Interdomain Routing (plus Transport Wrapup)

Tom Anderson

"A good network is one that I never have to think about" – Greg Minshall

TCP Known to be Suboptimal

Small to moderate sized connections

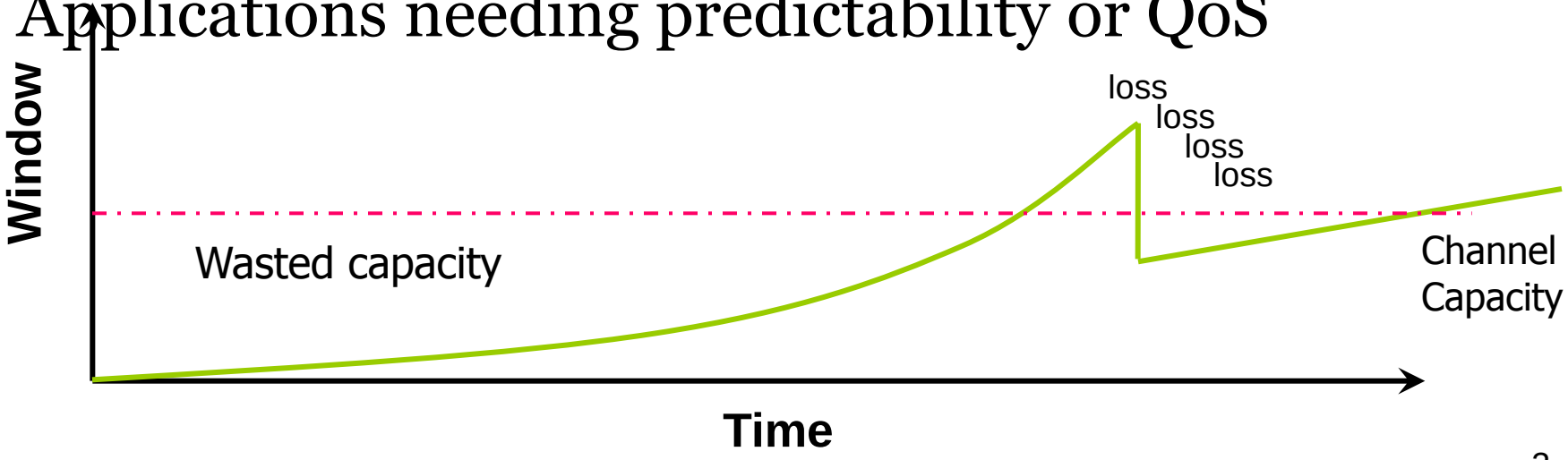
Intranets with low to moderate utilization

Wireless transmission loss

High bandwidth; high delay

Interactive applications

Applications needing predictability or QoS



Observation

Trivial to be optimal with help from the network; e.g., ATM rate control

- Hosts send bandwidth request into network
- Network replies with safe rate (min across links in path)

Can endpoint congestion control be near optimal with *no* change to the network?

- Assume: cooperating endpoints
- Router support only for isolation, not congestion control

PCP approach: directly emulate optimal router behavior!

Congestion Control Approaches

	Endpoint	Router Support
Try target rate for full RTT; if too fast, backoff	TCP, Vegas, RAP, FastTCP, Scalable TCP, HighSpeed TCP	DecBit, ECN, RED, AQM
Request rate from network; send at that rate	**PCP**	ATM, XCP, WFQ, RCP

PCP Goals

1. Minimize transfer time
2. Negligible packet loss, low queueing
3. Work conserving
4. Stability under extreme load
5. Eventual fairness

TCP achieves 3-5 (mostly)

PCP achieves all five (in the common case)

Probe Control Protocol (PCP)

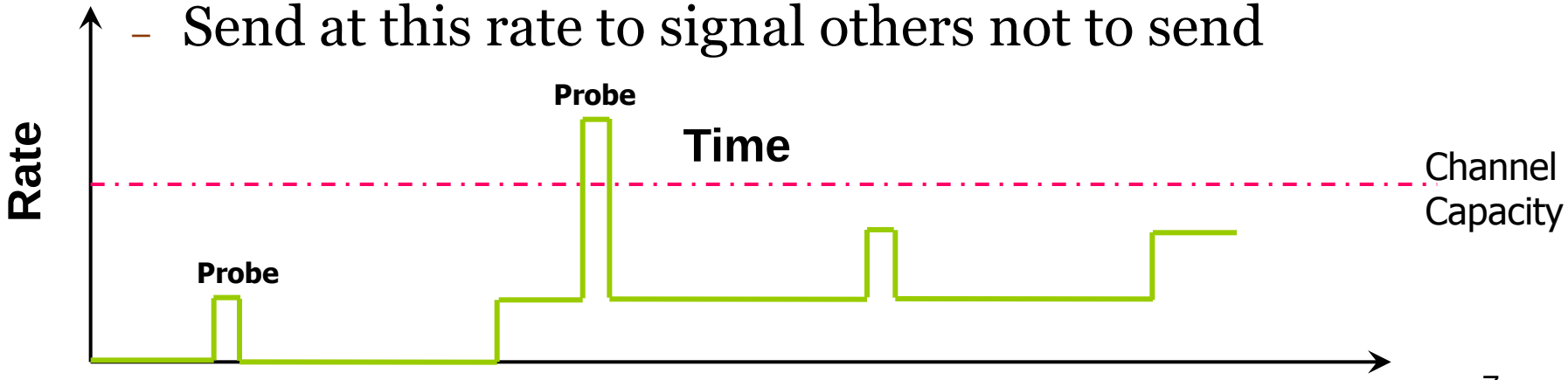
Probe for bandwidth using short burst of packets

- If bw available, send at the desired uniform rate (paced)
- If not, try again at a slower rate

Probe is a request

Successful probe sets the sending rate

- Send at this rate to signal others not to send

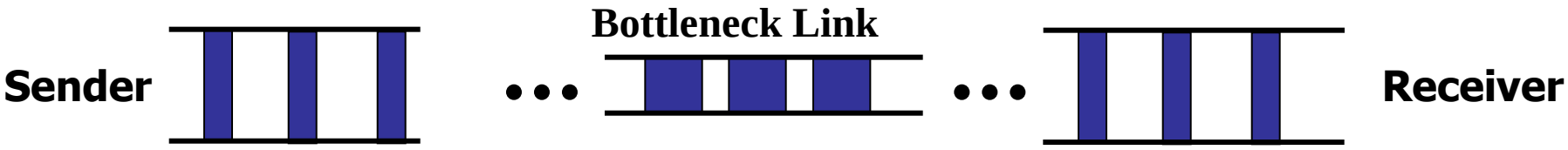


Probes

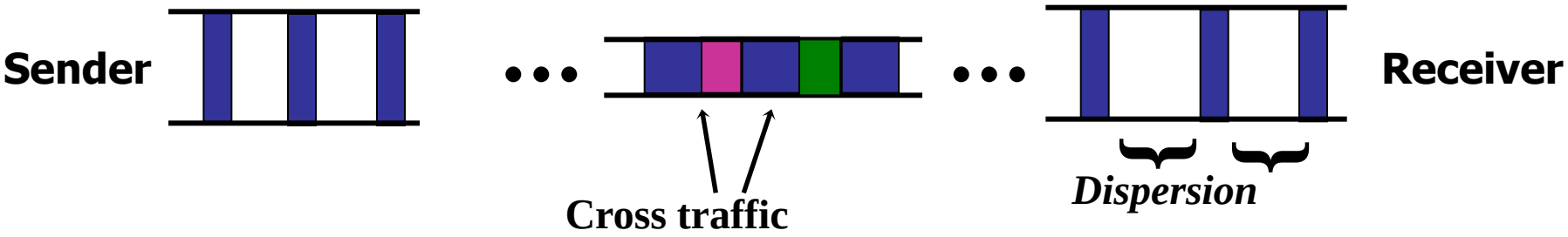
Send packet train spaced to mimic desired rate

Check packet dispersion at receiver

Successful probe:



Failed probe:



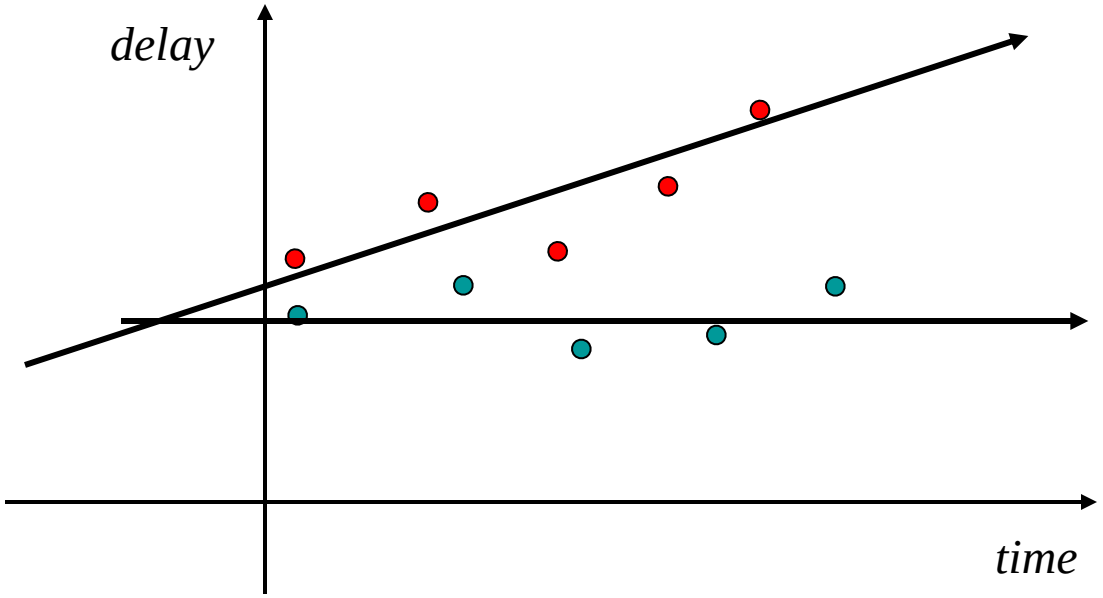
Probabilistic Accept

Randomly generate a slope consistent with the observed data

- same mean, variance as least squares fit

Accept if slope is not positive

Robust to small variations in packet scheduling



Rate Compensation

Queues can still increase:

- Failed probes, even if short, can add to queueing
- Simultaneous probes could allocate the same bw
- Probabilistic accept may decide probe was successful, without sufficient underlying available bandwidth

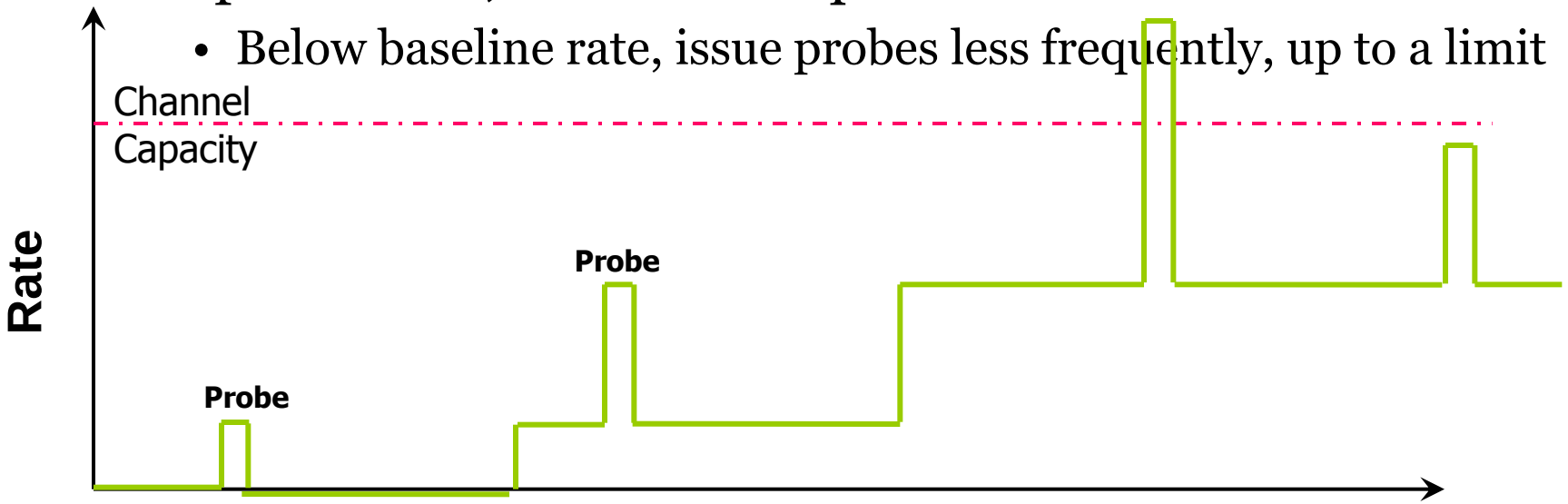
PCP solution

- Detect increasing queues by measuring packet latency and inter-packet delay
- Each sender decreases their rate proportionately, to eliminate queues within a single round trip
- Emulates AIMD, and thus provides eventual fairness

Binary Search

Base protocol: binary search for channel capacity

- Start with a baseline rate: One MSS packet per round-trip
- If probe succeeds, double the requested bandwidth
- If probe fails, halve the requested bandwidth
  - Below baseline rate, issue probes less frequently, up to a limit



History

Haven't we just reinvented TCP slow start?

