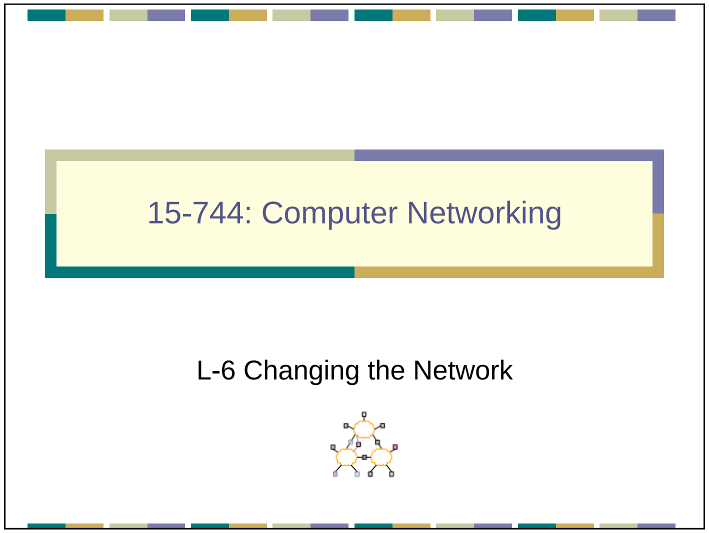
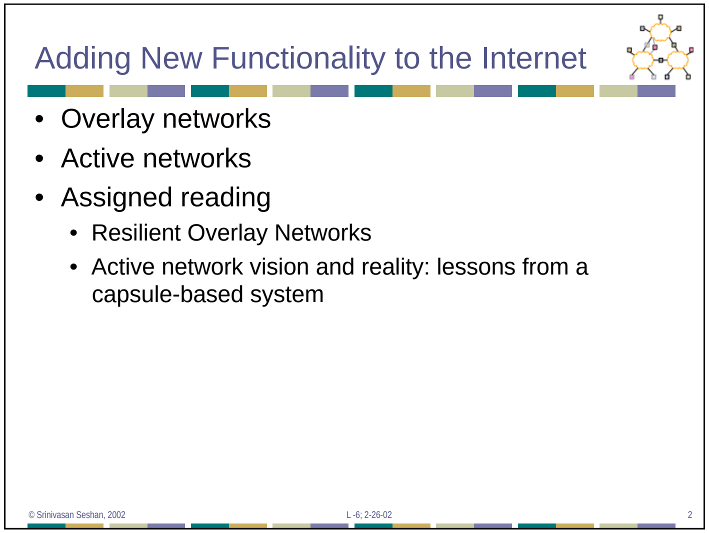


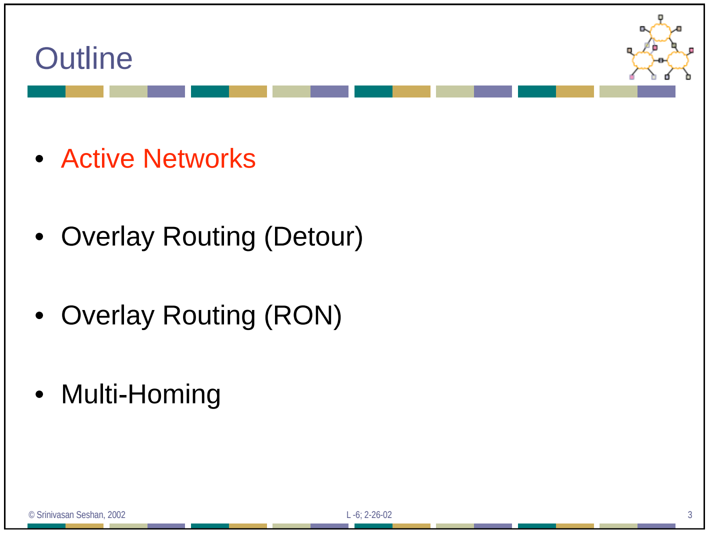
15-744: Computer Networking

L-6 Changing the Network



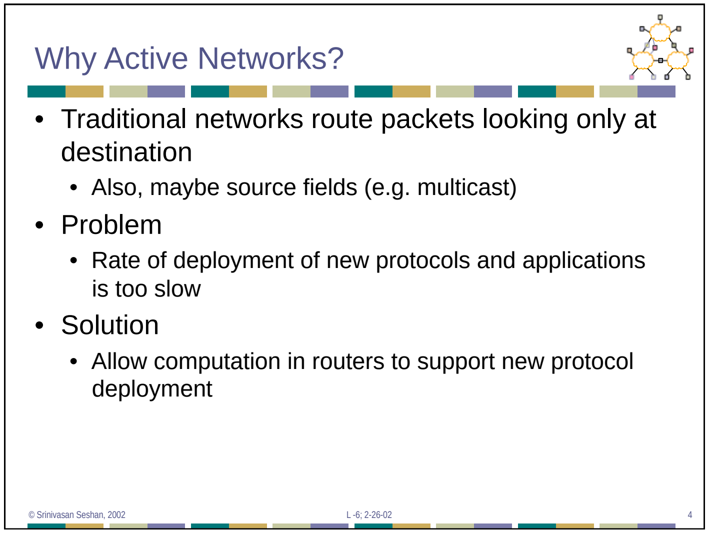
Adding New Functionality to the Internet

- Overlay networks
- Active networks
- Assigned reading
  - Resilient Overlay Networks
  - Active network vision and reality: lessons from a capsule-based system



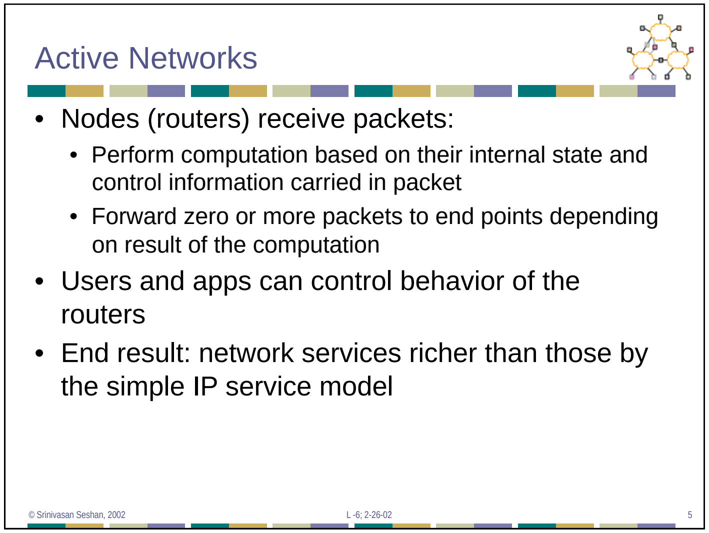
Outline

- Active Networks
- Overlay Routing (Detour)
- Overlay Routing (RON)
- Multi-Homing



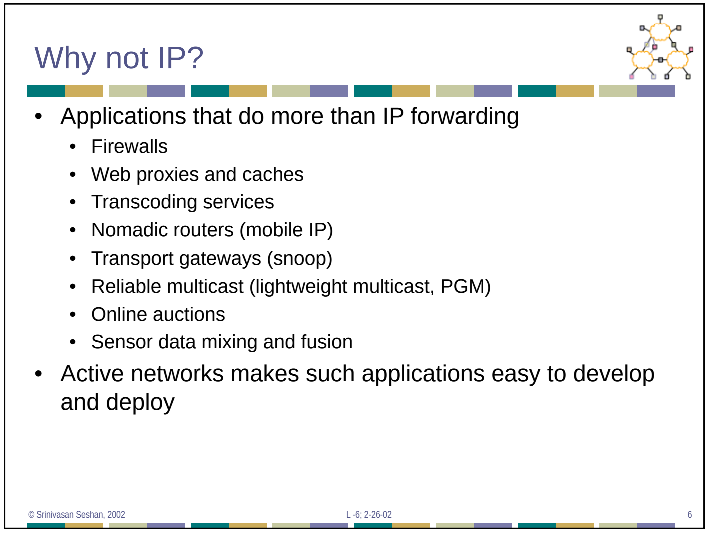
Why Active Networks?

