Multipath TCP Opportunities and Pitfalls

2009-03-06, IETF-74, San Francisco

Iljitsch van Beijnum

Multipath

- Usually, multiple paths exist between any two hosts (even after applying policy)
- But: routing only lets a flow use one path
- Using multiple paths would increase performance for the user, utilization of the network and robustness

Better speed/utilization

Robustness

Opportunities

- Get more use out of existing network, or
- Support current traffic loads with cheaper network
- Use cases:
  - cable + ADSL at home or in small office, get full aggregate speed for single flow
  - server farm with multiple uplinks can avoid congested paths towards end-users

Multipath IP

- Issues with simple network layer multipath:
  - now limited by the slowest path
    - 10 + 100 = 20, not 110!
  - reordering triggers TCP fast retransmit
    - cwnd = cwnd / 2
- So MIP, SHIM6, SCTP don't give us usable multipath (in their current state)

Multipath TCP

- Take a single TCP flow, split into <u>subflows</u>
  - packets for a subflow follow same path
  - flow control per session
  - congestion control per subflow
- Per-subflow cwnd = max rate on each path
- Ignore inter-subflow reordering

Path selection

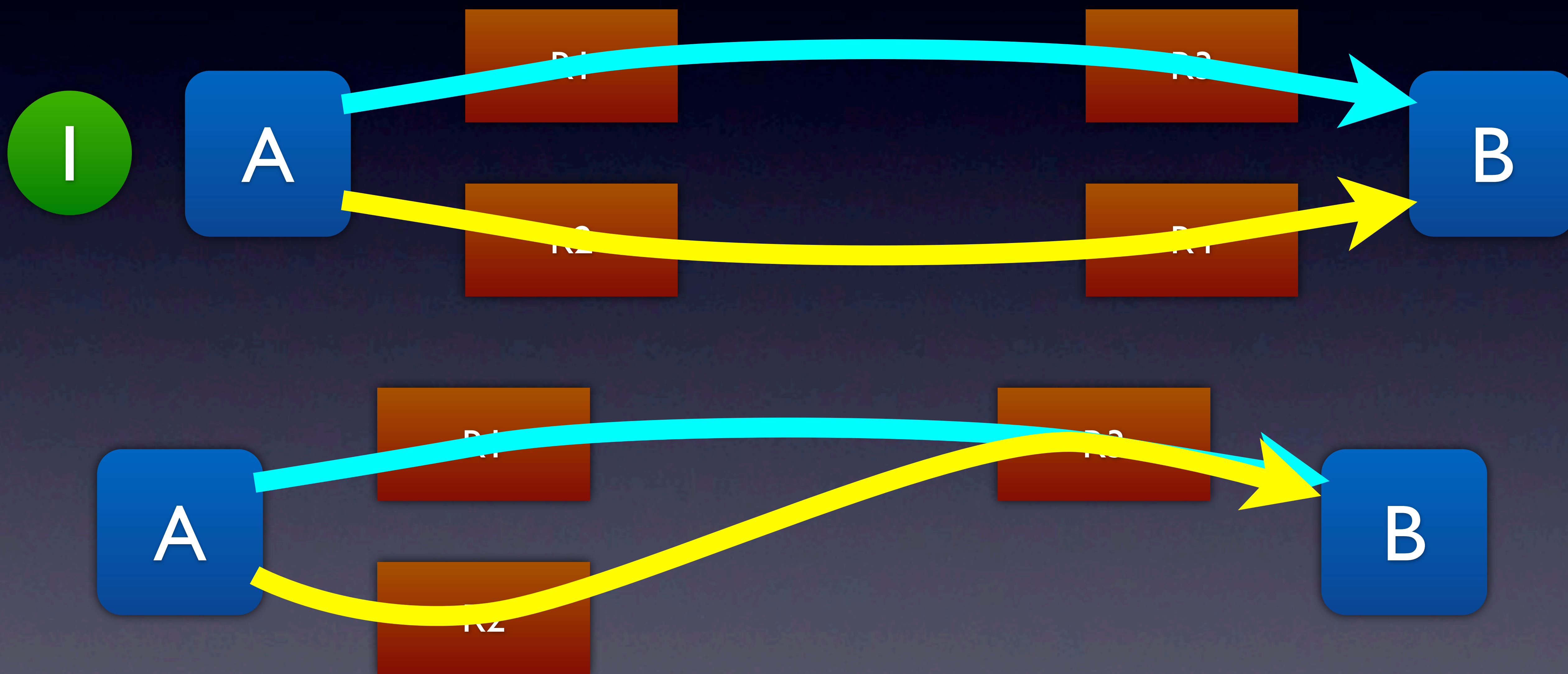
1. Multi-address:
   - paths are identified by source/dest address pair
2. First hop selection by host:
   - host has multiple NICs and/or multiple default routers with different connectivity
3. Path selector value:
   - value in packets selects the path (like source routing, ToS routing, MPLS)