- Still uses *O(log n)* steps to determine the bandwidth
- Does prevent losses, keeps queues small

Host keeps track of previous rate for each path

- Because probes are short, ok to probe using this history
- Currently: first try 1/3rd of previous rate
  - If prediction is inaccurate/accurate, we halve/double the initial probe rate

TCP Compatibility

TCP increases its rate regardless of queue size

- Should PCP keep reducing its rate to compensate?

Solution: PCP becomes more aggressive in presence of non-responsive flows

- If rate compensation is ineffective, reduce speed of rate compensation: “tit for tat”
- When queues drain, revert to normal rate compensation

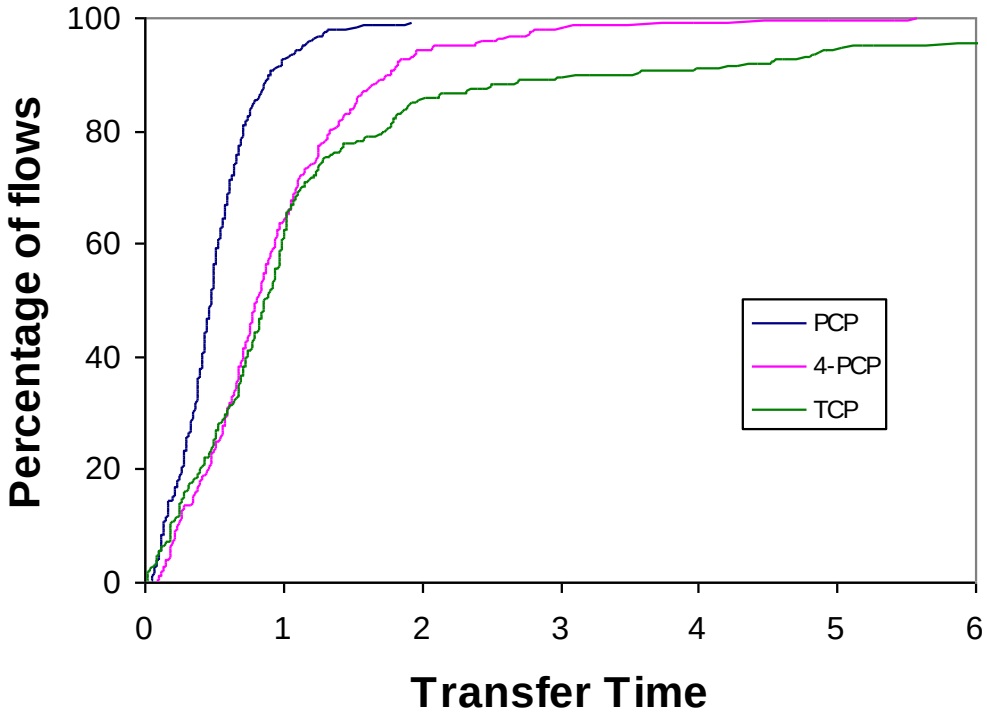
Otherwise compatible at protocol level

- Future work: PCP sender (receiver) induces TCP receiver (sender) to use PCP

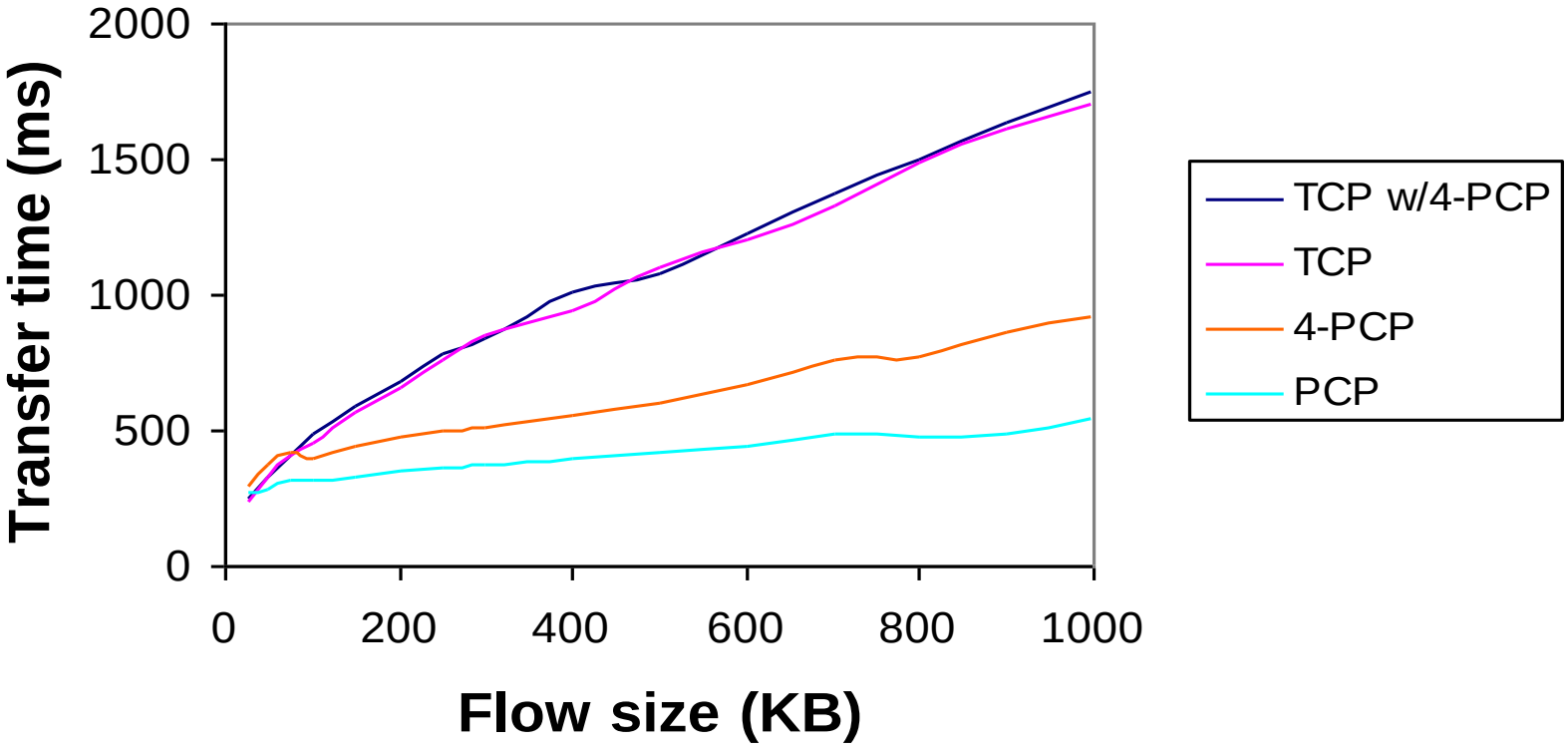
Performance

User-level implementation

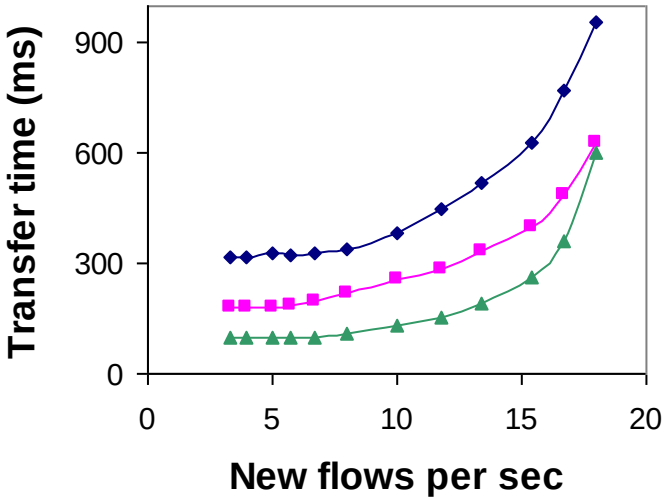
- 250KB transfers between every pair of RON nodes
- PCP vs. TCP vs. four concurrent PCP transmissions

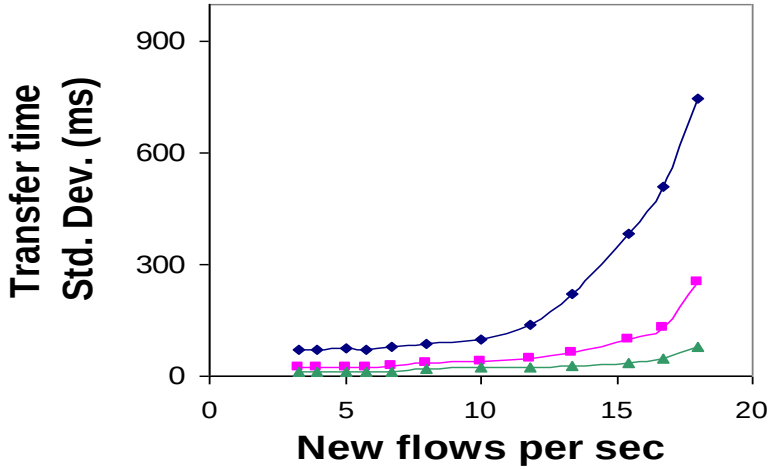


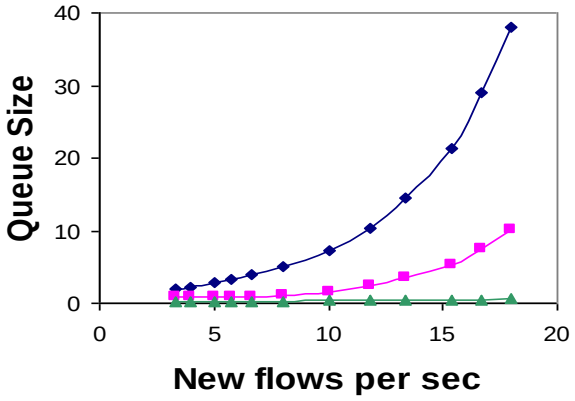
Is PCP Cheating?