- Traditional networks route packets looking only at destination
  - Also, maybe source fields (e.g. multicast)
- Problem
  - Rate of deployment of new protocols and applications is too slow
- Solution
  - Allow computation in routers to support new protocol deployment



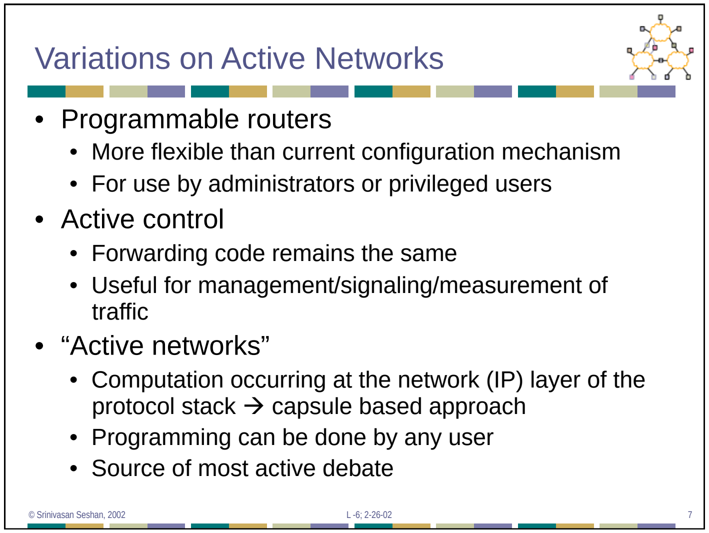
Active Networks

- Nodes (routers) receive packets:
  - Perform computation based on their internal state and control information carried in packet
  - Forward zero or more packets to end points depending on result of the computation
- Users and apps can control behavior of the routers
- End result: network services richer than those by the simple IP service model



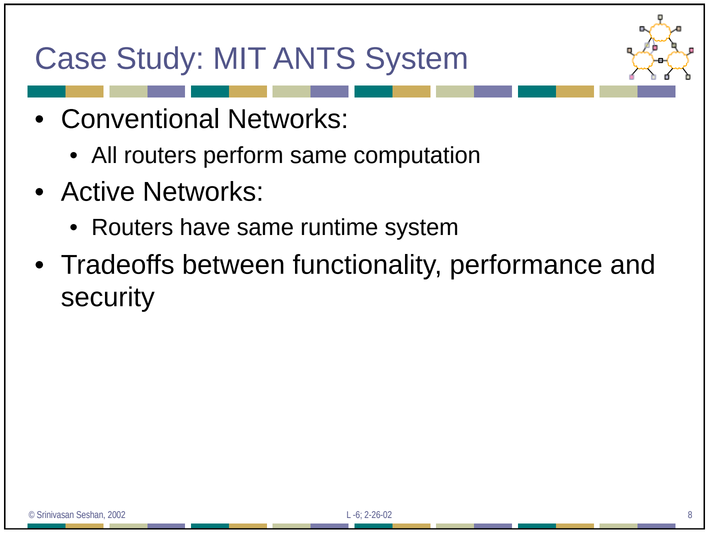
Why not IP?

- Applications that do more than IP forwarding
  - Firewalls
  - Web proxies and caches
  - Transcoding services
  - Nomadic routers (mobile IP)
  - Transport gateways (snoop)
  - Reliable multicast (lightweight multicast, PGM)
  - Online auctions
  - Sensor data mixing and fusion
- Active networks makes such applications easy to develop and deploy



Variations on Active Networks

- Programmable routers
  - More flexible than current configuration mechanism
  - For use by administrators or privileged users
- Active control
  - Forwarding code remains the same
  - Useful for management/signaling/measurement of traffic
- “Active networks”
  - Computation occurring at the network (IP) layer of the protocol stack → capsule based approach
  - Programming can be done by any user
  - Source of most active debate



Case Study: MIT ANTS System

- Conventional Networks:
  - All routers perform same computation
- Active Networks:
  - Routers have same runtime system
- Tradeoffs between functionality, performance and security

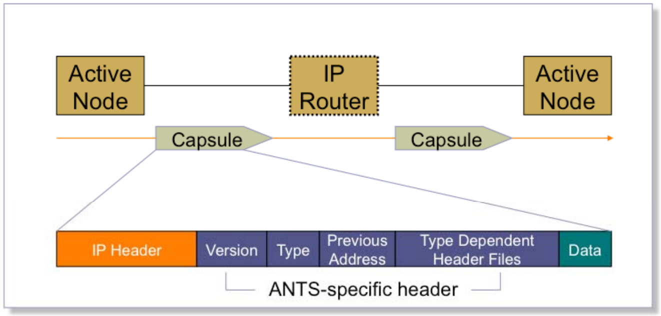
System Components

- Capsules
- Active Nodes:
  - Execute capsules of protocol and maintain protocol state
  - Provide capsule execution API and safety using OS/ language techniques
- Code Distribution Mechanism
  - Ensure capsule processing routines automatically/ dynamically transfer to node as needed

Capsules

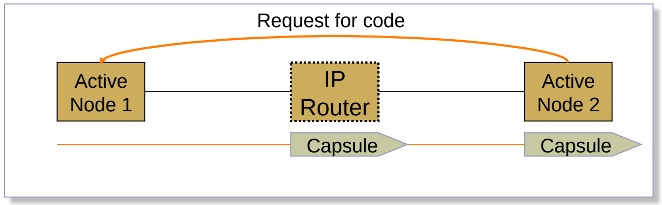
- Each user/flow programs router to handle its own packets
  - Code sent along with packets
  - Code sent by reference
- Protocol:
  - Capsules that share the same processing code
- May share state in the network
- Capsule ID (i.e. name) is MD5 of code

Capsules

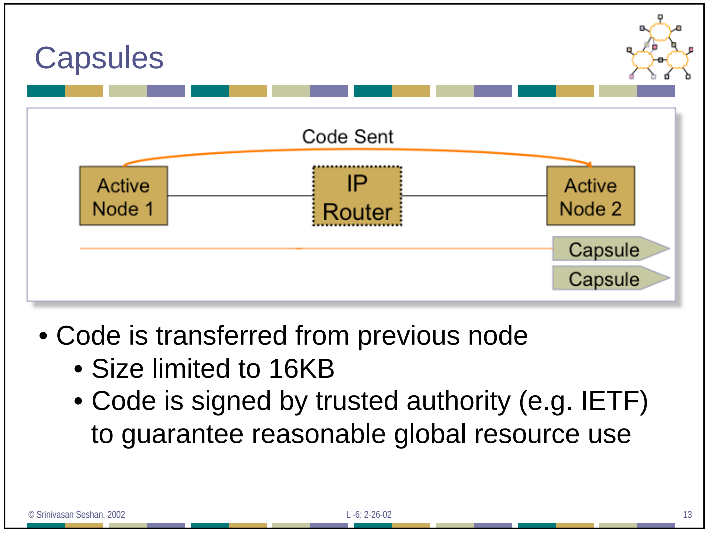


- Capsules are forwarded past normal IP routers

Capsules

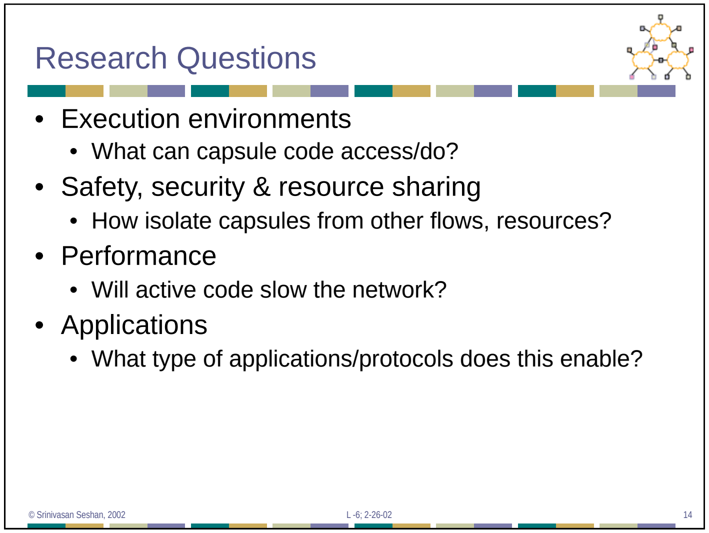


- When node receives capsule uses “type” to determine code to run
- What if no such code at node?
  - Requests code from “previous address” node
  - Likely to have code since it was recently used



