# CE693: Adv. Computer Networking

## L-14 Changing the Network

*Acknowledgments: Lecture slides are from the graduate level Computer Networks course thought by Srinivasan Seshan at CMU. When slides are obtained from other sources, a  a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.*

# Adding New Functionality to the Internet

- Overlay networks
- Active networks
- Assigned reading
  - Active network vision and reality: lessons from a capsule-based system
- Optional reading
  - Future Internet Architecture: Clean-Slate Versus Evolutionary Research
  - Resilient Overlay Networks

# Clean-Slate vs. Evolutionary

- Successes of the 80s followed by failures of the 90’s
  - IP Multicast
  - QoS
  - RED (and other AQMs)
  - ECN
  - …
- Concern that Internet research was dead
  - Difficult to deploy new ideas
  - What did catch on was limited by the backward compatibility required

# Outline

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

# 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)
  - 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

# 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

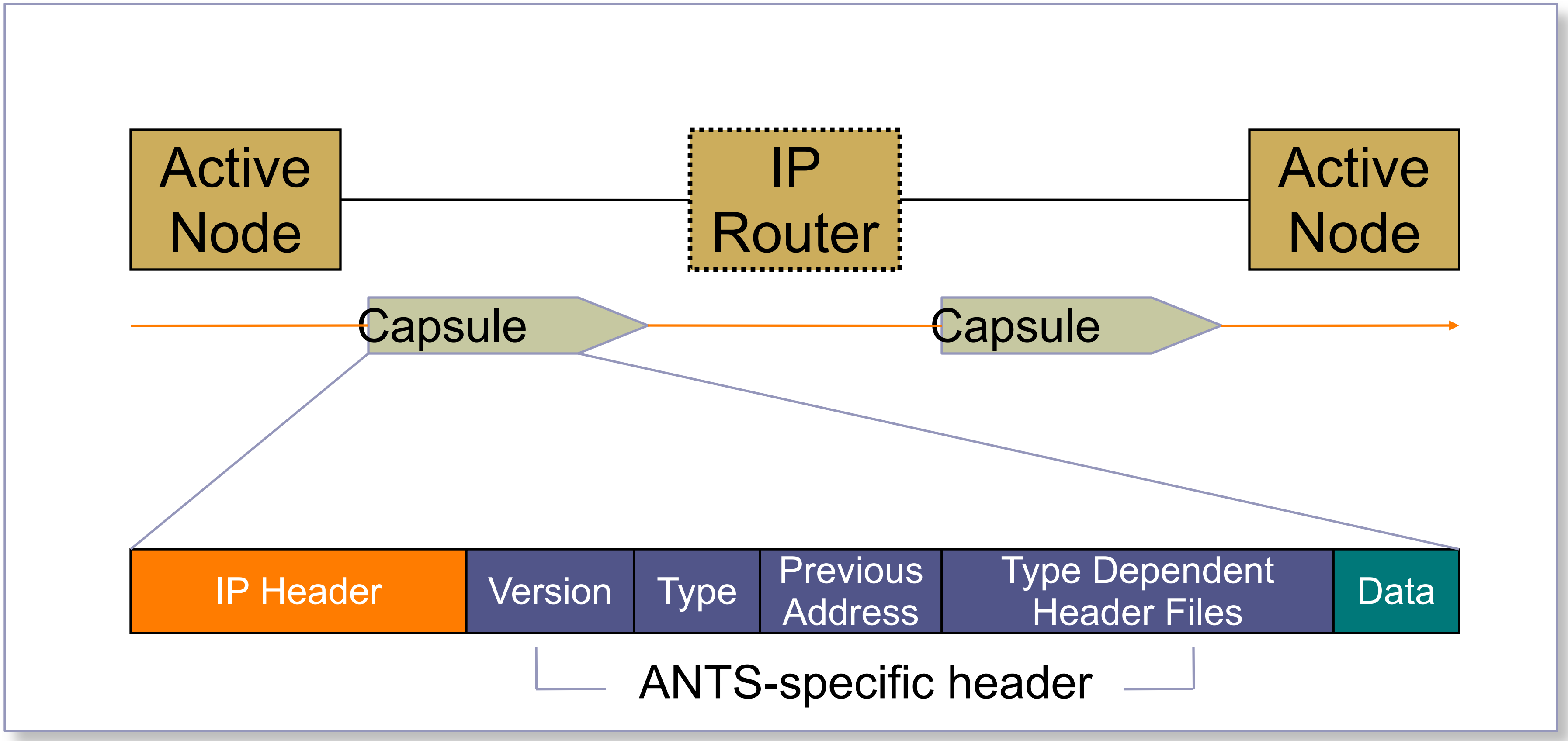
# System Components

- 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

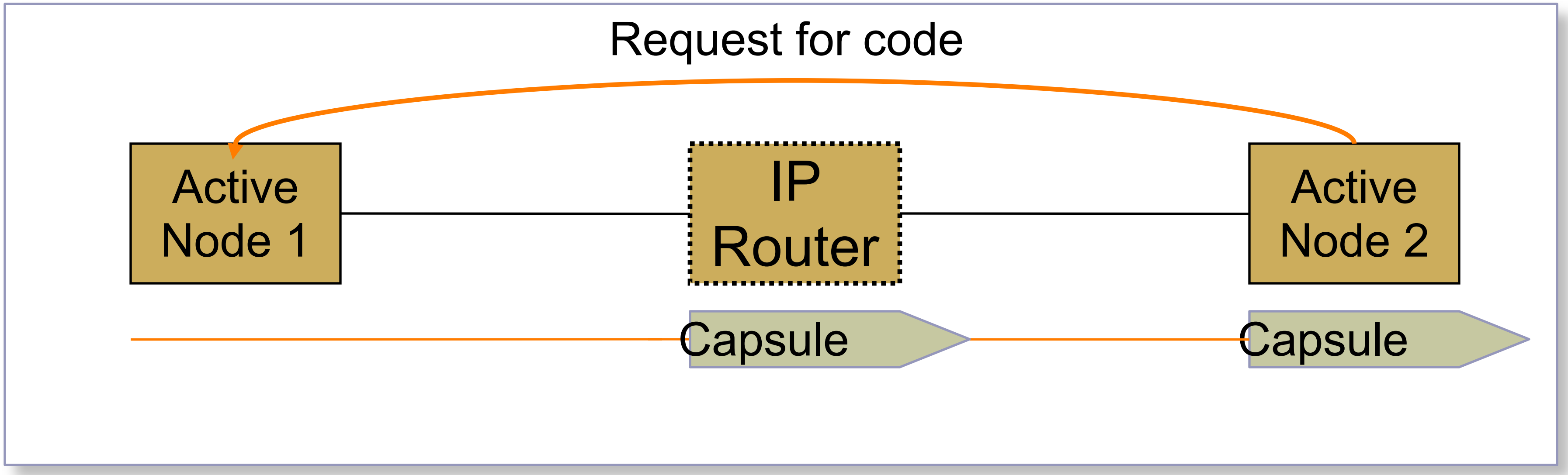
- Each user/flow programs router to handle its own packets
  - Code sent along with packets
  - Code sent by reference
- Protocol:
  - Capsules that share the same processing code
- May share state in the network
- Capsule ID (i.e. name) is MD5 of code

# Capsules



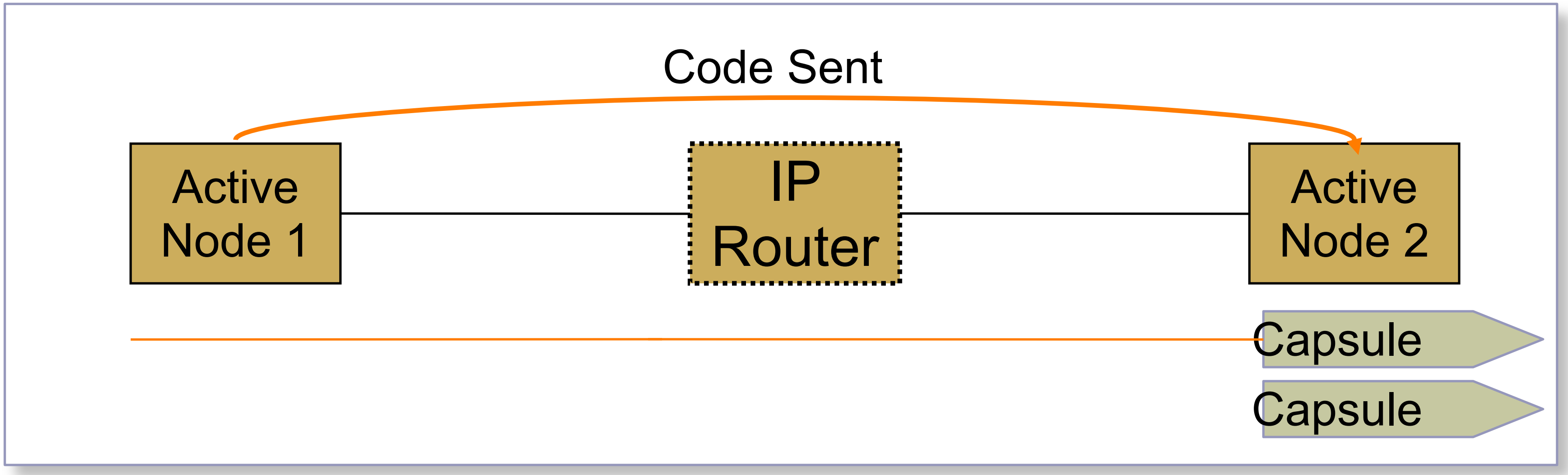
- Capsules are forwarded past normal IP routers