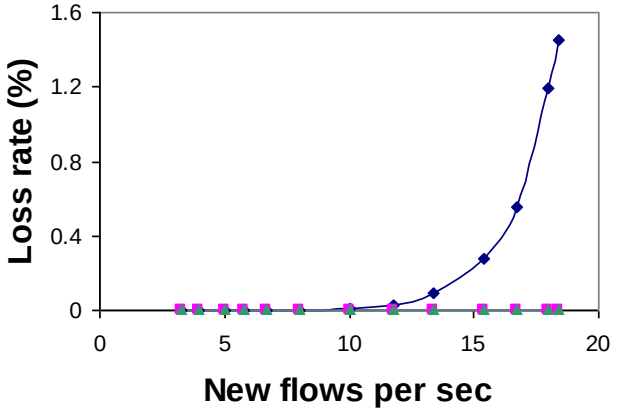


Simulation: Vary Offered Load

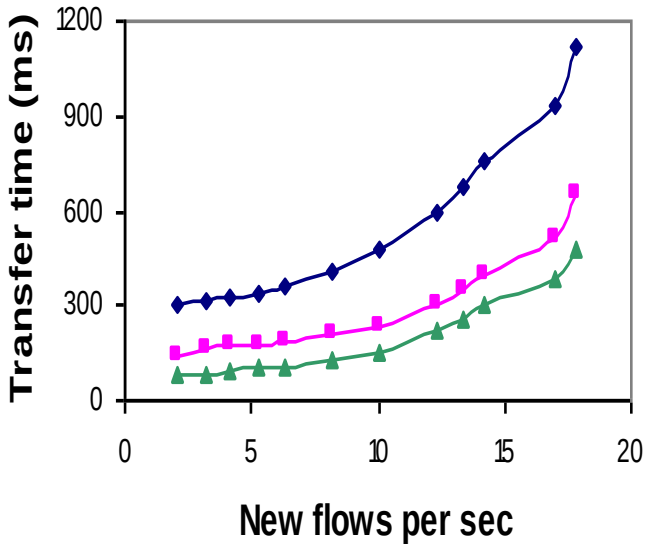


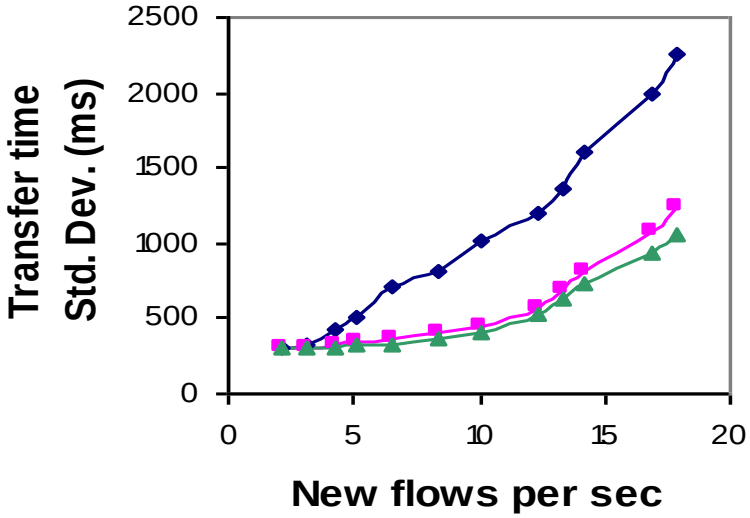


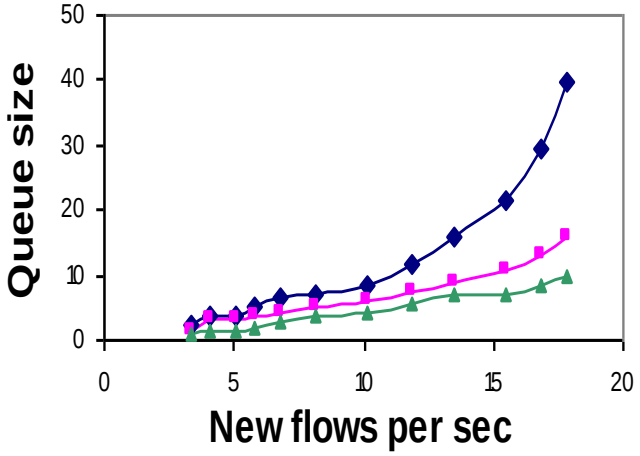


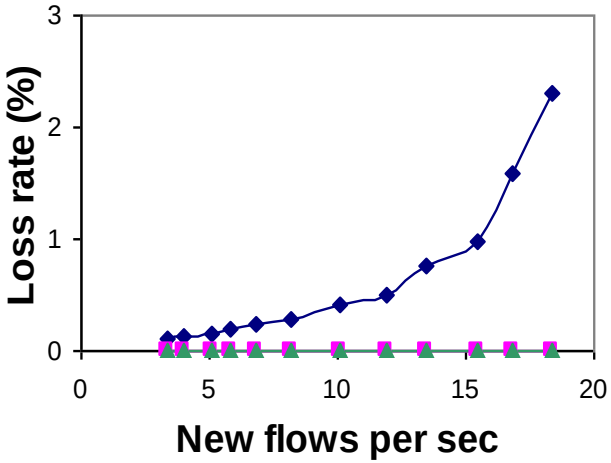


Simulation: Self-Similar Traffic

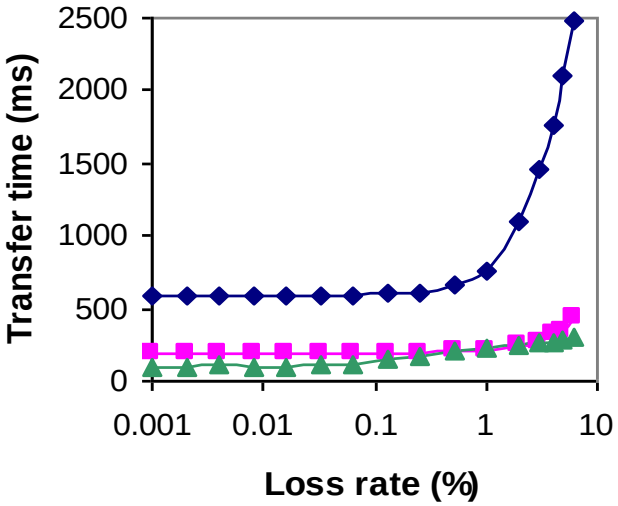


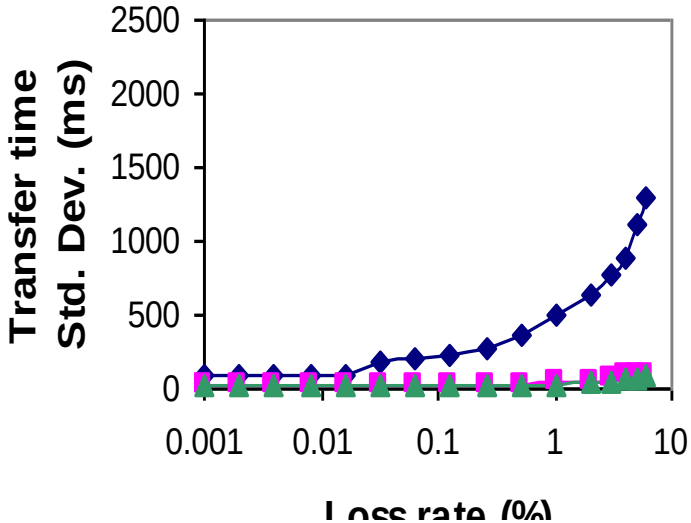


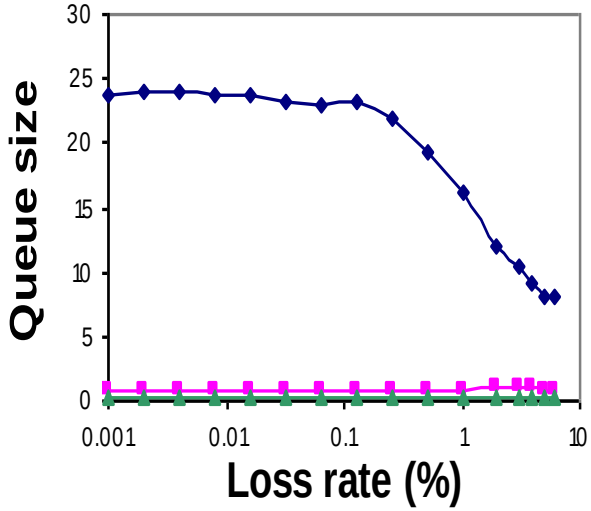


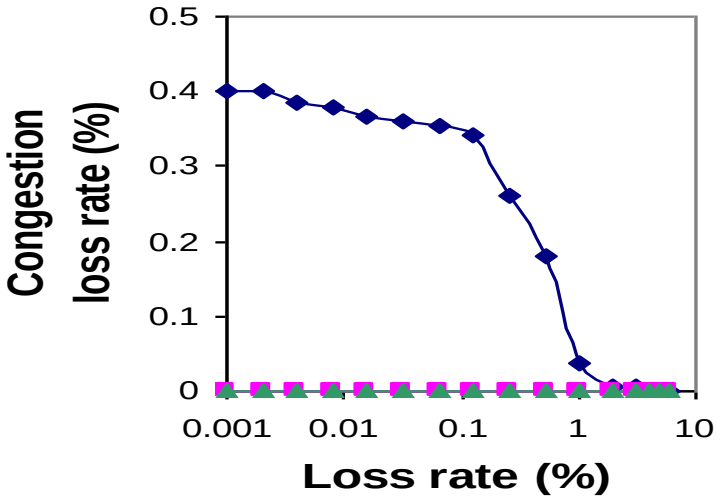


Simulation: Transmission Loss



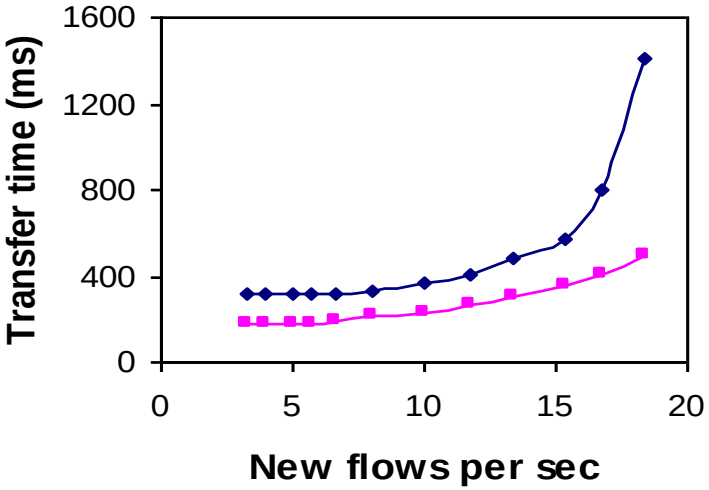


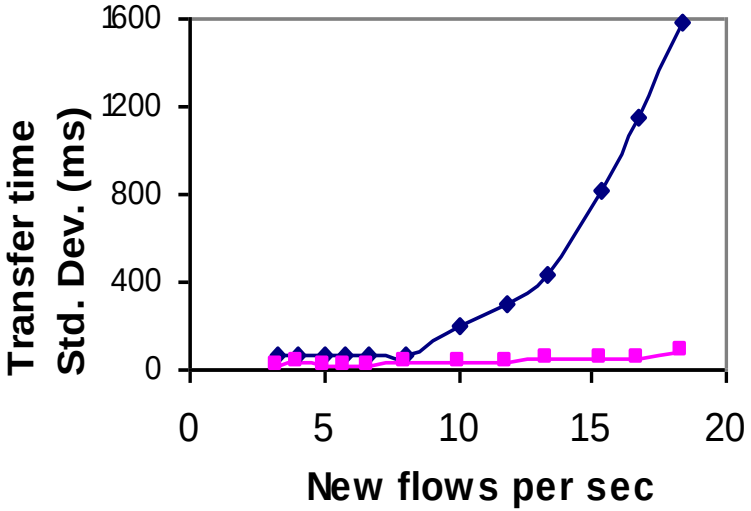


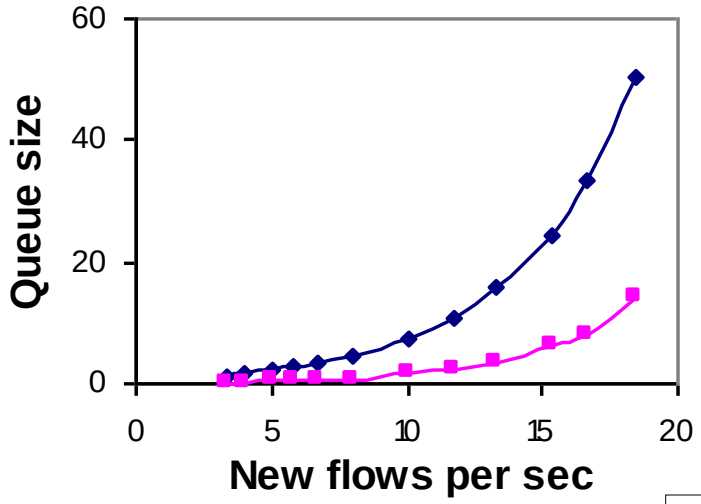


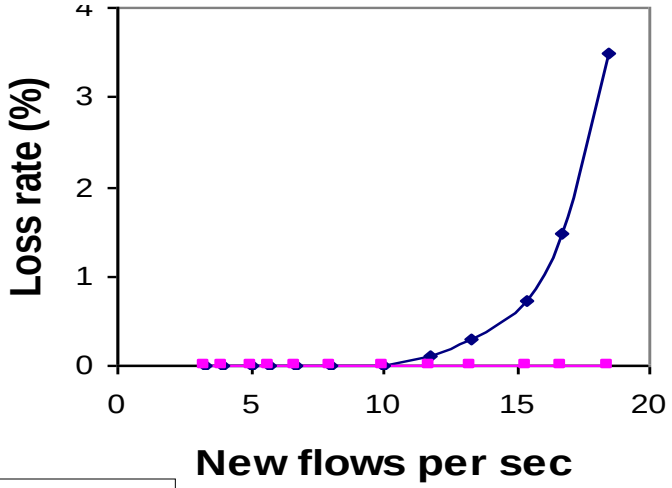
TCP PCP Fair Queueing

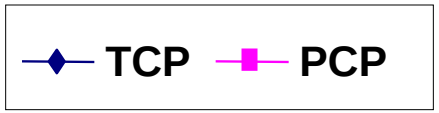
Simulation: Fair-Queued Routers











Related Work

Short circuit TCP's slow-start: TCP Swift Start, Fast Start

Rate pacing: TCP Vegas, FastTCP, RAP

History: TCP Fast Start, MIT Congestion Manager

Delay-based congestion control: TCP Vegas, FastTCP

Available bandwidth: Pathload, Pathneck, IGI, Spruce

Separate efficiency & fairness: XCP

PCP Summary

PCP: near optimal endpoint congestion control

- Emulates centralized control with no special support from network

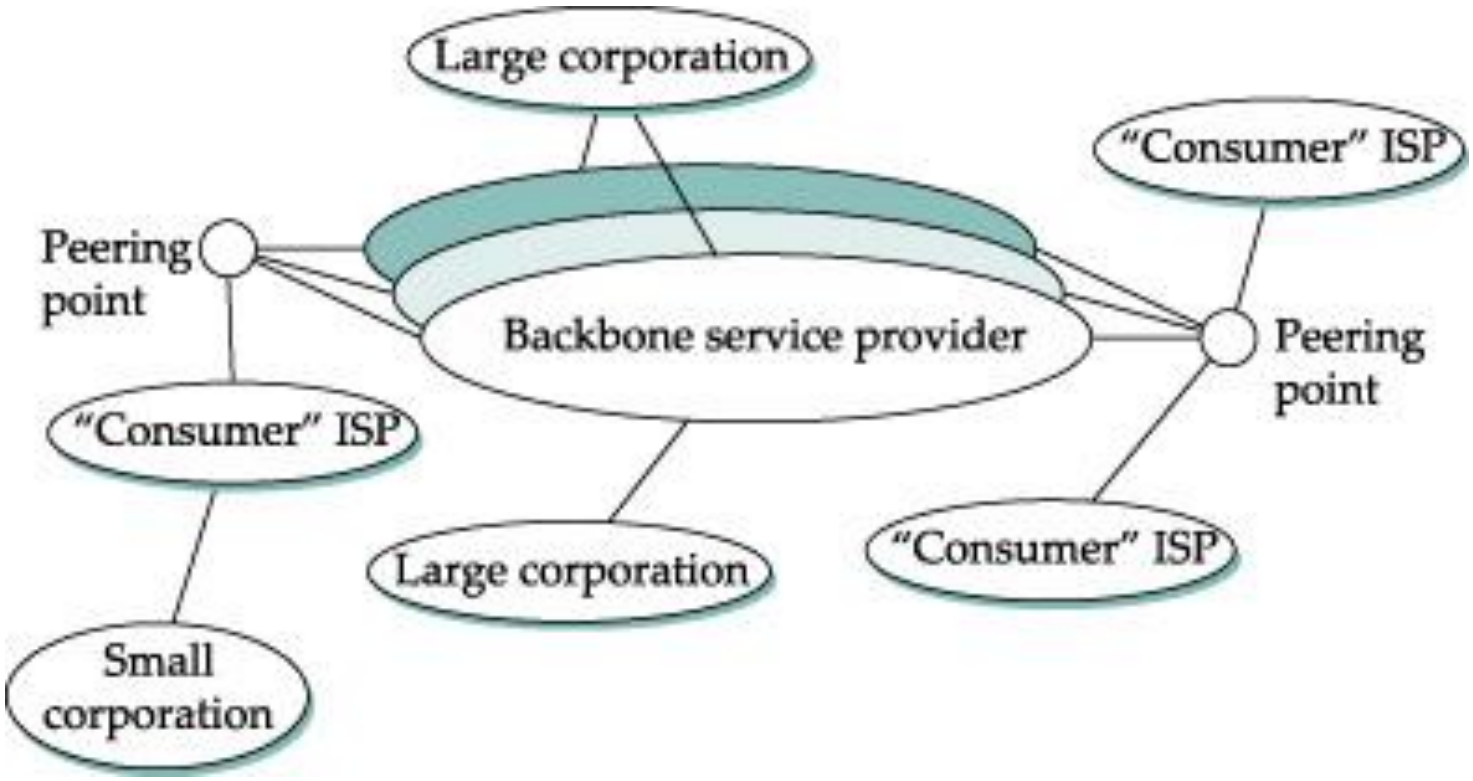
Better than TCP for today's common case

- Most paths are idle and have predictable performance
- Most flows are short-lived

User-level and kernel implementation available:

http://www.cs.washington.edu/homes/arvind/pcp

Internet today



