## 15-744: Computer Networking

L-15 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

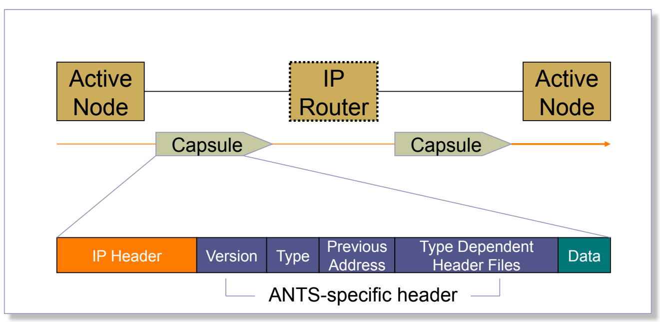


## Capsules

- 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



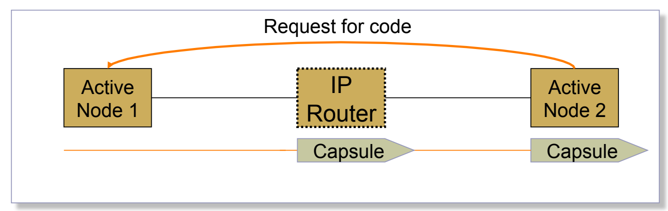
## Capsules




- Capsules are forwarded past normal IP routers

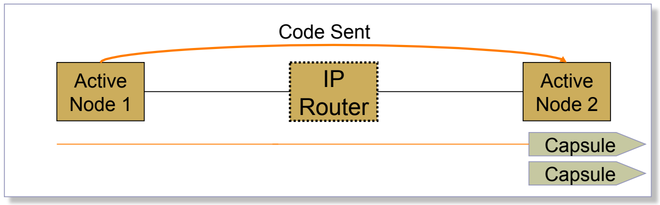


## Capsules




- 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.

17

## 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?

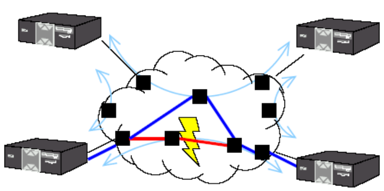
18

## Outline

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

19

## The Internet Ideal



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

20

## 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

25

## 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

26

## 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

27

## 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?

28

## 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<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 |

## Routing Convergence in Practice

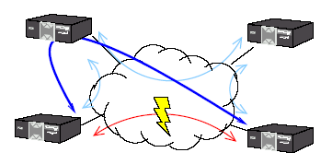
| Time | Prefix | Type | AS Path | Localpref MED | Community |
|---|---|---|---|---|---|
| 2005/11/01 00:06:23 | 195.78.38.0/23 | A | 174 5400 20703 28773 | | 174:21100 16631:1000 |
| 2005/11/01 00:06:39 | 195.78.38.0/23 | A | 3356 5400 20703 28773 | | 3356:2 3356:100 3356:123 3356:500 3356:2064 5400:46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

- Route withdrawn, but stub cycles through backup path…

33

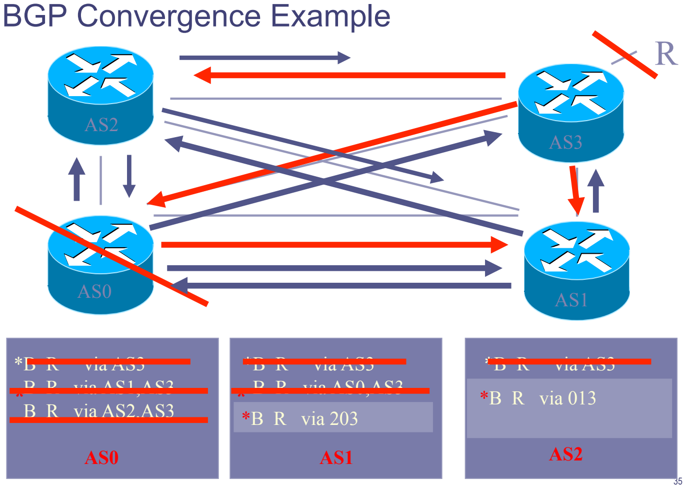
## Resilient Overlay Networks: Goal

- 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.