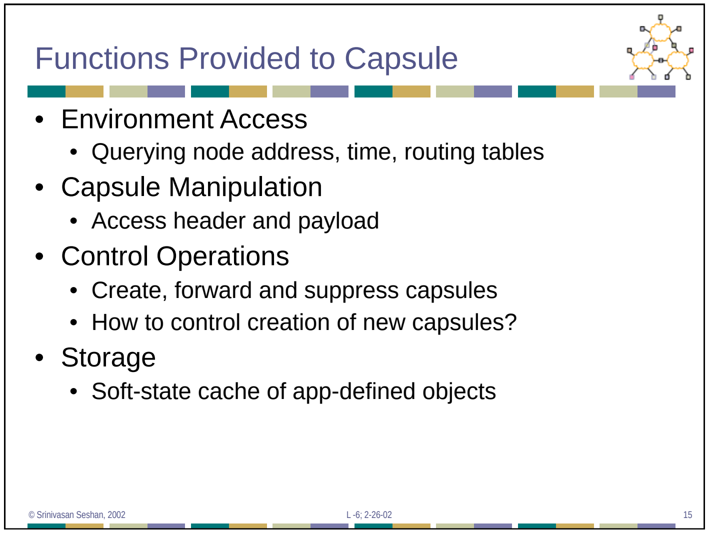
Capsules

- Code is transferred from previous node
  - Size limited to 16KB
  - Code is signed by trusted authority (e.g. IETF) to guarantee reasonable global resource use



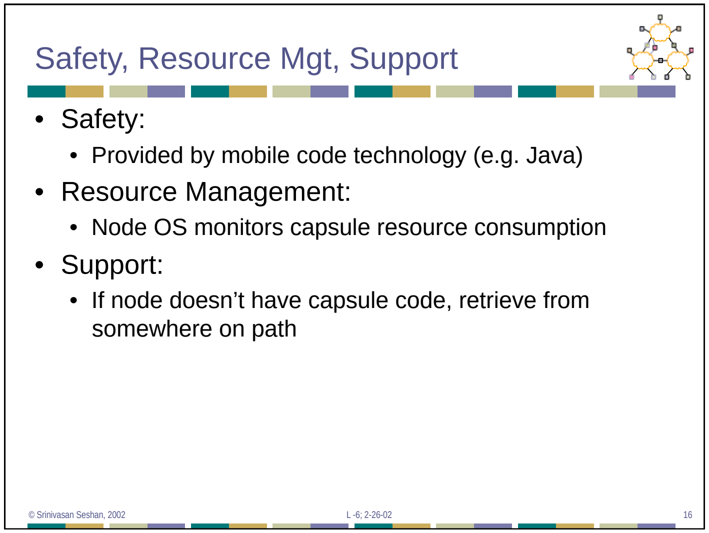
Research Questions

- Execution environments
  - What can capsule code access/do?
- Safety, security & resource sharing
  - How isolate capsules from other flows, resources?
- Performance
  - Will active code slow the network?
- Applications
  - What type of applications/protocols does this enable?



Functions Provided to Capsule

- Environment Access
  - Querying node address, time, routing tables
- Capsule Manipulation
  - Access header and payload
- Control Operations
  - Create, forward and suppress capsules
  - How to control creation of new capsules?
- Storage
  - Soft-state cache of app-defined objects



Safety, Resource Mgt, Support

- Safety:
  - Provided by mobile code technology (e.g. Java)
- Resource Management:
  - Node OS monitors capsule resource consumption
- Support:
  - If node doesn’t have capsule code, retrieve from somewhere on path

Applications/Protocols

- Limitations
  - Expressible → limited by execution environment
  - Compact → less than 16KB
  - Fast → aborted if slower than forwarding rate
  - Incremental → not all nodes will be active
- Proof by example
  - Host mobility, multicast, path MTU, Web cache routing, etc.

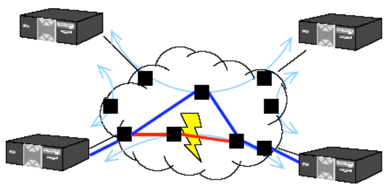
Discussion

- Active nodes present lots of applications with a desirable architecture
- Key questions
  - Is all this necessary at the forwarding level of the network?
  - Is ease of deploying new apps/services and protocols a reality?

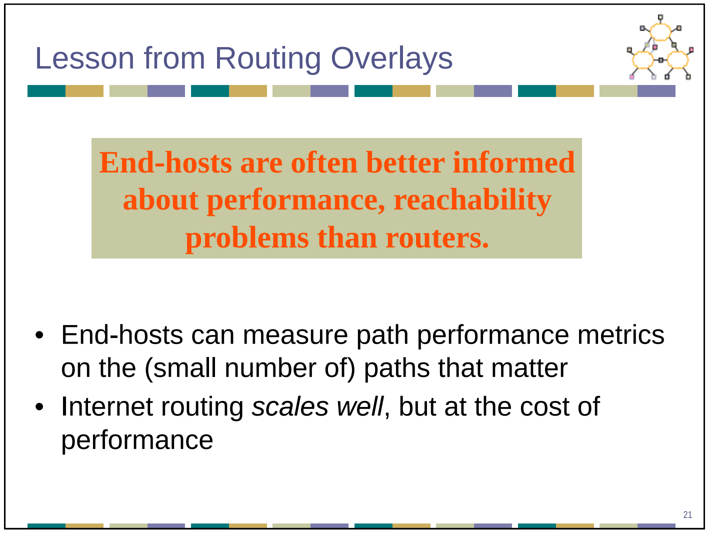
Outline

- Active Networks
- Overlay Routing (Detour)
- Overlay Routing (RON)
- Multi-Homing

The Internet Ideal



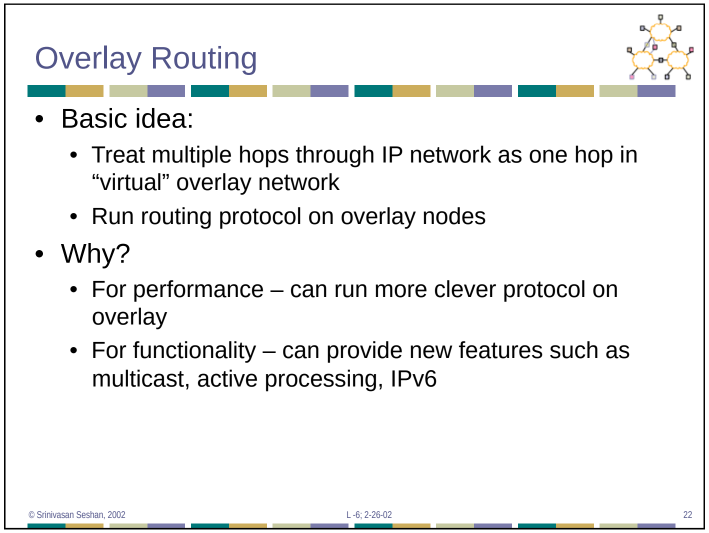
- Dynamic routing routes around failures
- End-user is none the wiser



Lesson from Routing Overlays

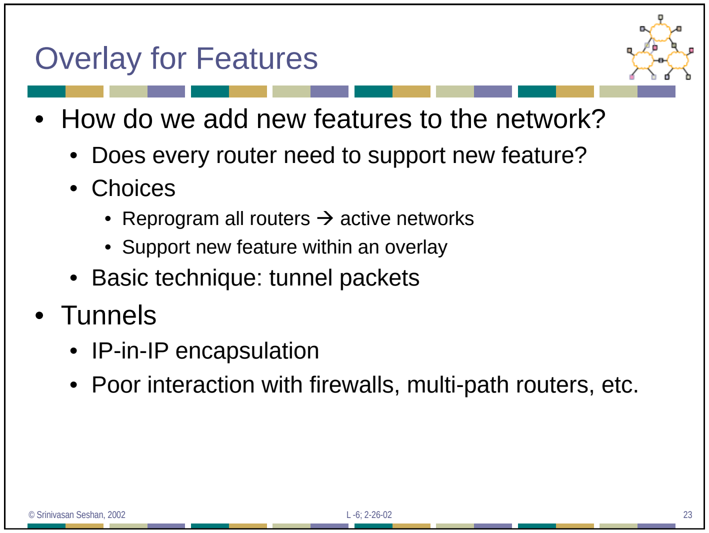
End-hosts are often better informed about performance, reachability problems than routers.

