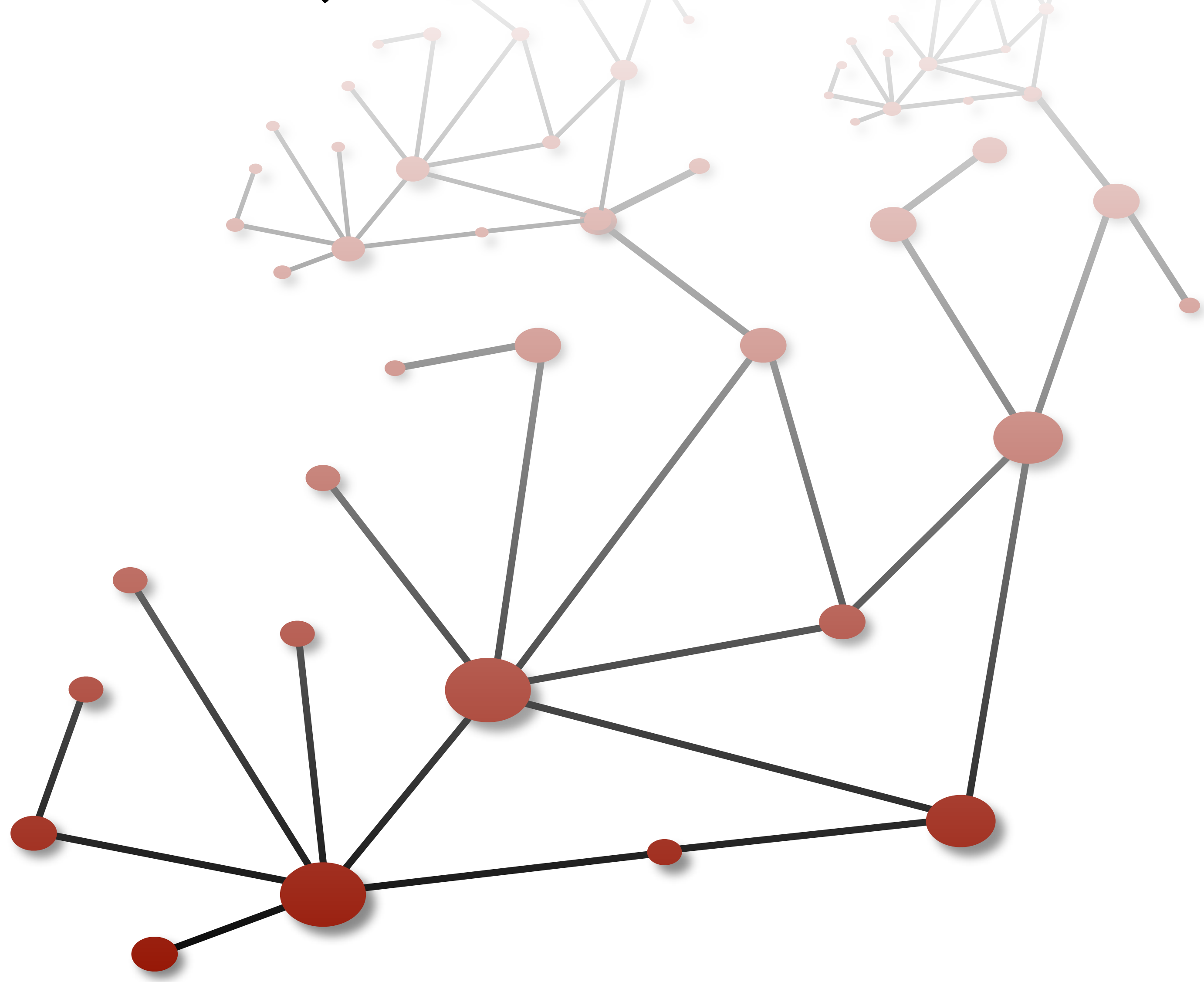
# CS 598: Advanced Internet

## Lecture 2: Project Ideas



Brighten Godfrey

pbg@illinois.edu

Fall 2009

# Announcements

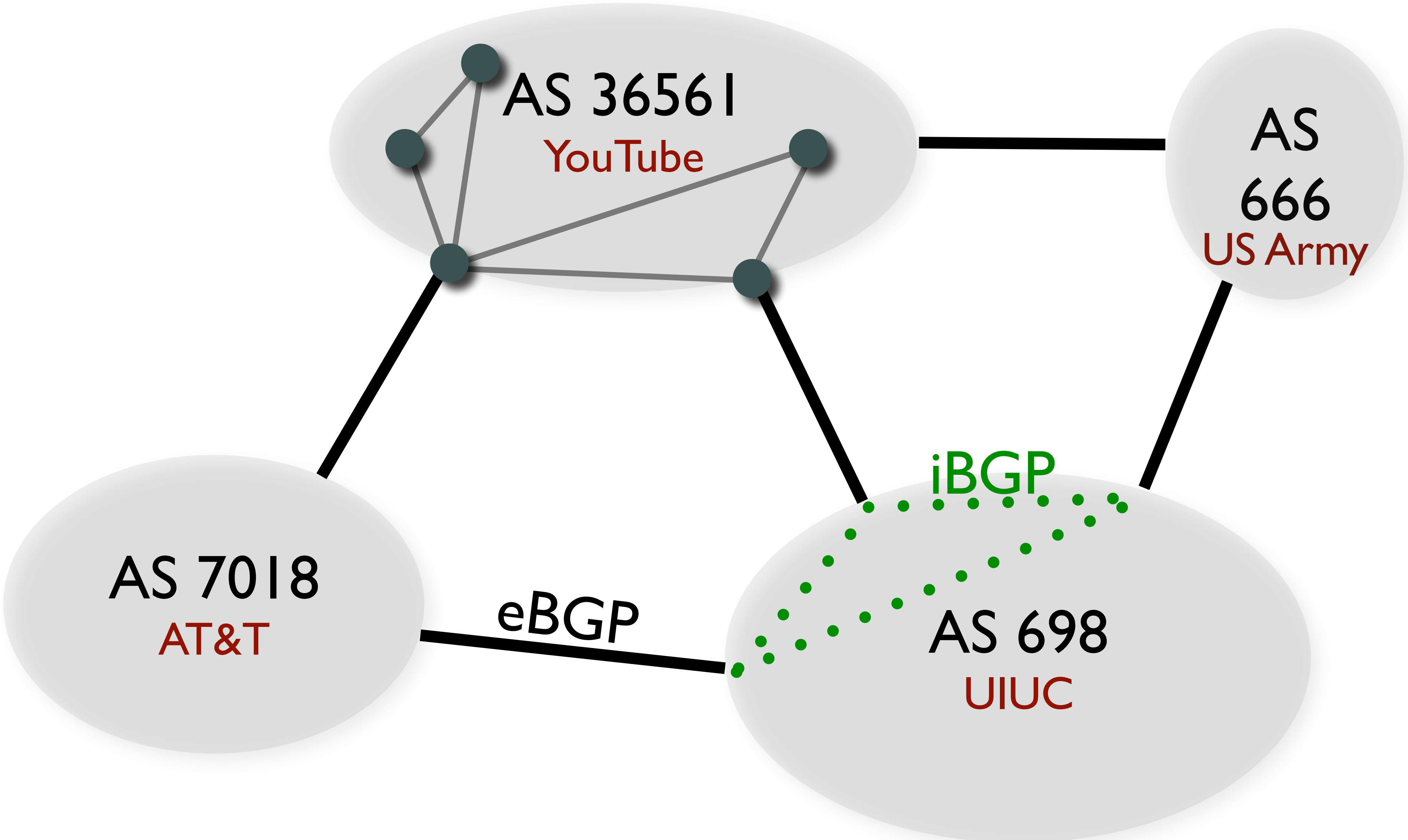
- Reminder: email me your name/email/background
- Slight change in office hours this week: 10-11a.m. Fri (instead of 10:30-11:30)
- Readings on web site (`www.cs.illinois.edu/~pbg/courses/cs598fa09/`)
- Cerf & Kahn, Clark paper reviews due before lecture Tuesday