Key goals for Internet routing

Scalability

Support arbitrary policies

- Finding “optimal” paths was less important

(Supporting arbitrary topologies)

Internet routing overview

Two-level hierarchy for scalability

- Intra-domain: within an ISP (OSPF, MPLS)
- Inter-domain: across ISPs (BGP)

Path vector protocol between Ases

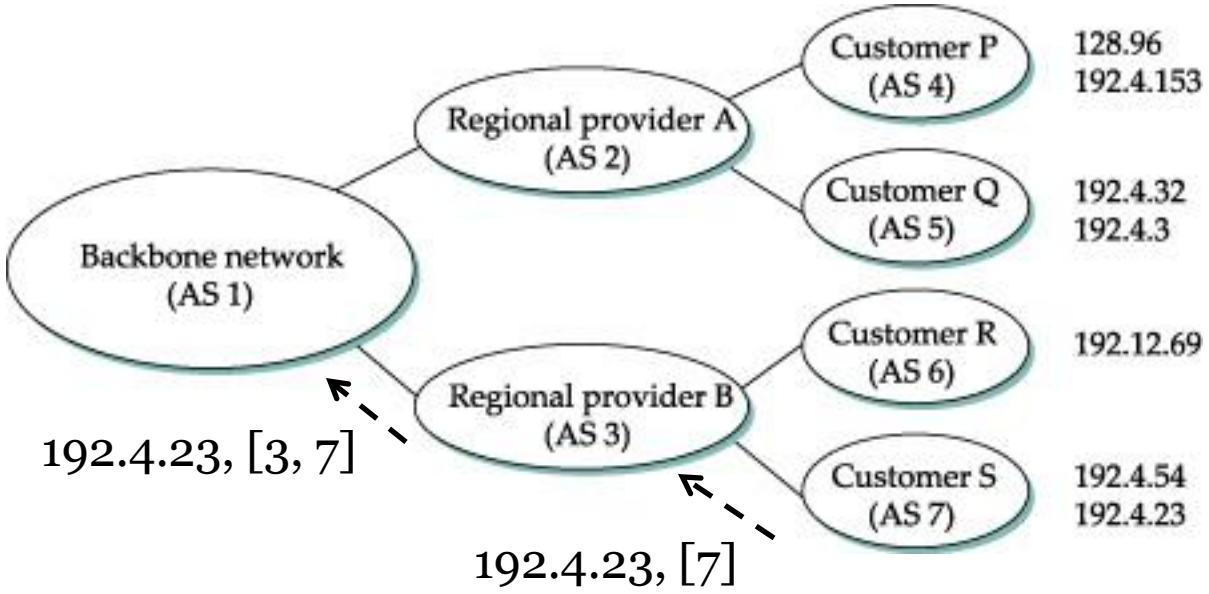
- Can support many policies
- Fewer messages in response to small changes
  - Only impacted routers are informed

Border Gateway Protocol

- Key idea: *opaque policy routing* under local control
  - Preferred routes visible to neighbors
  - Underlying policies are not visible
- Mechanism:
  - ASes send their most preferred path (to each IP prefix) to neighboring ASes
  - If an AS receives a new path, *start using it right away*
  - Forward the path to neighbors, with a *minimum inter-message interval*
    - essential to prevent exponential message blowup
  - Path eventually propagates in this fashion to all AS's

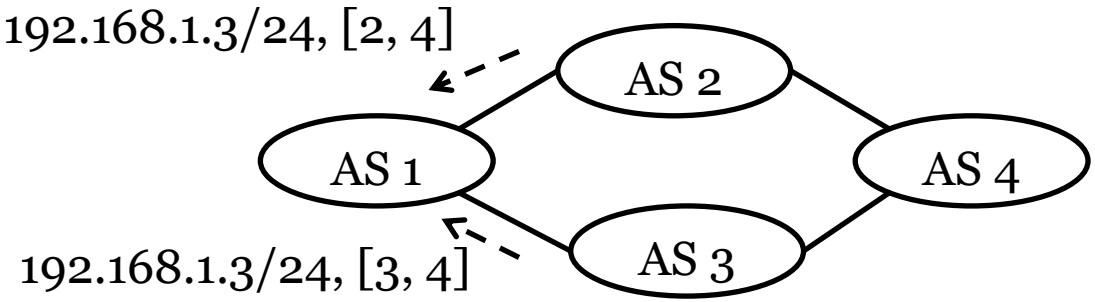
Path vector routing

Similar to distance vector routing info includes entire paths

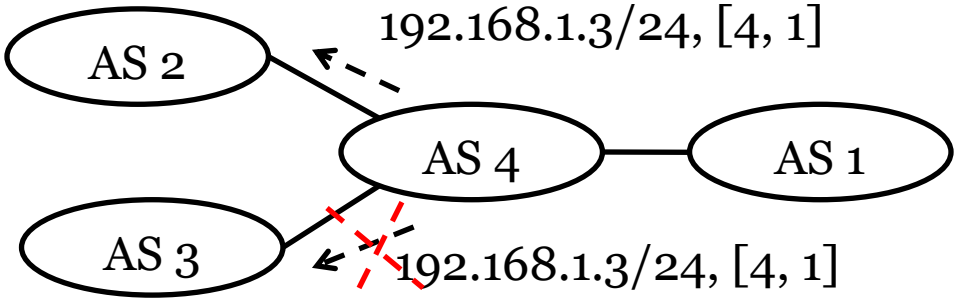


Policy knobs

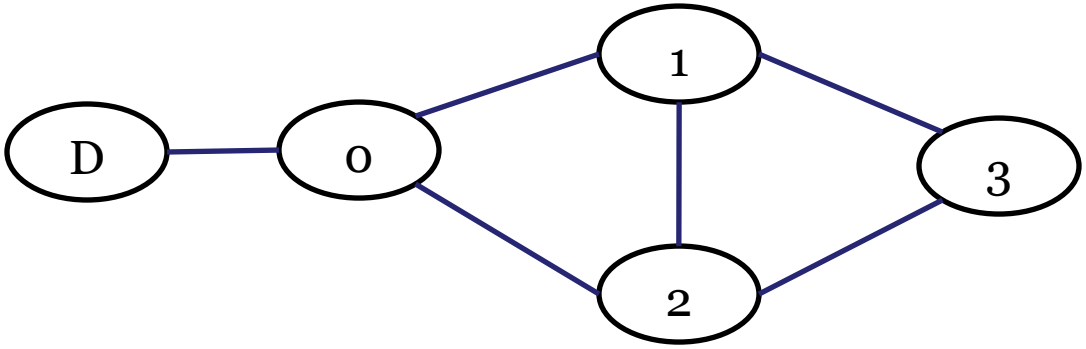
1. Selecting one of the multiple offered paths



2. Deciding who to offer paths



Path vector vs. link state vis-à-vis policy



AS3 preferences
[310]
[320]
[3210]
~~[3120]~~

With path vector, implementing the policy above requires only local knowledge at AS3