34

## BGP Convergence Example




35

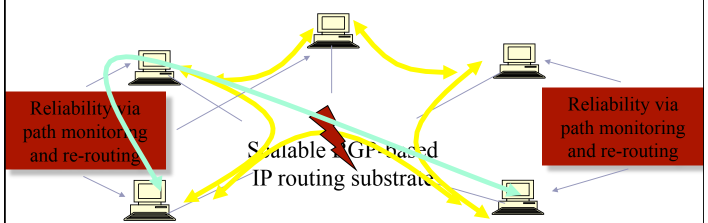
## The RON Architecture

- 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

36

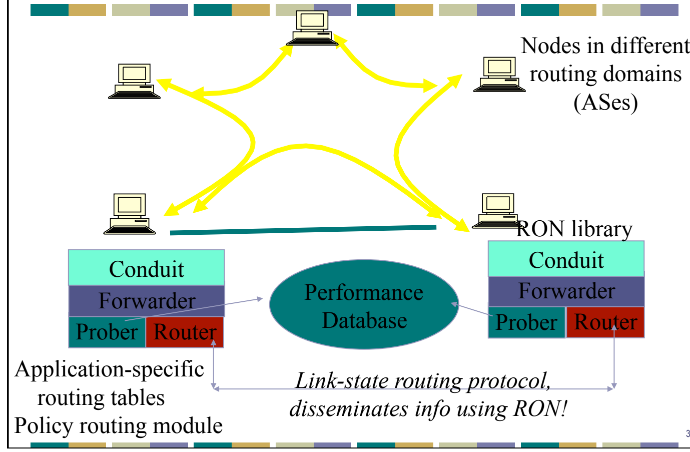
## RON: Routing Using Overlays

- 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.



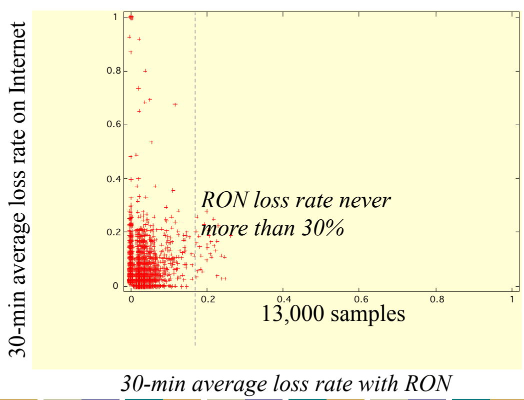

37

## RON Design




38

## RON greatly improves loss-rate




39

## 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!

40

## 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



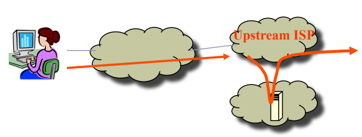
## Open Questions

- 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

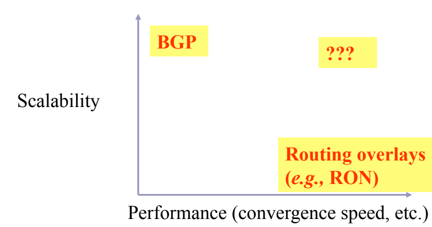



- Solution: in-network support for overlays
  - End-hosts establish reflection points in routers
    - Reduces strain on bottleneck links
    - Reduces packet duplication in application-layer multicast (next lecture)



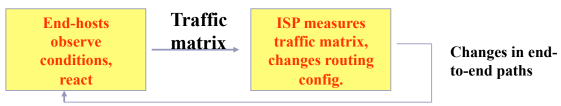
## Scaling

- 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?



## 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?




45

## 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

46

## 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

47

## Outline

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

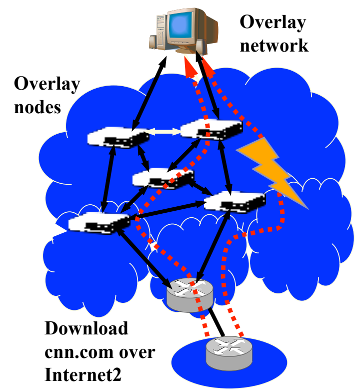
48

## 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




Compute Internet routes on the fly

Significantly improve Internet performance [Savage99, Andersen01]

Problems:

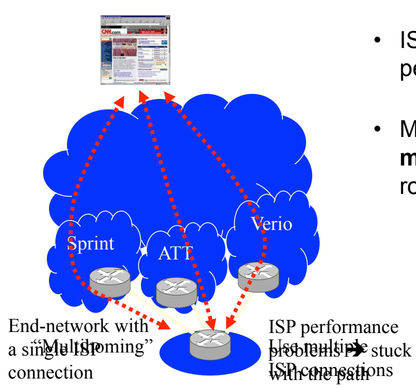
n! route choices; Very high flexibility

- Third-party deployment, application specific
- Poor interaction with ISP policies