# 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, etc.

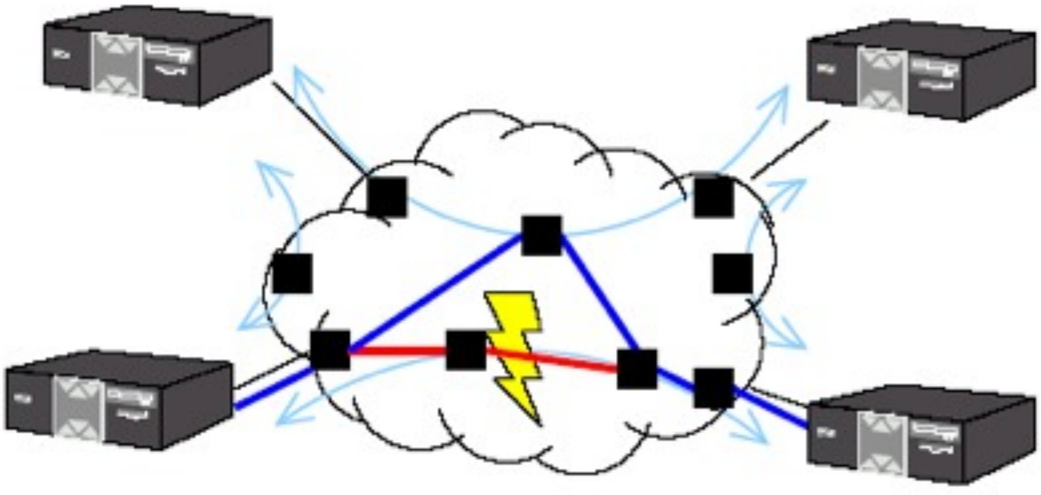
# Discussion

- 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)

# 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
- Alternate path characteristics
  - 30-55% of hosts had lower latency
  - 10% of alternate routes have 50% lower latency
  - 75-85% have lower loss rates

# Possible Sources of Alternate Paths

- A few really good or bad AS’s
  - Not really
- Better congestion or better propagation delay?
  - How to measure?
    - Propagation = 10th percentile of delays
  - Both contribute to improvement of performance
- What about policies/economics?

# Outline

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

# How Robust is Internet Routing?

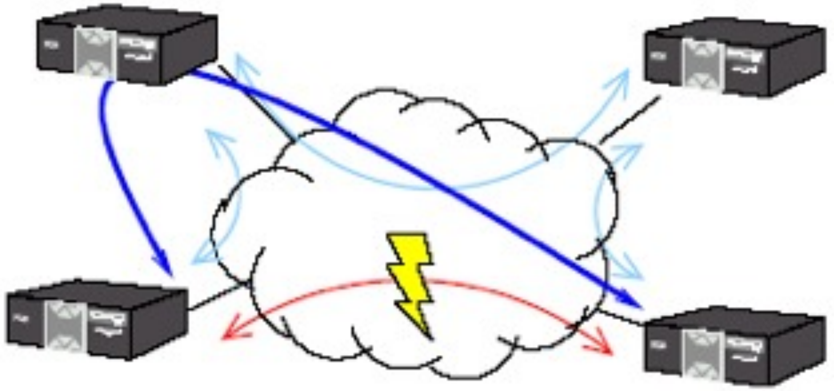
- Slow outage detection and recovery
- Inability to detect badly performing paths
- Inability to efficiently leverage redundant paths
- Inability to perform application-specific routing