With link state, AS3 would need to know the policies of other ASes as well

Typical routing policies

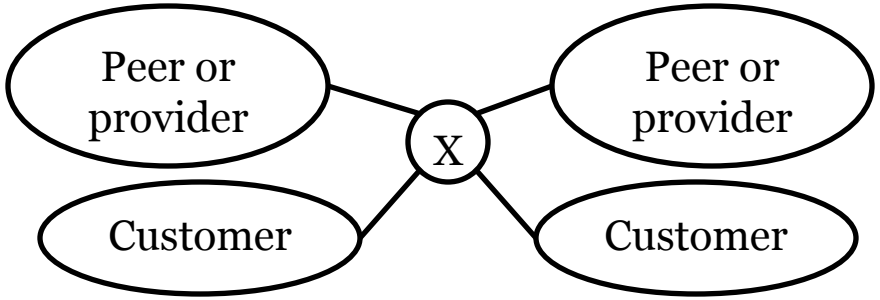
Driven by business considerations

Two common types of relationships between ASes

- Customer-provider: customer pays provider
- Peering: no monetary exchange

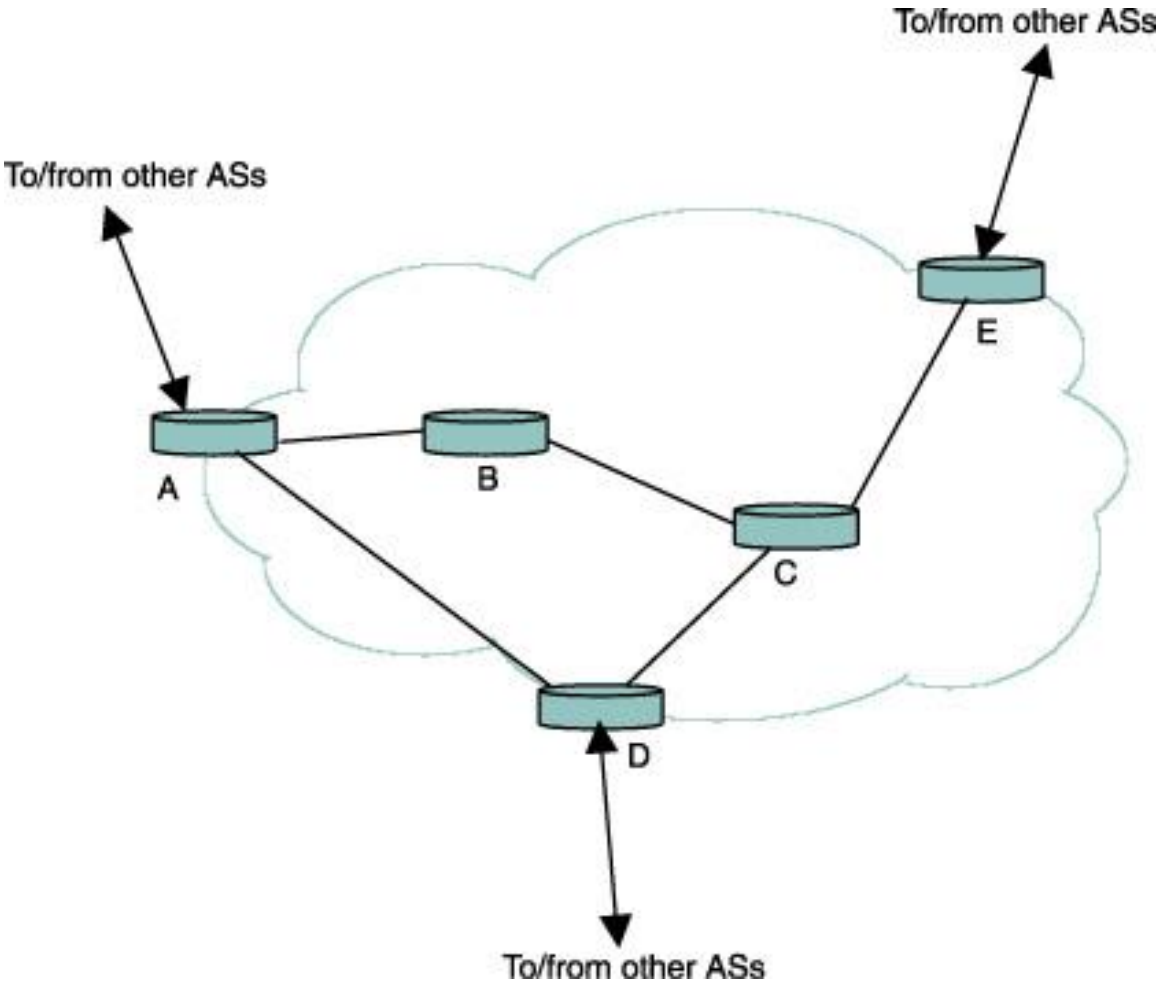
When selecting routes: customer > peer > provider

When exporting routes: do not export provider or peer routes to other providers and peers



Prefer routes with shorter AS paths

BGP at router level



BGP limitations

Path quality

Scale

Security

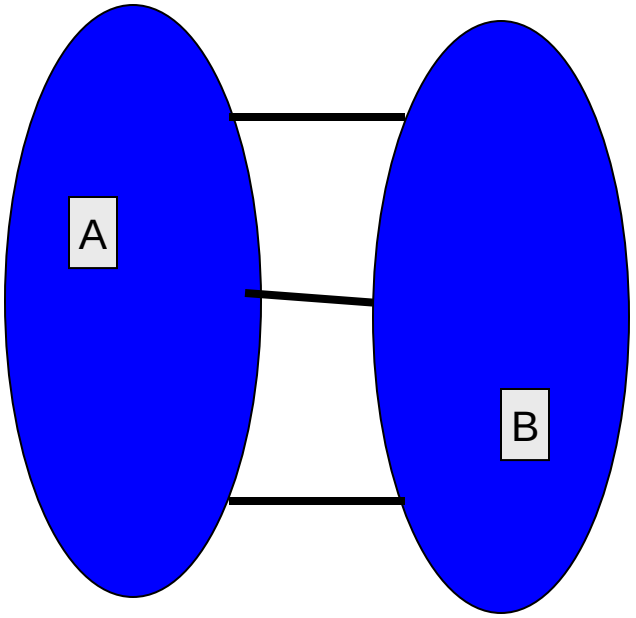
Convergence

Path quality with BGP

Combination of local policies may not be globally good

- Longer paths, asymmetric paths
- Shorter “detours” are often available

Example:
hot potato routing



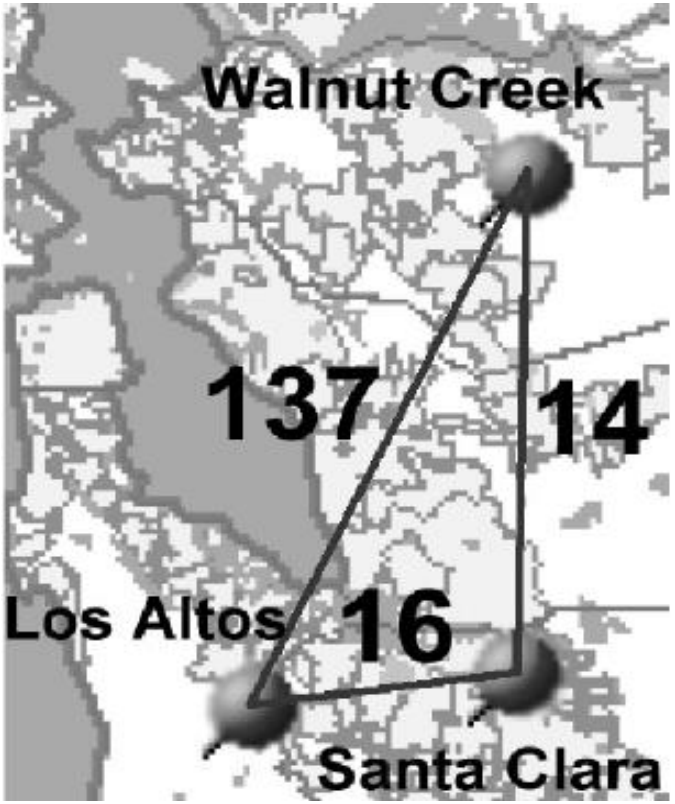
An Anecdote

In 1997, we observed that many routes in the Internet did not obey the triangle inequality

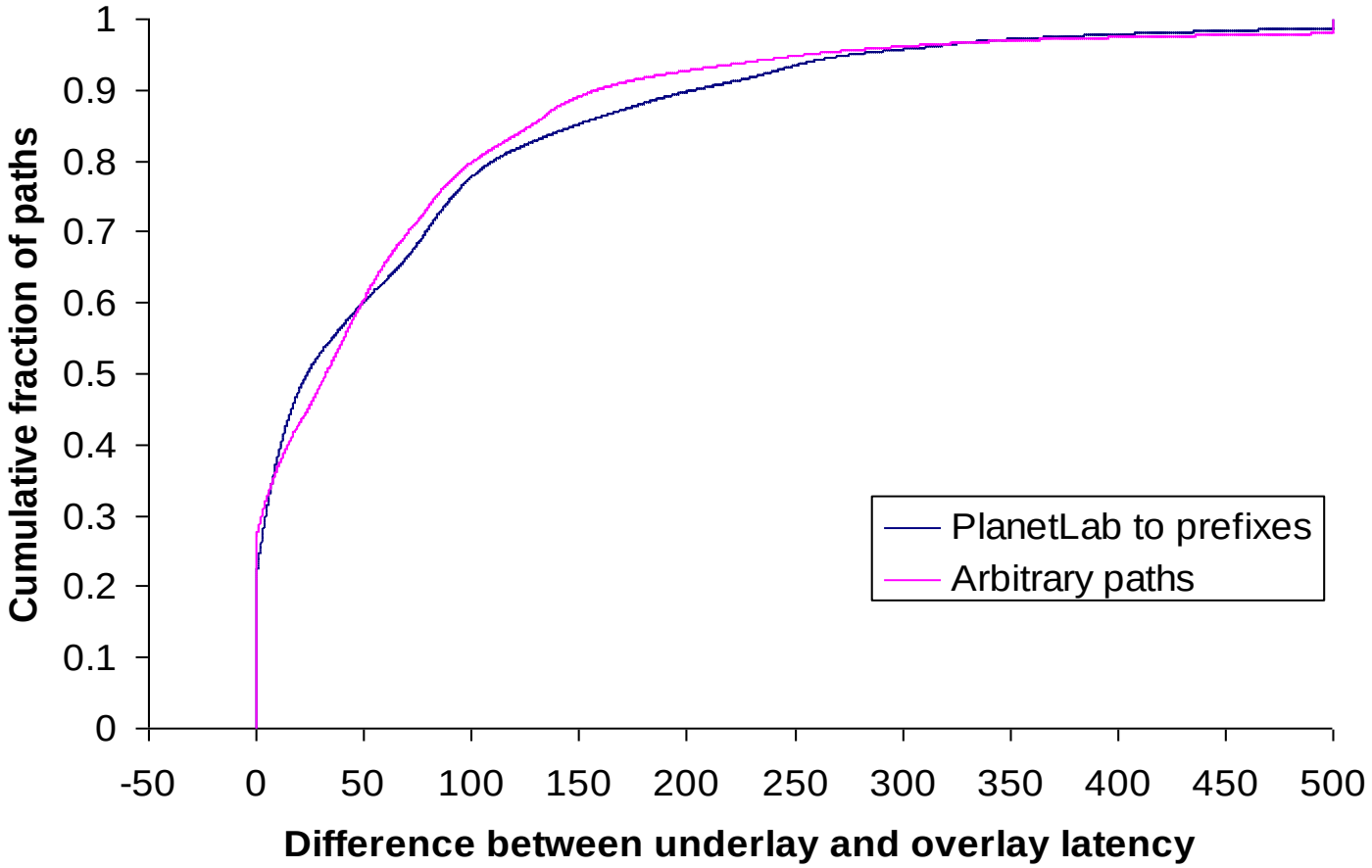
- 40% of all Internet routes
- 10% pathological

Fix via overlay routing?

Embarrass ISPs into improving their routing?



