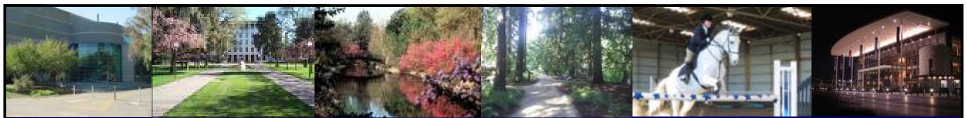


Overlay Networks
- Indirection & Virtualization

DIMACS Tutorial on Algorithms for Next Generation Networks

Chen-Nee Chuah
Robust & Ubiquitous Networking (RUBINET) Lab
http://www.ece.ucdavis.edu/rubinet
Electrical & Computer Engineering
University of California, Davis

Outline

- Overlay Networks: Overview
- Indirection: Overlay Routing Service
- Problematic Interactions between Multiple Overlays and with IP-layers
  – Equilibrium behavior: Game theoretic framework
  – Transient behavior
- Moving Forward
  – Productive co-existence of overlay/underlay
  – Combining multi-homing and overlay routing
- Discussions: Other Design Challenges
  – Resource sharing & network virtualization

Current Internet Infrastructure

- Network layer
  - Defines addressing, routing, and service model for communication between hosts
- Default IP-routing
  - Hierarchical structures (IGP vs. BGP)
    - Allow flexibility and distributed management
    - Achieve global reachability/connectivity
  - Dynamic re-routing around failures
  - CIDR allows route aggregation for announcements, leading to smaller routing tables

Why is it not good enough?

- Routing anomalies impact network/service availability
  - Failures, slow convergence, mis-configurations
- Trade-off performance of scalability
  - Internet paths are often sub-optimal
- New services need new capabilities
  - Mobility? Multicast service?

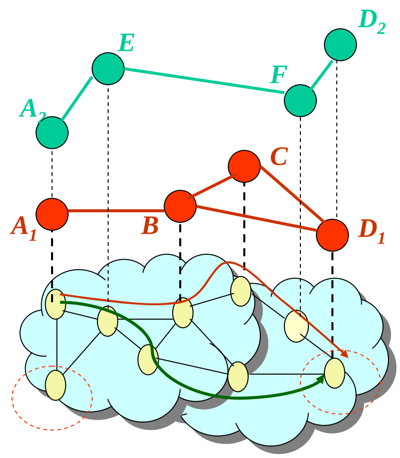
Solution Space:

- Change the existing network layer, or
- Build an overlay on top of existing networks

Overlay Networks

An overlay network

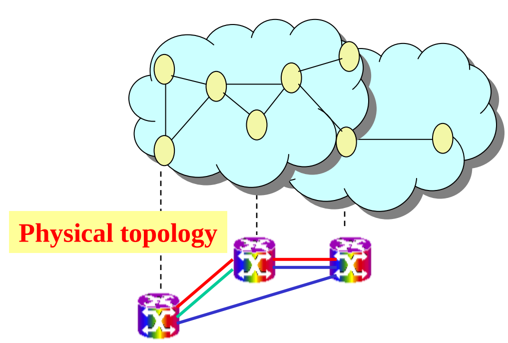
- Is built on top of one or more existing networks
- Adds an additional layer of <u>indirection</u> and/or <u>virtualization</u>
- Changes properties in one or more areas of underlying network



Resources at node A & D are shared among two overlays and the original network

Historical Example

- Internet is an overlay network
  - Goal: connect local area networks
  - Built on local area networks (e.g., Ethernet), phone lines
  - Add an Internet Protocol header to all packets



Application-Layer Overlay Networks

- Overlay networks are becoming popular
  - Allow application-level routing decisions, often designed to circumvent IP-layer routing problems
  - End-hosts and/or router nodes
  - Ad hoc vs. infrastructure-based (pre-selected common overlay nodes)
  - Application-specific, e.g., multicast like Splitstream [CD+03], DHT like Bamboo
  - Generic structured overlays, e.g., RON [AB+01], routing underlay [NPB03], Detour

Our discussion focused on infrastructure-based generic overlays …

Benefits

- Do not have to deploy new equipment, or modify existing software/protocols
  - Probably deploy new software on top of existing ones
    - E.g., adding IP on top of Ethernet does not require modifying Ethernet protocol or driver
  - Allows bootstrapping
    - Expensive to develop entirely new networking hardware/software
    - All networks after the telephone have begun as overlay networks

Benefits

- Do not have to deploy at every node
  - Not every node needs/wants overlay network service all the time
    - e.g., QoS guarantees for best-effort traffic
  - Overlay network may be too heavyweight for some nodes
    - e.g., consumes too much memory, cycles, or bandwidth
  - Overlay network may have unclear security properties
    - e.g., may be used for service denial attack
  - Overlay network may not scale (not exactly a benefit)
    - e.g. may require n^2 state or communication

Costs

- Adds overhead
  - Adds a layer in networking stack
    - Additional packet headers, processing
  - Sometimes, additional work is redundant
- Adds complexity
  - Layering does not eliminate complexity, it only manages it
  - **More layers of functionality → more possible unintended interaction between layers**

Outline

- Overlay Networks: Overview
- **Indirection: Overlay Routing Service**

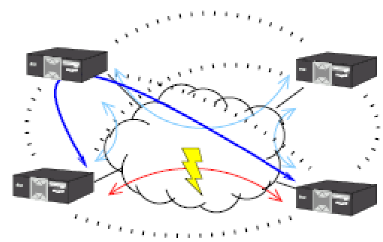
Overlay Routing (Indirection) Service

- Motivation: circumvent shortcomings of IP-layer routing
  - Suffers slow outage detection and recovery
  - Cannot detect badly performing paths
  - Cannot efficiently leverage redundant paths (e.g., AS-paths that do not conform to policies)
  - Cannot express sophisticated routing policy / metrics
  - Intra-AS routing is optimized for load balancing, not end-host or application-level performance

Example: Resilient Overlay Networks (RON)

D. G. Andersen, H. Balakrishnan, M. Frans Kaashoek, R. Morris, "Resilient Overlay Networks," *Proc. 18th ACM SOSP, Oct 2001*

- Goal: Increase reliability of communication for a small (< 50) set of connected hosts
- Basic idea: end hosts
  - Frequently measure *all* inter-node paths and detect outage
  - Exchange routing information
  - Route along app-specific best path consistent with routing policy



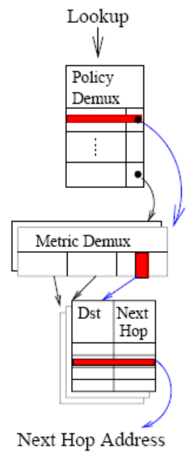
Probes and Routing

[And'01] Probing & Outage Detection

- Probe between nodes to measure path qualities
  - $O(n^2)$ active probes, UDP-based
  - Passive measurements
- Probing & Outage Detection
  - Probe every random(14) seconds
  - 3 packets, both sides get RTT and reachability
  - If "lost probe," send next immediately
  - If N lost probes, notify outage
  - Timeout based on RTT and RTT variance
- Store latency and loss-rate information in DB