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

# Resilient Overlay Networks: Goal

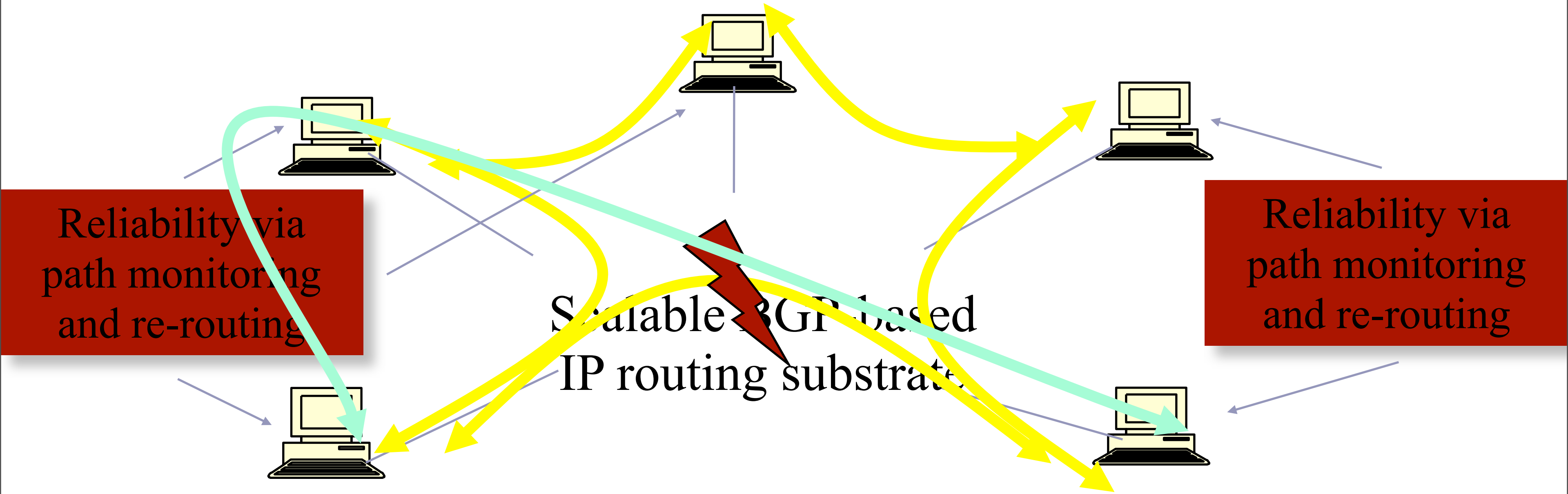
- Increase reliability of communication for a small (i.e., < 50 nodes) set of connected hosts

- Main idea: End hosts discover network-level path failure and cooperate to re-route.