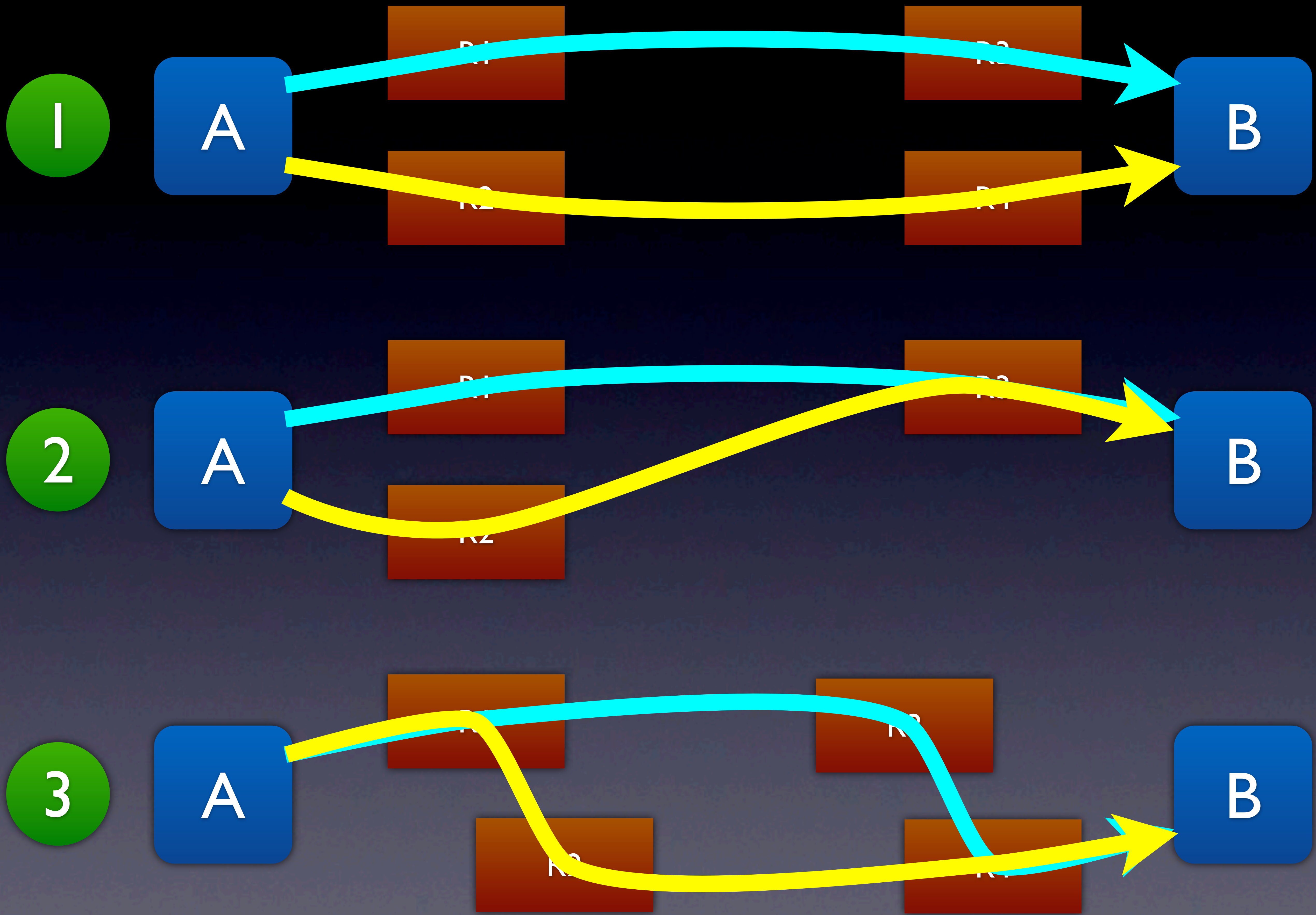
One or two ends

- Multi-address:
  - requires changes at hosts at both ends
- Single address:
  - only change sending host, compatitble with existing receivers