# Next Thursday's readings

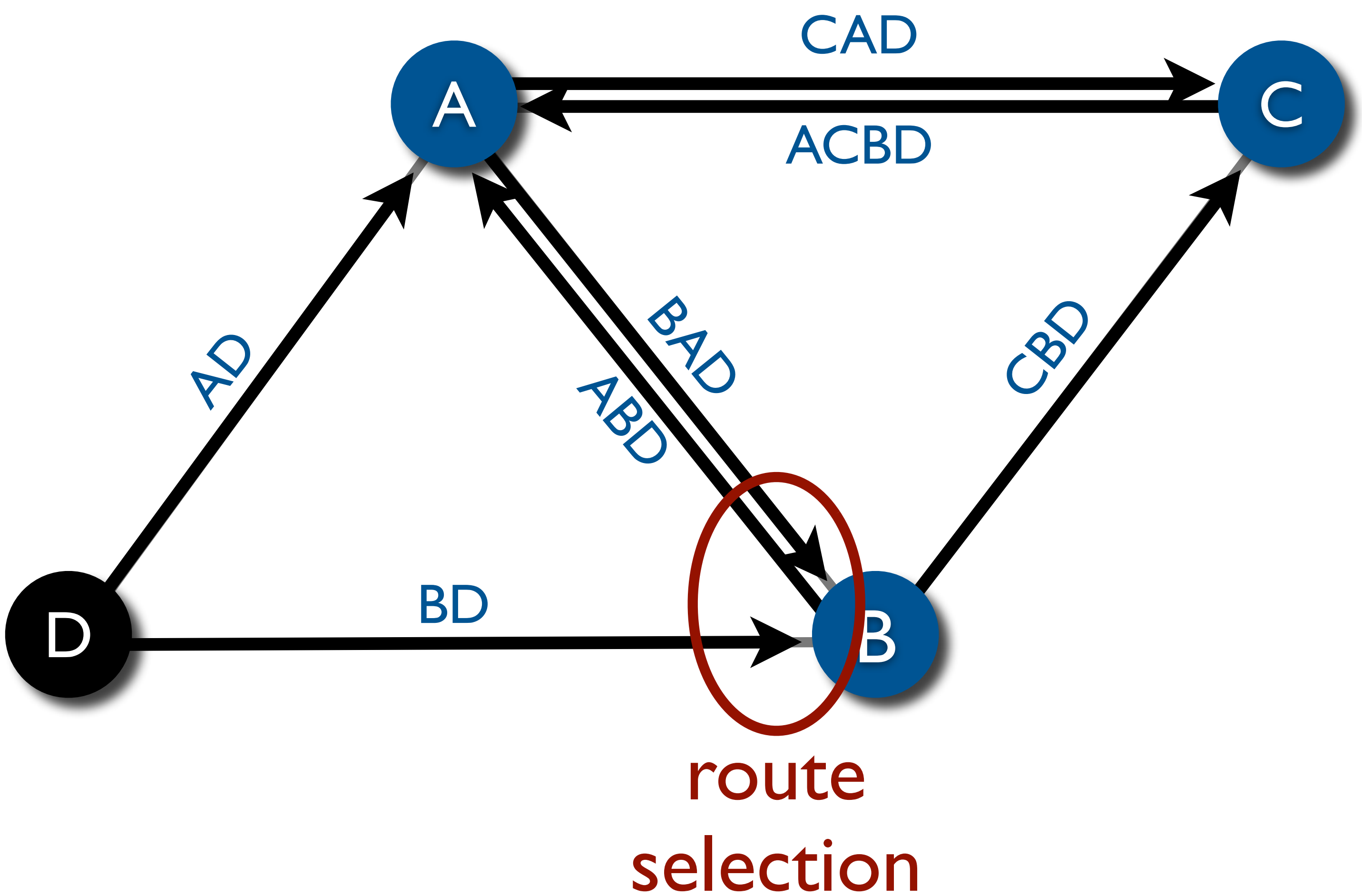
- Jon Postel. Internetwork protocol approaches. IEEE Transactions on Communications, April 1980.
- Saltzer, Reed and Clark, "End-to-End Arguments in System Design," ACM Trans. on Computer Systems, November 1984.
- Two volunteers?