[And'01] RON: Routing & Forwarding

- Link-state routing protocol between nodes
  - Disseminates info using the overlay
- Building forwarding tables
  - Policy routing
    - Restrict some paths from hosts, e.g., don't use Internet2 hosts to improve non-Internet2 paths
    - Generate table per policy
  - Metric optimization
    - App tags packets, e.g. "low latency"
    - Generate one table per metric



[And'01] Results & Implications

- Does the RON approach work?
  Probe-based outage detection seems effective
  - RON takes ~10s to route around failure, compared to BGP's several minutes
  - Many Internet outages are avoidable
  - RON often improves latency / loss / throughput

BUT

- Doesn't RON violate network policies?
- Can RON's routing behavior be stable?
  - Is large-scale deployment safe?
  - **Are there problematic interactions w/ lower-layer or other overlays?**

Outline

- Overlay Networks: Overview
- Indirection: Overlay Routing Service
- **Problematic Interactions between Multiple Overlays and with IP-layers**
  - **Equilibrium behavior: Game theoretic framework**
  - **Transient behavior**

Interactions between Overlays & IP-Layer

- Overlays compete with IP-layer to provide routing service
  - Both unaware of key things happening at the other layer
- Multiple overlay networks make independent decisions
- Multiple control mechanisms => problematic interactions
  - Seemingly independent periodic process can inadvertently become synchronized, e.g., routing update message [FJ94]
  - Multiple control loops reacting to same events => race conditions
- Big questions – ***How does all this affect ISPs & overlay networks and the traffic they carry?***

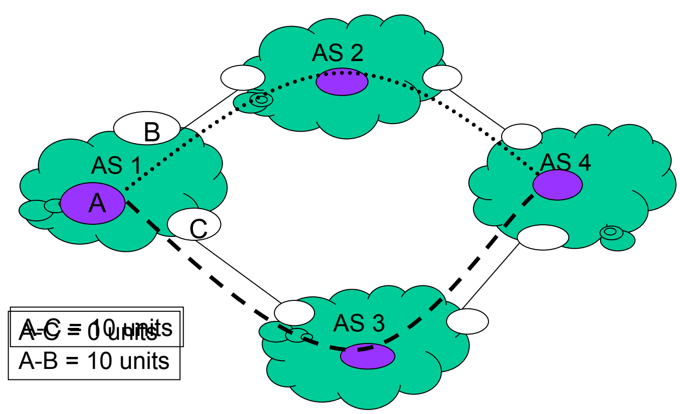
Potential "Side Effects" of Overlay Networks

R. Keralapura, N. Taft, C-N. Chuah, and G. Iannaccone, "Can ISPs take the heat from Overlay Networks?" *HotNets-III, November 2004*

1. Challenges to IP-layer traffic engineering (vertical interactions)
   – Overlays *shift* and/or *duplicate* TM values, increasing the dynamic nature of the TM
   – Harder to estimate Traffic Matrix (TM) essential for most TE tasks.

Problem 1: Challenges to IP-Layer Traffic Engineering (Vertical Interactions)

- Traffic Matrix Estimation



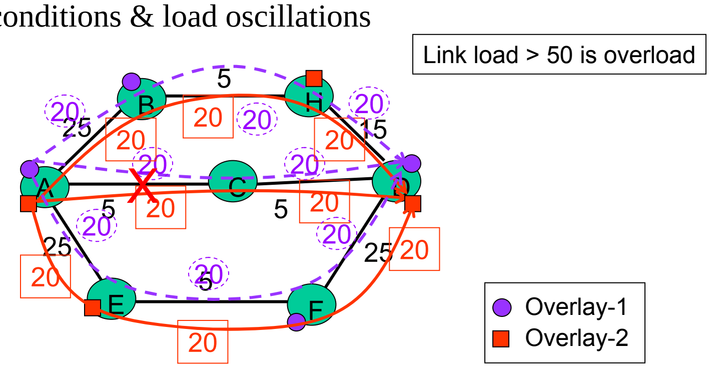
- *Shifts* TM values by changing the exit point
- Increases the dynamic nature of TM

Potential "Side Effects" of Overlay Networks

1. Challenges to IP-layer traffic engineering (vertical interactions)
2. Multiple overlays can get synchronized (horizontal interactions)
   - Can impact both overlay and non-overlay traffic
   - Interfere with load balancing or failure restoration, leading to oscillations

Problem 2: Synchronization btw Multiple Overlays (Horizontal Interactions)

- Multiple overlays can get synchronized!
  - Race conditions & load oscillations

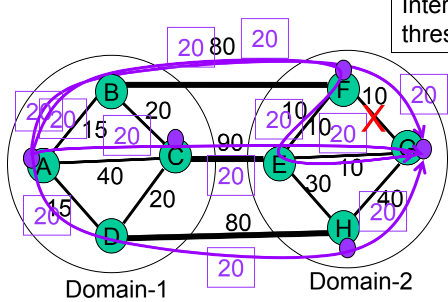


Potential "Side Effects" of Overlay Networks

1. Challenges to IP-layer traffic engineering
   (vertical interactions)
2. Multiple overlays can get synchronized
   (horizontal interactions)
3. Coupling of multiple ASes
   – Overlay Networks may respond to failures in an AS by shifting traffic in upstream AS.

Problem 3: Coupling Multiple AS Domains

Link load > 50 is overload
Interdomain links have higher thresholds



- Defeats one of the objectives of BGP to decouple different domains by insulating an AS from events in neighboring ASes

Problematic Interactions: Sample Studies

- Equilibrium behavior (game theoretic framework)
  - L. Qiu, Y. Yang, Y. Zhang, and S. Shenker (ICSI), "On Selfish Routing In Internet-like Environments," *ACM SIGCOMM 2003*.
  - Y. Liu, H. Zhang, W. Gong, D.Towsley, "On the Interaction Between Overlay Routing and Underlay Routing," *IEEE INFOCOM 2005*.
  - Joe W. J. Jiang, D. Chiu, John C.S. Lui, "On the Interaction of Multiple Overlay Routing," *Journal of Performance Evaluation*, 2005.
- Transient behavior
  - R. Keralapura, C-N. Chuah, N. Taft, and G. Iannaccone, "Can co-existing overlays inadvertently step on each other?" *Proc. IEEE ICNP*, November 2005.
- P2P vs. ISP
  - H. Wang, D. Chiu, John C.S. Lui, "Modeling the Peering and Routing Tussle between ISPs and P2P applications," *IEEE IWQoS 2006*.

