Routing & Addressing: Multihoming 10/25/04

Hong Tat Tong

Introduction

- Two attempts to control/ensure best performance over the Internet
  - Multihoming – from the endpoints
  - Overlay – with some help from middle-boxes – overlay routers
- Solutions both use multiple ISPs
- Reflects reality of traffic flows on Internet; traffic has to flow through multiple ISPs most of the time
- An early solution: multiple hierarchical addresses
  - Attempt to deal with scaling problem
  - Routing table inflation – one of the consequences of multihoming!

The two SIGCOMM papers

- *A comparison of overlay routing and multihoming route control – Akella et al @ SIGCOMM '04*
- Efficient and Robust Policy Routing Using Multiple Hierarchical Addresses – Paul Tsuchiya (Francis) @ SIGCOMM '91

A Comparison of Overlay Routing and Multihoming Route Control

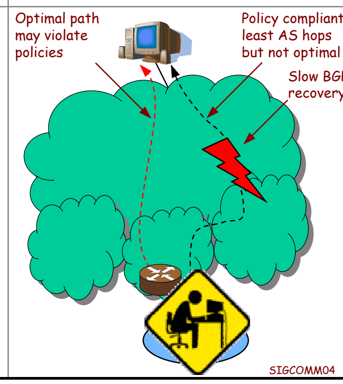
Aditya Akella
CMU

with Jeffrey Pang, Bruce Maggs, Srinivasan Seshan (CMU) and Anees Shaikh (IBM Research)

ACM SIGCOMM
Aug 31, 2004

BGP: Favorite Scapegoat!

BGP Inefficiencies

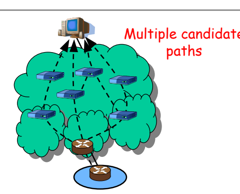


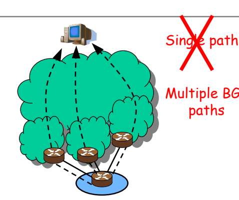
- Poor performance
  - Policy routing, Coarse-grained route selection
- Poor recovery times
  - BGP takes minutes to hours to recover

Overlay Routing

- Bypass BGP routes end-to-end
- Flexible control on end-to-end path
  - Improves performance
  - Better recovery times
- How do overlays address BGP inefficiencies?

Number of Route Choices

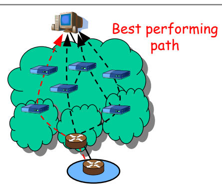


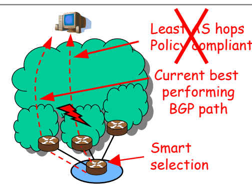


- Flexible control of end-to-end path ⇨ many route choices
- BGP: one path via each ISP ⇨ choices linked to #ISPs

Few more route choices via multihoming

Route Selection Mechanism





"Multihoming route control"

- Overlays: complex, performance-oriented selection
- BGP: simple, coarse metrics such as least AS hops, policy

Sophisticated selection among multiple BGP routes

Overlay Routing vs. Multihoming Route Control

Is multihoming route control competitive with the flexibility of overlay routing systems?

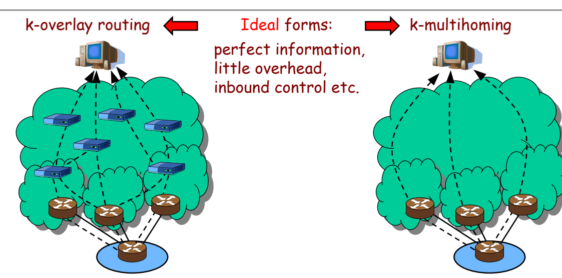
Yes ⇨ good performance and resilience achievable with BGP routing

No ⇨ bypass mechanisms or changes to BGP may be necessary for improved performance and resilience

Talk Outline

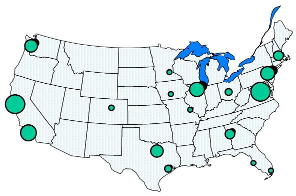
- Methodology of comparison
- Comparison results
- Discussion and summary

Comparison Methodology



- k-overlay performance depends on overlay size, node placement
  - Results based on the testbed chosen
- Ideal vs. practical forms: comparison likely to be unaffected
  - See our Usenix04 paper

Measurement Testbed

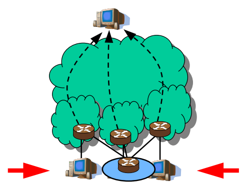


Area of dot ≈ number of nodes

68 nodes, US-based testbed…

- Attached to providers of different tiers
- Nodes in a city: singly-homed to distinct ISPs
- 17 cities

Multihoming emulation



- Stand-in for multihomed network
- Use testbed nodes also as intermediate overlay nodes

Key Comparison Metrics

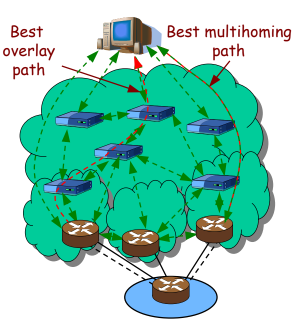
Compare overlay and multihoming paths from nodes in a city to other nodes in the testbed.