⇒ Expensive



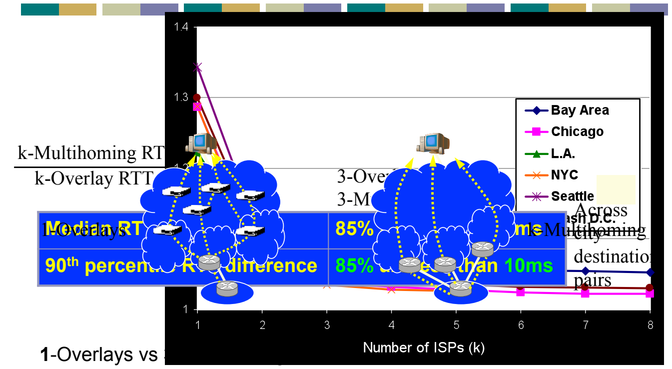
## Multihoming




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



## k-Overlays vs. k-Multihoming



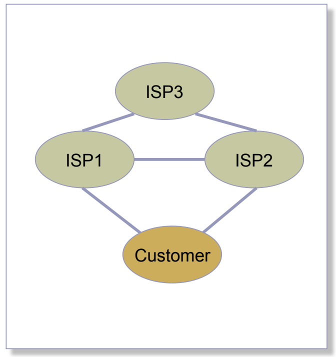

1-Overlays vs 3-Multihoming: Multihoming RTT 2% better in some cities, identical in others

Multihoming essential to overcome serious first hop ISP problems

3-Overlays vs 3-Multihoming (Throughput difference less than 3%)

## Multi-homing to Multiple Providers

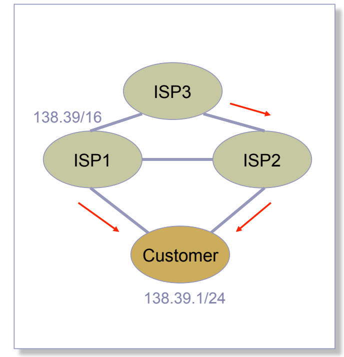
- Major issues:
  - Addressing
  - Aggregation
- Customer address space:
  - Delegated by ISP1
  - Delegated by ISP2
  - Delegated by ISP1 and ISP2
  - Obtained independently




53

## Address Space from one ISP

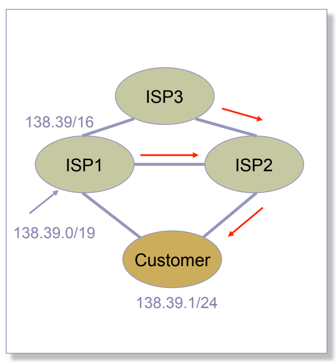
- Customer uses address space from ISP1
- ISP1 advertises /16 aggregate
- Customer advertises /24 route to ISP2
- ISP2 relays route to ISP1 and ISP3
- ISP2-3 use /24 route
- ISP1 routes directly
- Problems with traffic load?




54

## Pitfalls

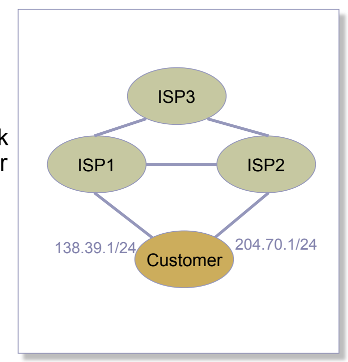
- ISP1 aggregates to a /19 at border router to reduce internal tables.
- ISP1 still announces /16.
- ISP1 hears /24 from ISP2.
- ISP1 routes packets for customer to ISP2!
- Workaround: ISP1 *must* inject /24 into I-BGP.




55

## Address Space from Both ISPs

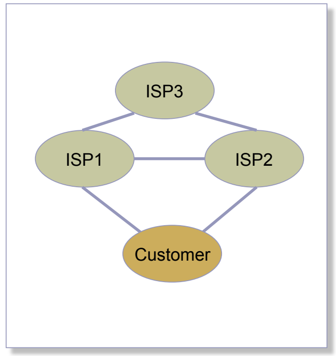
- ISP1 and ISP2 continue to announce aggregates
- Load sharing depends on traffic to two prefixes
- Lack of reliability: if ISP1 link goes down, part of customer becomes inaccessible.
- Customer may announce prefixes to both ISPs, but still problems with longest match as in case 1.