Kinds of multipath

Kinds of multipath



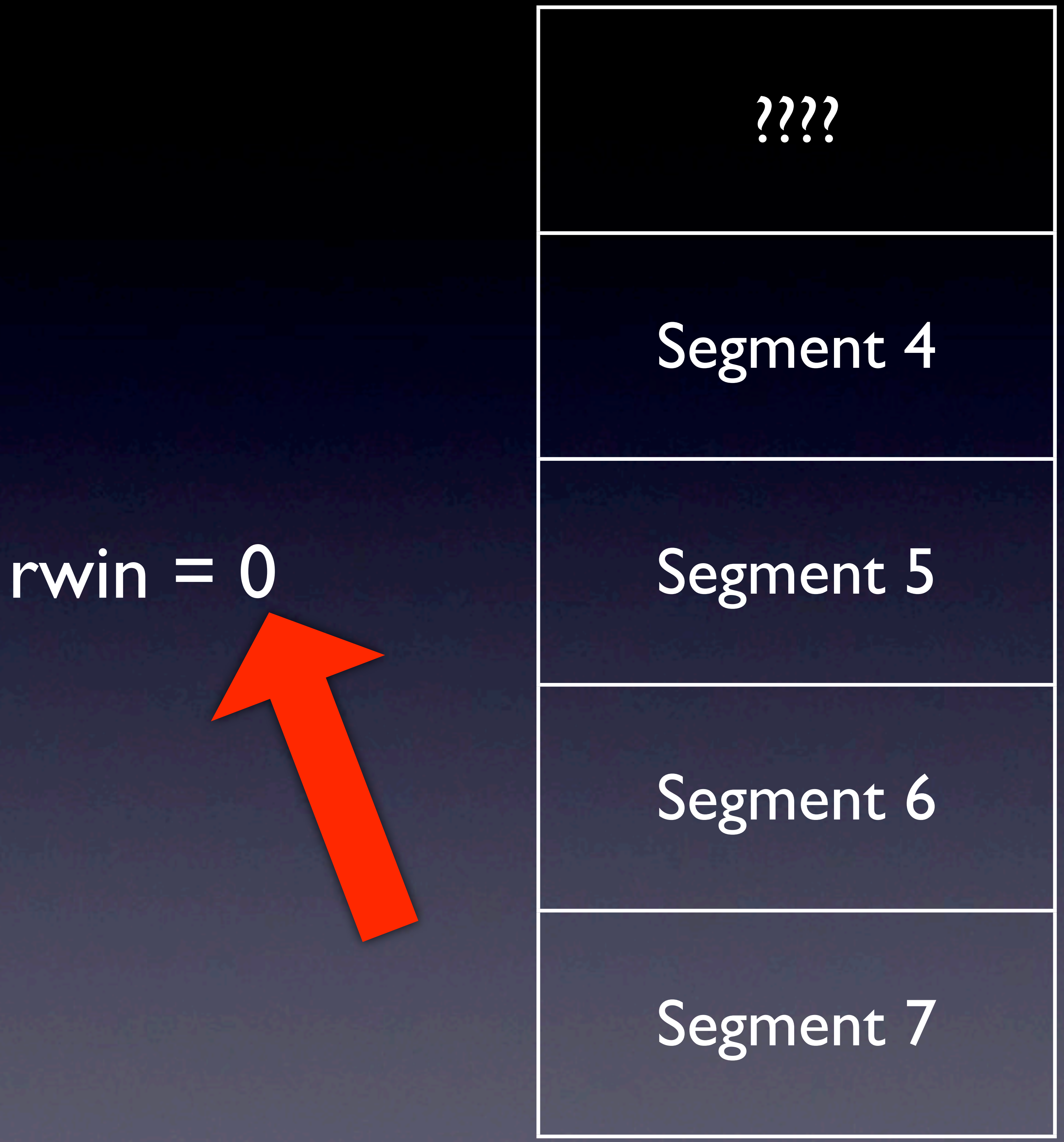


1
A
B
2
A
B
3
A
B

Multipath flow control

Receive buffer issues

- What if a path breaks?
- Packets keep flowing over other path
- But missing packets sent over broken path create hole in seqnum space
- Packets over working path are buffered
- When receiver buffer space runs out, session stalls
- Need to fix this or MPTCP is less robust



(caveat: much simplified!)

Multipath congestion control

Fairness, performance

- Congestion control determines how aggressively MPTCP competes with other flows
- Assumption: an MPTCP subflow can't be more aggressive than a NewReno flow
- Also: MPTCP must perform no worse than a NewReno flow on the best path

Either TCP fair or MP

- Crunching the numbers, these assumptions/restrictions give us either:
  - aggregate of MPTCP subflows is TCP fair but only one subflow can be used fully
  - multiple subflows used fully at the same time, but MPTCP is more aggressive than single flow NewReno

Why??