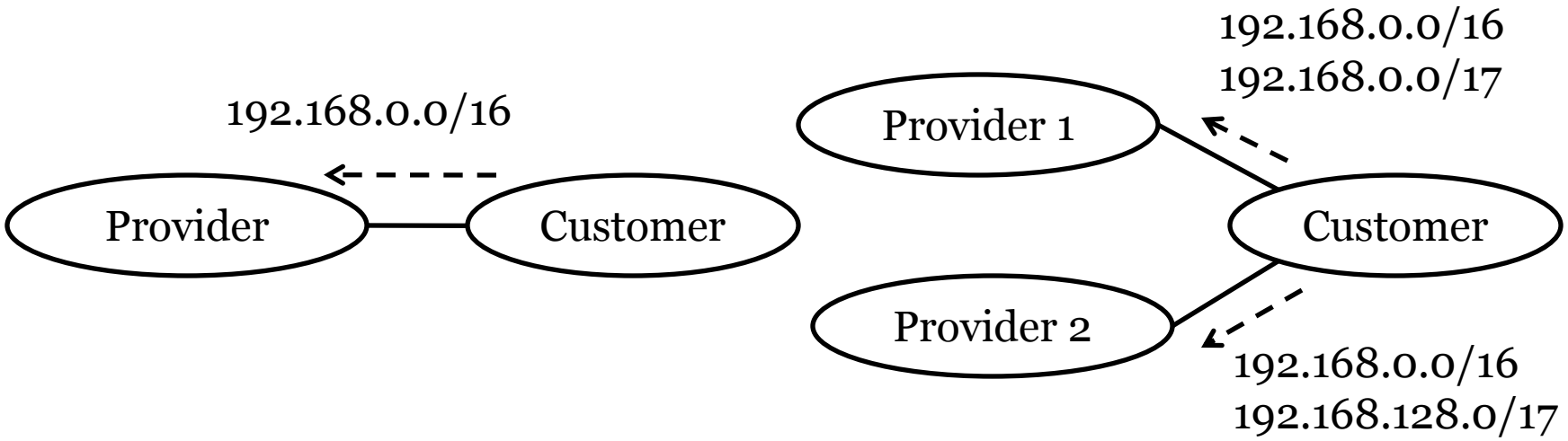
Detour Routing 2005



Scaling pressures on BGP

Too many prefixes (currently ~280K)

Major factors behind growth: multi-homing and traffic engineering



BGP security

Extreme vulnerability to attacks and misconfigurations

- An AS can announce reachability to any prefix
- An AS can announce connectivity to other Ases

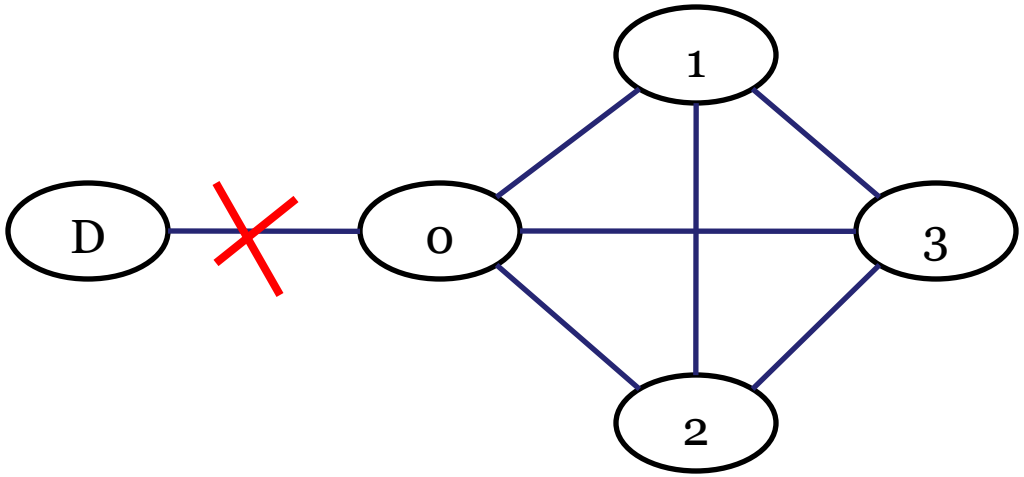
Many known incidents

- AS7007 brought down the whole internet in 1997
- 75% of new route adverts are due to misconfigs [SIGCOMM 2002]
- Commonly used for spamming

Technical solutions exist but none even close to deployment

- Incentives and deployability

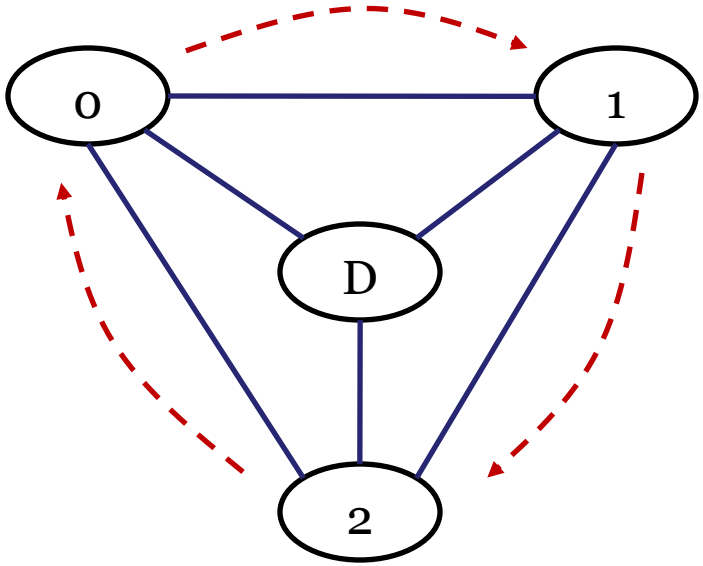
BGP convergence



Temporary loops during path exploration

Differentiating between failure and policy-based retraction can help but not completely

BGP convergence



To get to D, X prefers
[X, (X+1) mod 3]
[X]
Others

Persistent loops can also form in BGP

Fundamentally, the combination of local policies may not have a unique global solution

BGP convergence

Several other issues have been uncovered

- Interaction with intra-domain routing
- Interaction with traffic engineering extensions
- Interaction with scalability extensions

BGP Convergence

Most basic goal of the Internet is global reachability

- When an address is reachable from every other address
- "There is only one failure, and it is complete partition" Clark, *Design Philosophy of the Internet Protocols*

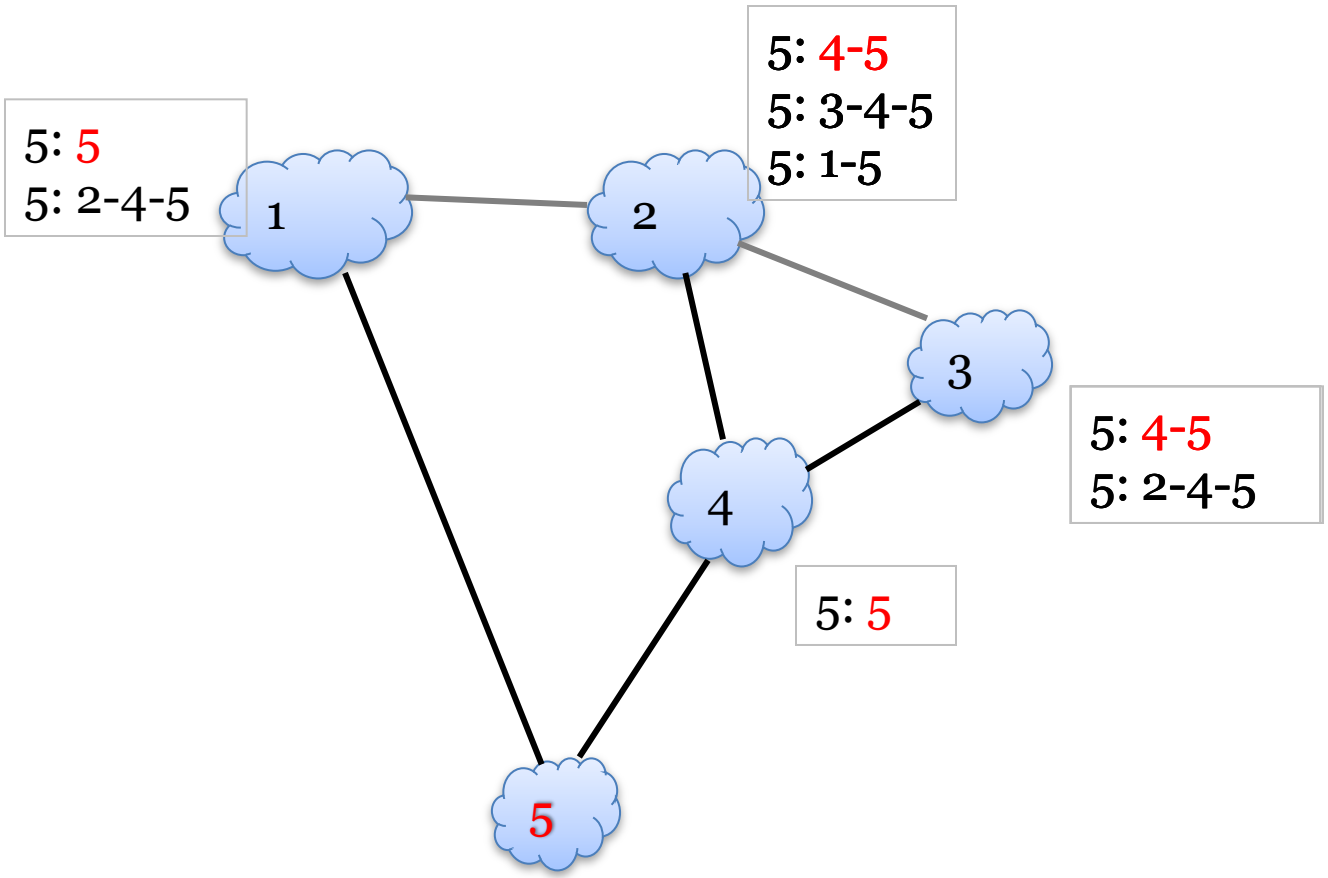
However, BGP does not come close to this goal:

- 10-15% of BGP updates cause loops and inconsistent routing tables
- Loops account for 90% of all packet losses
- Policy changes and traffic engineering also cause transient problems
- 100's of partial connectivity events/hour

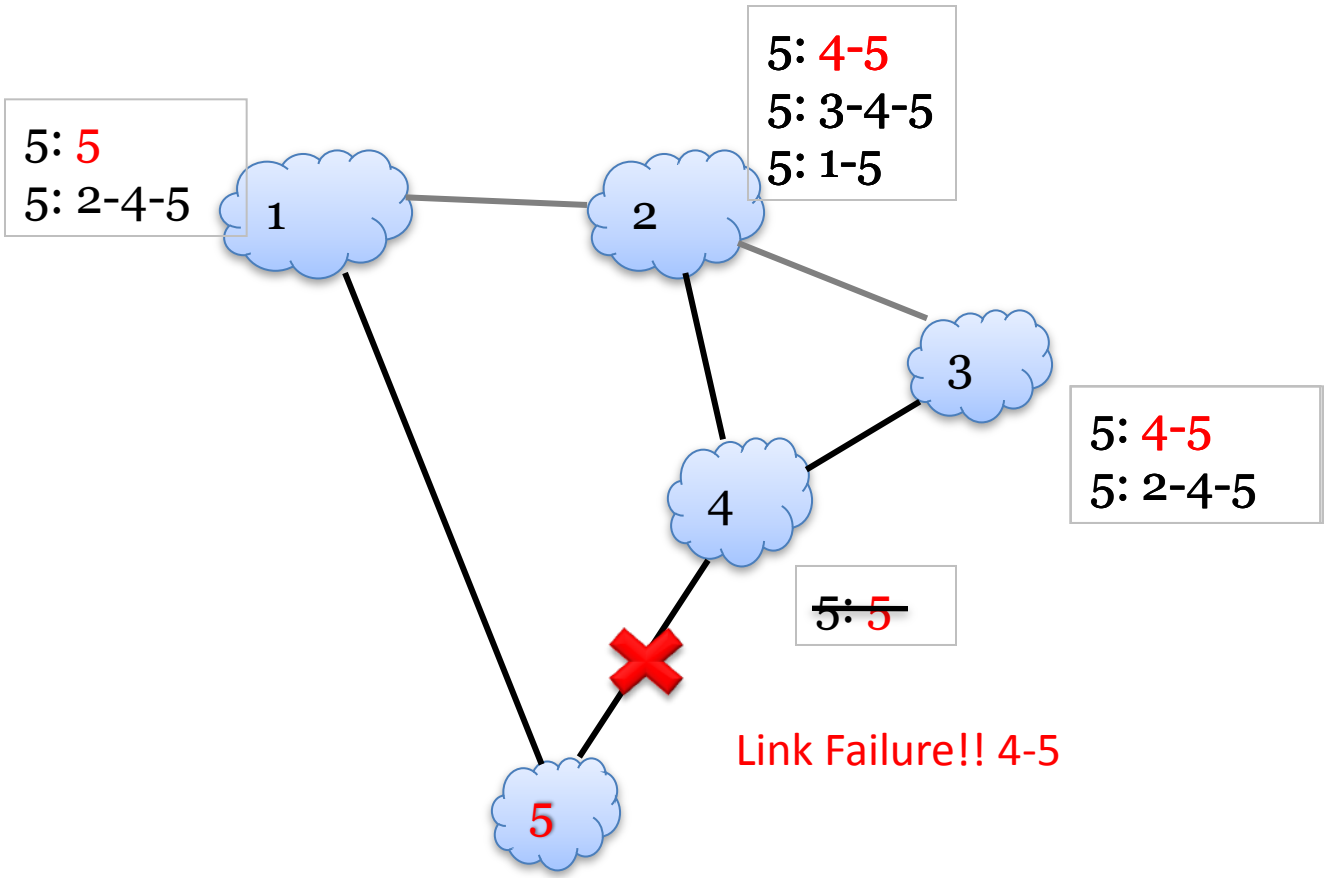
Border Gateway Protocol

- Key idea: *opaque policy routing* under local control
  - Preferred routes visible to neighbors
  - Underlying policies are not visible
- Mechanism:
  - ASes send their most preferred path (to each IP prefix) to neighboring ASes
  - If an AS receives a new path, *start using it right away*
  - Forward the path to neighbors, with a *minimum inter-message interval*
    - essential to prevent exponential message blowup
  - Path eventually propagates in this fashion to all AS's