- End-hosts can measure path performance metrics on the (small number of) paths that matter
- Internet routing *scales well*, but at the cost of performance



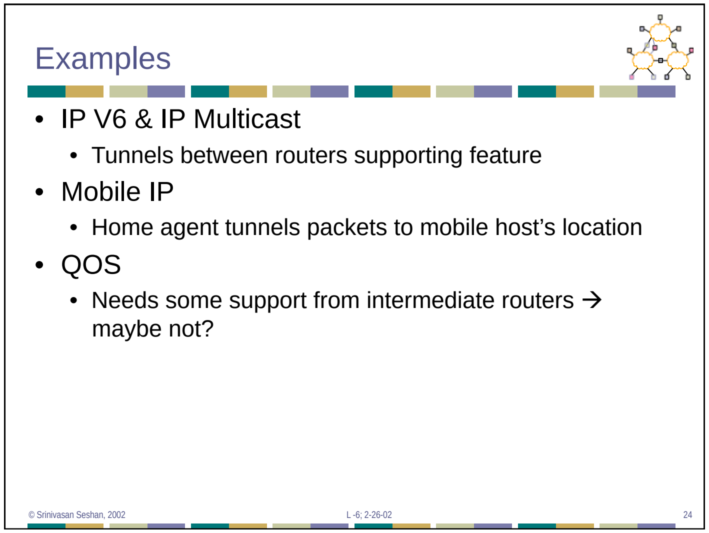
Overlay Routing

- Basic idea:
  - Treat multiple hops through IP network as one hop in “virtual” overlay network
  - Run routing protocol on overlay nodes
- Why?
  - For performance – can run more clever protocol on overlay
  - For functionality – can provide new features such as multicast, active processing, IPv6



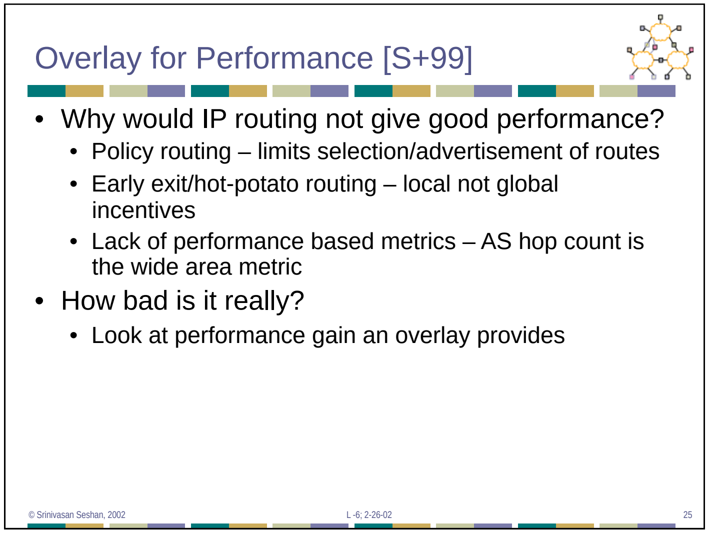
Overlay for Features

- How do we add new features to the network?
  - Does every router need to support new feature?
  - Choices
    - Reprogram all routers → active networks
    - Support new feature within an overlay
  - Basic technique: tunnel packets
- Tunnels
  - IP-in-IP encapsulation
  - Poor interaction with firewalls, multi-path routers, etc.



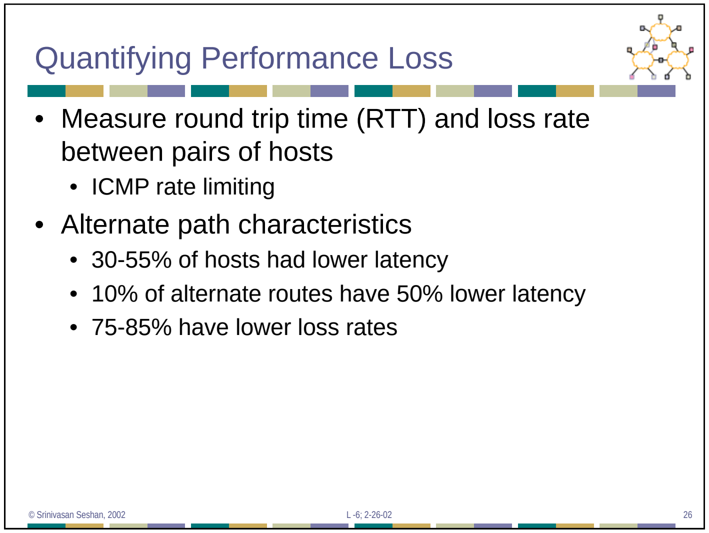
Examples

- IP V6 & IP Multicast
  - Tunnels between routers supporting feature
- Mobile IP
  - Home agent tunnels packets to mobile host’s location
- QOS
  - Needs some support from intermediate routers → maybe not?



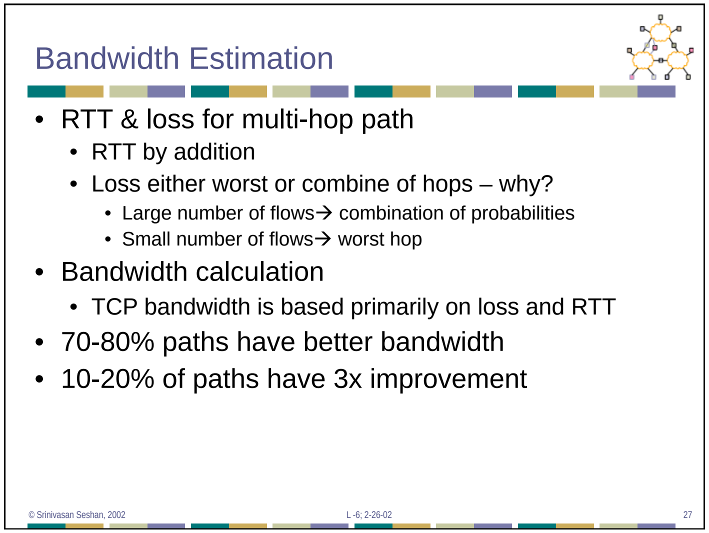
Overlay for Performance [S+99]

- Why would IP routing not give good performance?
  - Policy routing – limits selection/advertisement of routes
  - Early exit/hot-potato routing – local not global incentives
  - Lack of performance based metrics – AS hop count is the wide area metric
- How bad is it really?
  - Look at performance gain an overlay provides



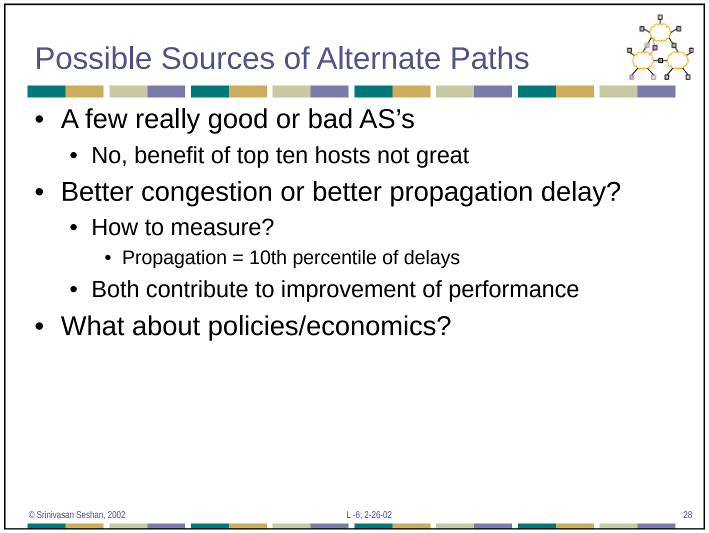
Quantifying Performance Loss

- Measure round trip time (RTT) and loss rate between pairs of hosts
  - ICMP rate limiting
- Alternate path characteristics
  - 30-55% of hosts had lower latency
  - 10% of alternate routes have 50% lower latency
  - 75-85% have lower loss rates



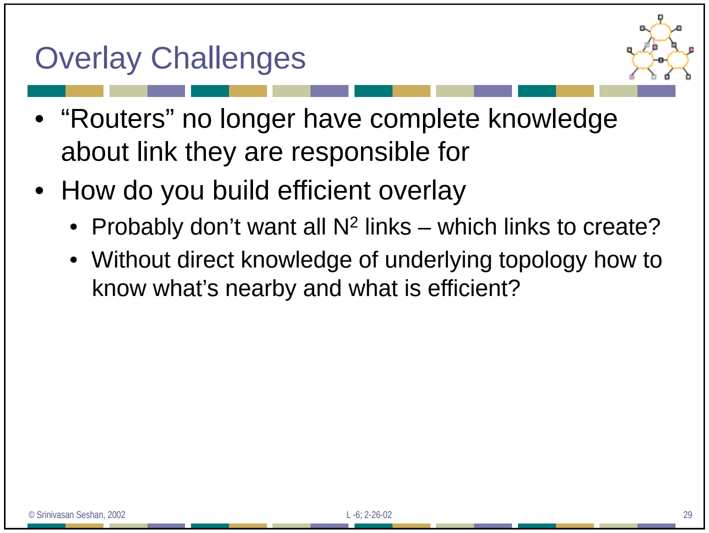
Bandwidth Estimation

- RTT & loss for multi-hop path
  - RTT by addition
  - Loss either worst or combine of hops – why?
    - Large number of flows→ combination of probabilities
    - Small number of flows→ worst hop
- Bandwidth calculation
  - TCP bandwidth is based primarily on loss and RTT
- 70-80% paths have better bandwidth
- 10-20% of paths have 3x improvement