# Abbreviated intro to interdomain routing
## and
## some associated problems

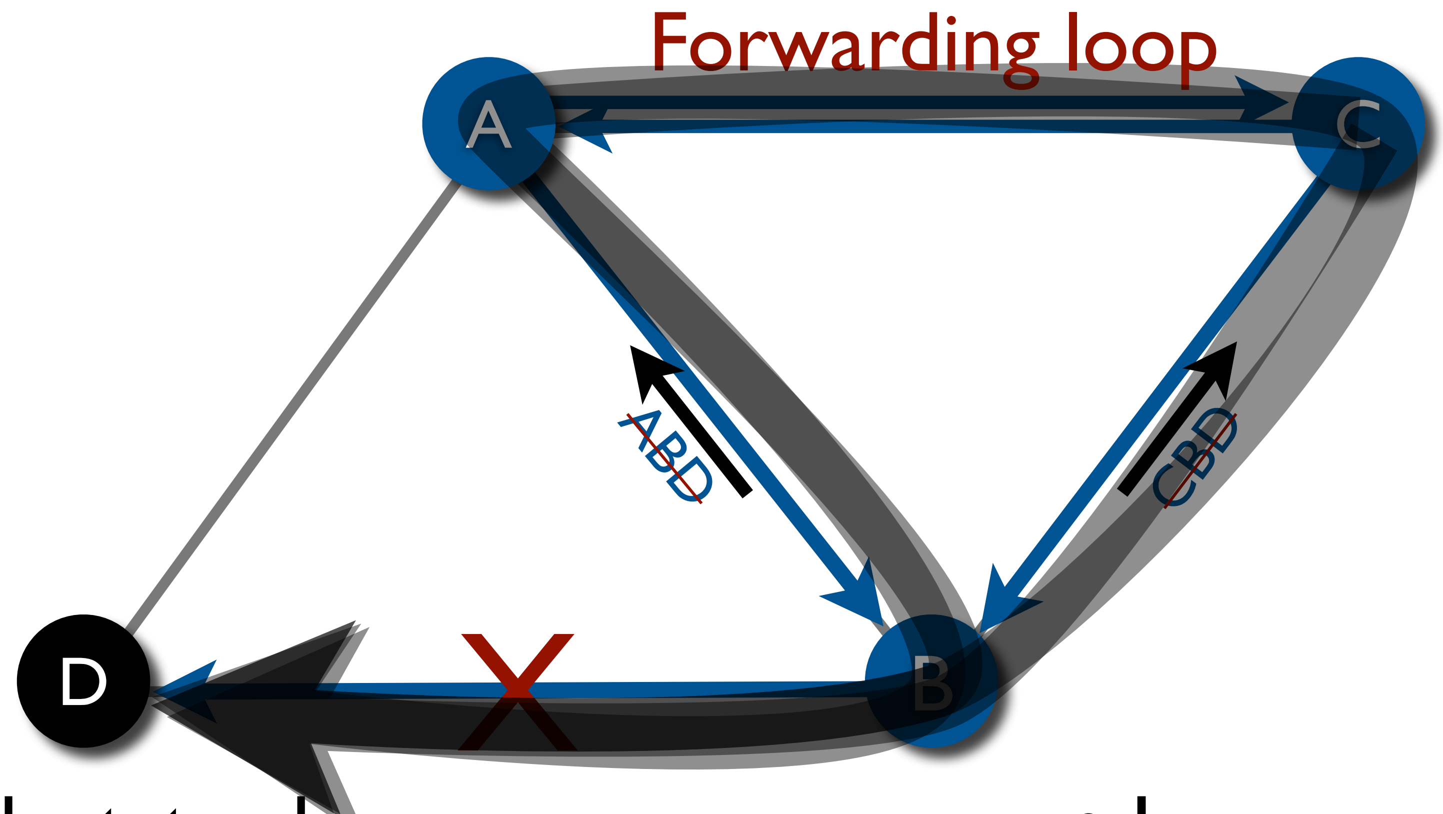
# Internet Routing



# Border Gateway Protocol



# Instability causes outages



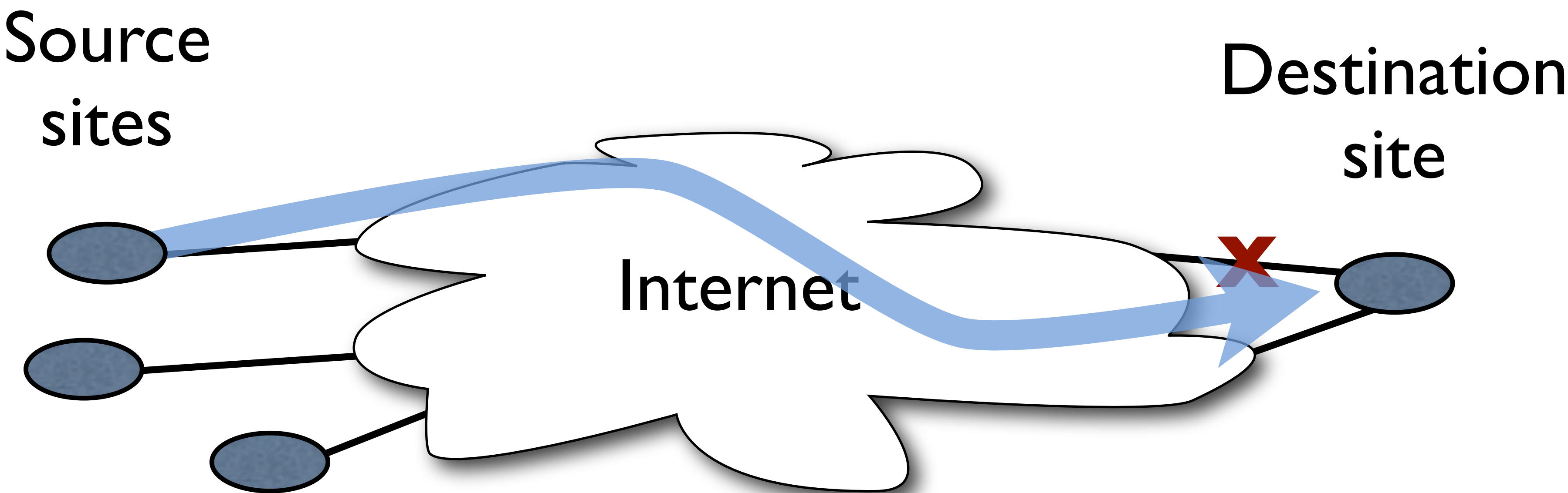

- Link state changes
- Router failures
- Config. changes
- ...