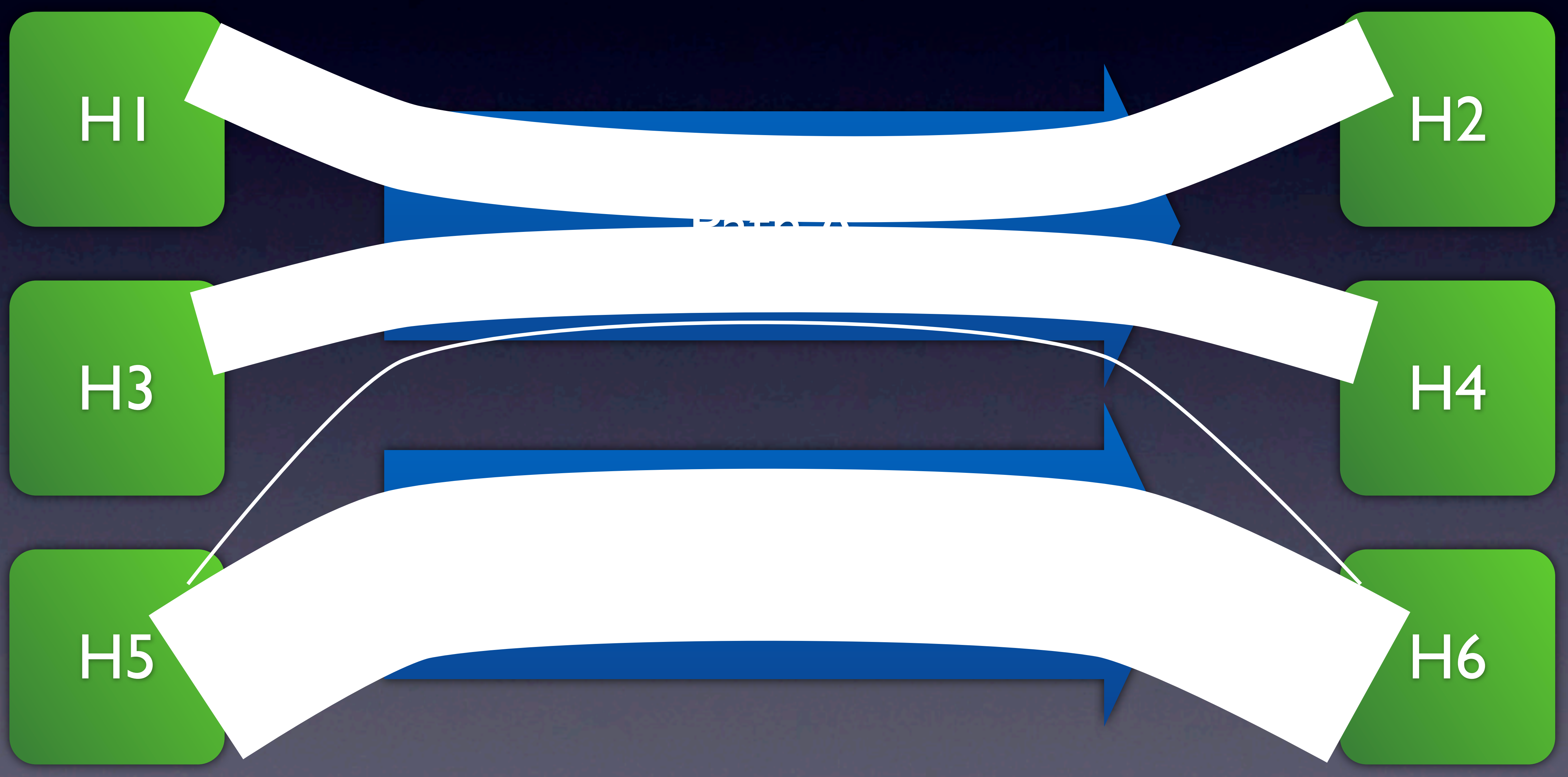
- Consider two airlines
  - one uses high RTT, MSS (seat size) and low *p* (~~loss probability~~ price)
  - other low RTT/MSS, high *p*
- Always choose fast/big/cheap if you can!

Resource pooling

- Have a collection of resources behave like a single, big one
- Only works if traffic can be moved from one path to another
- We do this through congestion control coupling: move traffic to less congested paths

Resource pooling

Resource pooling



Uncoupled CC

- Normal congestion control on *n* subflows
  - compete head-to-head with normal flows
  - weak resource pooling: only through faster completion
  - if several subflows share a bottleneck, multipath TCP takes a bigger share of the bottleneck than normal TCP (unfair?)

CC coupling

- Congestion control coupling necessary for resource pooling
- But this is tricky: can easily lead to reduced user performance or instability
- More work needed

That's it!

- Multipath is cool!
- TCP is a good place to implement it
- Join multipathtcp list:
  - https://www.ietf.org/mailman/listinfo/multipathtcp
- Questions?