- RTT performance
- Throughput performance
- Availability

Data collection & comparison results

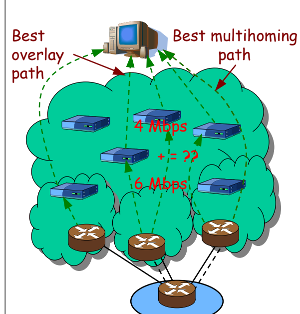
Summary of comparison results

Round-Trip Time Performance



- All-pairs HTTP transfers every 6min
  - 5-day trace
- Record HTTP-level RTT
- Compute delays of best direct and overlay paths
  - Overlay paths could have many hops
- Overlay paths superset of multihoming paths
  - Overlays always better

Throughput Performance

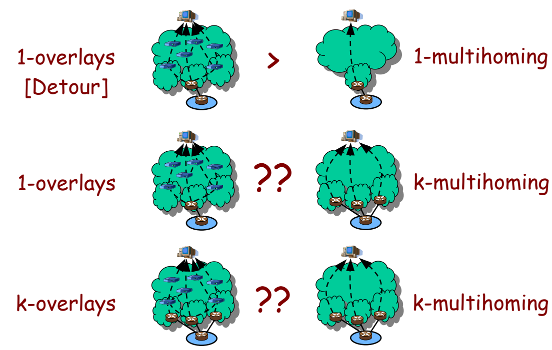


- All-pairs 1MB transfers every 18 min
  - 5-day trace
- Record throughput
- Compare best overlay and direct throughputs
- Overlays: Combine per-hop throughputs (like Detour does)
  - *Pessimistic* and *optimistic* combination functions
  - Consider one-hop overlay

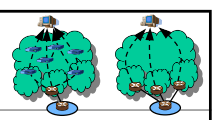
Talk Outline

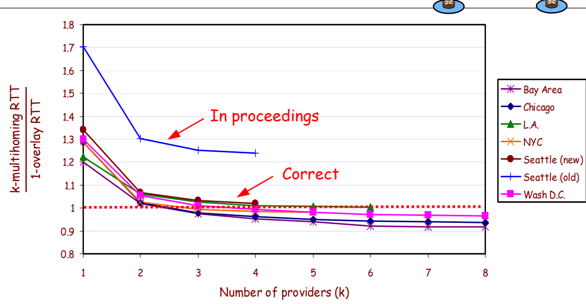
- Methodology of comparison
- **Comparison results**
- Discussion and summary

RTT and Throughput Comparison



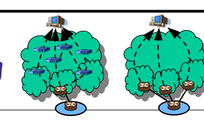
1-Overlays vs. k-Multihoming

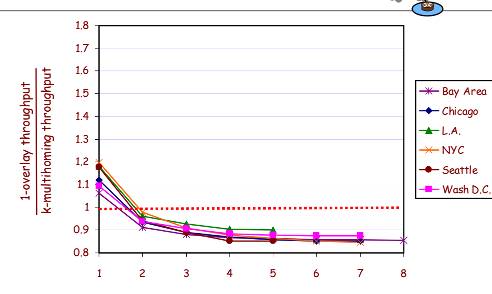




Benefits of 1-overlays significantly reduced compared to k-multihoming. 1-overlays cannot overcome first-hop ISP problems.

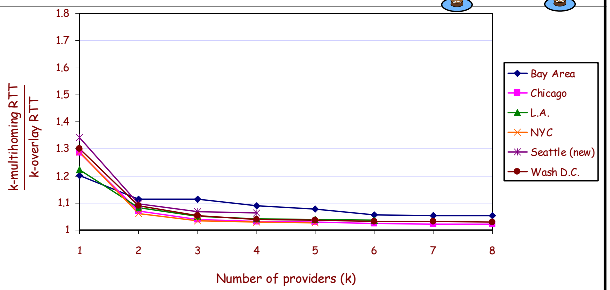
1-Overlays vs. k-Multihoming





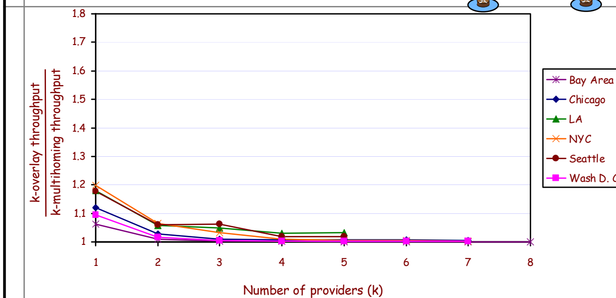
Benefits of 1-overlays significantly reduced compared to k-multihoming. 1-overlays cannot overcome first-hop ISP problems.