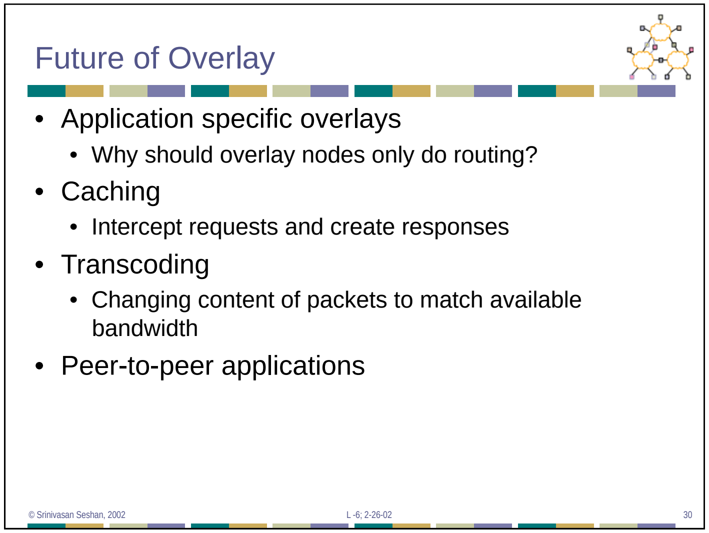
Possible Sources of Alternate Paths

- A few really good or bad AS's
  - No, benefit of top ten hosts not great
- Better congestion or better propagation delay?
  - How to measure?
    - Propagation = 10th percentile of delays
  - Both contribute to improvement of performance
- What about policies/economics?



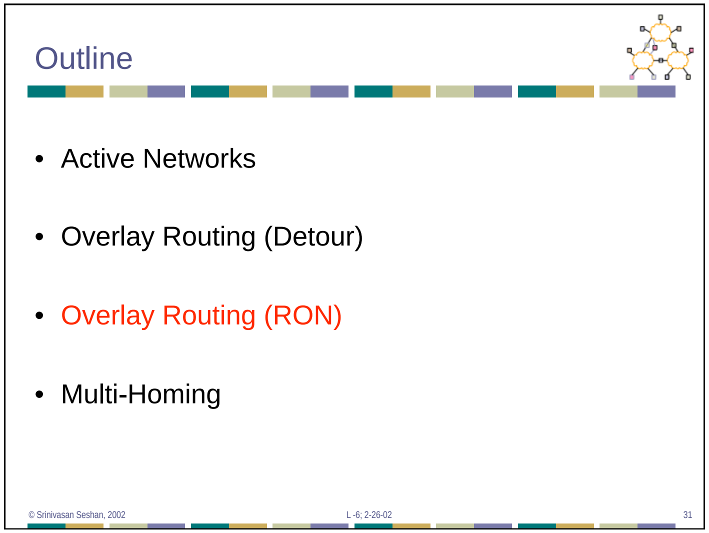
Overlay Challenges

- "Routers" no longer have complete knowledge about link they are responsible for
- How do you build efficient overlay
  - Probably don't want all N^2 links – which links to create?
  - Without direct knowledge of underlying topology how to know what's nearby and what is efficient?



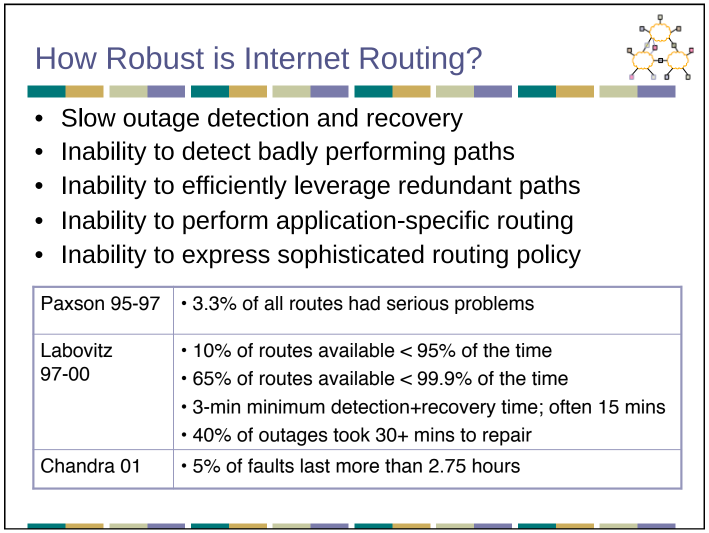
Future of Overlay

- Application specific overlays
  - Why should overlay nodes only do routing?
- Caching
  - Intercept requests and create responses
- Transcoding
  - Changing content of packets to match available bandwidth
- Peer-to-peer applications



Outline

- Active Networks
- Overlay Routing (Detour)
- Overlay Routing (RON)
- Multi-Homing



How Robust is Internet Routing?

- Slow outage detection and recovery
- Inability to detect badly performing paths
- Inability to efficiently leverage redundant paths
- Inability to perform application-specific routing
- Inability to express sophisticated routing policy

Paxson 95-97	• 3.3% of all routes had serious problems
Labovitz 97-00	• 10% of routes available < 95% of the time • 65% of routes available < 99.9% of the time • 3-min minimum detection+recovery time; often 15 mins • 40% of outages took 30+ mins to repair
Chandra 01	• 5% of faults last more than 2.75 hours

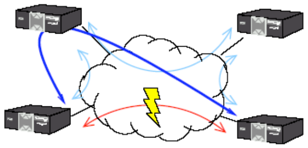
Routing Convergence in Practice

Time	Prefix	Type	AS Path	Localpref MED	Community
2005/11/01 00:06:23	195.78.38.0/23	A	174 5400 20703 28773		174:21100 16631:1000
2005/11/01 00:06:39	195.78.38.0/23	A	3356 5400 20703 28773		3356:2 3356:100 3356:123 3356:500 3356:2064 5400:46
2005/11/01 00:06:45	195.78.38.0/23	W			

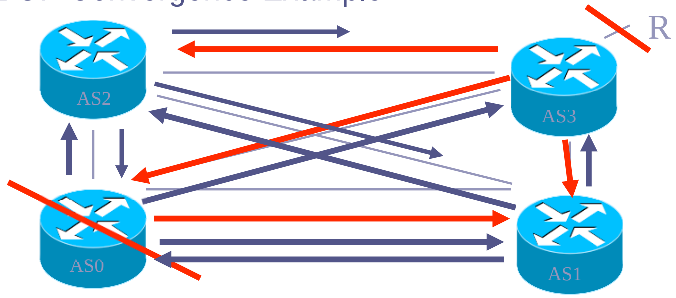
- Route withdrawn, but stub cycles through backup path…

Resilient Overlay Networks: Goal

- Increase reliability of communication for a small (i.e., < 50 nodes) set of connected hosts
- Main idea: End hosts discover network-level path failure and cooperate to re-route.