Overlay routing is selfish in nature

- IP routing is
  - Optimized for system-wide criteria (e.g., minimize maximum link utilization)
  - Often sub-optimal in terms of user performance
    - Because of policy routing, etc.
- Emerging trend: let end users choose their own routes
  - Example: Source routing, overlay routing
- Selfish nature
  - End hosts or routing overlays greedily select routes to optimize their own performance without considering system-wide criteria

Equilibrium Behavior of Selfish Routing

[Qiu'03] L. Qiu, Y. Yang, Y. Zhang, S. Shenker (ICSI), "On Selfish Routing In Internet-like Environments," *ACM SIGCOMM 2003*

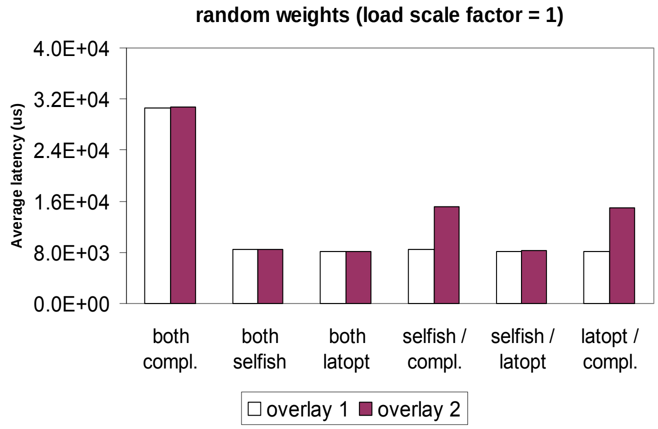
- Question: How does selfish routing perform in Internet-like environments?
- Approach: simulation study of equilibrium behavior
  - Focus on intra-domain environments
    - Realistic topologies (from ISP, Rocketfuel, random power law)
    - Traffic demands (real & synthetic traces)
    - Latency functions (propagation & queuing delay)
  - Apply game theory to compute traffic equilibria and compare results with global optima & default IP routing
    - In each round, each overlay computes its best response by fixing the other overlays' traffic; then the best response and the previous state are merged using decreasing relaxation factors.

[Qiu'03] Selfish Overlay Routing

- Routing schemes considered
  - Overlay source routing: individual minimize own delay
  - Overlay latency optimal routing
    - Cooperative within an overlay, but selfish across overlays
  - Compliant (i.e. default) routing: OSPF
    - Unit, optimized, and random weights
- Performance metrics
    - User: Average latency
    - System: Maximum link utilization, network cost [FRT02]

Courtesy of L. Qiu

[Qiu'03] Horizontal Interactions



- Different routing schemes coexist well without hurting each other
  - –achieves close to optimal average latency
- Optimal average latency is achieved at the cost of overloading some links

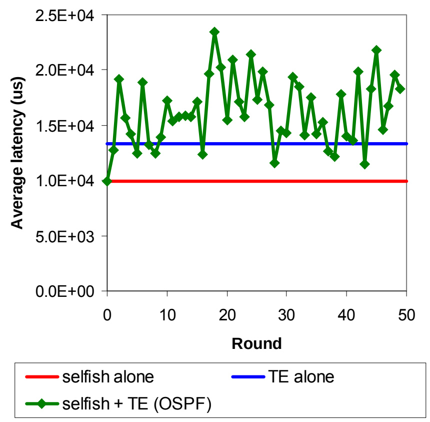
Courtesy of L. Qiu

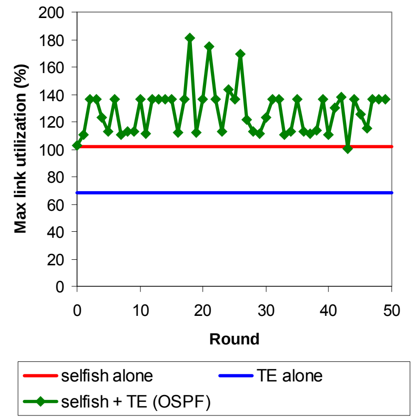
[Qiu'03] Vertical Interactions

- Vertical interaction:
  - Selfish overlays: minimize user latency
  - Traffic engineering: minimize network cost
- Question:
  - Will the system reach a state with both low latency and low network cost?

Courtesy of L. Qiu

[Qiu'03] Selfish Overlays vs. OSPF Optimizer





OSPF optimizer interacts poorly with selfish overlays because it only has very coarse-grained control.

Courtesy of L. Qiu

Interactions between Overlay & Underlay Routing

[Liu'05] Y. Liu, H. Zhang, W. Gong, D. Towsley, "On the Interaction Between Overlay Routing and Underlay Routing," *Infocom 2005*.

overlay traffic demand

Overlay Routing Optimizer
To minimize overlay cost

Player 1

flow allocation on logical links: "X"
traffic demand for underlay

flow allocation on physical routes: "Y"

Underlay Routing Optimizer
To minimize overall network cost

Player 2

non-overlay traffic demand

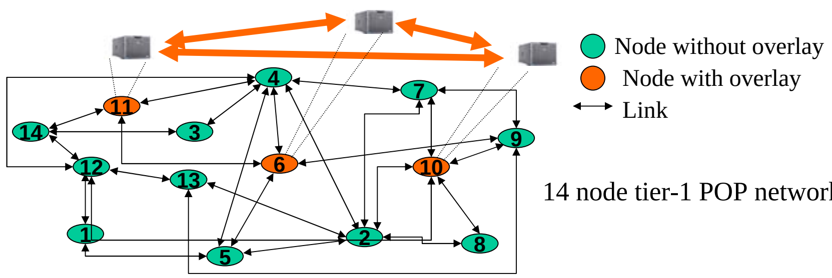
Iterative Dynamic Process

- Equilibrium: existence? uniqueness?
- Dynamic process: convergence? oscillations?
- Performance of overlay and underlay traffic?

Courtesy of Yong Liu

[Liu'05] Similar setup as previous paper

- Focus on interactions in a single AS
- Routing models:
  - Optimal underlay routing (minimize total delay for all network traffic)
  - Optimal overlay routing (minimize total delay for all overlay traffic)
  - Selfish overlay source routing
- Study interactive dynamic process in Game-theoretic framework



Courtesy of Yong Liu

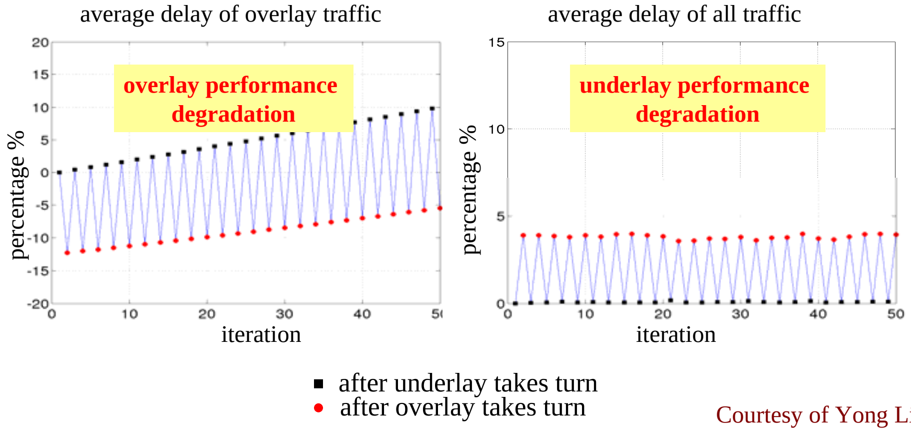
[Liu'05] Simulation Results

Iterative process

- Underlay takes turn at step 1, 3, 5, …
- Overlay takes turn at step 2, 4, 6, …

$$\frac{\text{Delay}(k) - \text{Delay}(1)}{\text{Delay}(1)} 100\%$$

$$k = 1,2,3,4,5,...$$



Courtesy of Yong Liu

[Liu'05] Game-theoretic Study

- Two-player non-cooperative, non-zero sum game

Overlay

$$\min_X J^{\text{overlay}}(X,Y)$$
$$\text{s.t. } G_{\text{overlay}}(X,Y) \geq 0$$

Underlay

$$\min_X J^{\text{underlay}}(X,Y)$$
$$\text{s.t. } G_{\text{underlay}}(X,Y) \geq 0$$

X : strategy of "overlay"
traffic allocation on logical links

Y : strategy of "underlay"
traffic allocation on physical links

$J^{\text{overlay}}(X,Y)$: cost of "overlay"

$J^{\text{underlay}}(X,Y)$: cost of "underlay"

$G_{\text{overlay}}(X,Y)$: constraint of "overlay"

$G_{\text{underlay}}(X,Y)$: constraint of "underlay"

Courtesy of Yong Liu

[Liu'05] Game-theoretic Study

- Best-reply dynamics
  - Overlay & TE take turns computing optimal strategies based on response of other players

$$X(k+1) = \arg\min_X J^{\text{overlay}}(X, Y(k))$$
$$Y(k+1) = \arg\min_Y J^{\text{underlay}}(X(k), Y)$$

$X(k)$: strategy of "overlay" at step k

$Y(k)$: strategy of "underlay" at step k

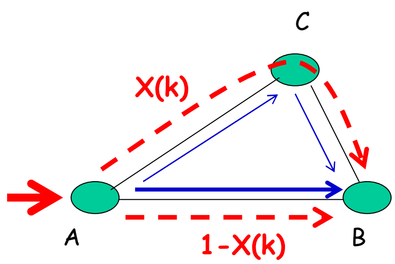
- Nash Equilibrium (X^*, Y^*)

$$X^* = \arg\min_X J^{\text{overlay}}(X, Y^*)$$
$$Y^* = \arg\min_Y J^{\text{underlay}}(X^*, Y)$$

Courtesy of Yong Liu

[Liu'05] Analysis: Optimal Underlay Routing v.s. Optimal Overlay Routing

- Overlay
  - One central entity calculates routes for all overlay demands, given current underlay routing
  - Assumption: it knows underlay topology and background traffic



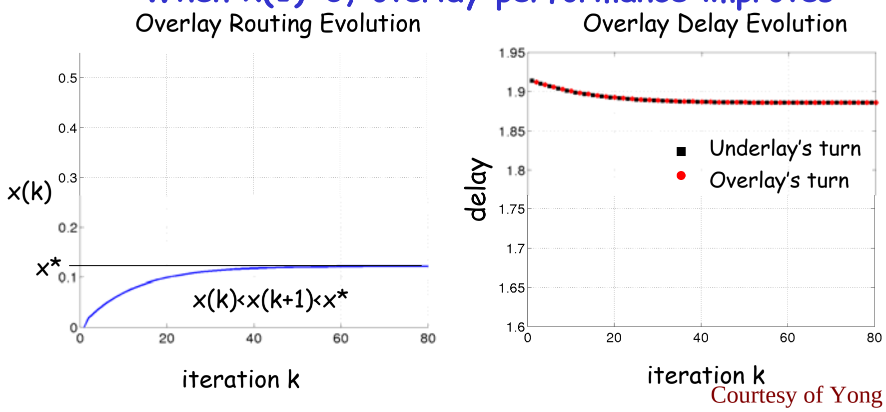
Overlay's routing decision is denoted as a single variable X(k): overlay's flow on path ACB after round k

Courtesy of Yong Liu

[Liu'05] Best-reply Dynamics

- There exists unique Nash Equilibrium Point (NEP), x*
- x* globally stable: x(k) →x*, from any initial x(1)
- Is the NEP efficient?

When x(1)=0, overlay performance improves



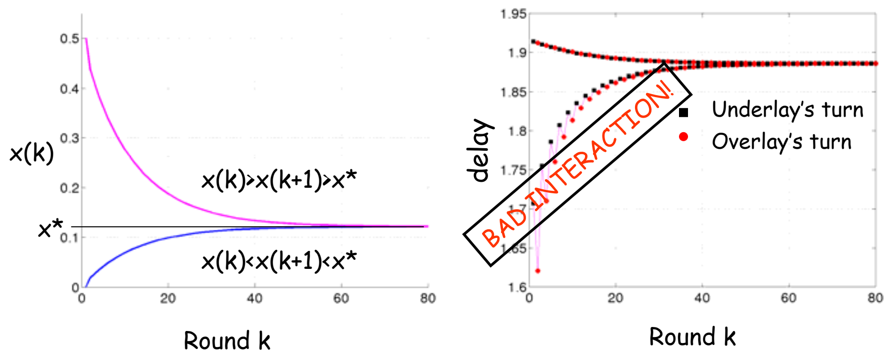
Courtesy of Yong Liu

[Liu'05] Best-reply Dynamics

- There exists unique Nash equilibrium x*,
- x* globally stable: x(k) →x*, from any initial x(1)

When x(1)=0.5, overlay performance degrades

Overlay Routing Evolution   Overlay Delay Evolution



Courtesy of Yong Liu

[Liu'05] Conclusions

- Interactions between blind optimizations at two levels may lead to lose-lose situation
  - Nash Equilibrium Point can be inefficient: overlay cost can increase even if it optimizes its routing at each round
- Selfish overlay routing can degrade performance of network as a whole
  - Overlay routing never improves TE performance
  - Average cost increase to TE depends on fraction of overlay traffic
    - Maximum cost & variation when half of the network demand is overlay traffic
  - Impact on TE cost is reduced when link capacity increases

Open Issues

- Time scales of interaction
  - TE usually happens at slower time scales than overlays
- Existence of NEP depends on topology, traffic demand patterns, etc.
  - Logical link coupling of overlay networks
- *What about the dynamics in the transient period before system stabilizes?*
- *What happens when both underlay & overlays react to external triggers like link/router failures that lead to dynamic re-routing?*

What about transient behavior?

[Ker'05] R. Keralapura, C-N. Chuah, N. Taft, and G. Iannaccone, "Can co-existing overlays inadvertently step on each other?" *IEEE ICNP*, Nov. 2005

Goals:

- Identify conditions of race conditions and compute the likelihood of synchronizations through an analytical model
  - Assuming overlay traffic is a significant portion of overall traffic
  - Validation via simulations
- Explore techniques to avoid or limit harmful synchronizations
- Provide guidelines for large-scale deployments of overlays

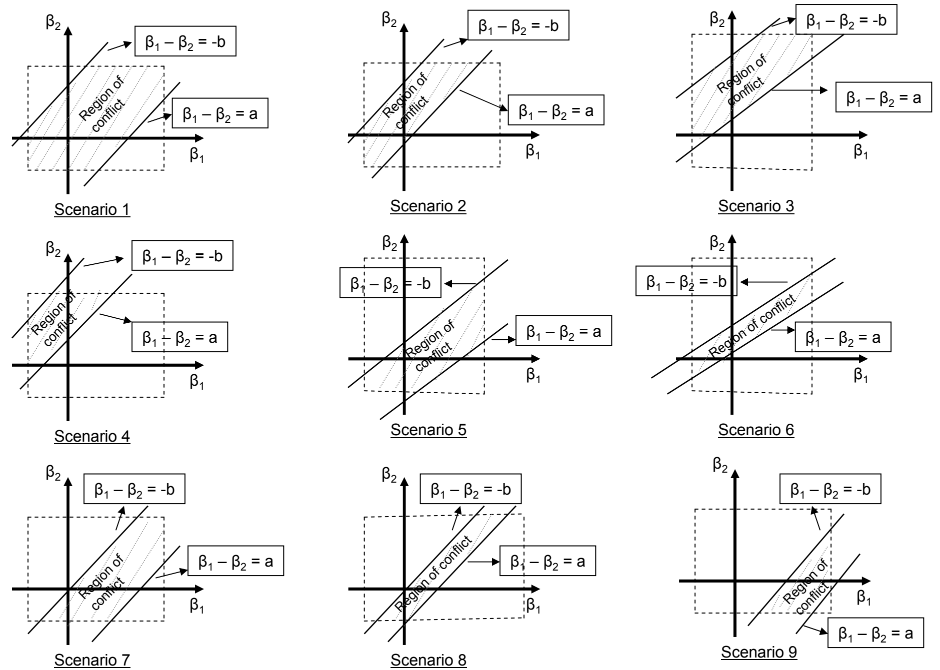
[Ker'05] Synchronization of Multiple Overlays

- Three main conditions for synchronization
  - Path performance degradation due to external triggers (e.g., failures, flash crowds)
  - Topology, i.e. partially overlapping primary and backup paths
  - Periodic path probing processes
- The first two conditions are beyond control of overlays
  - Frequent events that degrade path performance
  - Overlay node placement determines path overlap
- Focus on overlay path probing
  - How likely do two overlays get synchronized based on the parameters of their path probing procedures
    - Is it pathological or a more general problem?
  - Predicting how long the oscillations last before they disentangle

Modeling Overlay Path Probing Process

- For overlay network, *i*
  - Probe Interval – $P_{i,}$
  - Timeout – T_i
  - High Frequency Probe Interval – Q_i
  - Number of High Frequency Probes – N_i
- Additional parameter: round trip time R_{ij} over path *j*
- By definitions: $P_i \geq Q_i \geq T_i \geq R_{ij}$
- Consider two overlay networks
  - Time of occurrence of probes: β_i, $i=1,2$
  - Final high frequency probes: $f_i = \beta_i + N_i Q_i$
  - Overlays synchronize when:
    - O_1 moves traffic first. O_2 sends out the last high freq probe before O_1 moves its traffic, decides the path is bad, and move its traffic shortly after.

      $$f_1 < f_2 \quad \text{and} \quad f_2 - f_1 < T_1$$
    - Or vice versa



Probability of Synchronization/Oscillations

- Probability of Synchronization – Nine cases

$$A_C = A - A_1 - A_2$$

$$P(S) = \frac{A_C}{A}$$

– For the simplest case:

$$A = P_1 P_2$$

$$A_1 = 0.5(P_1 - R_1/2 - a + R_2/2)^2$$

$$A_2 = 0.5(P_2 - R_2/2 - b + R_1/2)^2$$

– For identical overlays

$$P(S) = T(2P - T)/P^2$$

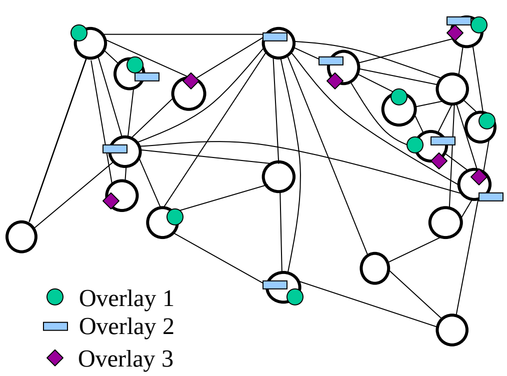
How long do oscillations last?

- Oscillations last until overlay networks
  - "Disentangle" themselves
  - "Influenced" by external event (e.g., link recovery)
- Assuming no external events
  - Bounds on the duration of oscillations and hence quantify the impact (in a probabilistic sense) on both overlay and IP traffic
- Length of oscillations

$$k = \left\lceil \frac{T_1 + T_2}{\left| P_1 - P_2 + N_1 Q_1 - N_2 Q_2 + T_1 - T_2 \right|} \right\rceil$$

Simulation Study

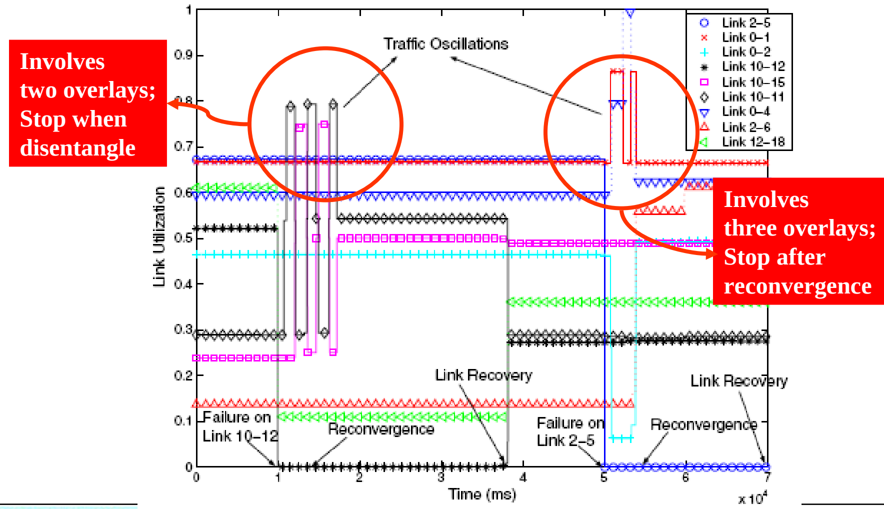
- Consider a Tier-1 ISP's pop-level topology
- Deploy five overlay networks on top of it
  - Different probing parameters, RTTs, and traffic demand



Timer	P(ms)	Q(ms)	T(ms)	N
O_1	2000	600	300	3
O_2	2000	1000	350	3
O_3	1000	500	200	3
O_4	800	400	120	3
O_5	700	300	100	3

Illustrating Race Conditions

- Oscillations in link load



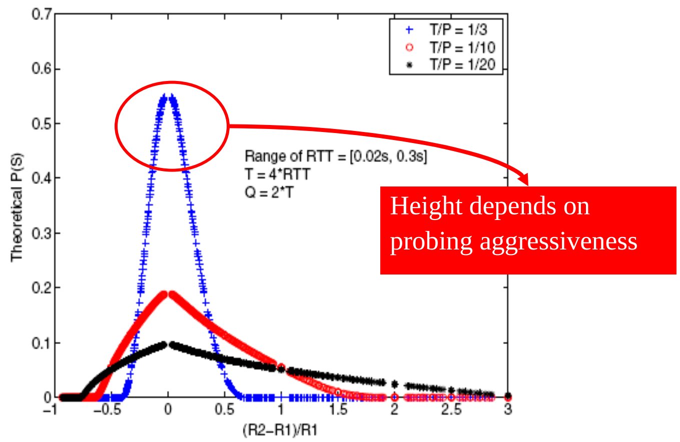
Sensitivity to Probe Parameters

- Does the inherent randomness/variation in RTT help reduce *P(S)*?
- Is *P(S)* non-negligible in common Internet operating regions?
  - Consider it non-negligible if *P(S)* > 10%
- How do we choose the parameter settings to drive P(S) low?

First, some definitions …

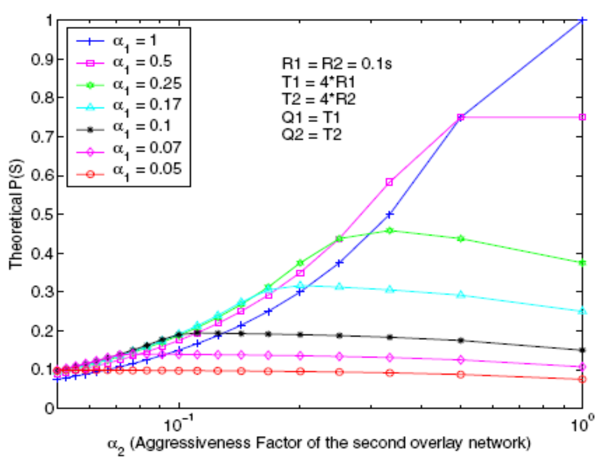
- Aggressiveness factor: $\alpha_i = T_i / P_i$
- Assume *T*=4**RTT*
- Proportional overlays:
  P & *Q* multiples of *T* (different per path)
- Fixed overlays:
  P & *Q* values are set independent of *T* and *RTT*

Proportional Overlays: Influence of RTTs



- When one RTT is more than twice the other, P(S) is close to zero.
- If two overlays span similar geographic region (similar RTTs), P(S) is non-negligible.

Proportional Overlays: Impact of Relative Aggressiveness on P(S)



- As long as one overlay is non-aggressive, P(S) is low
- Caveat: Fairness issue

How to mitigate oscillations?

- Less aggressive probing to avoid synchronization
  - Cons: fairness issues, slower reactions
- Break synchronization through randomization
  - Simply randomizing probe intervals or time-out values does *NOT* help
  - Back-off approach works better
    - i.e., successively increase the time out/probe parameters each time an overlay decides to switch to the same destination

Open Problems

- Large-scale deployment issues
  - What overlay topologies are most likely to have these problems?
  - What are the general design rules-of-thumb?
- How to share information between the IP layer and the overlays as well as among multiple overlay networks?
  - How to resolve conflicts?
  - What if one player can predict the other player's response?
- Overlay routing and inter-domain routing
  - How to contain oscillations/instability in one domain?

Outline

- Overlay Networks: Overview
- Indirection: Overlay Routing Service
- Problematic Interactions between Multiple Overlays and with IP-layers
- **Moving Forward**
  - **Productive co-existence of overlay/underlay**
  - **Combining multi-homing and overlay routing**

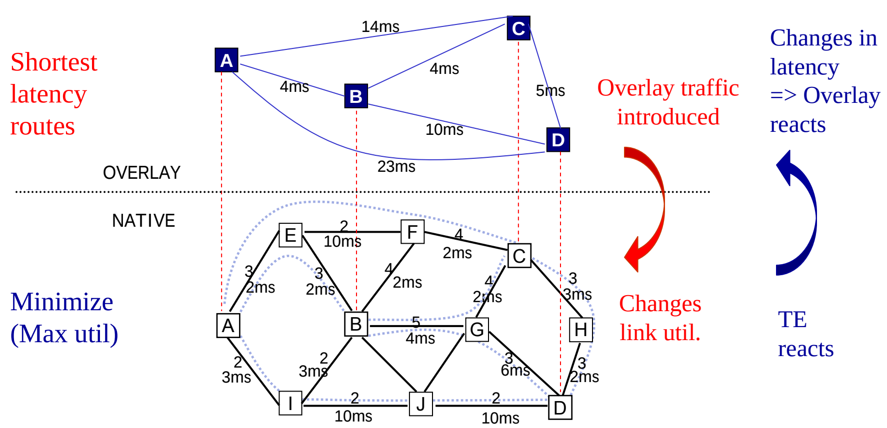
Moving Forward

- Strategies for resolving conflicts
  - S. Seetharaman, V. Hilt, M. Hofmann, and M. Ammar, "Preemptive Strategies to Improve Routing Performance of Native and Overlay Layers," *IEEE INFOCOM 2007*.
  - C. Wu, B.Li , "Strategies of Conflict in Coexisting Streaming Overlays," *IEEE INFOCOM 2007*.
- Spanning multiple AS domains
  - Z. Li, P. Mohapatra, and C-N. Chuah, "Virtual Multi-Homing: On the Feasibility of Combining Overlay Routing with BGP Routing," *IFIP Networking Conference, LNCS series, vol. 3462, pp. 1348-1352,* May 2005
  - Y. Zhu, C. Dovrolis, M. Ammar, "Combining multihoming with overlay routing (or, how to be a better ISP without owning a network)," *IEEE INFOCOM 2007*.
  - Y. Li, Y. Zhang, L. Qiu, S. Lam, "SmartTunnel: Achieving Reliability in the Internet," *IEEE INFOCOM 2007*.