k-Overlays vs. k-Multihoming



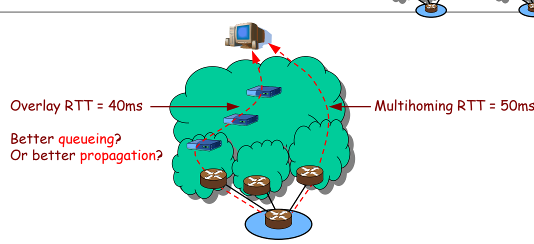
k-overlay routing offers marginal benefits over k-multihoming.

k-Overlays vs. k-Multihoming



k-overlay routing offers marginal benefits over k-multihoming.

3-Overlays vs. 3-Multihoming

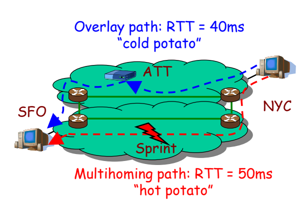


- Congestion vs. propagation
  - Better indirect overlays paths are physically shorter → 66% of cases
  - But, largest improvements (> 50ms) due to overlays avoiding congestion

3-Overlays vs. 3-Multihoming

Better indirect paths	Percentage
Violate inter-domain policies	67*
Conform to inter-domain policies	25*
Same AS-level Path as a multihoming path	15

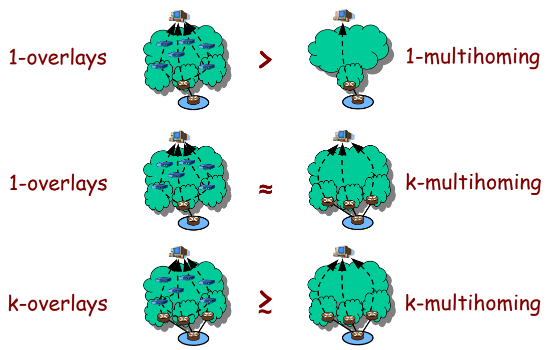
* 8% of paths could not be mapped to an AS level path



- Inter-domain and peering policy violation
  - Most indirect paths violate inter-domain policies

ISP Cooperation: BGP can realize 15% of "indirect" paths

RTT and Throughput: Summary



Availability Comparison: Summary

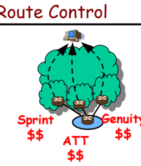
- Use active ping measurements and RON failure data
- k-overlays offer almost perfect availability
  - Multihoming may be necessary to avoid first-hop failures
- k-multihoming, k > 1, is not as perfect
  - 3-multihoming: availability of 100% on 96% of city-dst pairs
    - 1-multihoming: only 70% of pairs have 100% availability
  - May be good enough for practical purposes

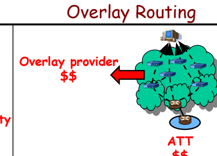
Talk Outline

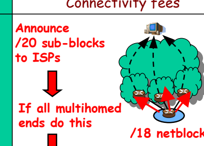
- Methodology of comparison
- Comparison results
- **Discussion and summary**

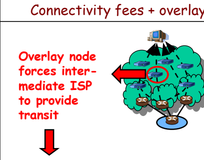
Overlay Routing vs. Multihoming Route Control

	Route Control	Overlay Routing
Cost	Connectivity fees	Connectivity fees + overlay fee
Operational issues	Routing table expansion	Bad interactions with policies









Summary

- Route control similar to overlay routing for most practical purposes
- Overlays very useful for deploying functionality
  - Multicast, VPNs, QoS, security
- But **overlays may be overrated** for end-to-end performance and resilience
- Don't abandon BGP – there's still hope

Comments

- Overall, a well-constructed study
- Good sample size – but US-centric (what about international links?)
- "The most marked improvements in RTT were due to overlay paths avoiding congestion"
  - Will the performance gap between overlay and multihoming be greater in more congested networks?
  - Problems of oscillation?
- Problem: study only deals with snapshots, does not see trends over time (e.g. oscillatory behavior) that might be caused by these route control mechanisms…

Efficient and Robust Poliy Routing Using Multiple Hierarchical Addresses

Paul Tsuchiya
Bellcore

ACM SIGCOMM
1991

Problems

- IP Address/routing algorithms do not scale well - O(N^2)
- Policy routing increasingly required
- Scaling vs policy – problem!
  - Hierarchical addresses for scaling
  - Hierarchy restricts policy control options – not able to send packet via different network
  - For policy – routers have to keep track of individual networks – not scalable!