BGP Convergence Example



~~*B R via AS3~~
~~B R via AS1,AS3~~
~~B R via AS2.AS3~~
AS0

~~*B R via AS3~~
~~*B R via AS0,AS3~~
*B R via 203
AS1

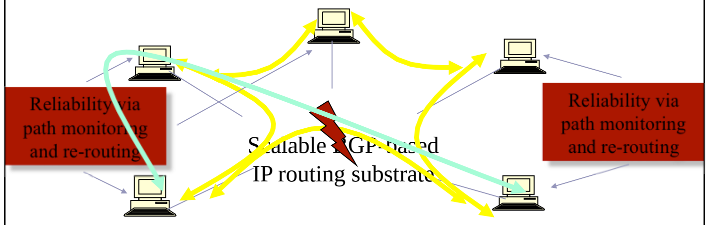
~~*B R via AS3~~
*B R via 013
AS2

The RON Architecture

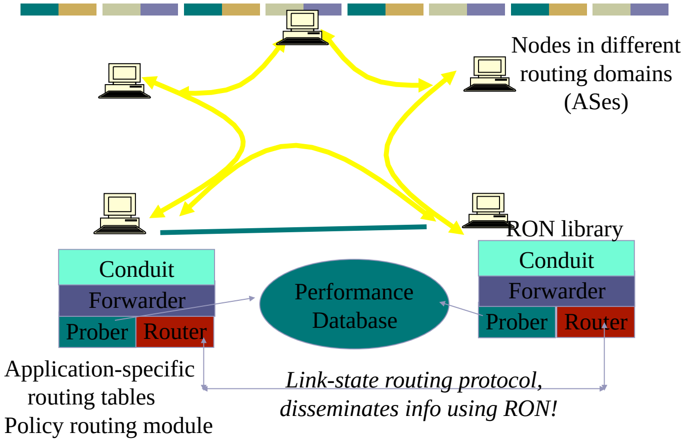
- Outage detection
  - Active UDP-based probing
    - Uniform random in [0,14]
    - O(n^2)
  - 3-way probe
    - Both sides get RTT information
    - Store latency and loss-rate information in DB
- Routing protocol: Link-state between overlay nodes
- Policy: restrict some paths from hosts
  - E.g., don’t use Internet2 hosts to improve non-Internet2 paths

RON: Routing Using Overlays

- Cooperating end-systems in different routing domains can conspire to do better than scalable wide-area protocols
- Types of failures
  - Outages: Configuration/op errors, software errors, backhoes, etc.
  - Performance failures: Severe congestion, DoS attacks, etc.

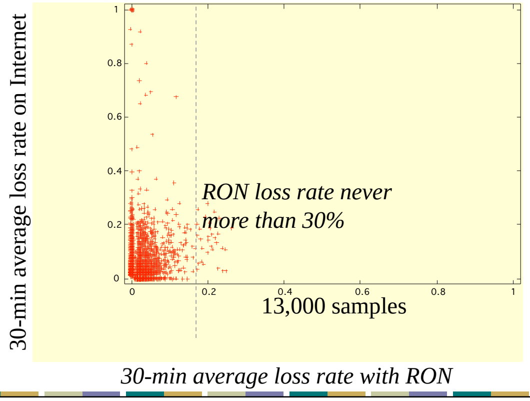

Reliability via path monitoring and re-routing
Scalable BGP-based IP routing substrate
Reliability via path monitoring and re-routing

RON Design



Link-state routing protocol, disseminates info using RON!

RON greatly improves loss-rate



RON loss rate never more than 30%

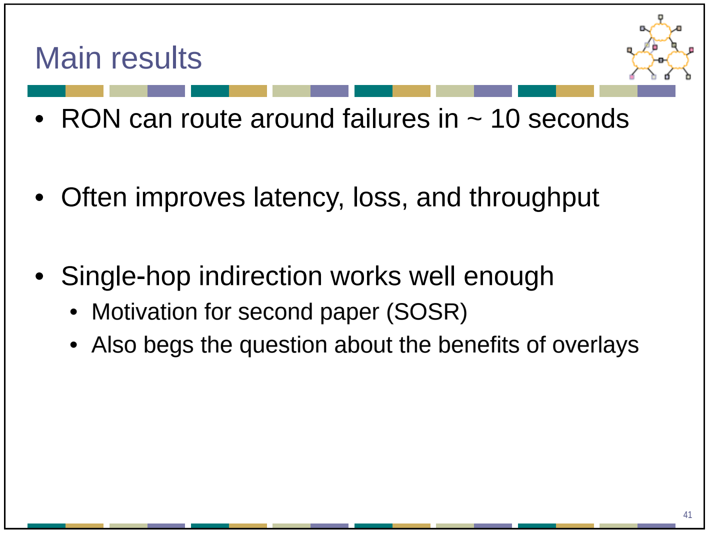
13,000 samples

An order-of-magnitude fewer failures

30-minute average loss rates

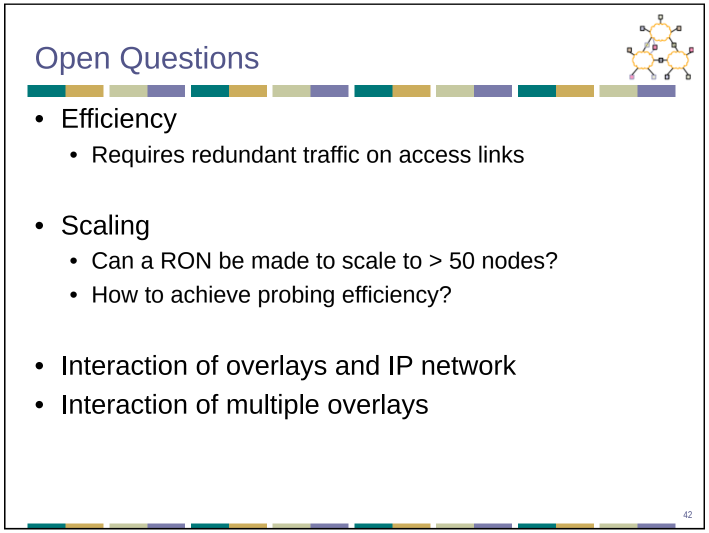
Loss Rate	RON Better	No Change	RON Worse
10%	479	57	47
20%	127	4	15
30%	32	0	**0**
50%	20	0	**0**
80%	14	0	**0**
100%	10	0	**0**

6,825 "path hours" represented here
12 "path hours" of essentially complete outage
76 "path hours" of TCP outage
RON routed around all of these!
One indirection hop provides almost all the benefit!



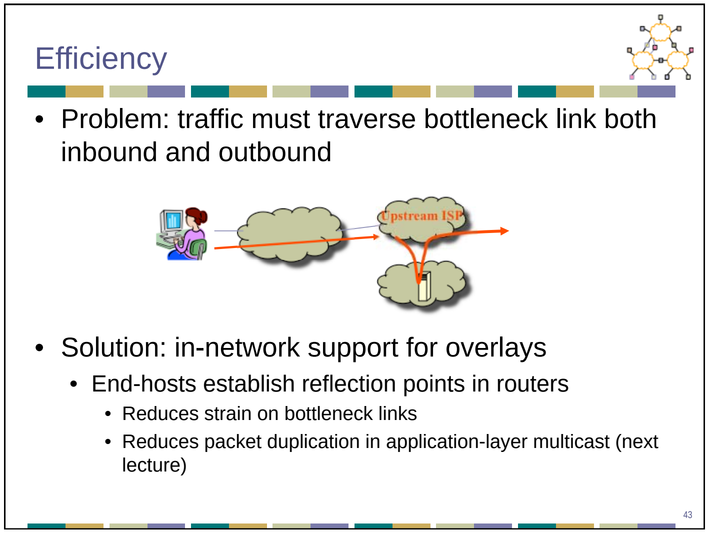
Main results

- RON can route around failures in ~ 10 seconds
- Often improves latency, loss, and throughput
- Single-hop indirection works well enough
  - Motivation for second paper (SOSR)
  - Also begs the question about the benefits of overlays



Open Questions

- Efficiency
  - Requires redundant traffic on access links
- Scaling
  - Can a RON be made to scale to > 50 nodes?
  - How to achieve probing efficiency?
- Interaction of overlays and IP network
- Interaction of multiple overlays