56

## 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




57

## The "Price of Anarchy"

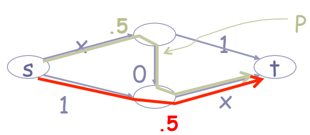
$$\frac{\text{cost of worst Nash equilibrium}}{\text{"socially optimum" cost}}$$

- A directed graph $G = (V,E)$
- source–sink pairs $si,ti$ for $i=1,..,k$
- rate $ri \geq 0$ of traffic between $si$ and $ti$ for each $i=1,..,k$
- For each edge $e$, a latency function le(•)

58

## Flows and Their Cost

- Traffic and Flows:
- A flow vector f specifies a traffic pattern
  - $f_P$ = amount routed on $s_i$-$t_i$ path P



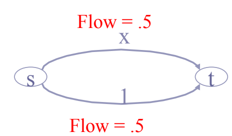

$l_P(f) = .5 + 0 + 1$

The Cost of a Flow:

- $\ell_P(f)$ = sum of latencies of edges along P (w.r.t. flow f)
- $C(f)$ = cost or total latency of a flow f: $\Sigma_P f_P \bullet \ell_P(f)$

59

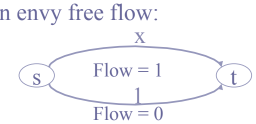
## Example




Cost of flow = .5•.5 +.5•1 =.75

Traffic on lower edge is "envious".

An envy free flow:




Cost of flow = 1•1 +0•1 =1

60

## Flows and Game Theory

- Flow: routes of many noncooperative agents
  - each agent controlling infinitesimally small amount
    - cars in a highway system
    - packets in a network
- The toal latency of a flow represents social welfare
- Agents are selfish, and want to minimize their own latency

61

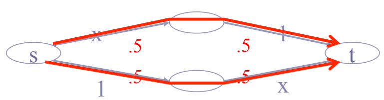
## Flows at Nash Equilibrium

- A flow is at Nash equilibrium (or is a Nash flow) if no agent can improve its latency by changing its path
  - **Assumption:** edge latency functions are continuous, and non-decreasing
- **Lemma:** a flow f is at Nash equilibrium if and only if all flow travels along minimum-latency paths between its source and destination (w.r.t. f)
- **Theorem:** The Nash equilibrium exists and is unique

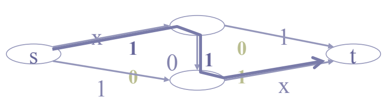
62

## Braess's Paradox

Traffic rate: r = 1




Cost of Nash flow = 1.5




Cost of Nash flow = 2

All the flows have increased delay

63

## Existing Results and Open Questions

- Theoretical results on bounds of the price of anarchy: 4/3
- **Open question:** study of the dynamics of this routing game
  - Will the protocol/overlays actually *converge* to an equilibrium, or will the oscillate?
- **Current directions:** exploring the use of taxation to reduce the cost of selfish routing.

64

## Intuition for Delayed BGP Convergence

- There exists a message ordering for which BGP will explore all possible AS paths
  - Convergence is O(N!), where N number of default-free BGP speakers in a complete graph
  - In practice, exploration can take 15-30 minutes
  - Question: What typically prevents this exploration from happening in practice?
- Question: Why can't BGP simply eliminate all paths containing a subpath when the subpath is withdrawn?



## When (and why) does RON work?

- Location: Where do failures appear?
  - A few paths experience many failures, but many paths experience at least a few failures (80% of failures on 20% of links).
- Duration: How long do failures last?
  - 70% of failures last less than 5 minutes
- Correlation: Do failures correlate with BGP instability?
  - BGP updates often coincide with failures
  - Failures near end hosts less likely to coincide with BGP
  - Sometimes, BGP updates precede failures (why?)

Feamster *et al., Measuring the Effects of Internet Path Faults on Reactive Routing, SIGMETRICS 2003*



## Location of Failures

- Why it matters: failures closer to the edge are more difficult to route around, particularly last-hop failures
  - RON testbed study (2003): About 60% of failures within two hops of the edge
  - SOSR study (2004): About half of failures potentially recoverable with one-hop source routing
    - Harder to route around broadband failures (why?)