Solution: Multiple Addresses per node!

But what are addresses anyway?

What are addresses?

- Shoch
  - Name
  - Address
  - Route
- Tsuchiya
  - Identifying
  - Routing

What are addresses?

- Shoch
  - Routing tables should hold multiple paths
  - Able to pick new paths quickly
  - Addresses should be as static as names
- Tsuchiya
  - Yup!
  - Yup!
  - Nah…

Taxonomy: Shoch vs Tsuchiya

Shoch

name | **address** | **route**

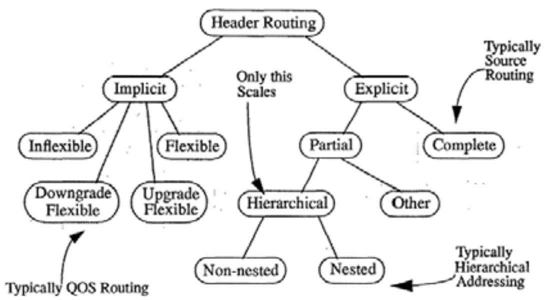
Tsuchiya

identifying | **routing**

Tsuchiya elaborated

naming | **ID** | **header routing** | **table routing**

Taxonomy for Header Routing



Issues and choices

- Use telephone-style multiple hierarchy for network addressing?
  - Provider access code – area code – switch - id
  - Inappropriate – can't assume that terminal can be reached through a particular backbone
- Where to put routing information in header?
  - Source route field? encoding is inefficient
  - QoS field? Not commonly implemented
  - DNS/X.500 return addresses, not these fields!
  - ADDRESSES – the most expedient/compatible place to put this information

Hierarchical Routing – division of labor

- Scalable – each router needs RHR^(1/H) entries instead of R^2
- Table routing finds the paths to the backbones
- Header (directory) routing defines the path from the backbone to the destination
  - Directory service works well if response is independent of source

Hierarchical routing + policy routing

- Need P paths between routers!
- Hence, need P paths from source to backbone by **table routing**
- But – common policy to have multiple backbones
- Hence, we also need P addresses (one associated with each backbone) by **header routing**

Static vs Dynamic Addressing

- Statically pre-assign routes for destination; choose from this set of routes
  - Can only handle a certain set of failures
  - But Internet has generally stable topology/good reliability
  - Since topology is stable, no great need to use dynamic addresses for header routing (backbones don't normally change that frequently)
- Dynamically calculate routes from scratch
  - Can handle arbitrary set of failures
  - BUT dynamic addressing is beyond state of the art

Proposed connection steps

1. Source gets address set from directory service
2. Prune address set based on policy
3. Negotiate address set with destination
   - Need change in TCP for this
4. Establish communications with preferred address
5. Change address if current one fails
   - Need change in TCP for this

Changes required in TCP

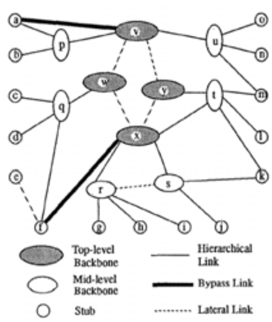
- Initiator sends connection request packet with list of possible addresses
  - But what address is this packet sent to? Unclear…
  - What if the address is invalid? Have to try again…
- Receiver prunes list and responds
- On ICMP unreachable error, host tries next address in list instead of giving up

Policy routing

- Find path that:
  - Satisfies minimum performance requirements of application
  - Satisfies constraints placed on path by sender, receiver, or backbone
  - Gives best price/performance ratio to whoever is paying (sender/receiver/both)

Policy routing

- Up-across-down
- Destination address determines exit backbone and part of down-path
- Source address (if looked at by router) determines entry backbone and part of up-path
- Across path? Determined by table routing!
- Trivial because backbones coordinate amongst themselves to form contiguous system



Problems with across path

- Non contiguous policy
  - Some stubs just don't want to go through backbone X
  - Billing policies
  - Such cases are not very common or plausible
- Different services on same backbone
  - Certain stubs have high speed access but not others
  - QoS parameter
  - 2 address spaces

Problems with multiple hierarchical addresses

- Added burden on forwarding algorithm in browsers
  - Optimize search: check routing table entry for "internal" address space first
  - If most traffic stays within private domain… (?)
- Address assignments to hosts may change often
  - Better network management systems to configure addresses?
  - Incorporate address assignment into intra-domain routing protocol

Problems with multiple hierarchical addresses

- Proliferation of addresses due to multiple backbones
  - But hierarchies tend to be shallow
  - No need to have one address for every path!
- How does source know the type of backbone associated with an address?
  - Either DNS returns this type, or source keeps table
- Idea of shifting burden to ends is consistent with end-to-end principle, but is problematic
  - No incentive for ends to solve a problem of the middle!