$\Rightarrow$

- Loops
- Detection delay
- Black holes

# Instability causes outages

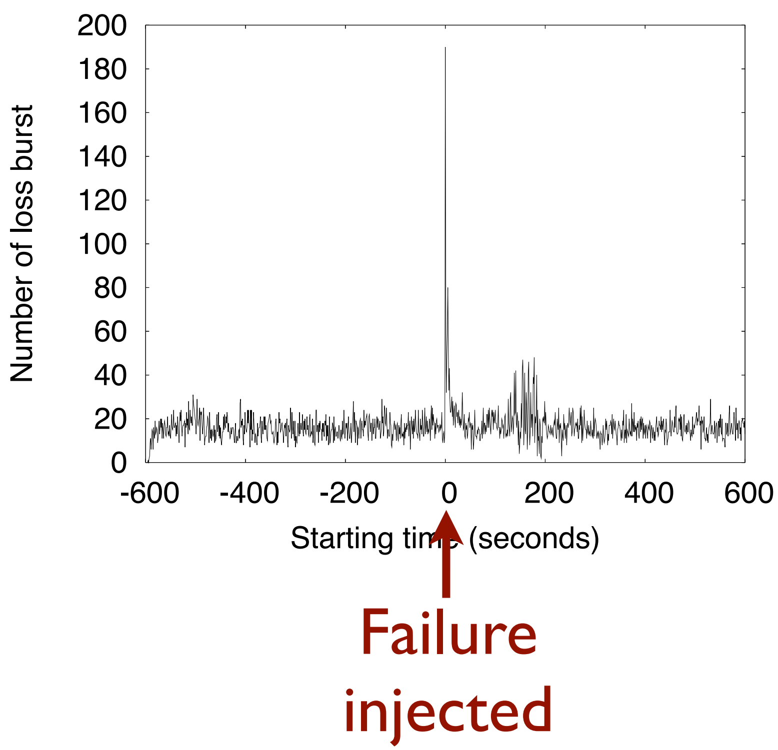
[F. Wang, Z. M. Mao, J. Wang, L. Gao, R. Bush SIGCOMM'06]



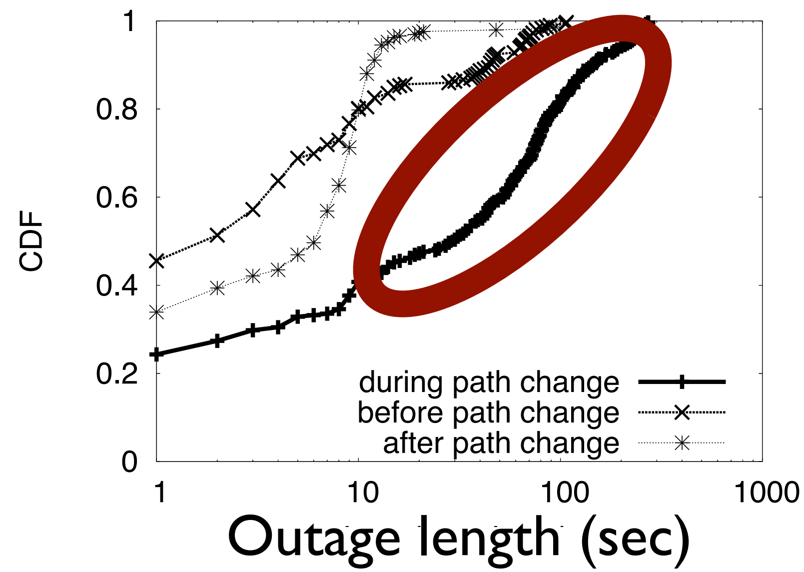
# Instability causes outages

[F. Wang, Z. M. Mao, J. Wang, L. Gao, R. Bush SIGCOMM'06]

## More outages




## Longer outages




...and higher latency, packet reordering, router CPU load during instability.

# Instability affects VoIP

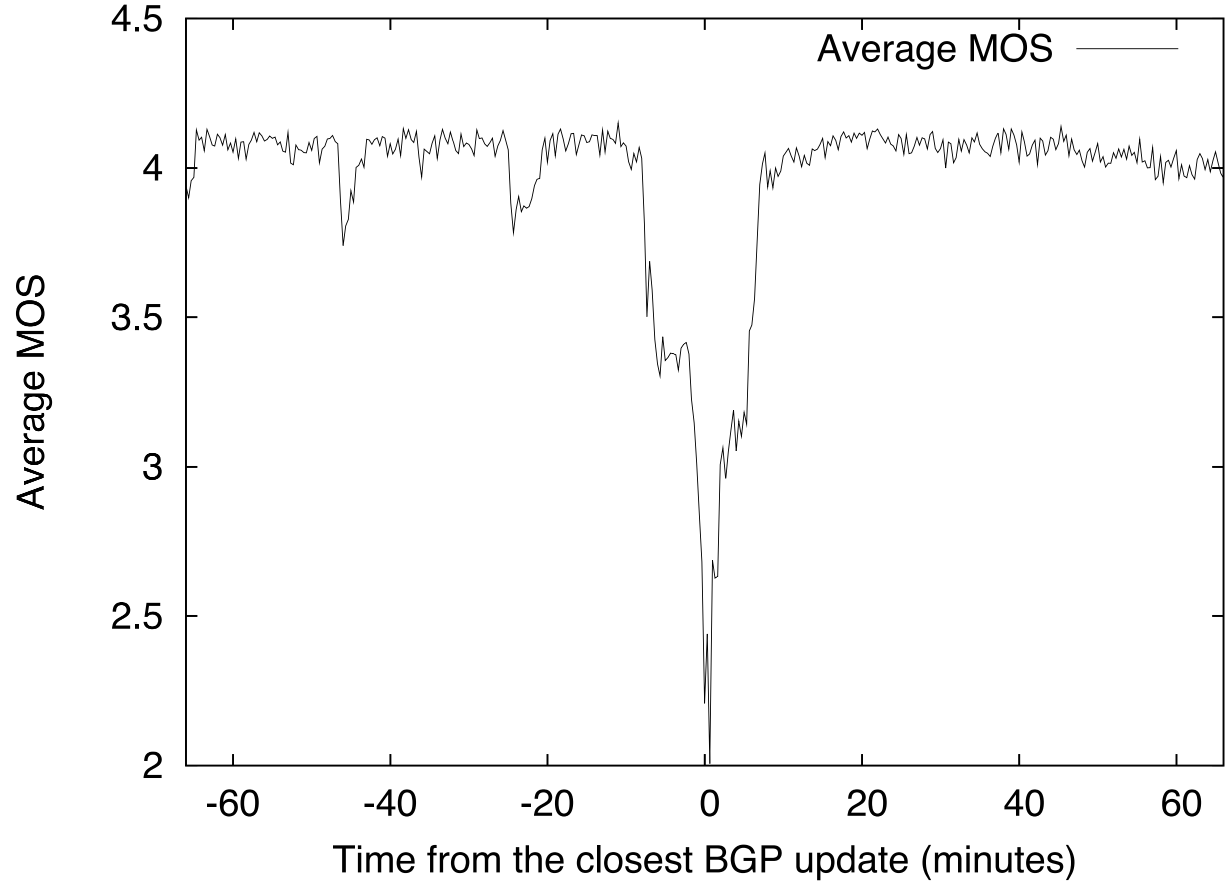
[Kushman, Kandula, Katabi '07]

Toll quality

Cell phone quality

Unacceptable

Unintelligible or outage

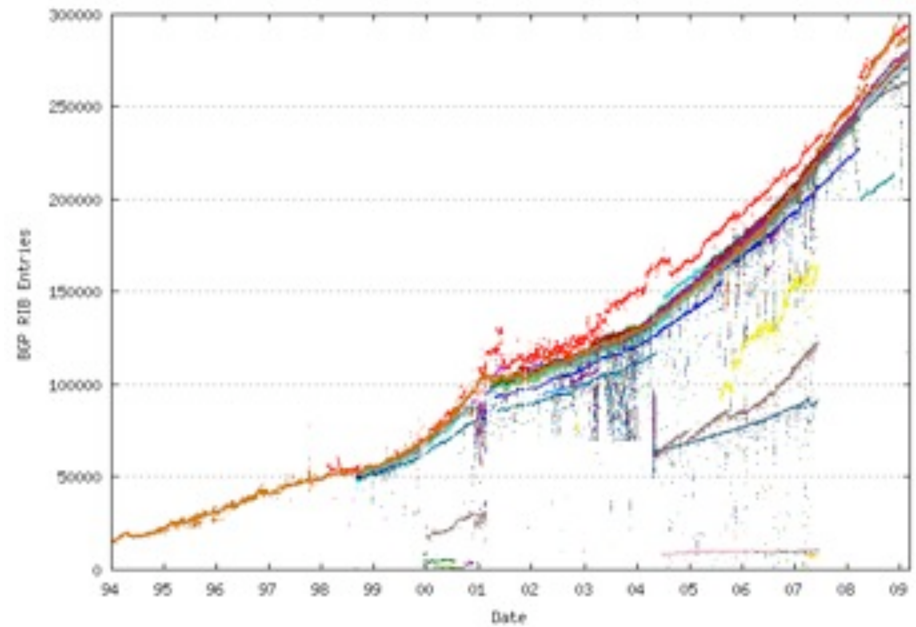



43% within 10 mins of BGP update

# Scaling

- One entry per destination prefix in forwarding table
- In control plane, multiply this by number of neighbors
  - leads to use of route reflectors
- Need to process many (bursty) update messages
- How much of a problem will this be?

Internet forwarding table size vs. time, 1994-2009




[Huston ’09]

# Small range of expressible policies

- You get to pick one path to each destination, from among one path offered from each neighbor
- No multipath
- Difficult to directly express complex policies (e.g., virtual peering)
- Rigid granularity of aggregation: IP prefix

# Lack of extensibility

- One service offered by IP: I will deliver your packet to the designated endhost (somehow).
- A fixed set of IP options are the only way to specify a different kind of service.

# And more...

- Visibility (where is this traffic coming from? What caused a certain problem? ... )
- Mobility
- Security (later in this course)

# Project suggestions

# First a quick overview of pathlet routing

upon which several of the project suggestions are based

# Pathlet Routing

P. Brighten Godfrey
pbg@illinois.edu
Igor Ganichev, Scott Shenker, and Ion Stoica
{igor,shenker,istoica}@cs.berkeley.edu

SIGCOMM 2009

# Pathlet routing

**vnode** virtual node

**pathlet** fragment of a path: a sequence of vnodes

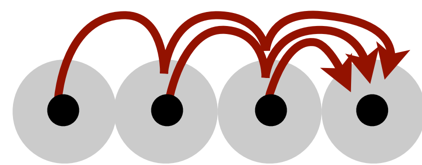
virtual graph: flexible way to define policy constraints

Source routing over pathlets.

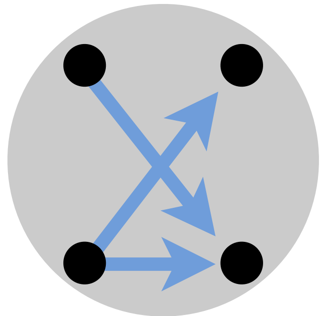
provides many path choices for senders

# Flexibility

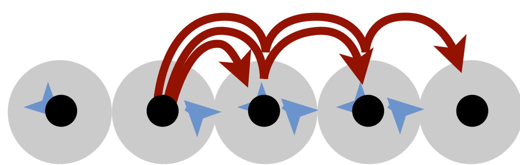
- can emulate BGP, source routing, MIRO, LISP, NIRA



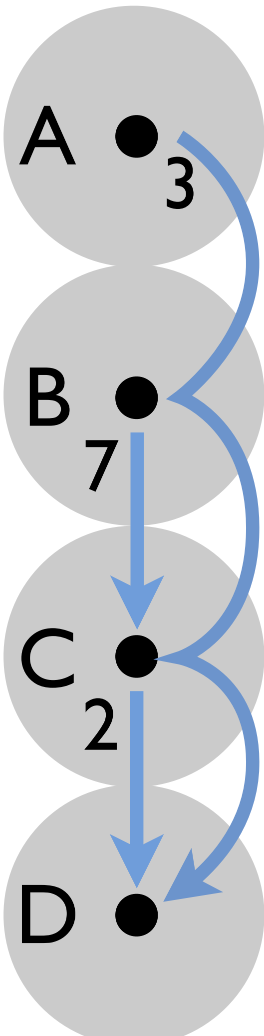
- local transit policies provide multipath and small forwarding tables



- coexistence of different styles of routing policy



# Pathlets




| Packet route field | Forwarding table |
|---|---|
| 3 | |
| 7,2 | |
| 2 | |
| | delivered! |

| | |
|---|---|
| … | … |
| 3 | push 7,2; fwd to B |

| | |
|---|---|
| … | … |
| 7 | fwd to C |

| | |
|---|---|
| … | … |
| 2 | fwd to D |

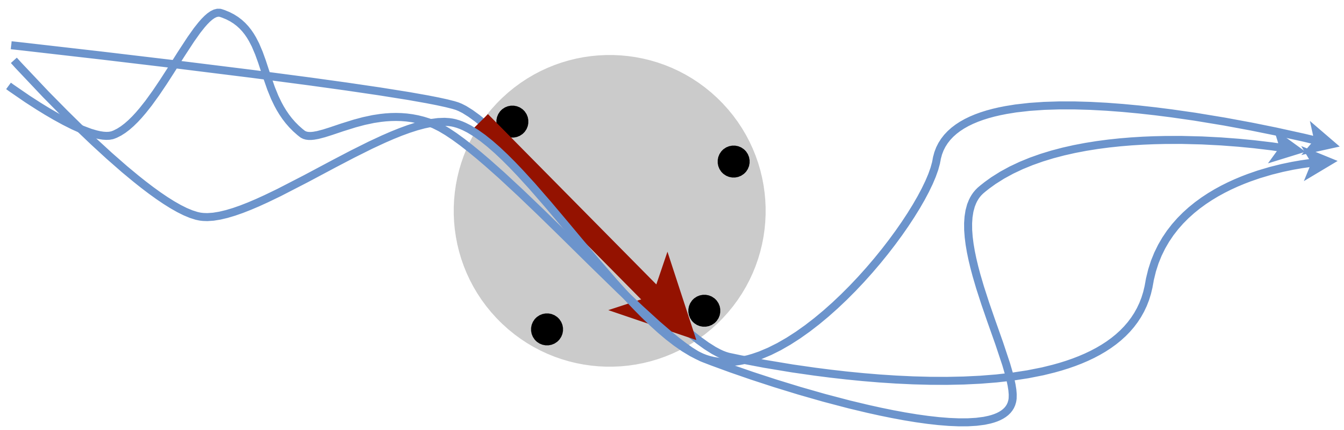
delivered!

# Dissemination

- Global gossip fine, except for scalability
- So, let routers choose not to disseminate some pathlets
- Leads to (ironic) use of path vector — only for pathlet dissemination, not route selection

# Local transit policies

Each ingress ➔ egress pair
is either allowed or disallowed.

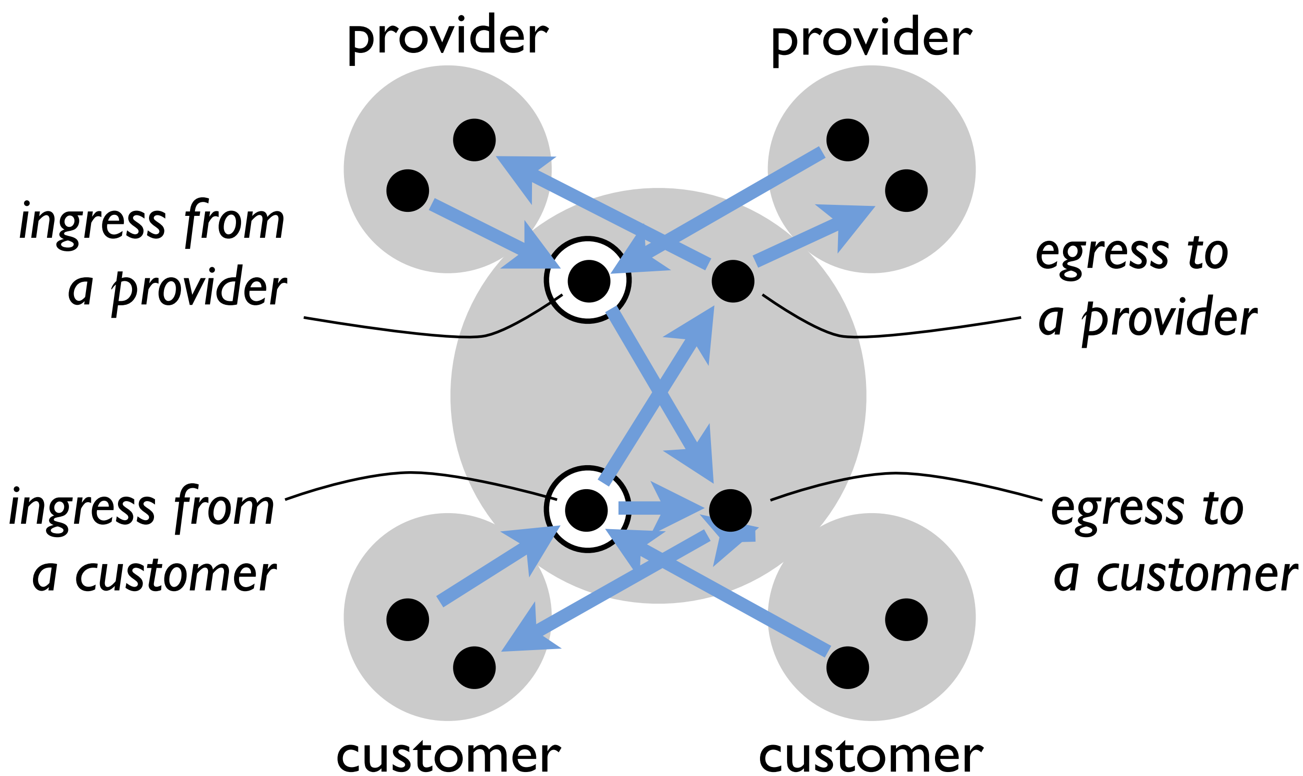


Subject to this, any path allowed!

Represented with few pathlets: small FIB

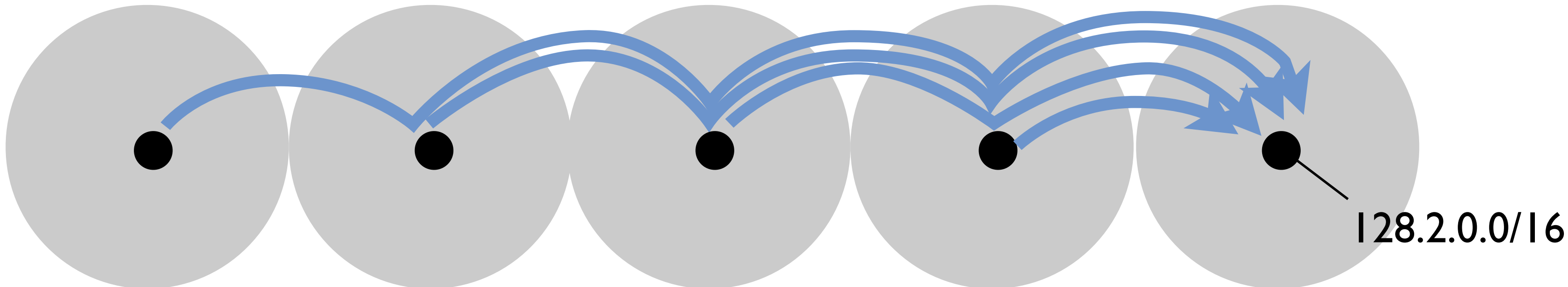
# “All valley-free” is local

“customers can route to anyone; anyone can route to route to customers”



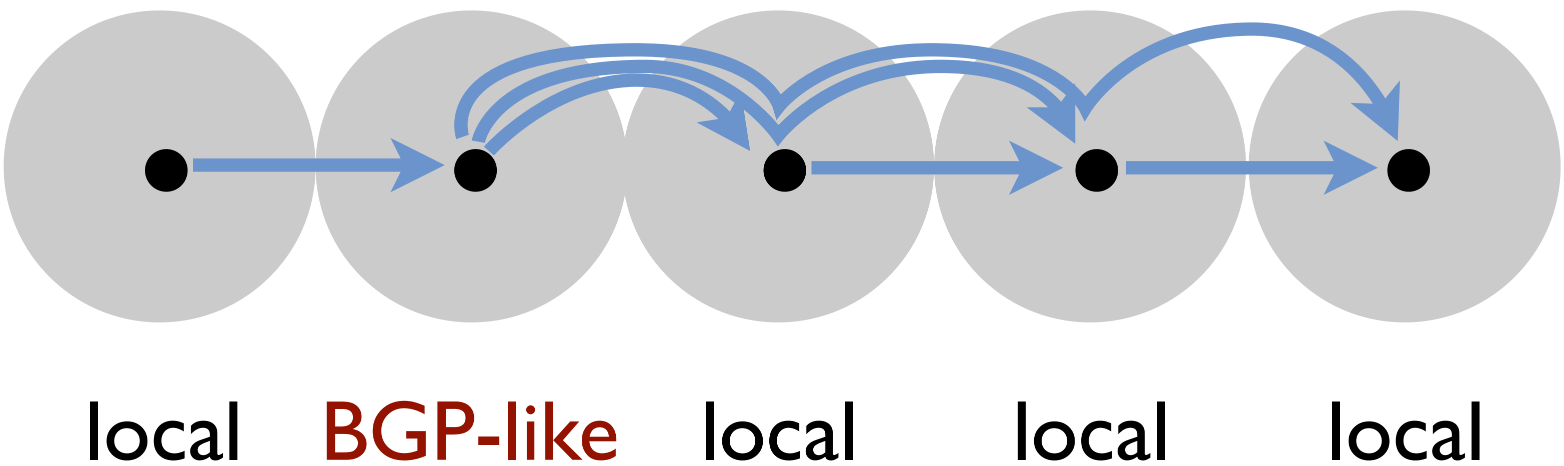

Forwarding table size: 3 + #neighbors

# Emulating BGP




Make this real?

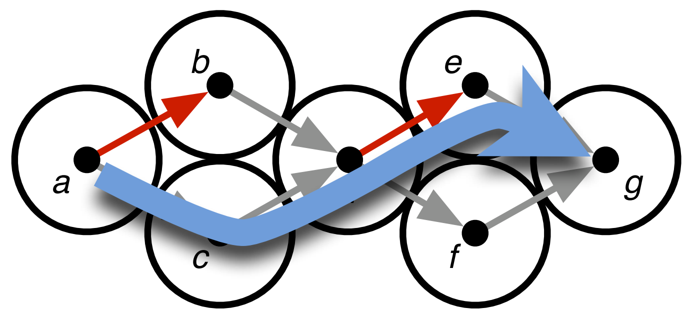
# Mixed policies



# Pathlet-related projects

# Lightweight pathlet dissemination

- Our path vector-based dissemination protocol requires O(DL) control plane state per pathlet, where D = degree and L = mean path length
- Is it possible to reduce this, maybe to O(1)?
- Challenges:
  - Routers must not be required to disseminate all pathlets
  - Tricky multiple-failure case:



# Stability of pathlet routing

- BGP can be unstable due to policy conflicts
- Pathlet routing generalizes BGP, so this can clearly happen
- Can anything *worse* happen? (e.g., maybe destinations become unreachable -- even worse than the control plane not converging.) Can you develop rules to limit the damage to being no worse than BGP?

# Small FIB even with complex policy

- Traditional IP LPM forwarding requires one entry per prefix
- Idea: change packet format to be path, rather than address. Separates forwarding info from policy-checking info.
- Then, check policy on slow path or in more compact way (Bloom filter)?
- Challenge: if you have false positives or only check some traffic, how do you deal with malicious users?

# Per-packet payment

- Pathlet routing lets you use multiple paths
- But why would a network offer multiple paths, beyond the “cheapest” to any destination?
- Several possible answers, but what about if we had a scheme to pay per packet based on the utilized route (rather than by total volume of packets)?
- Design such a system
- Note: this is pretty challenging! (Big security implications, for example)

# Other routing projects

# Route control following payment

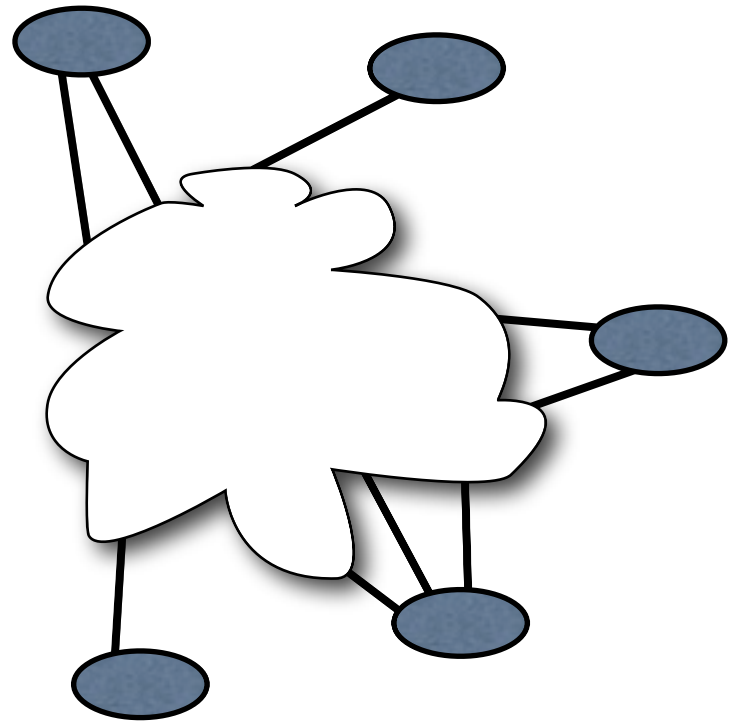
- High level principle (similar to Yang et al's NIRA): if I am paying for part of a packet's path, I should get control over that part of the route
- Design a system which permits this
- E.g., given a spec of where payment is flowing. This determines what portions of rotues different parties can control.

# Multipath with per-destination policy

- Deflection Routing and Path Splicing provide multiple paths, but providers can control which next-hops for each destination
- But for scalability, not explicitly source routed: source can't see path, and PS can encounter loops
- How can you get this policy control but with explicit source routing -- and make it scale?
- Challenge: representing all usable links can take O(n) state per destination in the worst case -- way too much! Need a compact representation, and maybe a tradeoff with how many paths are available to use.

# Scalability of LISP

- LISP (Locator/Identifier Split Protocol) separates routes into a portion crossing the "core" and a final hop to the edge
- Currently working its way through IETF standardization
- Does this fundamentally improve scalability of routing? e.g. in power law graph, are forwarding tables asymptotically smaller? How much smaller in a large set of measured graphs?



# Clean traffic engineering

- Current interdomain traffic engineering is clunky: prefix deaggregation, AS prepending, ...
- Design new architecture which does traffic engineering "cleanly": fine-grained, automatic control over ingress/egress points of inbound and outbound traffic

# Security-related projects

# Suffix and prefix route control

- Given a packet's route (vn, vn-1, ..., v0, x, w1, ..., wn)
- Pathlet routing roughly allows x to control a prefix of what comes after (w1, w2, ...)
- For security, we may want to control a *prefix* of what comes *before* (e.g., v2, v1, v0). I.e., policy specified as whitelists/ blacklists of regexps of the form .*BxA.* where B is a portion of the path required to come before x, and A is a portion required to come after
- Simple to state --- but how do you use it? Given a set of such whitelists / blacklists, how do you compute shortest policy-compliant paths? Can you extend to general regexps?

# Checking forwarding behavior

- Given a network, directly inspect forwarding plane state, in order to answer reachability queries (is there a way to get from A to B? without going through C first?)
- Note this is about checking behavior, not about checking configuration files
- Challenges:
  - state: many possible (input, output) pairs at each box
  - may need to infer what function the forwarding plane is computing

# Random and Weird projects

# No-setup TCP

- Every TCP connection involves a "3-way handshake"
- Incurs latency penalty of one RTT
- Instead, how about sending request (e.g., HTTP GET) immediately with first packet?
- Need to deal somehow with stale connection problems that 3-way handshake was meant to handle
- Implement and measure how much this improves performance in practice

# Latency-optimized replica placement

- You have servers in a bunch of datacenters; they can be quite close or various degrees of very far away.
- If you want fast, dependable storage, how do you optimize it for this scenario?
- e.g., a quorum system needs to contact certain sets of nodes to complete operations. You have some flexibility in what these sets are; how can you pick them best?

# Mine the NANOG list

- NANOG mailing list frequently discusses Internet failures as they happen
- Mine the list archives: What kind of failures are most common? What are the causes?
- Correlate with anomalies in Route Views data (logs of updates from real Internet routers)
- A good launching pad for future research questions

# New projects added Tuesday Sep. 1

# Optimally hierarchical distributed systems

- Distributed systems frequently hierarchical: some set of nodes are picked for greater responsibilities (e.g., content distribution systems, Skype, distributed hash tables)
- Larger set of these “superpeers” brings more capacity (good) but potentially greater overhead and worse service quality (bad!)
- How do you balance these tradeoffs optimally? (e.g., if $n$ superpeers incur $\log(n)$ overhead factor, and you know the distribution of node capacities, what is the optimal set of superpeers?)

# emailfs

- We use email a lot like a filesystem
- Admit it, and design an email system that has the best of both worlds. Compared with email,
  - avoid explicit duplication of content
  - integrated versioning of files?
  - ideas from distributed filesystems to deal with large files?

# Incentive compatibility of congestion control

- What congestion control schemes are both efficient and incentive compatible?
- Intermediate problem: convergence with feedback effects
- Simulate these effects using ns2 or similar packet-level evaluation, working with Brighten and coauthors

# Announcements reminder!

- Reminder: email me your name/email/background
- Slight change in office hours this week: 10-11a.m. (instead of 10:30-11:30)
- Readings on web site (`www.cs.illinois.edu/~pbg/courses/cs598fa09/`)
- Cerf & Kahn, Clark paper reviews due before lecture Tuesday
- See you next week!