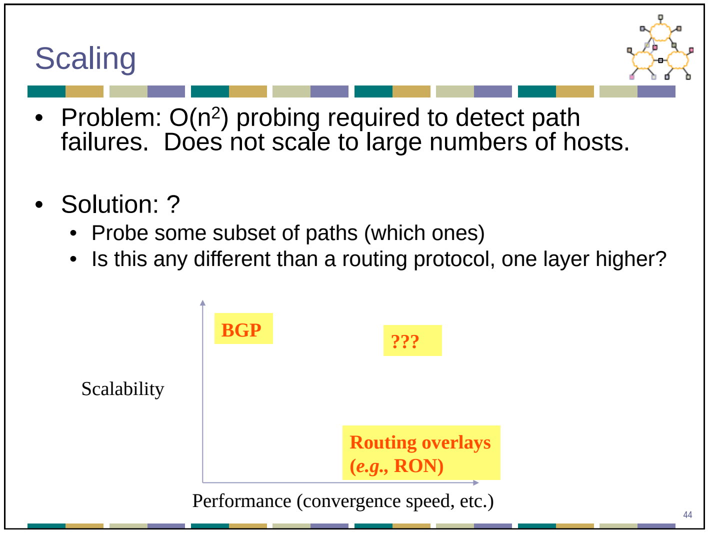


Efficiency

- Problem: traffic must traverse bottleneck link both inbound and outbound

Upstream ISP

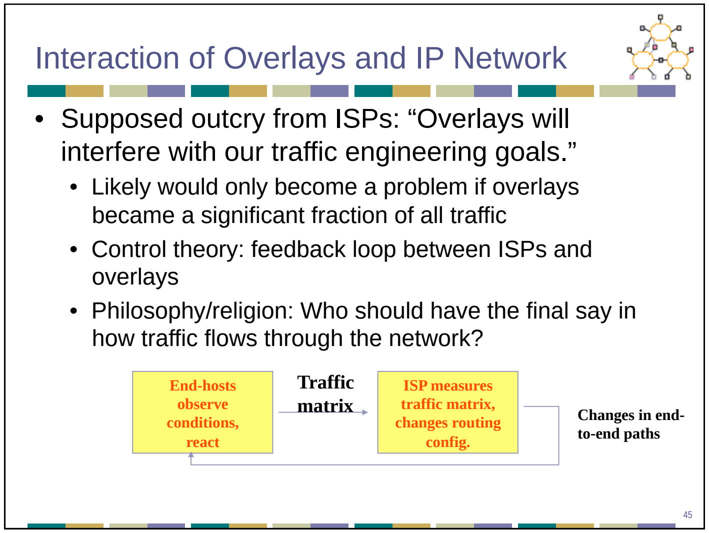
- Solution: in-network support for overlays
  - End-hosts establish reflection points in routers
    - Reduces strain on bottleneck links
    - Reduces packet duplication in application-layer multicast (next lecture)



Scaling

- Problem: O(n^2) probing required to detect path failures. Does not scale to large numbers of hosts.
- Solution: ?
  - Probe some subset of paths (which ones)
  - Is this any different than a routing protocol, one layer higher?

BGP
???
Scalability
Routing overlays (*e.g.,* RON)
Performance (convergence speed, etc.)



Interaction of Overlays and IP Network

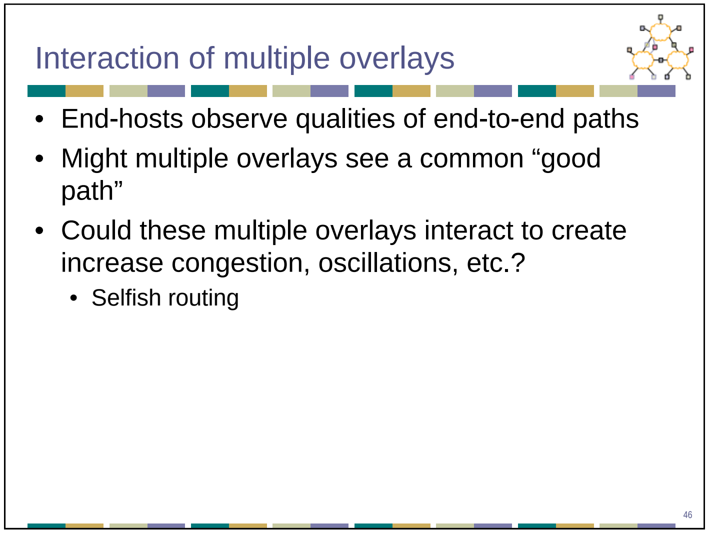
- Supposed outcry from ISPs: “Overlays will interfere with our traffic engineering goals.”
  - Likely would only become a problem if overlays became a significant fraction of all traffic
  - Control theory: feedback loop between ISPs and overlays
  - Philosophy/religion: Who should have the final say in how traffic flows through the network?

End-hosts observe conditions, react

Traffic matrix

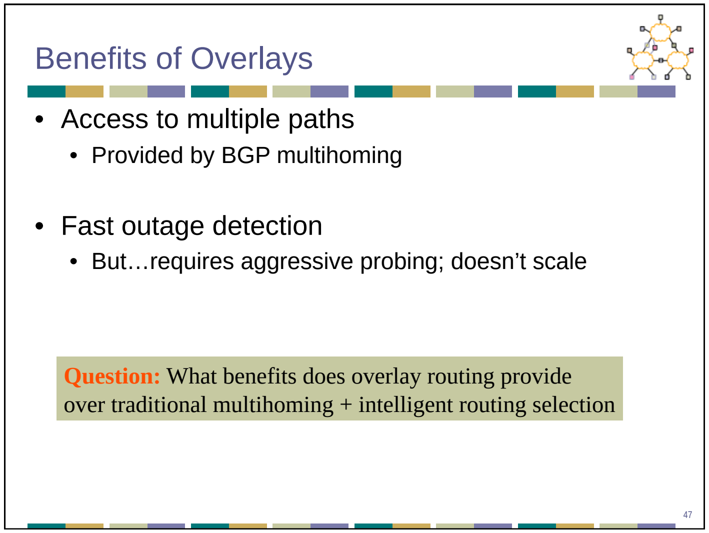
ISP measures traffic matrix, changes routing config.

Changes in end-to-end paths



Interaction of multiple overlays

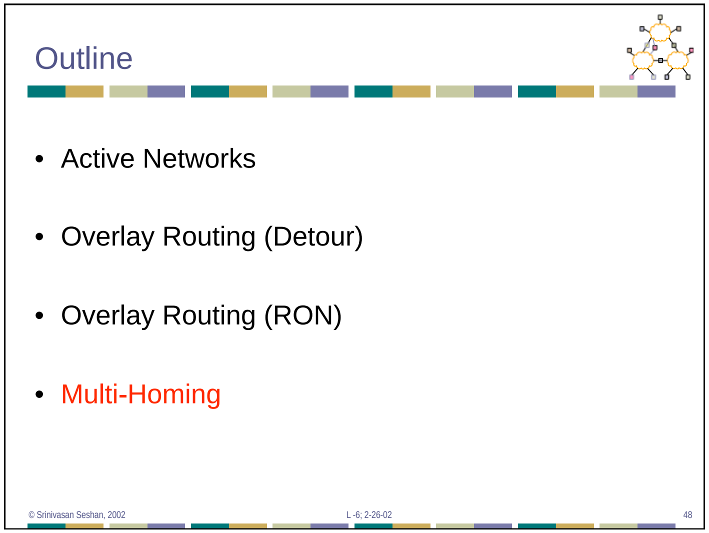
- End-hosts observe qualities of end-to-end paths
- Might multiple overlays see a common “good path”
- Could these multiple overlays interact to create increase congestion, oscillations, etc.?
  - Selfish routing



Benefits of Overlays

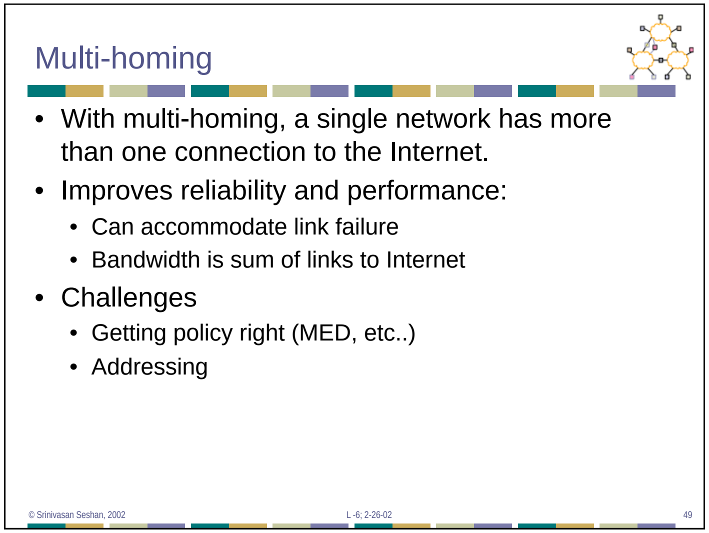
- Access to multiple paths
  - Provided by BGP multihoming
- Fast outage detection
  - But…requires aggressive probing; doesn’t scale

Question: What benefits does overlay routing provide over traditional multihoming + intelligent routing selection



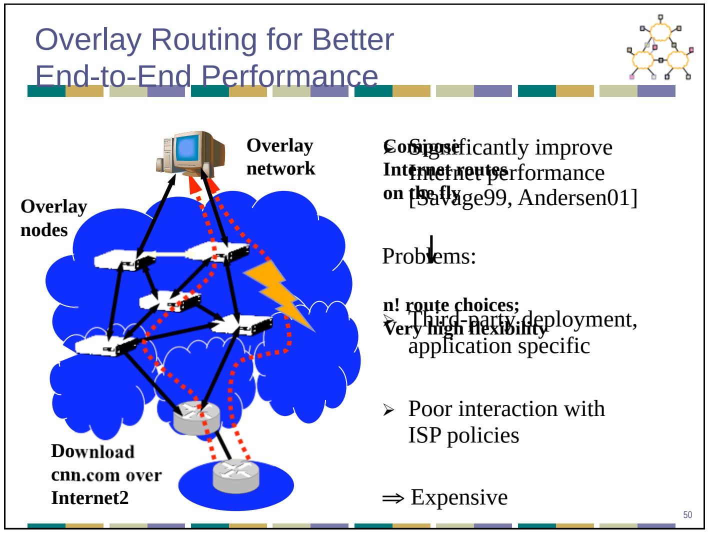
Outline

- Active Networks
- Overlay Routing (Detour)
- Overlay Routing (RON)
- Multi-Homing