Preemptive Strategies to Resolve Conflicts

- Overlay/underlay problematic interactions caused by
  - Mismatch of routing objectives
  - Misdirection of traffic matrix estimation

Illustration: Overlay Routing vs TE



The system suffers from prolonged route oscillations and sub-optimal routing costs

[See'07] Preemptive Strategies to Resolve Conflicts

- Overlay/underlay problematic interactions caused by
  - Mismatch of routing objectives
  - Misdirection of traffic matrix estimation

S. Seetharaman, V. Hilt, M. Hofmann, and M. Ammar, "Preemptive Strategies to Improve Routing Performance of Native and Overlay Layers," *IEEE INFOCOM'07*.

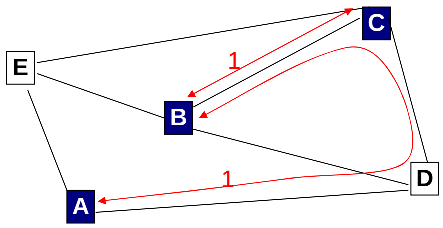
- Goals
  - Obtain the best possible performance for a particular layer
    … while steering the system towards a stable state
- Proposed solution: designate leader / follower
  - Leader will act after predicting or counteracting the subsequent reaction of the follower
  - Similar to the Stackelberg approach

[See'07] Resolving Conflict

- Challenges:
  - Incomplete information
  - Unavailable relation between the objectives
  - NP-hard prediction
- Simplifications:
  - Assume: Each layer has a general notion of the other layer's selfish objective
  - Operate leader such that
    a. Follower has no desire to change → Friendly
    b. Follower has no alternative to pick → Hostile
  - Constitutes a preemptive action
  - Use history to learn desired action gradually.

[See'07] Overlay Strategy - Friendly

- Native layer only sees a set of src-dest demands

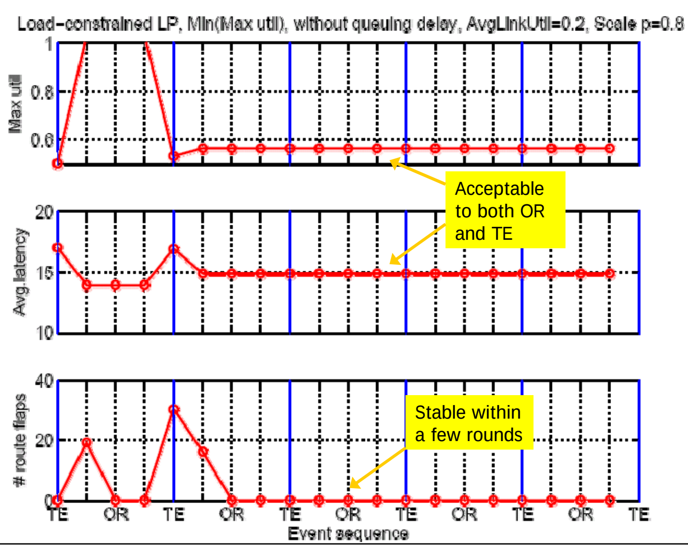


Overlay link	Traffic (Mbps)
A → B	0
A → C	1
B → C	2

- Improve latency of overlay routes, while retaining the same load pressure on the native network!

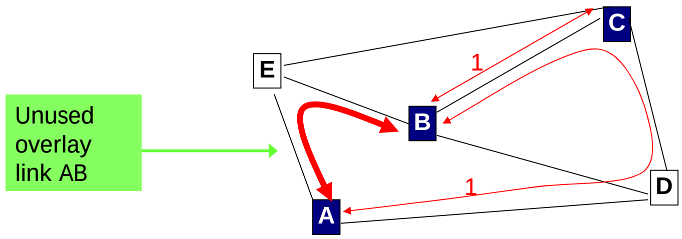
⇨ Load-constrained LP

[See'07] Overlay Strategy – Friendly (contd.)



[See'07] Overlay Strategy - Hostile

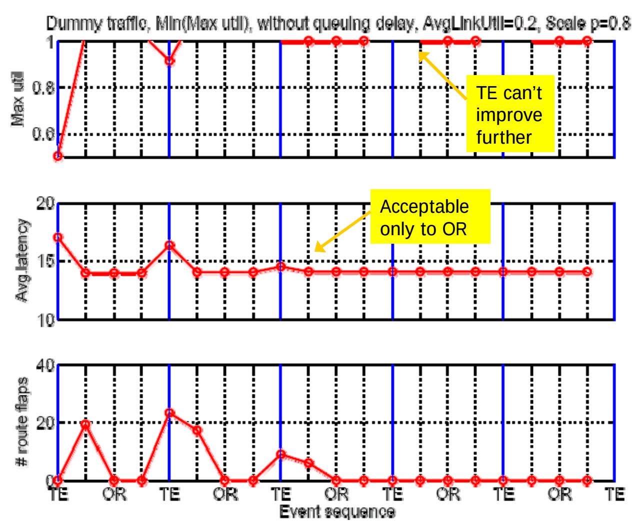
- Push TE to such an extent that it does not reroute the overlay links after overlay routing



- Send dummy traffic in an effort to render TE ineffective

⇨ Dummy traffic injection

[See'07] Overlay Strategy - Hostile (contd.)



[See'07] Preemptive Strategies: Summary

- Inflation factor
  = Steady state obj value with strategy / Best obj value achieved

← Inflation →

Leader	Strategy	Overlay	TE
Overlay	Friendly: Load-constrained LP	1.082	1.122
	Hostile: Dummy traffic injection	1.023	1.992
Native	Friendly: Hop count-constrained LP	1.027	1.184
	Hostile: Load-based Latency tuning	1.938	1.072

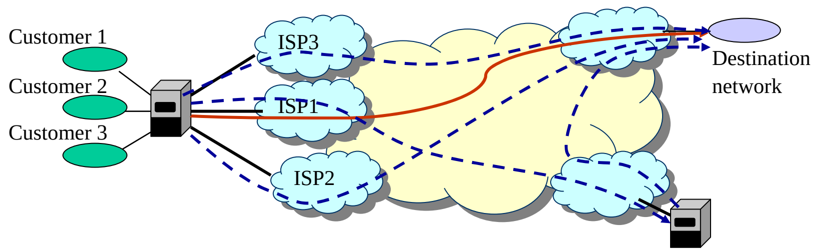
- Each strategy achieves best performance for the target layer
  - within a few rounds
  - with no interface between the two layers
  - with all information inferred through simple measurements
- If both layers deploy preemptive strategies, the performance of each layer depends on the other layer's strategy.