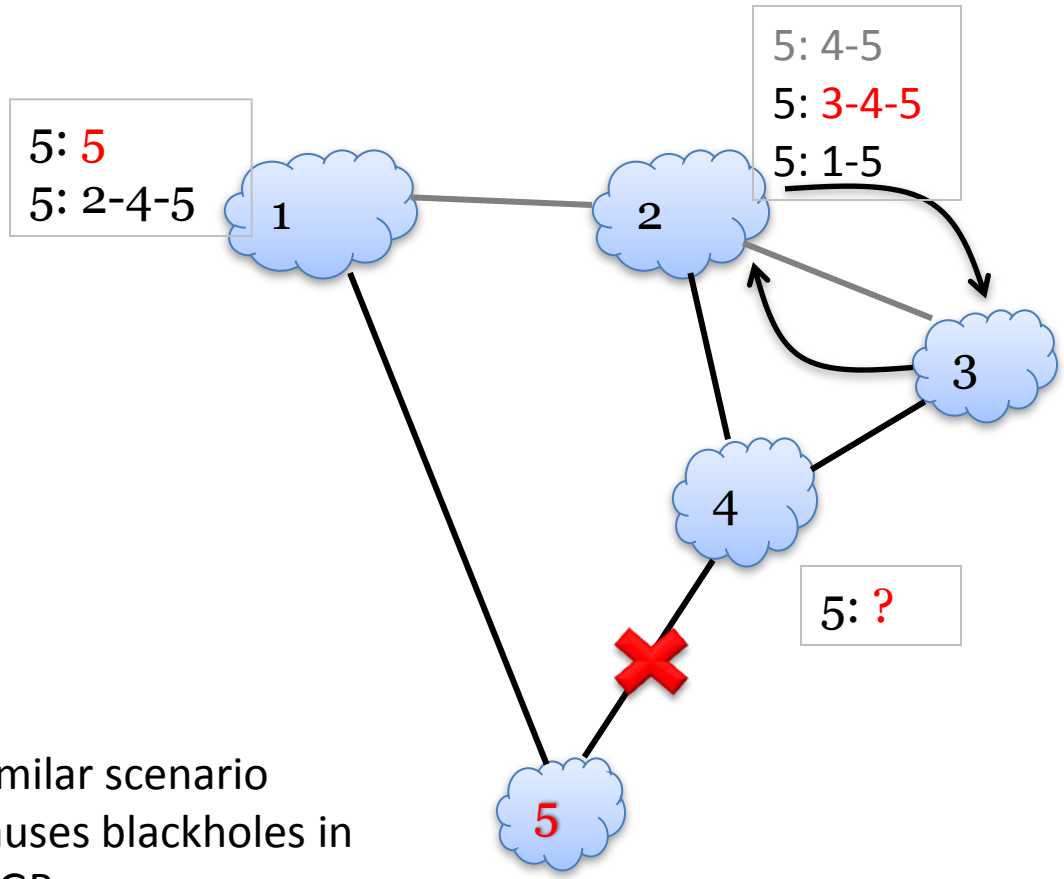
Failures Cause Loops in BGP



Failures Cause Loops in BGP



Failures Cause Loops in BGP

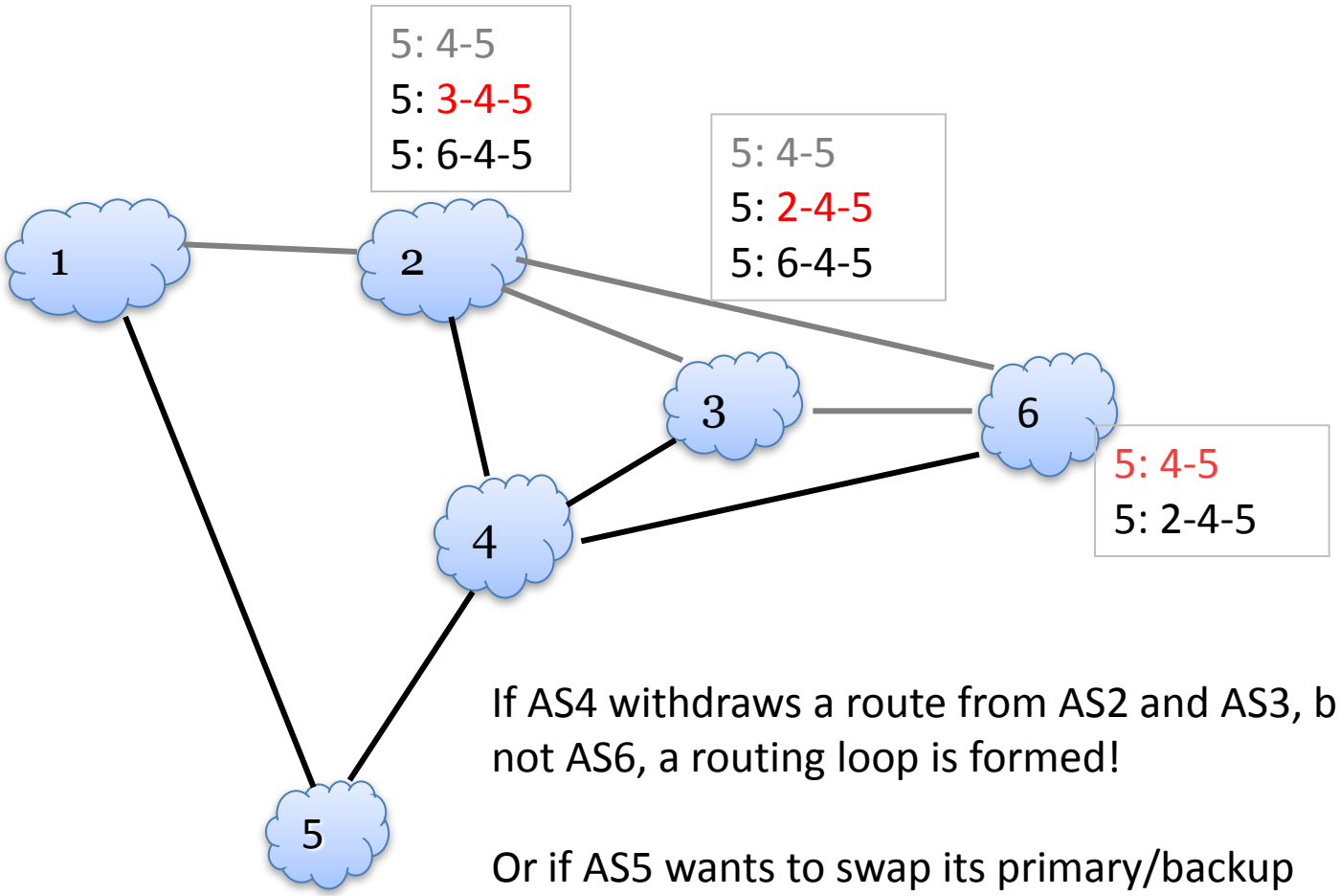


AS2 and AS3 now switch to next best path

A routing loop is formed between AS2 and AS3!

Similar scenario causes blackholes in iBGP

Policy Changes Cause Loops in BGP



If AS4 withdraws a route from AS2 and AS3, but not AS6, a routing loop is formed!

Or if AS5 wants to swap its primary/backup provider from 4 -> 1, or 1->4, a loop is formed

The Internet as a Distributed System

BGP mixes liveness and safety:

- Liveness: routes are available quickly after a change
- Safety: only policy compliant routes are used

BGP achieves neither!

- Messages are delayed to avoid exponential blowup
- Updates are applied asynchronously, forming temporary loops and blackholes

This is a distributed state management problem!

Consensus Routing

Separate concerns of liveness and safety

- Different mechanism is appropriate for each

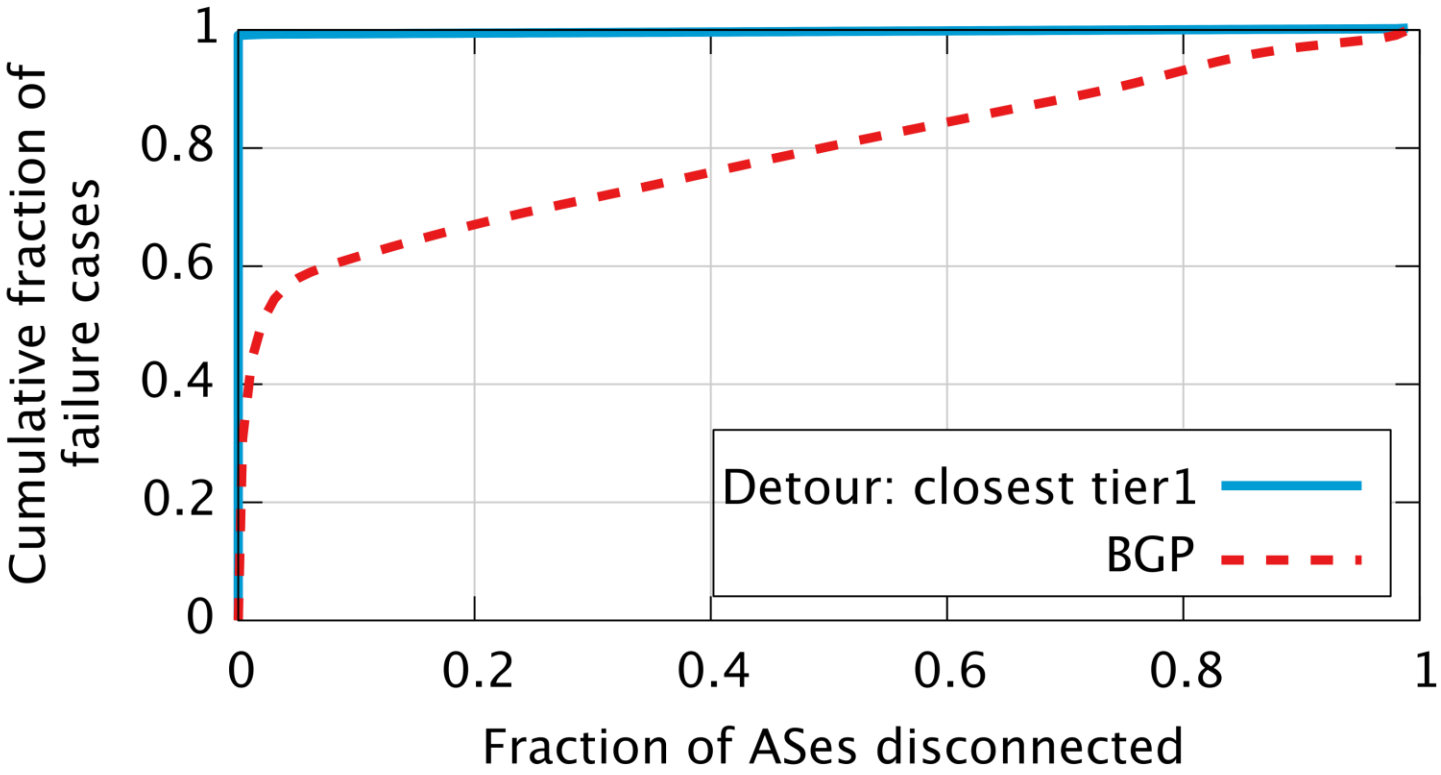
Liveness: routing system adapts to failures quickly

