s \cdot \min_{r \in s} \sum_{j \in r} \hat{p}_j - \sum_j L_j^*(\hat{p}_j) \quad \text{is maximal}$$

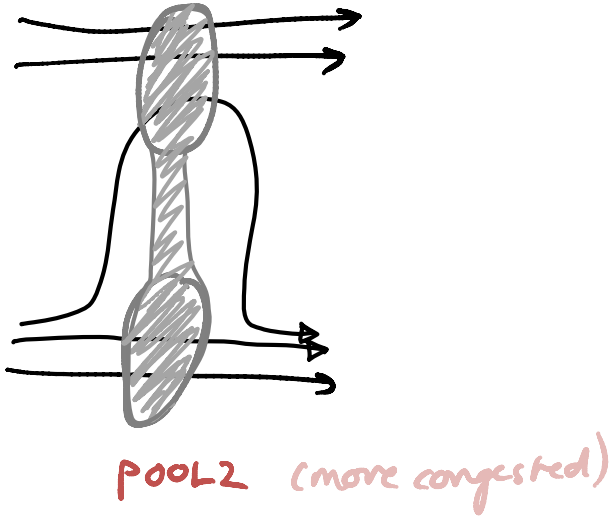
where $\hat{p}_j = \begin{cases} p_j & \text{if } j \text{ is in the subset} \\ 0 & \text{otherwise} \end{cases}$

$$L_j^*(p) = \sup_z \; pz - L_j(z)$$

If there are multiple resource pools, we need better path diversity



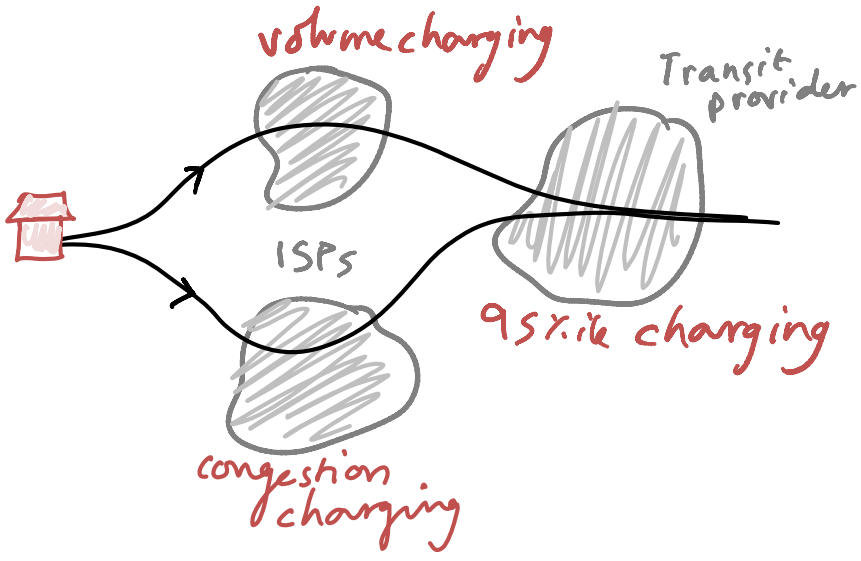
behaves like

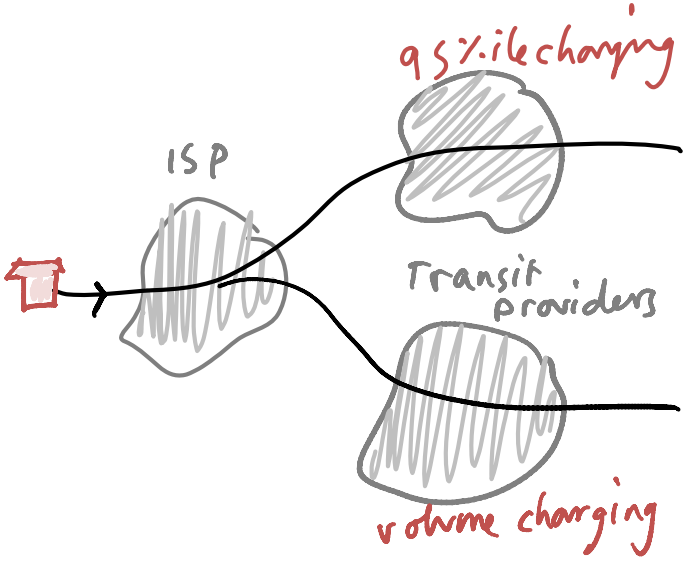


- Incomplete resource pooling is a sign that there isn't enough traffic with enough good path choices.
- Giving more choice can shift traffic onto the less congested pool
- If enough traffic can shift, the pools will merge
- Can we get good resource pooling with multihoming alone? Or with BitTorrent? If not, how much network support is needed?

What are the economic implications of multipath transport?

Multipath transport should improve transparency and accountability. It probably already does.





- How should ISPs signal congestion to multipath users, to manage their traffic?
- Would an ISP who uses congestion pricing gain a competitive advantage? Would a transit provider who uses congestion pricing gain a competitive advantage?
- Is it in the interests of an ISP to offer multipath routing support?
- Is BitTorrent already pushing us in the direction of congestion charging?

The Resource Pooling Principle

- **Observation** Resource pooling is often the only practical way to achieve resilience at acceptable cost. It is also a cost-effective way to achieve flexibility and high utilization.

- **Consequence** At every place in a network architecture where sufficient diversity of resources is available, designers will attempt to design their own resource pooling mechanisms.

- **Principle** A network architecture is effective overall, only if the resource pooling mechanisms used by its components are effective and do not conflict with each other.

- **Corollary** The most effective way to do resource pooling in the Internet is to harness the responsiveness of the end systems in the most generic way possible, as this maximizes the benefits while minimizing the conflicts.