Remaining Open Questions

- Will such preemptive strategies work in practice?
  - With multiple co-existing overlays *and * multiple competing ISPs?
- Are there fundamental limitations in terms of overlay topologies that determine stability conditions and/or overlay performance?
  - How many overlays sharing the same native paths?
  - How many overlays per physical node?
- How dynamic can an overlay be?
  - Semi-static overlay
    vs.
    Totally on-demand, ad hoc peer-to-peer swarming

Beyond Individual AS: Inter-Domain Routing

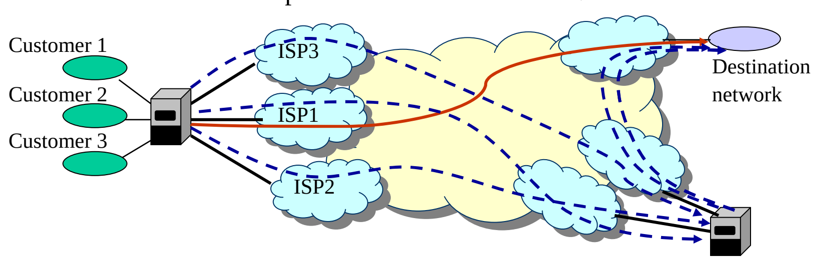
- Can improve inter-domain routing by leveraging redundant AS paths
  - Multi-homing: subscribe to multiple upstream ISPs
    - InterNAP, route science; cost $$$
  - Overlay routing: leverage redundant AS paths not permitted by IP-layer policies



Combining Multi-homing with Overlay

Y. Zhu, C. Dovrolis, M. Ammar, "Combining multihoming with overlay routing (or, how to be a better ISP without owning a network)," *IEEE INFOCOM 2007*.

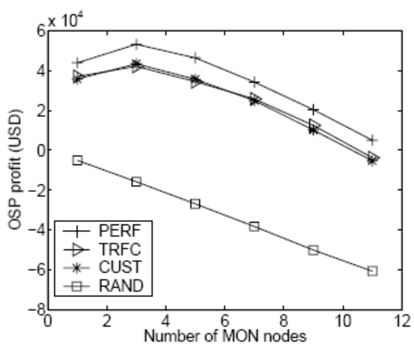
- Overlay Service Providers that manage multi-homed overlay network (MON)
  - K ISPs, N MON nodes => $K^2(N-1)$ MON indirect paths
- Questions:
  - Where to place MON nodes
  - How to select upstream ISPs for each node?

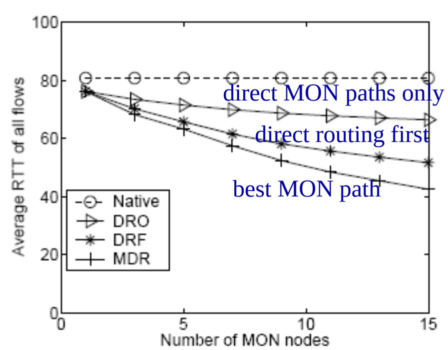


[Zhu'07] Problem Formulation & Design Heuristics

- Semi-static overlays, optimized over larger time-scales
  - Key performance metric: propagation delay
  - Input
    - Distributions of customers & traffic
    - Cost: fixed cost to operate OSP node, and cost of upstream capacity from multiple upstream ISPs
    - Profit: customer subscription cost
- Problem is NP-hard
- Design heuristics
  - RAND: Randomly select *N* MON nodes, and up to *K* ISPs
  - CUST: Place MON nodes at *N* locations with maximum number of customers. Select *K* ISPs with maximum coverage.
  - TRFC: Place MON nodes at *N* location with largest aggregate traffic volume. Select up to *K* that receive maximum customer traffic.
  - PERF: Select *N* locations and up to *K* ISPs that will turn as many flows to OSP-preferred paths (w/ lower delay) as possible.

[Zhu'07] Subset of Results





- PERF outperforms other heuristics
- OSP has lower profit when traffic is more dispersed
- OSP can reduce RTT relative to native routing with any of the three routing strategies

Issues

- How does this interact with inter-domain traffic engineering?
  - Tuning of BGP attributes and community fields
    - More effective at controlling outgoing traffic
  - Multiple players – each AS runs its own TE optimization!

Outline

- Overlay Networks: Overview
- Indirection: Overlay Routing Service
- Interactions between multiple overlays and with IP-layers
- **Discussions: Other Design Challenges**
  - **Resources Sharing & Network Virtualization**

Resource Sharing & Allocation

- Important challenge: ***How to allocate resources on the same physical nodes/paths among multiple overlays and native layer?***
  - Bandwidth, storage, compute power
  - QoS guarantees
    => need to isolate one overlay from the other
    => need to provision for faults, overloads, etc.
- Virtualization
  - Servers & storage have been virtualized to support adaptable and scalable functionalities at application-side

What about Network Virtualization?

Network Virtualization

- Decouple network functionalities from underlying infrastructure and incorporate application interests
  - Characterization related to QoS
    - Task-specific service resolution (e.g., where to find DNS)
  - Requires automated remediation and provisioning
- Challenges
  - End-to-end network path composed of many distributed elements
  - Limited means for sharing state between network entities
  - Constrained by security and trust issues
  - Lack of automated diagnosis and troubleshooting
- Example large-scale projects
  - PlanetLab Project, http://www.planet-lab.org/

PLANETLAB

- Global research network that supports the development of new network services
  - Started in 2003
  - Currently consists of 808 nodes at 401 sites
- An overlay network testbed
  - Experiment with planetary-scale services under real-world condition
  - Examples: file sharing and network-embedded storage, content distribution networks, routing and multicast overlays, QoS overlays, scalable object location, anomaly detection mechanisms, and network measurement tools

NSF GENI Initiative

- Global Environment for Network Innovations (GENI)
  - Promote innovative research without constraints of existing Internet design (ability to start from scratch!)
  - Global experimental facility that may evolve into the next Internet
  - Enable multiple researchers to run experiments across all layers

Sounds like overlays!?

- GENI-related development efforts, http://www.geni.net/dev.html
  - VINI: Virtual Network Infrastructure. J. Rexford and L.Peterson
  - Prototyping for Wireless Virtualization and Wired-Wireless Virtualization. D. Raychaudhuri, S. Paul, M. Gruteser, and I. Seskar
  - Time-Based Wireless Virtualization. S.Banerjee.

Other Overlay Services & Applications

- Content distributions, e.g., *Akamai*
- Overlay multicast & streaming
- Mobility support
- Collaborative overlays to improve reliability/security
  - Co-DNS: make DNS lookup faster and more reliable http://codeen.cs.princeton.edu/codns/
  - DoX: detect and prevent DNS cache poisoning
    - L. Yuan, K. Kant, P. Mohapatra, and C-N. Chuah, “A Proxy View of Quality of Domain Name Service,” *IEEE INFOCOM’07*.

Questions & Comments?

- E-mail: chuah@ucdavis.edu