- Dynamically re-route around problem using known, stable routes (e.g., with backup paths or tunnels)

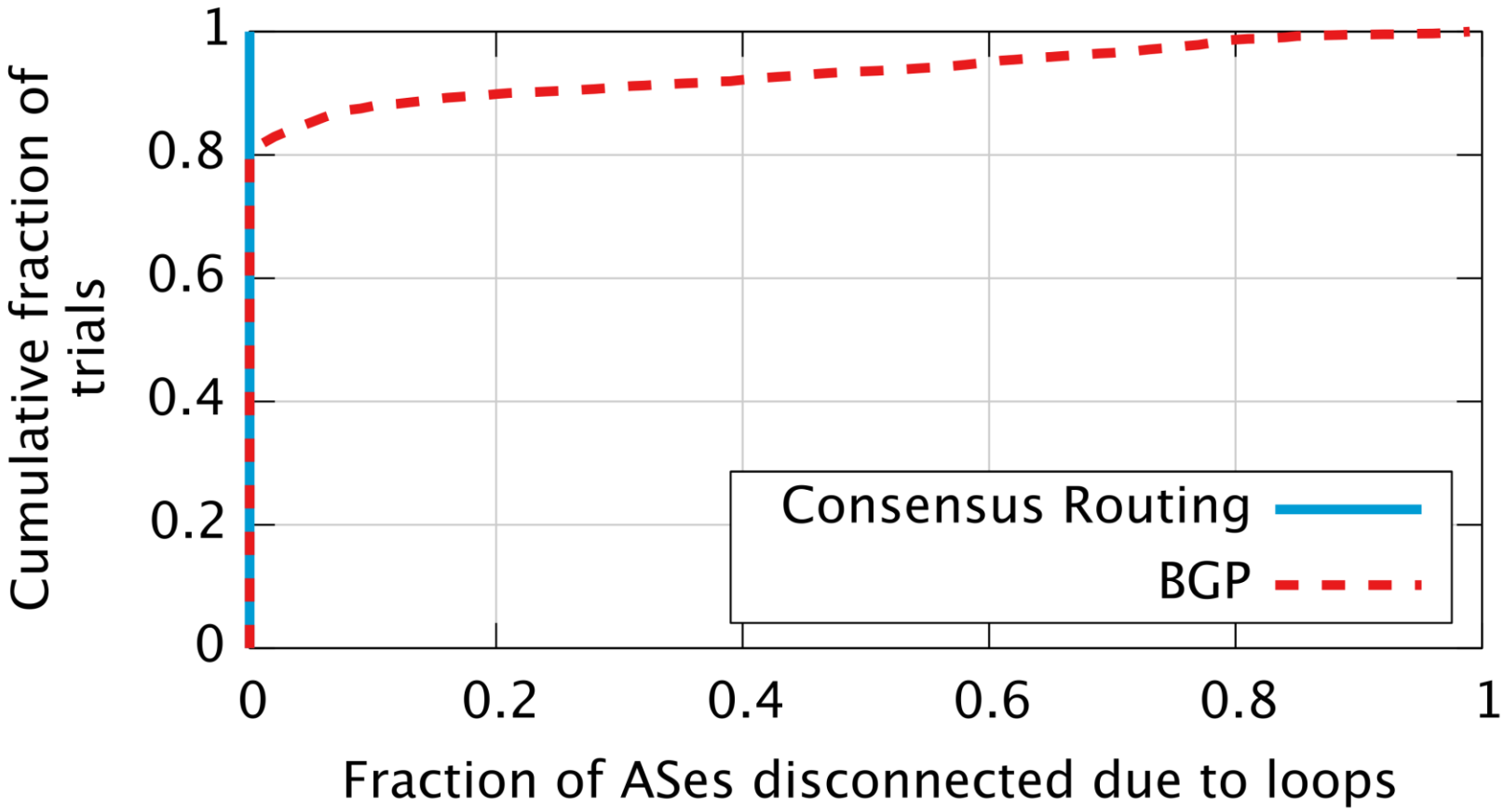
Safety: forwarding tables are always consistent and policy compliant

- AS's compute and forward routes as before, including timers to reduce message overhead
- Only apply updates that have reached everywhere
- Apply updates at the same time everywhere

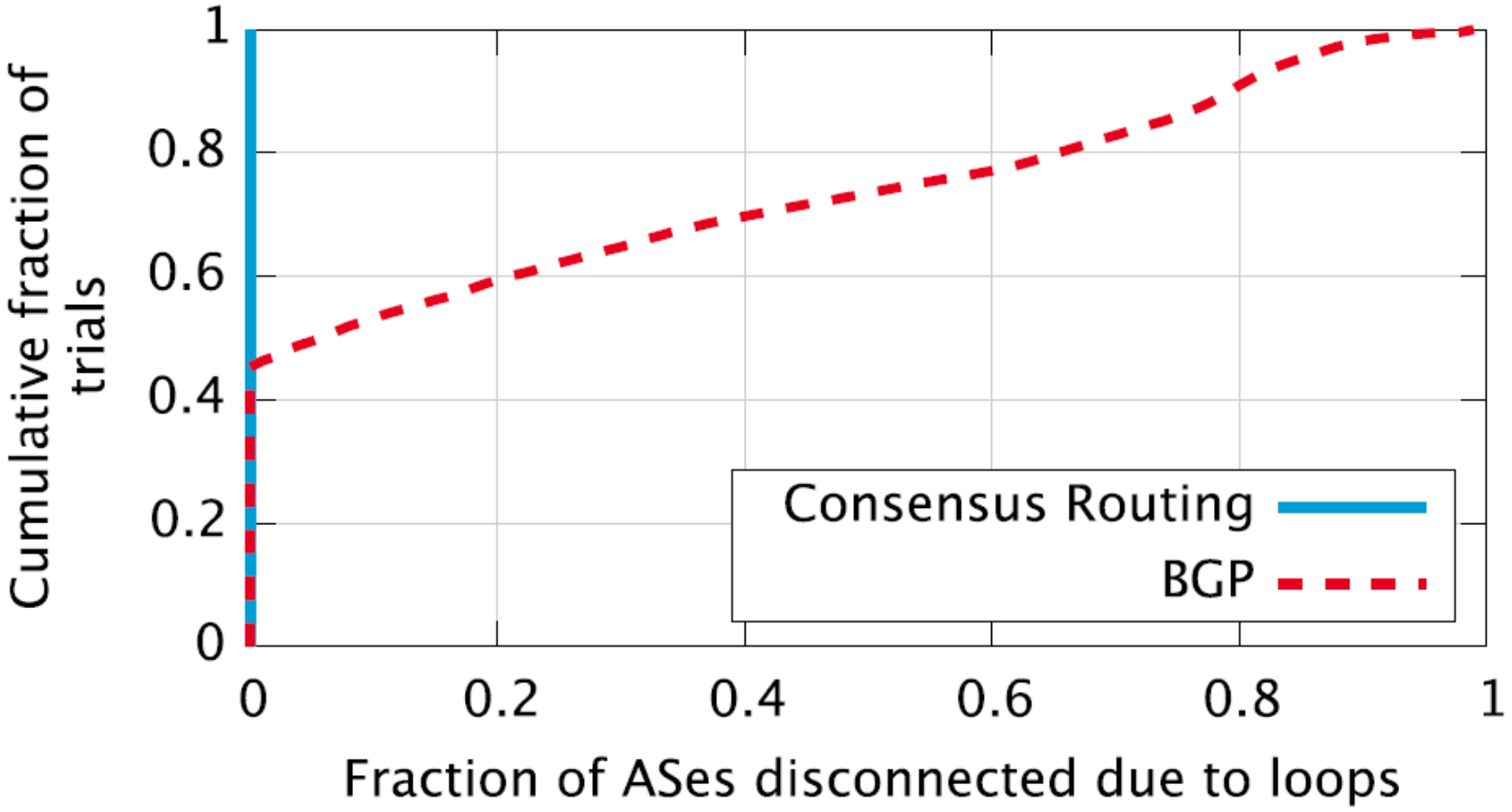
Availability After Failure



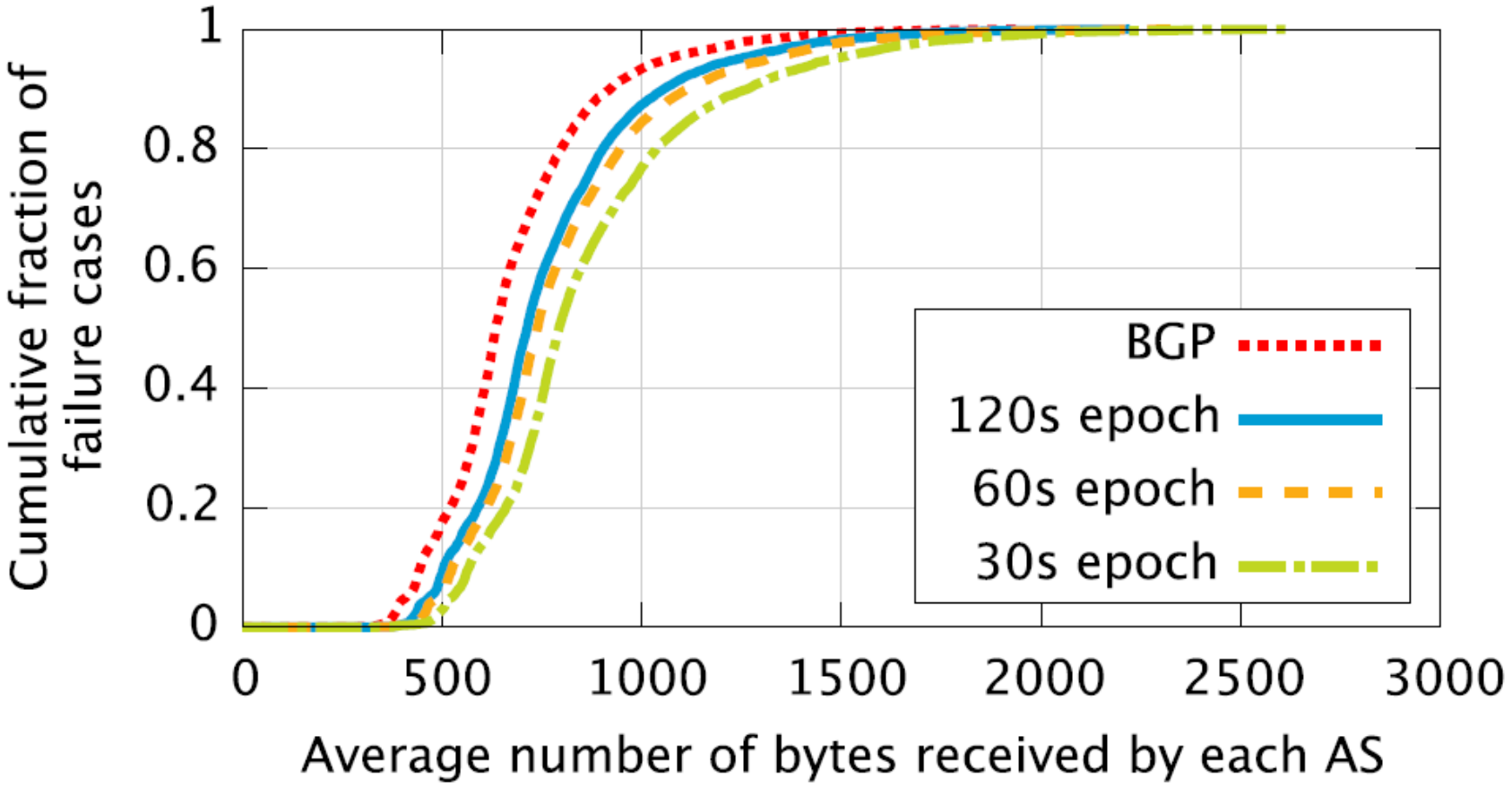
BGP loops, path prepending



BGP loops, prefix engineering



Control traffic overhead



Average delay in reaching consensus