Multi-homing

- With multi-homing, a single network has more than one connection to the Internet.
- Improves reliability and performance:
  - Can accommodate link failure
  - Bandwidth is sum of links to Internet
- Challenges
  - Getting policy right (MED, etc..)
  - Addressing



Overlay Routing for Better End-to-End Performance

Compose Internet routes on the fly

➢ Significantly improve Internet performance [Savage99, Andersen01]

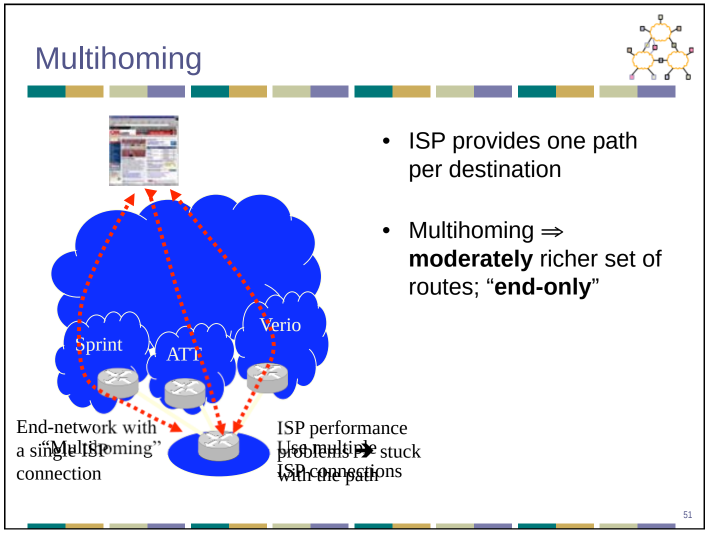
n! route choices; Very high flexibility

Problems:

➢ Third-party deployment, application specific

➢ Poor interaction with ISP policies

⇒ Expensive



Multihoming

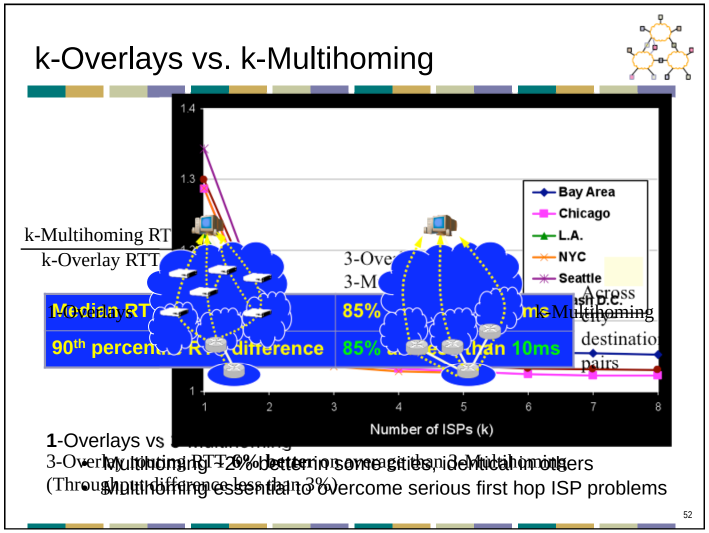
End-network with a single ISP connection

“Multihoming”

ISP performance problems → stuck with the path

Use multiple ISP connections

- ISP provides one path per destination
- Multihoming ⇒ **moderately** richer set of routes; “**end-only**”



k-Overlays vs. k-Multihoming

Median RTT difference	85% ... ms
90th percentile RTT difference	85% ... than 10ms

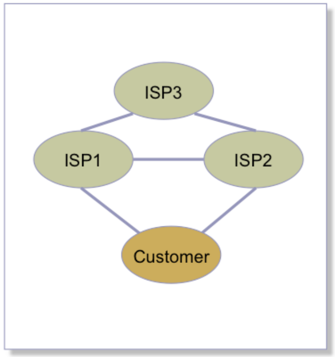
1-Overlays vs 3-Multihoming

- Multihoming ~2% better in some cities, identical in others
- Multihoming essential to overcome serious first hop ISP problems

3-Overlay routing RTT **6% better** on average than 3-Multihoming (Throughput difference less than 3%)

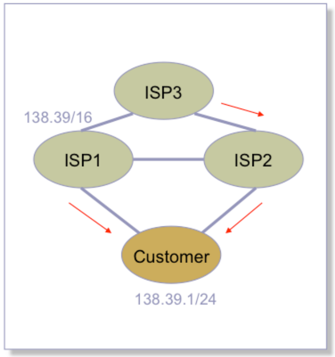
Multi-homing to Multiple Providers

- Major issues:
  - Addressing
  - Aggregation
- Customer address space:
  - Delegated by ISP1
  - Delegated by ISP2
  - Delegated by ISP1 and ISP2
  - Obtained independently



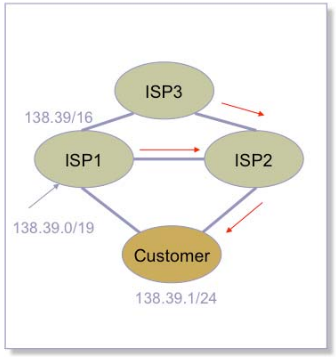
Address Space from one ISP

- Customer uses address space from ISP1
- ISP1 advertises /16 aggregate
- Customer advertises /24 route to ISP2
- ISP2 relays route to ISP1 and ISP3
- ISP2-3 use /24 route
- ISP1 routes directly
- Problems with traffic load?



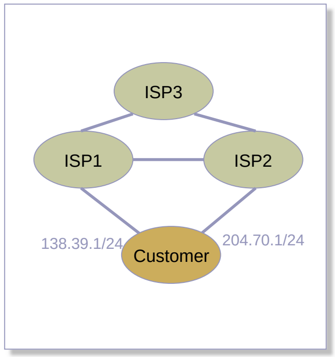
Pitfalls

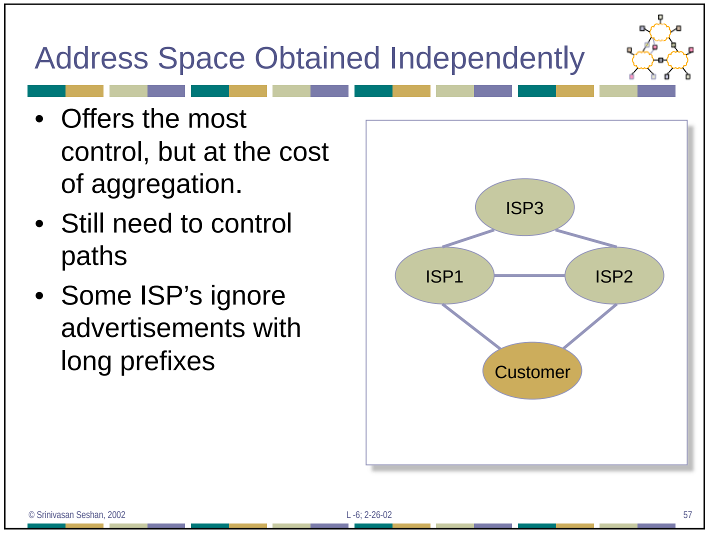
- ISP1 aggregates to a /19 at border router to reduce internal tables.
- ISP1 still announces /16.
- ISP1 hears /24 from ISP2.
- ISP1 routes packets for customer to ISP2!
- Workaround: ISP1 *must* inject /24 into I-BGP.



Address Space from Both ISPs

- ISP1 and ISP2 continue to announce aggregates
- Load sharing depends on traffic to two prefixes
- Lack of reliability: if ISP1 link goes down, part of customer becomes inaccessible.
- Customer may announce prefixes to both ISPs, but still problems with longest match as in case 1.





Address Space Obtained Independently

- Offers the most control, but at the cost of aggregation.
- Still need to control paths
- Some ISP’s ignore advertisements with long prefixes

ISP3
ISP1
ISP2
Customer