# 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.

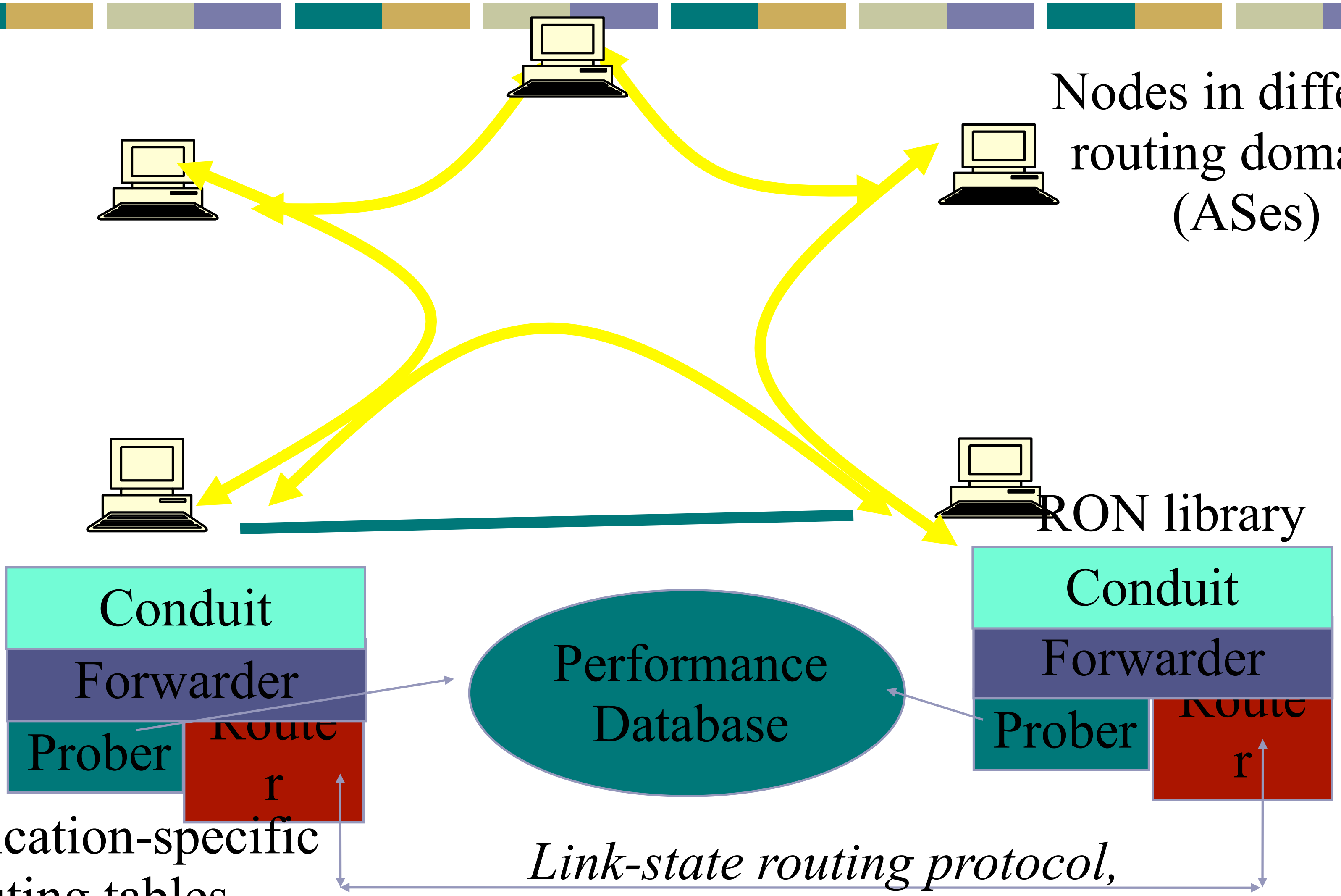


Reliability via path monitoring and re-routing

Scalable BGP-based IP routing substrate

Reliability via path monitoring and re-routing

# RON Design



Nodes in different routing domains (ASes)

Application-specific routing tables

Policy routing module

*Link-state routing protocol, disseminates info using RON!*

# 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

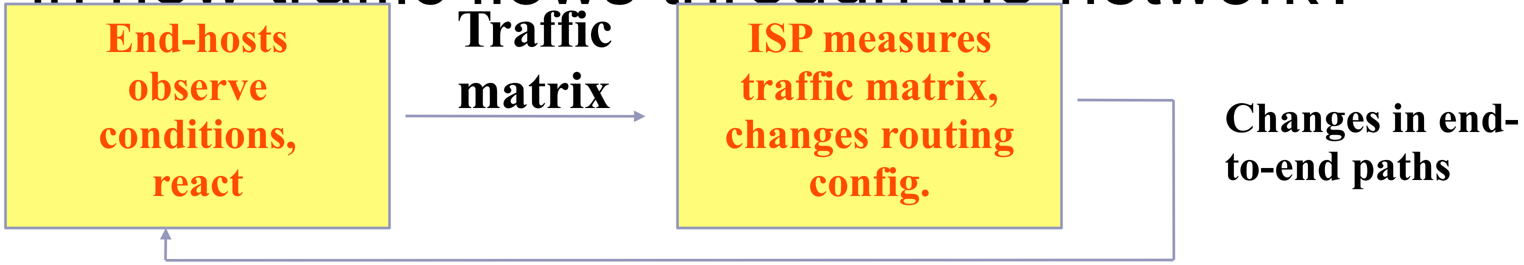
- RON can route around failures in ~ 10 seconds
- Often improves latency, loss, and throughput
- Single-hop indirection works well enough
  - Motivation for another paper (SOSR)
  - Also begs the question about the benefits of overlays

# Open Questions

- Scaling
  - Probing can introduce high overheads
  - Can use a subset of $O(n^2)$ paths → but which ones?
- Interaction of multiple overlays
  - End-hosts observe qualities of end-to-end paths
  - Might multiple overlays see a common “good path”

# 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

# 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

# Next Lecture

- Distributed hash tables
- Required readings:
  - Looking Up Data in P2P Systems
  - Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications