

Multipath Resource Management in Overlay Networks

Ph.D. Dissertation Defense

Zheng Ma

Advisor: Arvind Krishnamurthy

Committee: Y. Richard Yang

Joan Feigenbaum

Ravi Sundaram (Northeastern)

Collaborators (2003-2006)

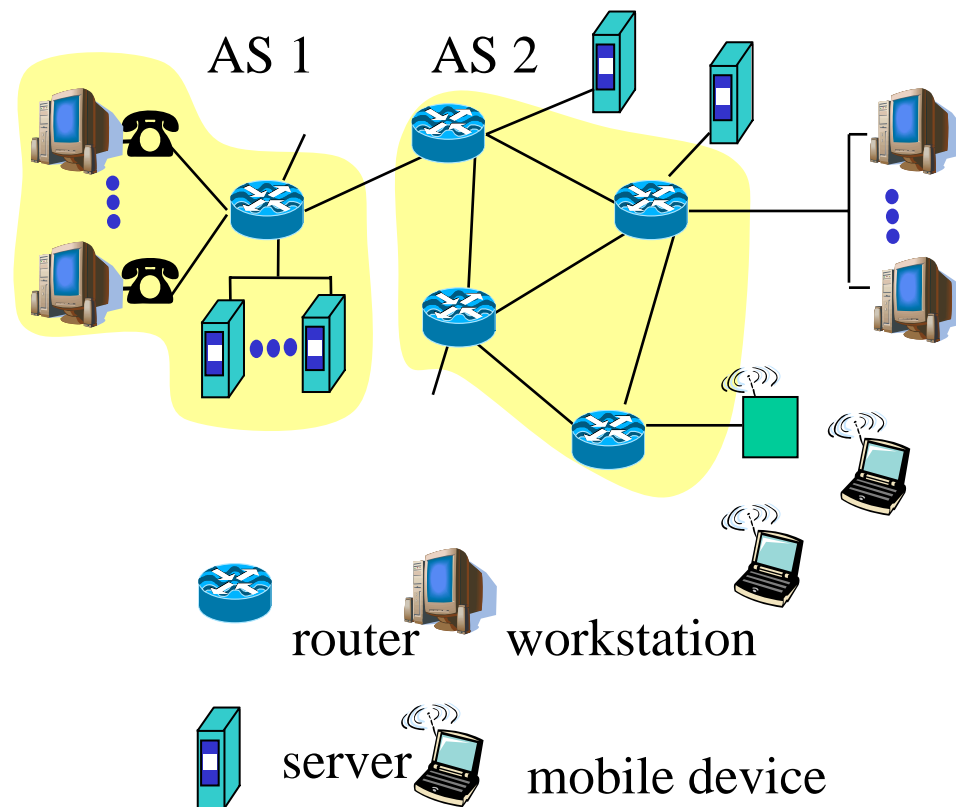
- Daria Antonova (Northeastern)
- Jiang Chen
- Arvind Krishnamurthy
- Larry Peterson (Princeton)
- Huai-Rong Shao (Mitsubishi Electric Research Laboratories)
- Chia Shen (Mitsubishi Electric Research Laboratories)
- Avi Silberschatz
- Ravi Sundaram (Northeastern)
- Hao Wang
- Randolph Y. Wang
- Anthony Young
- Y. Richard Yang

Outline

- Background and motivation of using overlay networks
- Challenges of using overlays and our solutions
 - Resource management in a single service overlay
 - Resource allocation for multiple overlays
 - Integrated traffic engineering in underlay with overlay support
- Conclusions and future work

Current Internet

- Internet is a worldwide, publicly accessible network of networks
 - 439 Mil **end hosts** in July 2006: running various applications
 - Interconnected with **physical links** (fiber, copper, radio, satellite) and **routers**
 - Routers control the **routing** for end-hosts



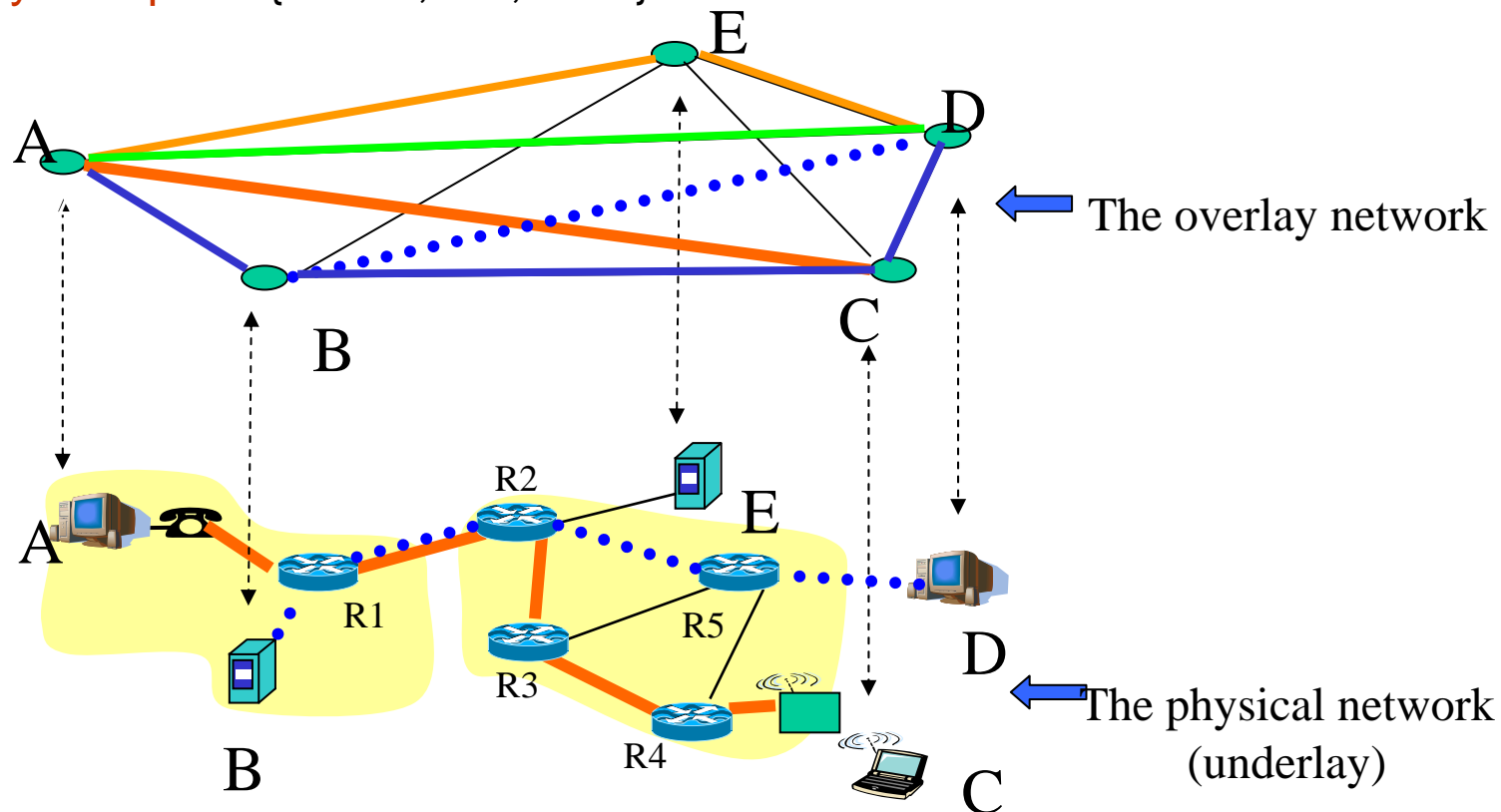
Overlay network illustration

Overlay network: A virtualized network built over an existing network

Overlay link (physical path): AC ($AR_1R_2R_3R_4C$), BD ($BR_1R_2R_3R_5D$)

Overlay path: AB+BC+CD and AE+ED

Overlay multipath: {ABCD, AD, AED}

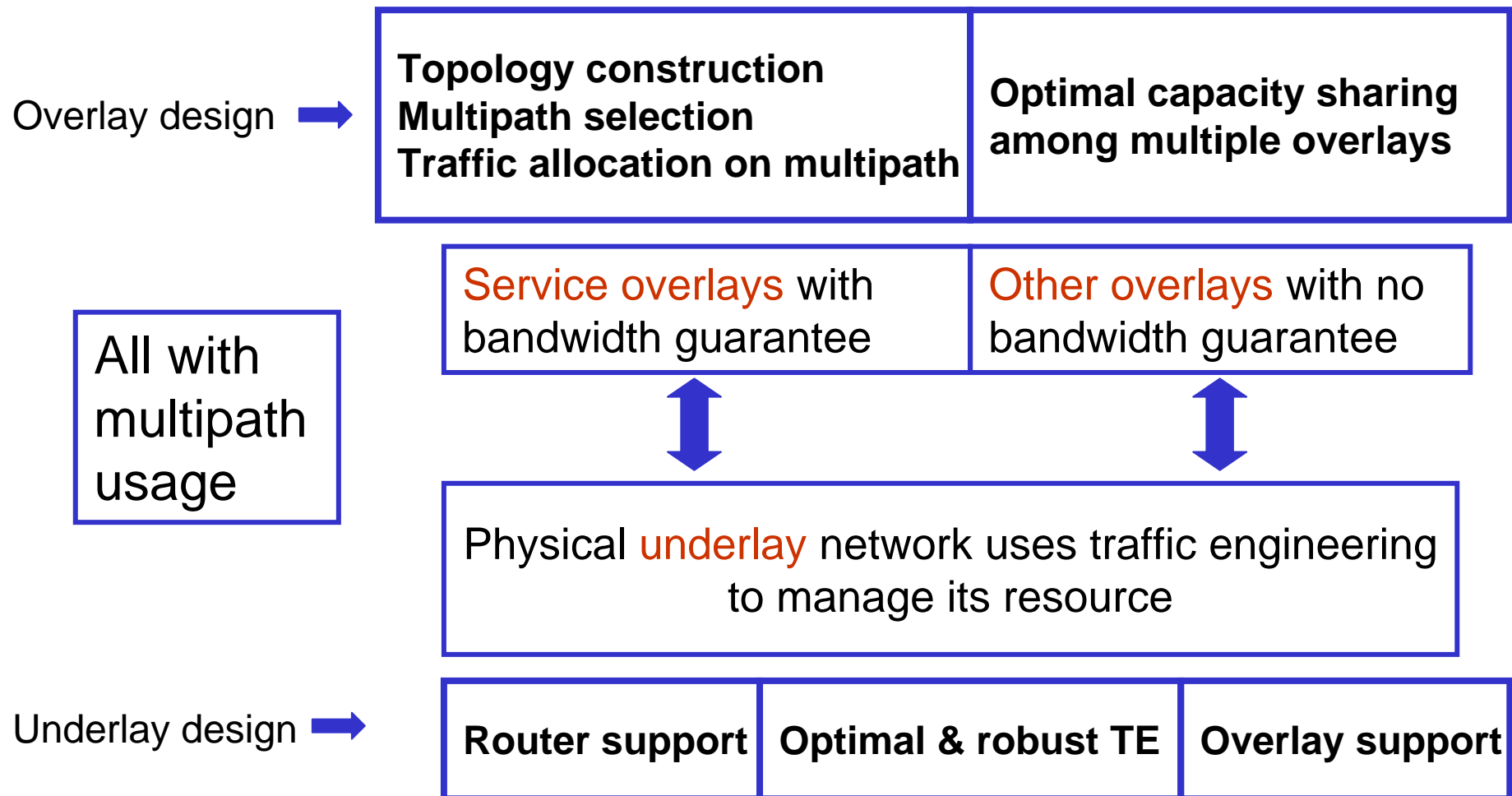


Overlay network: examples

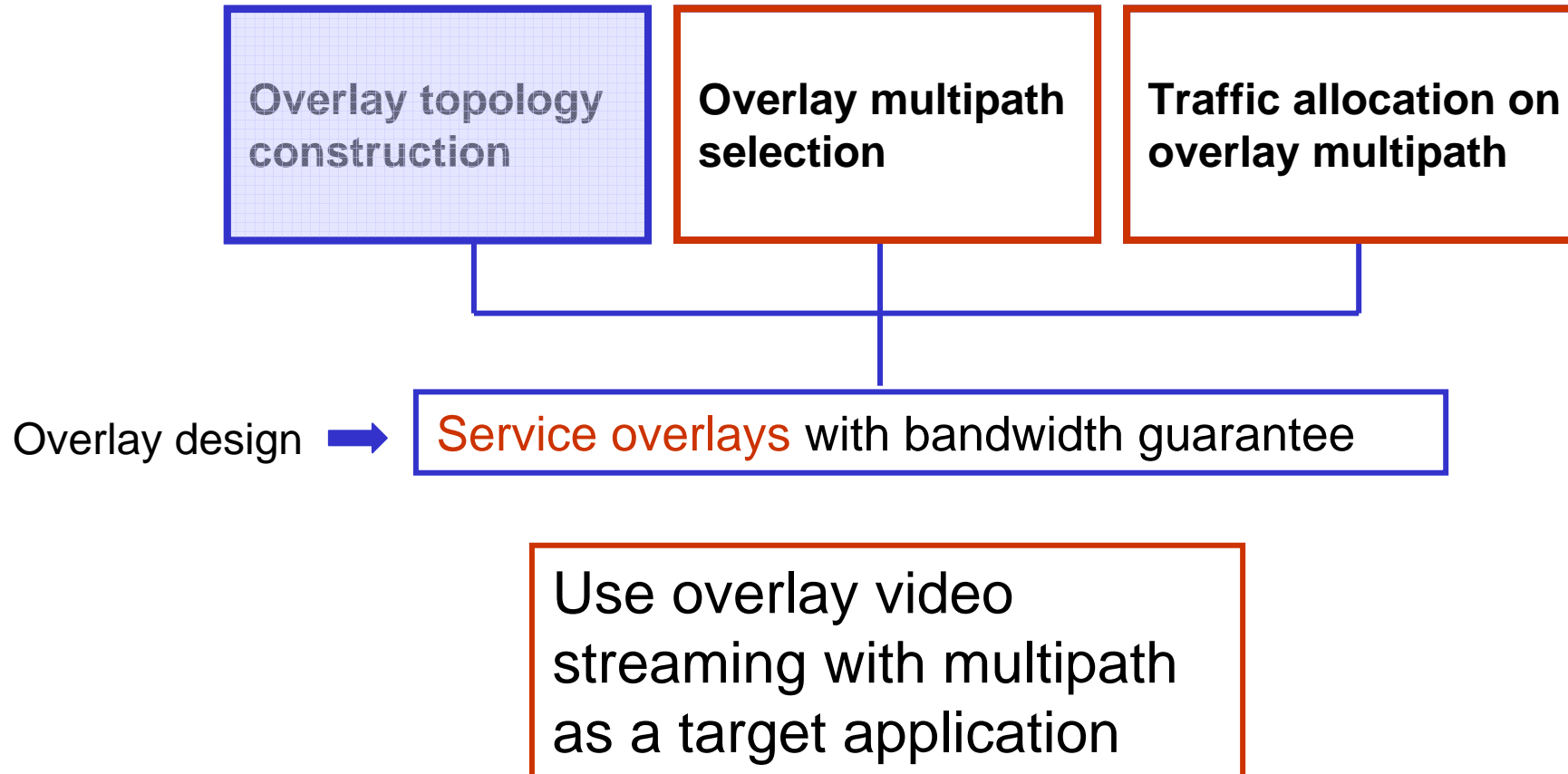
- *Akamai*: content distribution
 - *Virtual Private Networks*: provide secure and guaranteed service
 - *Emulab*: testbed networks
- *KaZaA, Bittorrent, Skype*: file sharing, IP Phone
 - *ESM, Overcast, PPlive*: multicast, video streaming, IP TV
 - *Onion routing* (anonymity), *SOS* (security), *i3*(mobility) and many more ...

Roadmap of the dissertation

Resource management in overlay networks is to achieve efficient and effective resource discovery, selection and allocation

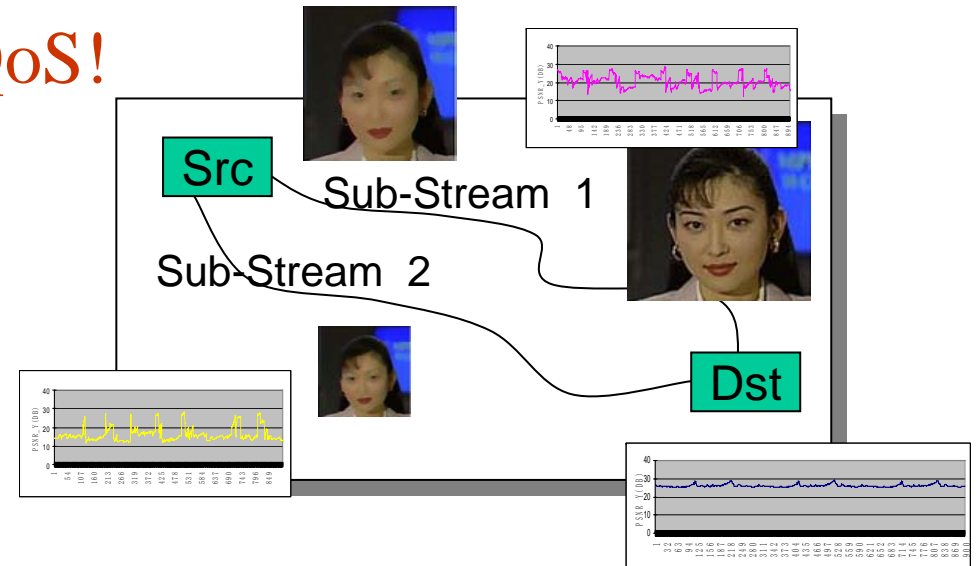


What's next?



Overlay video streaming with multipath

- Why multipath? **Improve QoS!**
 - Overcome burst losses
 - Load balancing among paths



Source coding part:

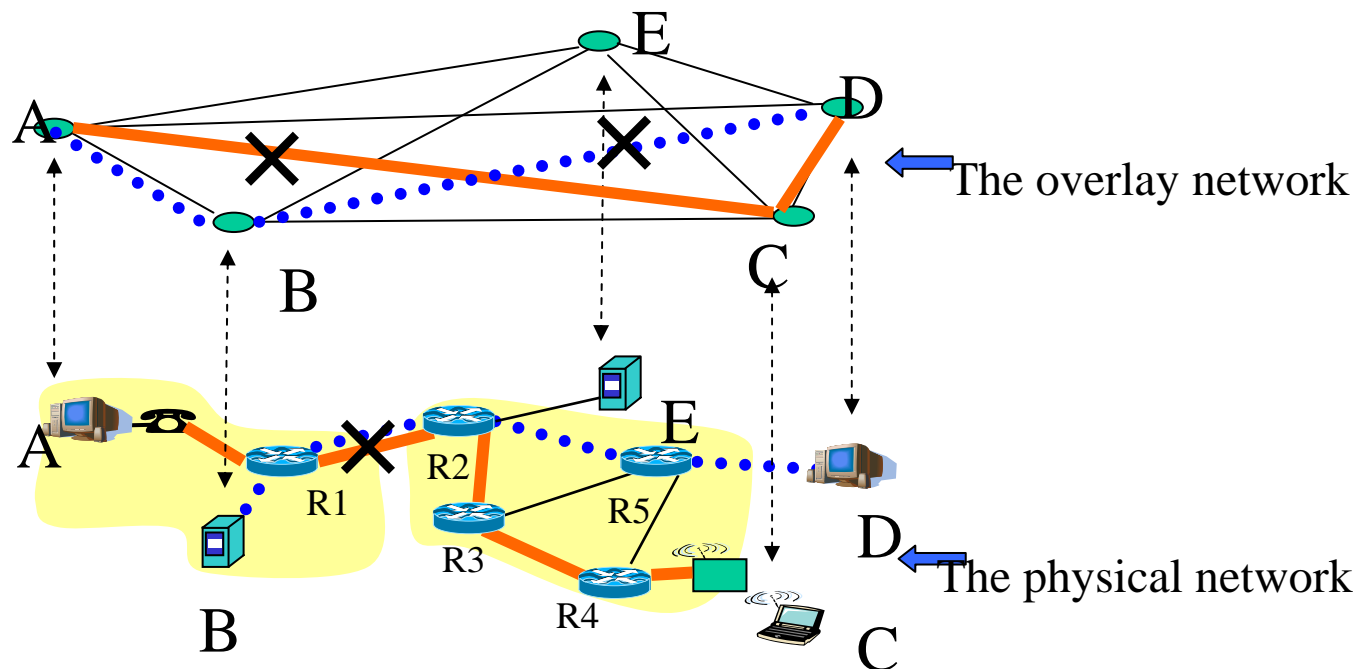
- How to generate multiple sub-streams

Network support part:

- How to find multiple paths with QoS provisioning
- How to allocate traffic on multiple paths

Challenges of selecting overlay multipath

- End-host don't know enough information about **underlying network** such as topology, available bandwidth, latency, loss rate, shared bottleneck and etc.



Multipath selection: previous methods

- Try to minimize sharing of physical links
 - Totally disjoint overlay paths [Begen *et. al* '03]
 - Heuristic which gives penalty on sharing physical links [Apostolopoulos *et. al* '02]
- Limitations of previous methods:
 - Correlation between paths as link jointness.
 - Good links with enough bandwidth are only used once.
 - Inefficient path selection
 - Usually enumerate all paths from source to destination, then find the minimal disjoint pair of paths

Our approach [MSS'04]

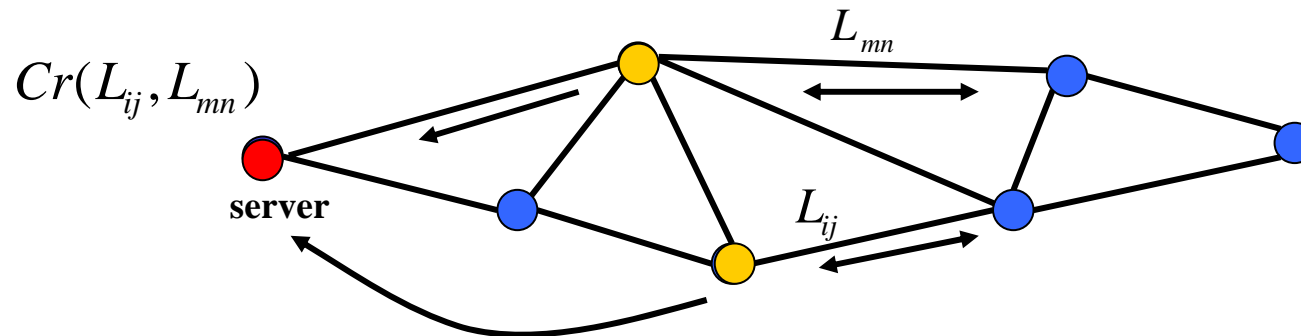
- A new efficient multipath selection scheme
 - The scheme is based on our proposed *path correlation model*
 - *New link QoS correlation metric*
 - *Path correlation model based on this metric*
 - An efficient path selection algorithm:
 - Correlation Cost Routing

A new QoS metric

- Define a new QoS metric, *overlay link correlation* for each overlay link pair:

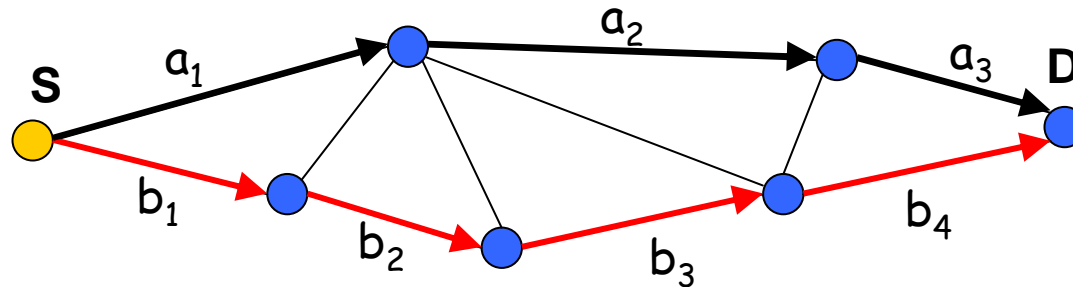
$$Cr(L_{ij}, L_{mn}) = \frac{E[(L_{ij} - \bar{L}_{ij})(L_{mn} - \bar{L}_{mn})]}{\sqrt{E(L_{ij}^2) - (\bar{L}_{ij})^2} \sqrt{E(L_{mn}^2) - (E(\bar{L}_{mn}))^2}} = \frac{Cov(L_{ij}, L_{mn})}{\sqrt{Var(L_{ij})} \sqrt{Var(L_{mn})}}$$

- This can be done by a monitoring system:



Path correlation model

- Define path correlation:

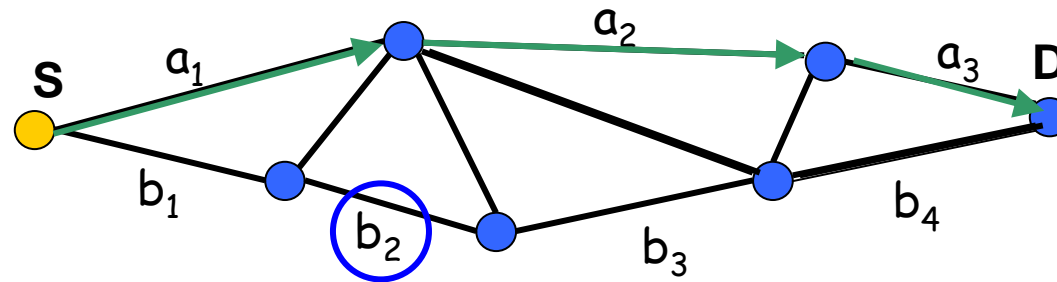


$$Cr(\overline{Path_a}, \overline{Path_b}) = \sum_{i=1}^3 \sum_{j=1}^4 Cr(a_i, b_j)$$

- The minimal correlated multipath selection problem is *NP-Hard*

Correlation Cost Routing

- Our *correlation cost routing* algorithm:
 - Key idea: finding the paths one by one by updating the link metric info
 - Update link metric using **Correlation Cost** for each link after first path



Update b_2 **Correlation Cost** respect to previous paths:

$$Cr_{b_2}^a = Cr(a_1, b_2) + Cr(a_2, b_2) + Cr(a_3, b_2)$$

- New path calculated by Dijkstra algorithm

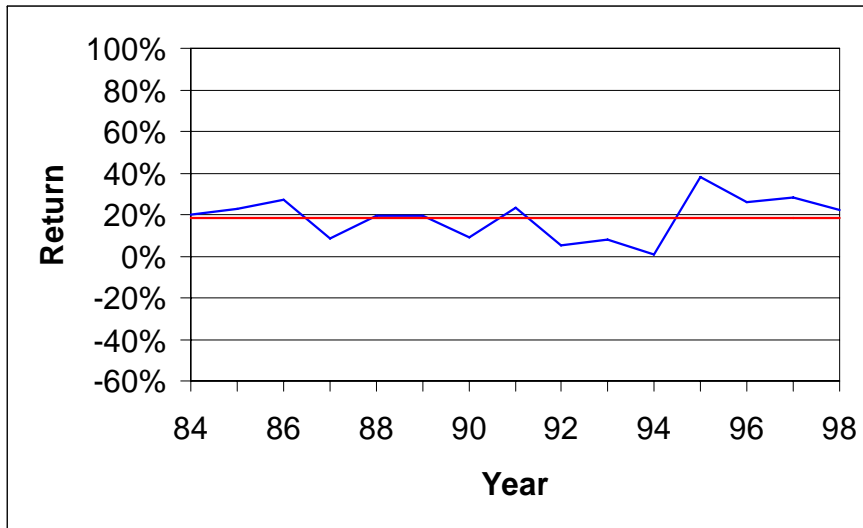
Traffic allocation on overlay multipath

[AKMS'04]

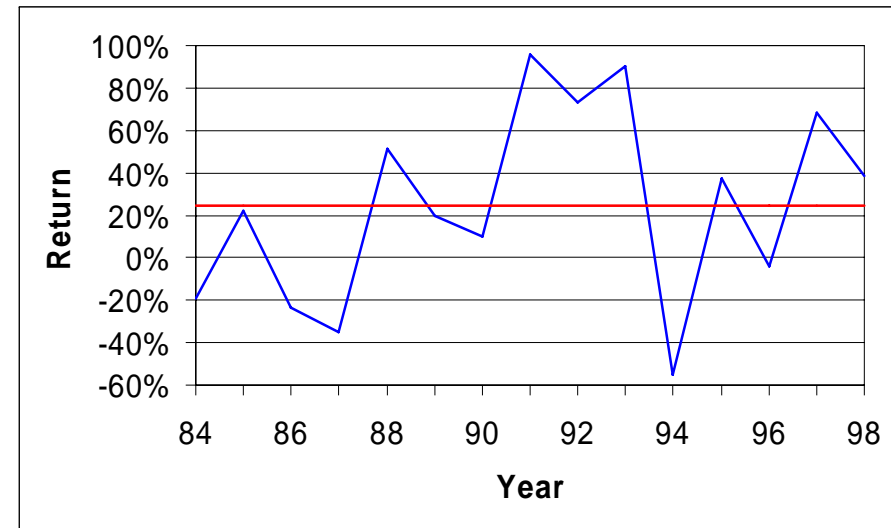
- After selecting the path, we need to decide how much traffic to send along each path
- Assume that statistical information of correlation among overlay paths is available
 - Such as expected value, variance, and covariance of a desired metric are given using previous monitoring system
- How do we allocate traffic on correlated network paths?
 - **Some analogy from portfolio theory**: build a more robust combination from constituent stocks that suffer from high variance

Portfolio theory

Exxon



Southwest Airlines



$$\mu_{\text{Exxon}} = 18\%, \mu_{\text{Southwest}} = 25\%$$

$$\sigma_{\text{Exxon}} = 20\%, \sigma_{\text{Southwest}} = 40\%$$

- Correlation of the two stocks is 0.2
- Low risk portfolio will have higher portion of Exxon stocks
- High return portfolio will have higher portion of Southwest stocks

In our setup

- Problem Setting
 - sending non-redundant data through multiple paths
 - application has a desired expectation for average latency
 - application wants to minimize variance in latency
 - has a candidate set of paths, want to determine the allocation of traffic to paths
- Notation:
 - path latencies (vector of random variables): L
 - expected values of path latencies: $E(L)$
 - covariance matrix of path latencies: $C(L)$
 - desired average latency: μ
 - path allocation decision (fractions of weights allocated to different paths): w

Optimization problem

- Solve the following problem:

$$\text{minimize } \mathbf{w}^T \mathbf{C}(\mathbf{L}) \mathbf{w} \quad (\text{minimize variance})$$

subject to:

$$\mathbf{w}^T \mathbf{E}(\mathbf{L}) = \mu \quad (\text{meet expectation})$$

$$\mathbf{w}^T \mathbf{1} = 1 \quad (\text{allocation constraint})$$

$$\mathbf{w} \geq 0 \quad (\text{no "shorting"})$$

- Instance of a quadratic programming problem
- Fortunately, it is solvable in polynomial time if $\mathbf{C}(\mathbf{L})$ is a positive semi-definite matrix (or $\mathbf{x}^T \mathbf{C}(\mathbf{L}) \mathbf{x} \geq 0$)
- $\mathbf{w}^T \mathbf{C}(\mathbf{L}) \mathbf{w}$ is simply the variance of the latency of the multipath and is guaranteed to be non-negative

Evaluation: Foreman sequence

Original Video



Transmitted by Single Path



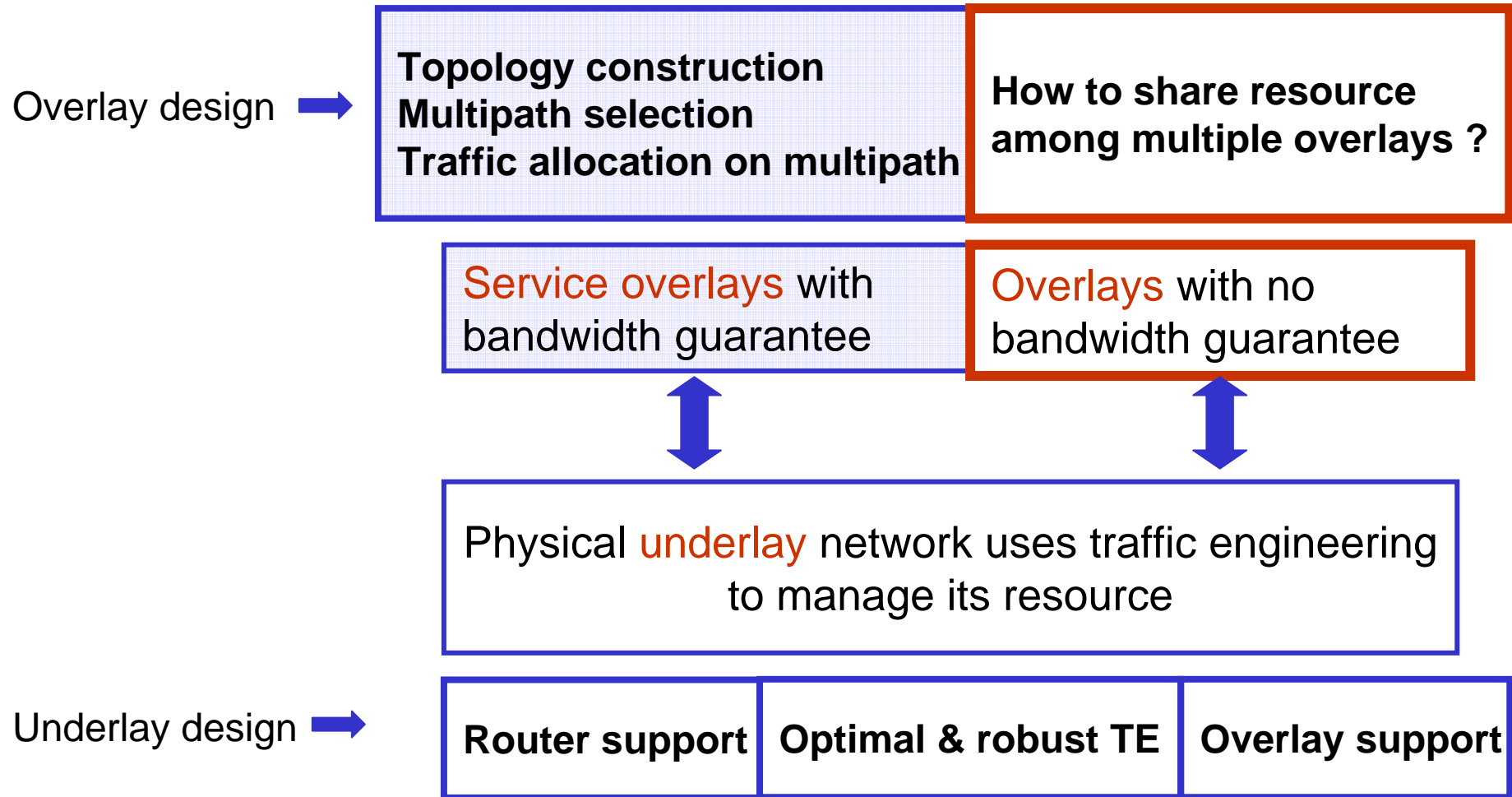
Link Disjoint Multi-Path



Minimal Correlated Multi-path

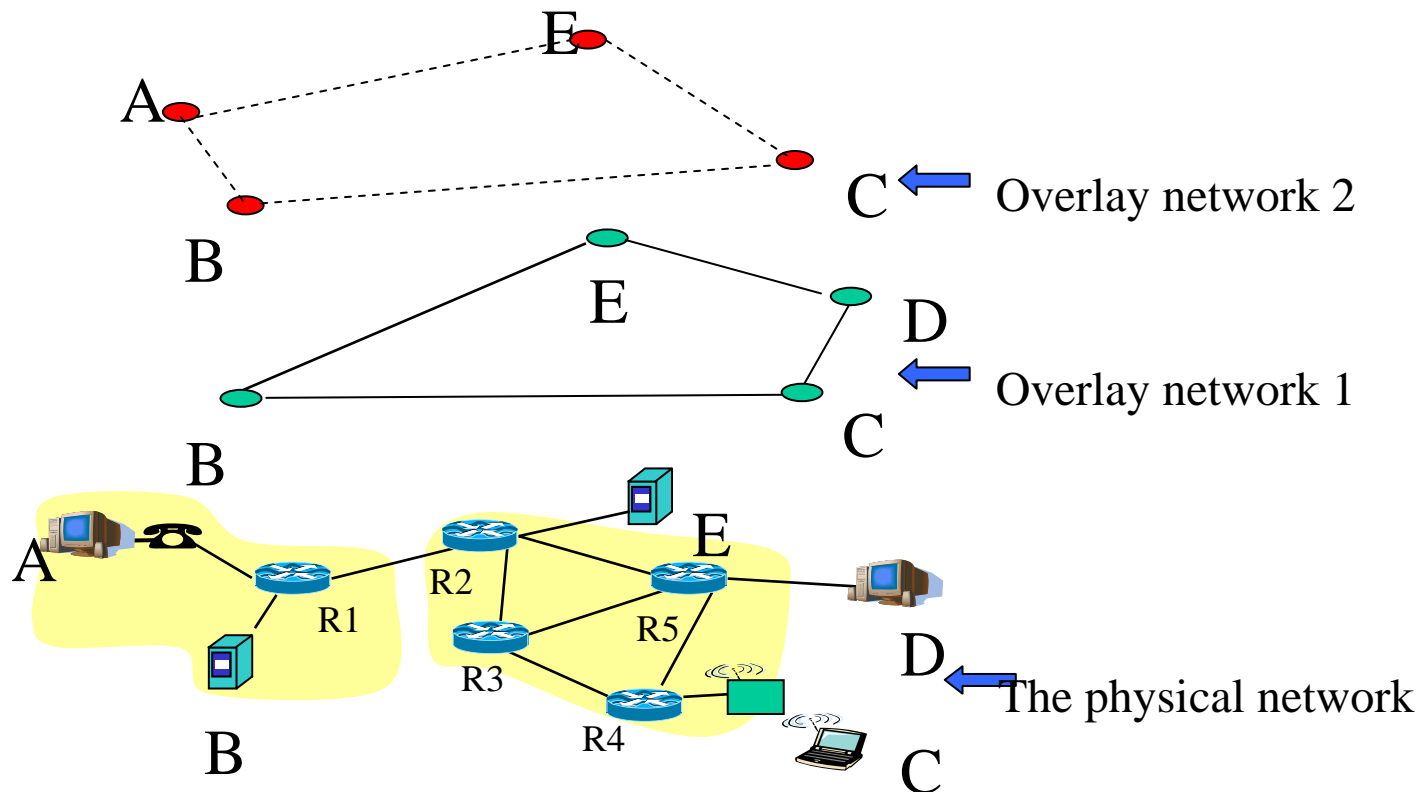


What's next



Motivation

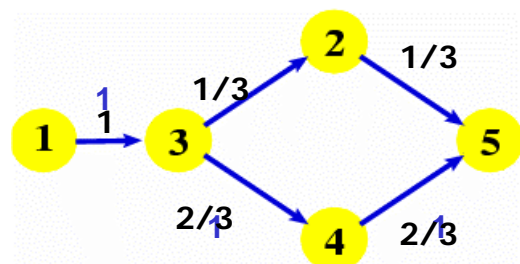
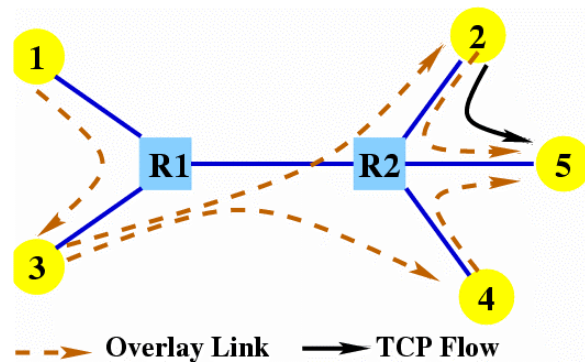
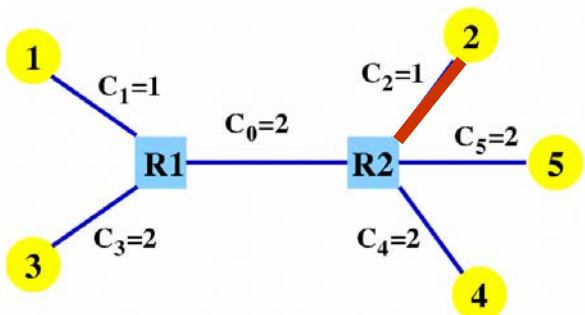
- Overlay traffic without bandwidth guarantee is continuously on the rise, e.g. p2p traffic (> 60% of overall Internet traffic)
- Each overlay uses complicated rate control based on self-interest. How can they share the resource?



Rate control of overlays: current practice

- No congestion control
 - Each overlay individually probes for available network resource and makes the rate control decision on their own
 - Network collapses
 - ISP will limit the rate
- Use TCP at each overlay link
 - e.g. Skype and BitTorrent
 - If the flow rate on each link is controlled by TCP without coordinating with other links of the same overlay application, we refer to such a scheme as *end-to-end* rate control
 - *Is this enough?* **NO!**

Sub-optimal sharing using only end-to-end control: overlay max flow example



Topology of Overlay O_1

$$x_1 = (x_{13}^1, x_{32}^1, x_{34}^1, x_{25}^1, x_{45}^1)$$

$$x_2 = x_{tcp} = x_{25}^2$$

$$f_1(x_1) = x_{25}^1 + x_{45}^1$$

$$f_{tcp}(x_{tcp}) = x_{25}^2$$

$$U_1(f_1(x_1)) = \log(x_{25}^1 + x_{45}^1)$$

$$U_{tcp}(f_{tcp}(x_{tcp})) = \log(x_{25}^2)$$

The system optimum is

$$x_1 = (1, 0, 1, 0, 1), x_2 = 1, \text{ total utility } 0$$

With only **end-to-end rate control**:

$$\text{Equilibrium: } x_1 = (1, 1/3, 2/3, 1/3, 2/3), x_2 = 1/3, \text{ total utility } -0.48$$

Our contributions [MCYK'06]

- We propose *overlay flows* control to coordinate the rate control of each overlay link distributively
- Key idea:
 - Solving the overlay utility maximization *system problem* in a distributed and iterative way
 - Using a pricing based method
 - We don't require the knowledge of the underlay networks. Instead we use a “try-and-back-off” approach

$$P: \text{maximize} \quad \sum_{i=1}^n U_i(f_i(x_i))$$

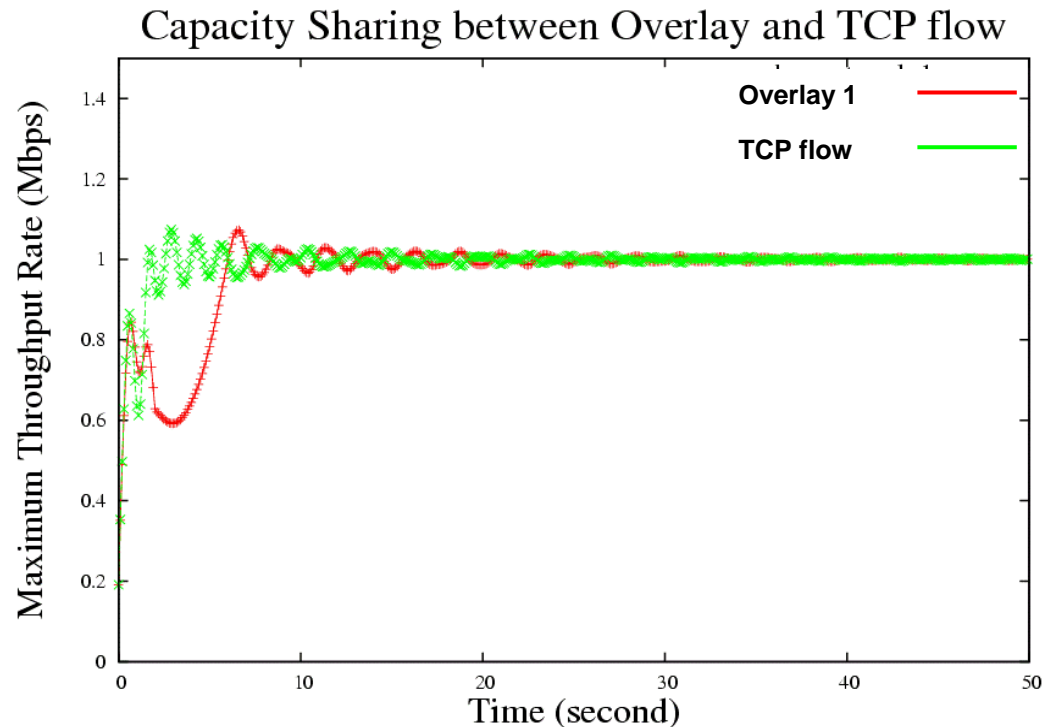
$$\text{subject to} \quad Ax \leq C$$

$$Fx = 0$$

$$\text{over} \quad x_e \geq 0, \forall e.$$

Evaluation: convergence

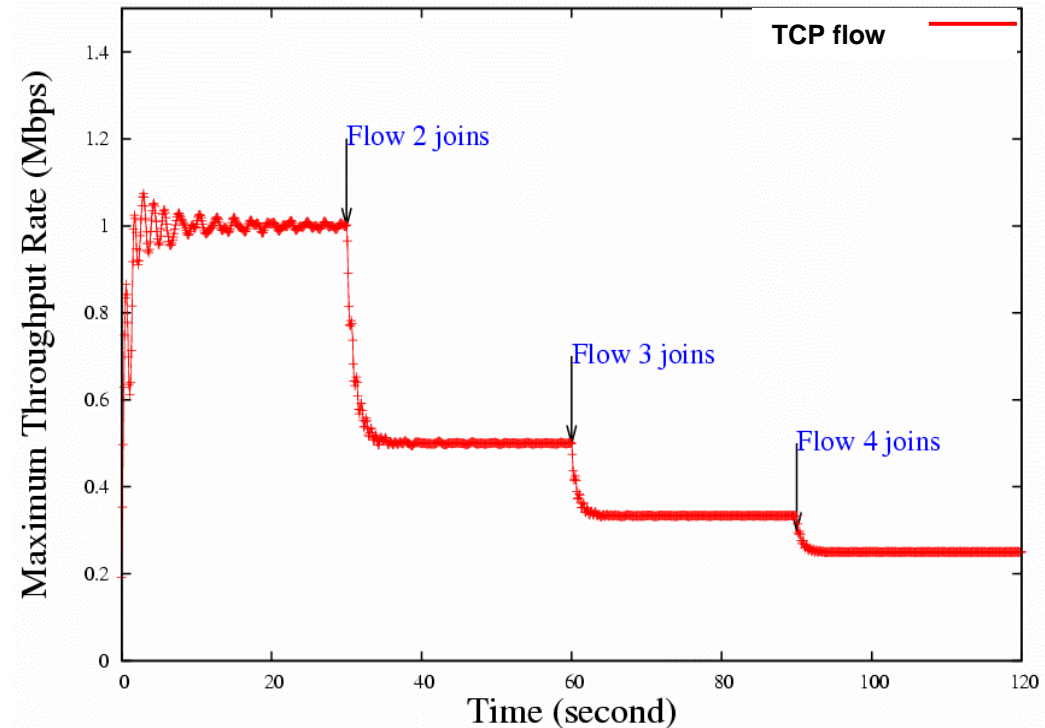
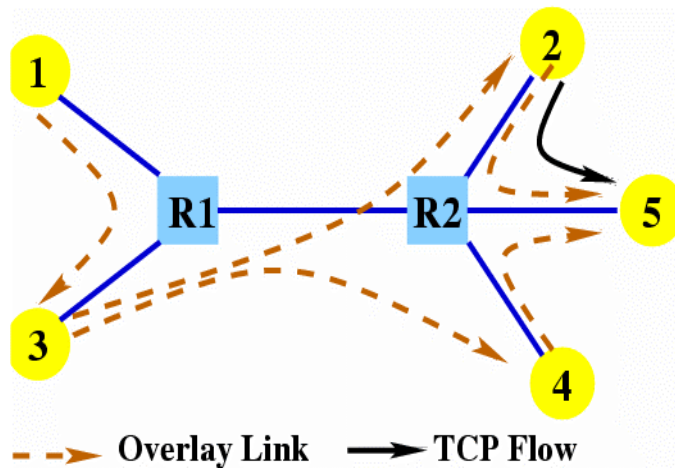
- Use the previous algorithm for overlay maximum flow example. More experiments on other topologies show similar results



The algorithm can converge to the system optimum quickly.
Proof of convergence using Lyapunov theory is in the dissertation

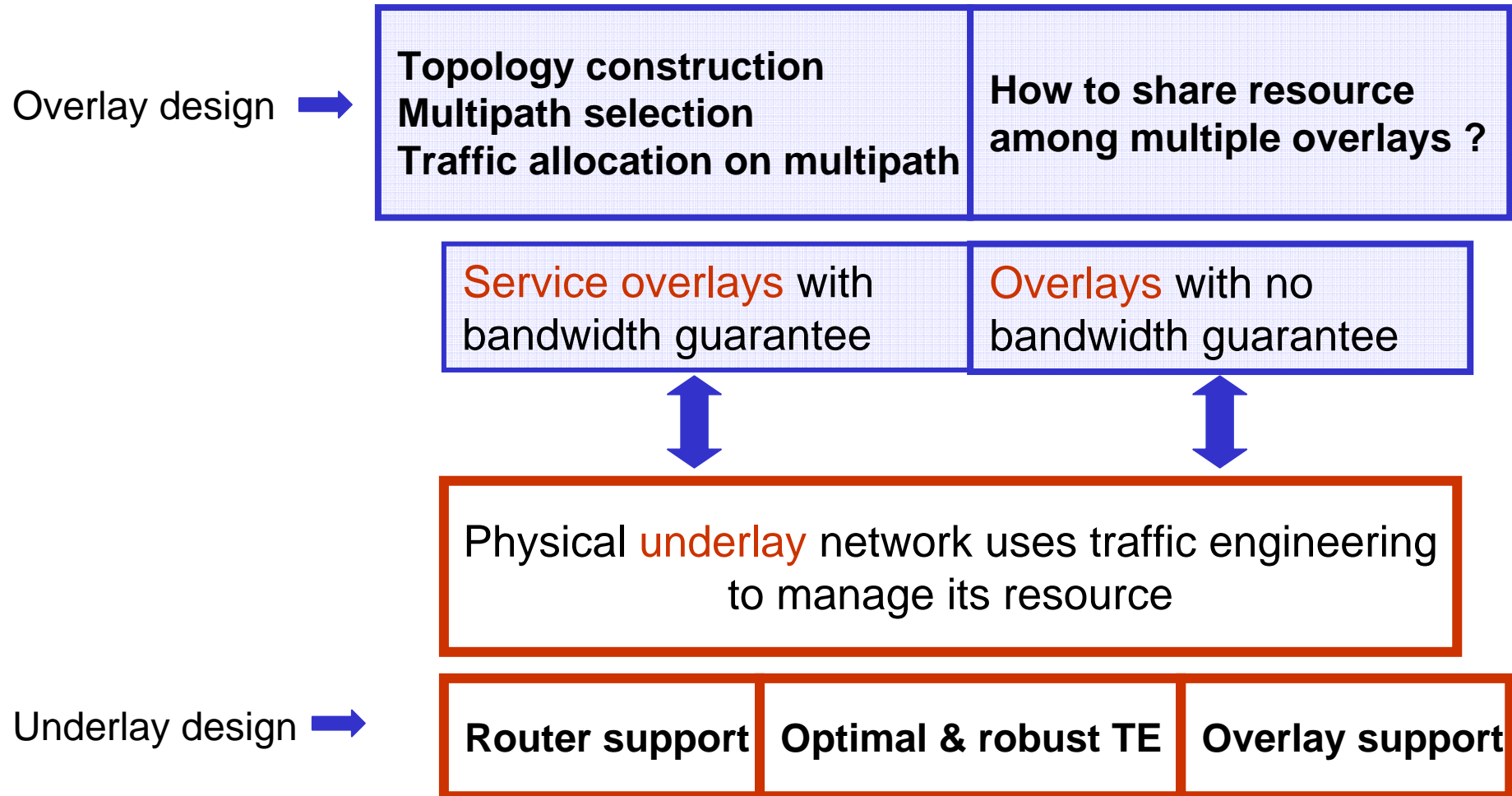
Evaluation: dynamics

In previous example, add more TCP flows between node 2 and node 5 at different time



The algorithm can react to the change and converge to the new fair share quickly. More evaluations can be found in the dissertation.

What's next



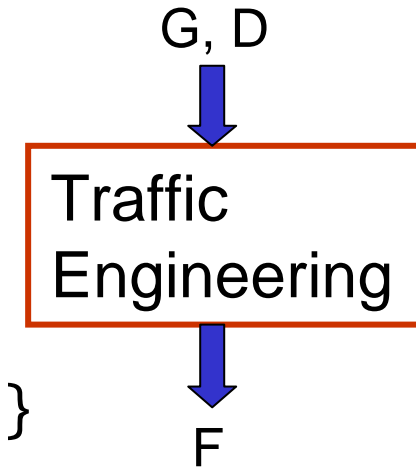
Overlay's influence on underlay

- Overlay traffic makes the traffic demand more dynamic [Qiu *et. al* '03]
 - Overlay constantly probes for available network resource and use overlay routing to redirect traffic
 - Internet traffic is highly unpredictable already! We identified sudden traffic spikes in real traces of several networks
 - *Need to handle the unpredicted traffic*
- Service overlays need the underlay to *reserve the bandwidth* for all of their possible traffic

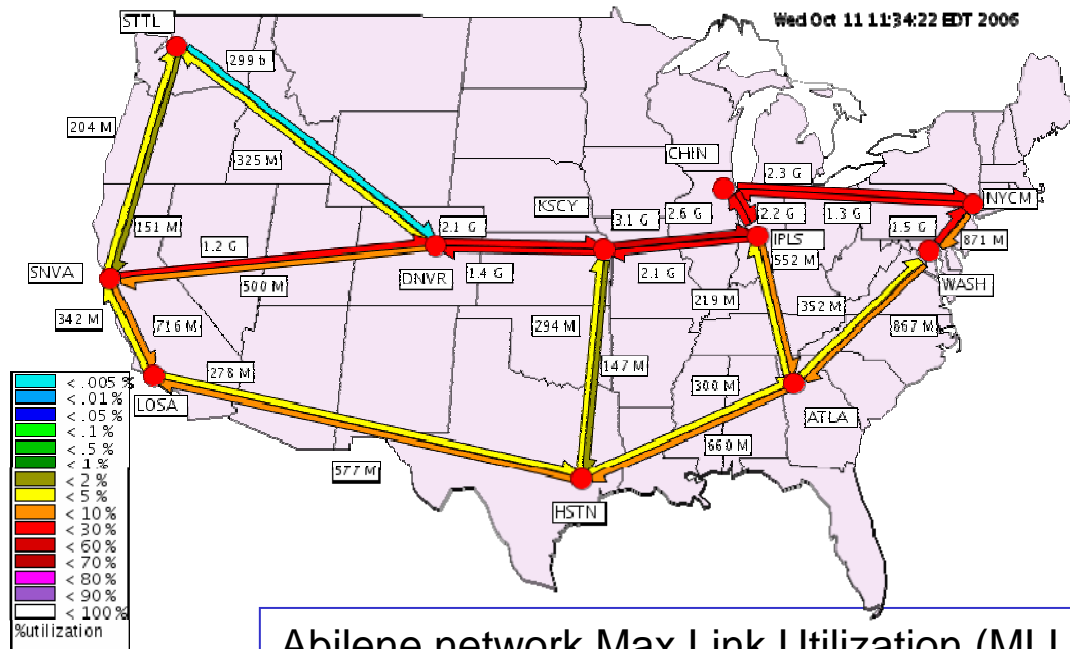
Traffic Engineering (TE) in underlay

A black box view:

- Input 1: Network topology, $G = (V, E)$
- Input 2: Traffic matrices (TMs), $D = \{ d_{ab} \mid a, b \in V \}$
- Output: Link-based routing, $F = \{ f_{ab}(i, j) \mid a, b \in V, (i, j) \in E, 0 \leq f_{ab}(i, j) \leq 1 \}$



Good for overall network objective
e.g. MLU



Abilene network Max Link Utilization (MLU) <http://abilene.internet2.edu/>

TE algorithm examples

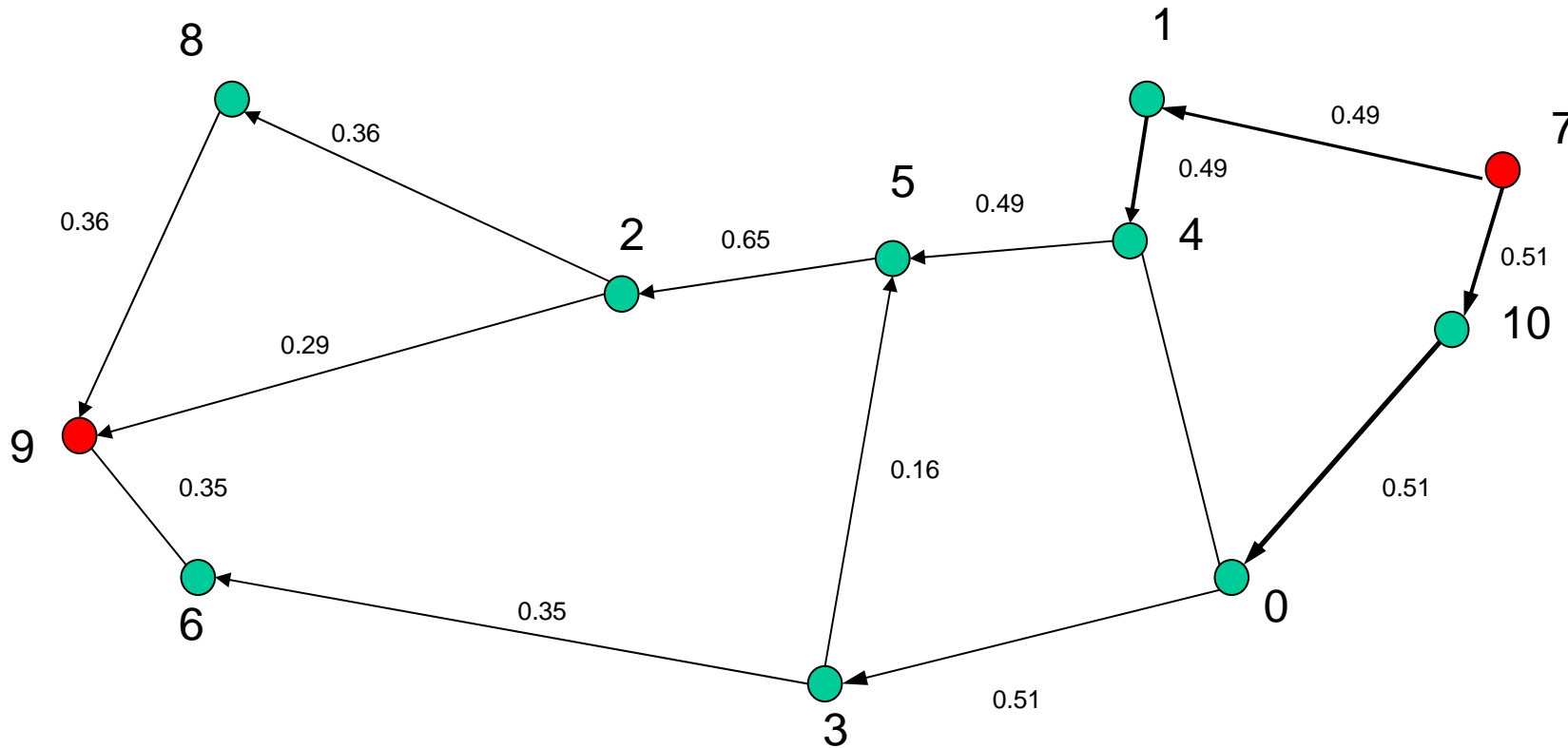
Optimal: given a TM, computes the routing that gives the minimal MLU [Fortz *et. al* '00]

Dynamic: uses optimal routing of previous interval for this interval, changes routing at each interval [Elwalid *et. al* '01]

Oblivious: uses only one routing for any TMs. [Applegate *et. al* '03]

Optimal and robust: optimizes a routing for predicted TMs but with a worst case performance bound for any TM [Wang *et. al* '06]

A link-based routing



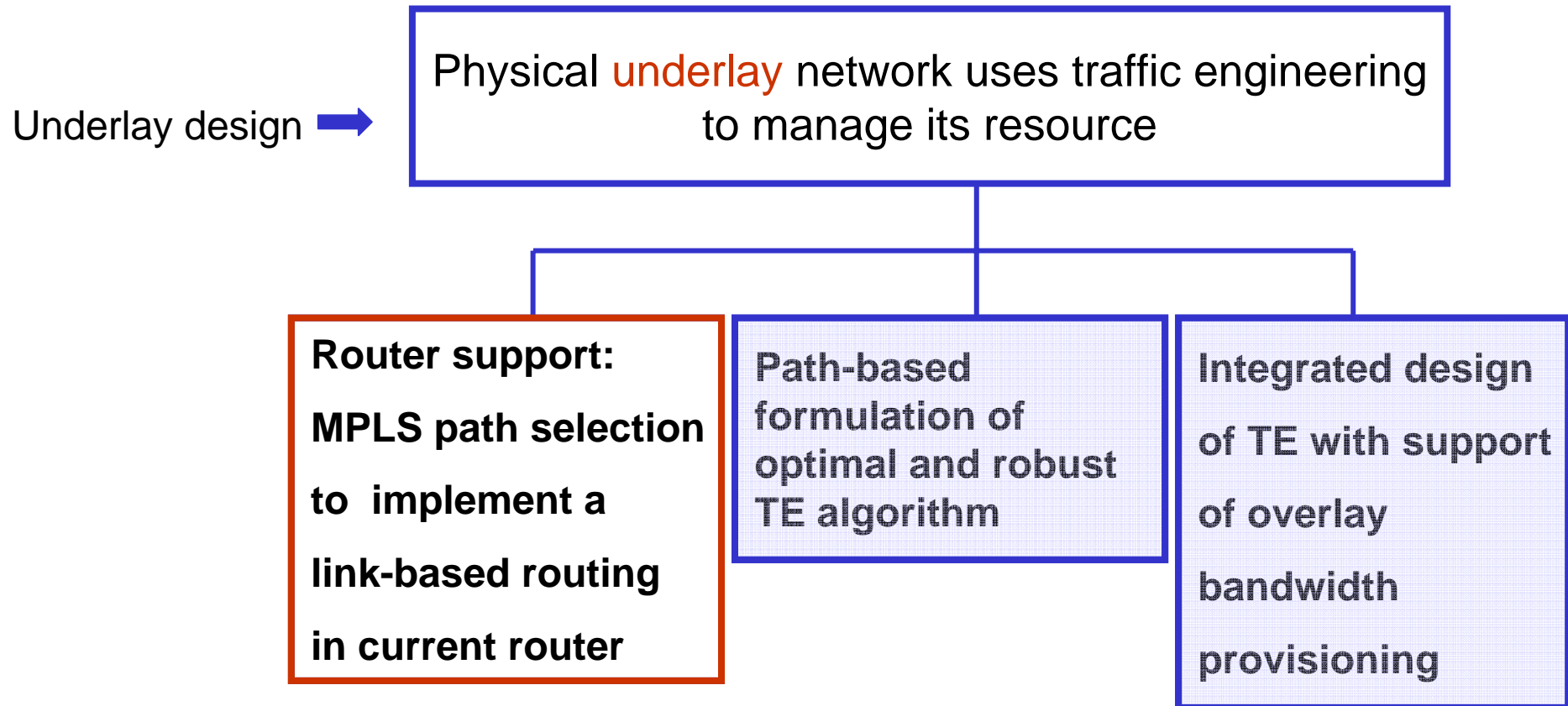
$f_{79}\{i, j\}$ on each link

What is the problem?

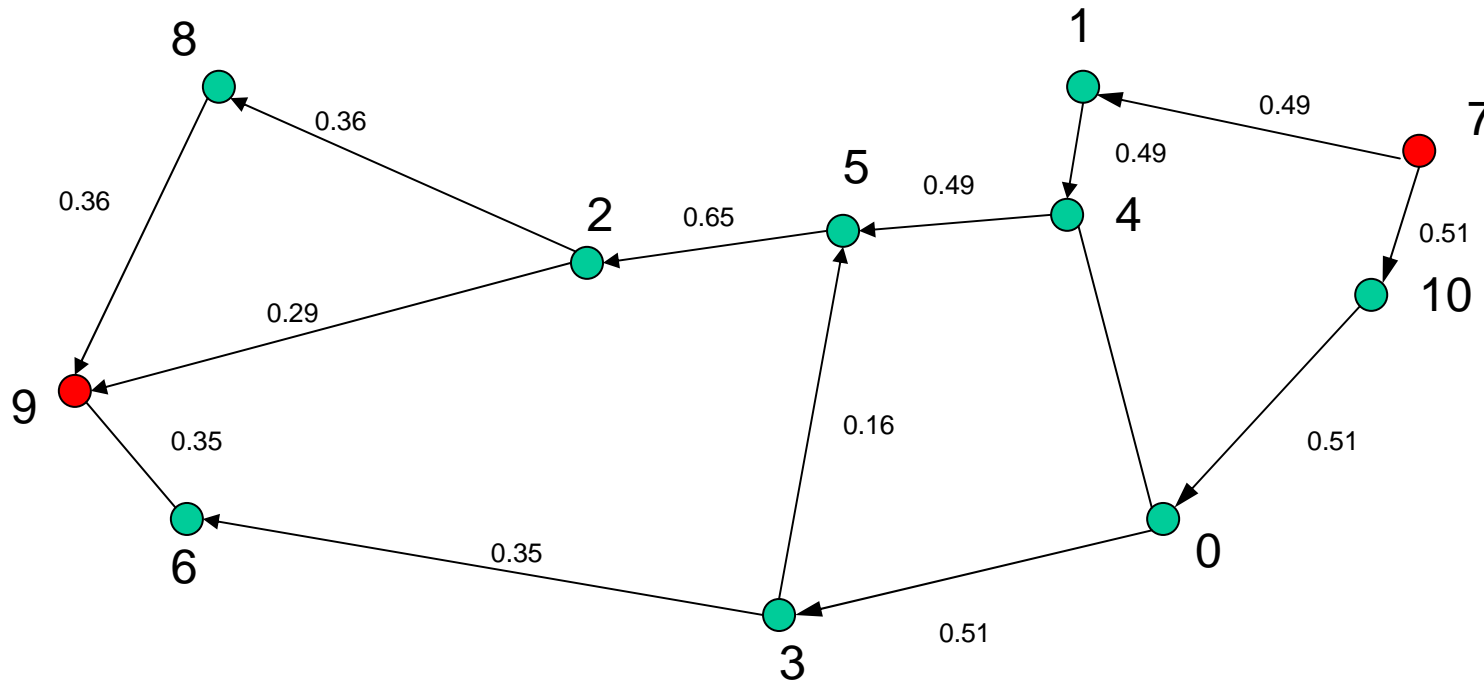
- Routers *cannot support* this type of complicated arbitrary splitting of traffic for each OD pair at each link
- Hard to be implemented with OSPF [Fortz *et.al* 00]
- Instead, routers support *path-based routing* for each OD pair along different paths using MPLS
 - Path-based routing, $F = \{f_{ab}^k \mid a, b \in V, k \text{ is integer}\}$
$$\sum_{i=1}^k f_{ab}^k = 1, 0 \leq f_{ab}^k \leq 1$$
 - MPLS (Multiprotocol label switching) is a widely supported technology by modern routers

Our approach: *Morsel* [MWYKA'06]

MPLS-based Optimal and Robust TE with path SELection



Implementation of a link-based routing



- In theory: convert arc-flow to path-flow.
- Use standard *flow decomposition method*, we need at most $|E|$ paths.
- However, $|E|$ could be very large for large ISPs.

Setup paths from 7->9:

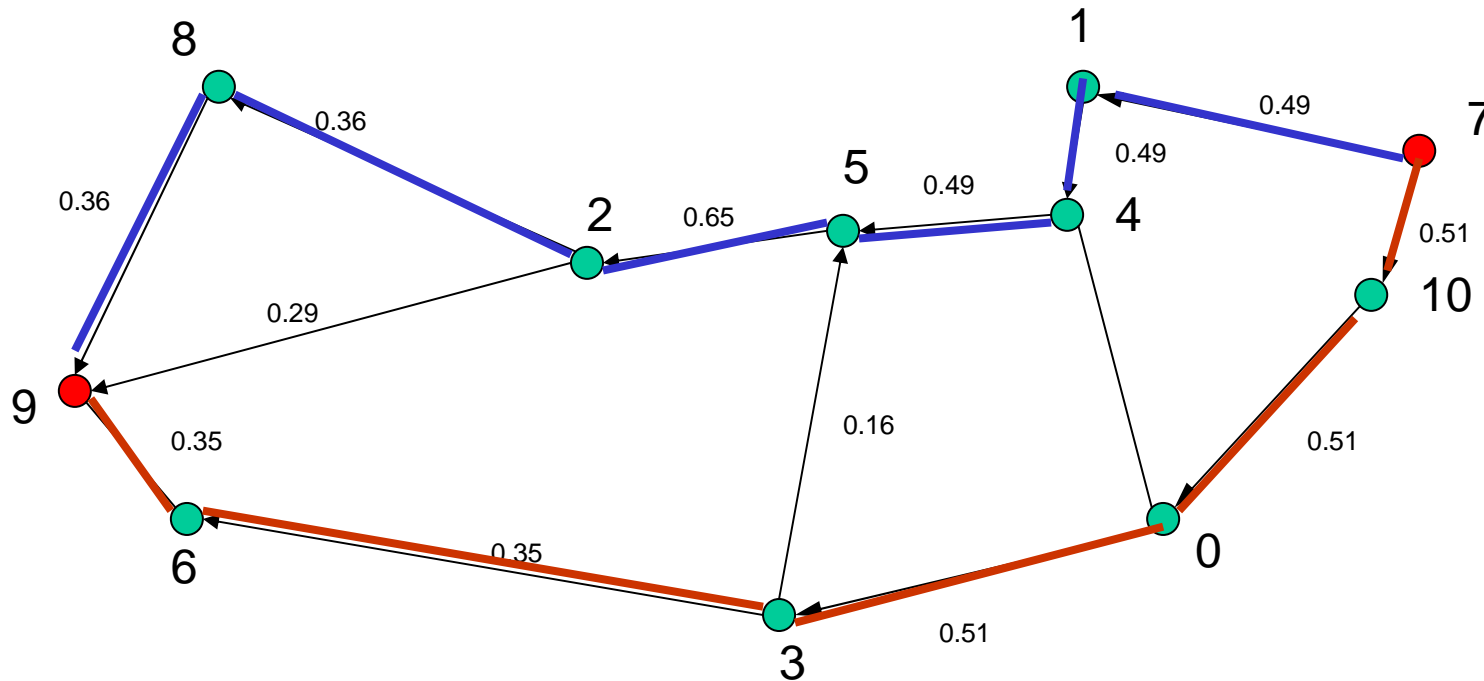
7->1->4->5->2->9: 0.13

7->1->4->5->2->8->9: 0.36

7->10->0->3->5->2->9: 0.16

7->10->0->3->6->9: 0.35

Coverage-based path set selection



- Coverage-based method: a greedy flow-decomposition algorithm to select path one by one until we reach the required *coverage* (total path flow rate).

Select paths from 7→9 with coverage 0.7:

7→1→4→5→2→8→9: 0.36

7→10→0→3→6→9: 0.35

Total coverage > 0.7, we are done

Properties of q -Coverage path sets

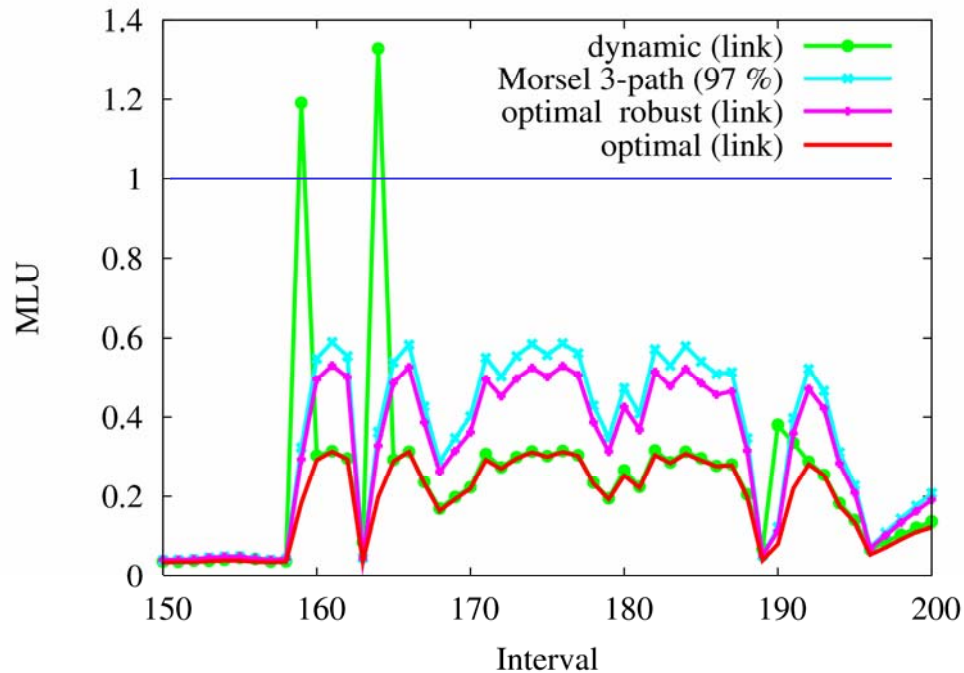
- *Theorem:* Given a link-based routing f and a q -coverage path set for f , there is a path-based routing over the q -coverage path set such that for any demand D , the **MLU** under the path-based routing is bounded by $1/q$ of the **MLU** achieved by f .

Evaluation methodology

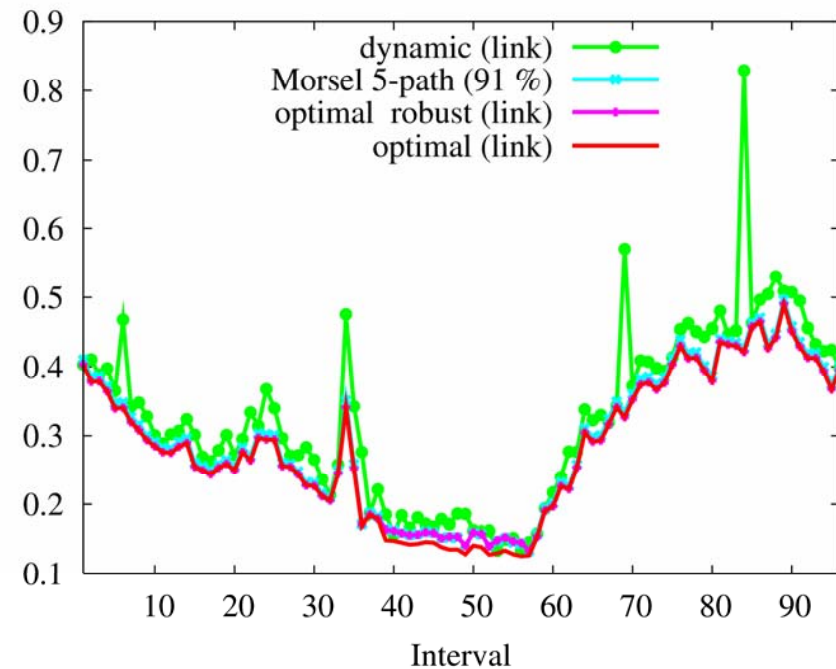
- TE Algorithms
 - **Optimal**: requires an oracle
 - **Dynamic**: optimizes routing for TM in previous interval
 - **Optimal and robust (link)**: the link-based routing optimize for previous day and the same day last week, bound the worse case traffic for any TM
 - **Morsel**: implement the above optimal and robust routing with selected MPLS paths
- Dataset
 - G'EANT (AS20965) 23 nodes, 74 edges
 - 15-min router-level TMs on G'EANT (4 months: May - Aug. 2005)
 - Abilene (AS11537) 11 nodes, 28 edges
 - 5-min router-level TMs on Abilene (6 months: Mar - Sep. 2004)
 - Abovenet (AS 6461) 22 nodes, 84 edges

Evaluation: Max Link Utilization

Abilene trace

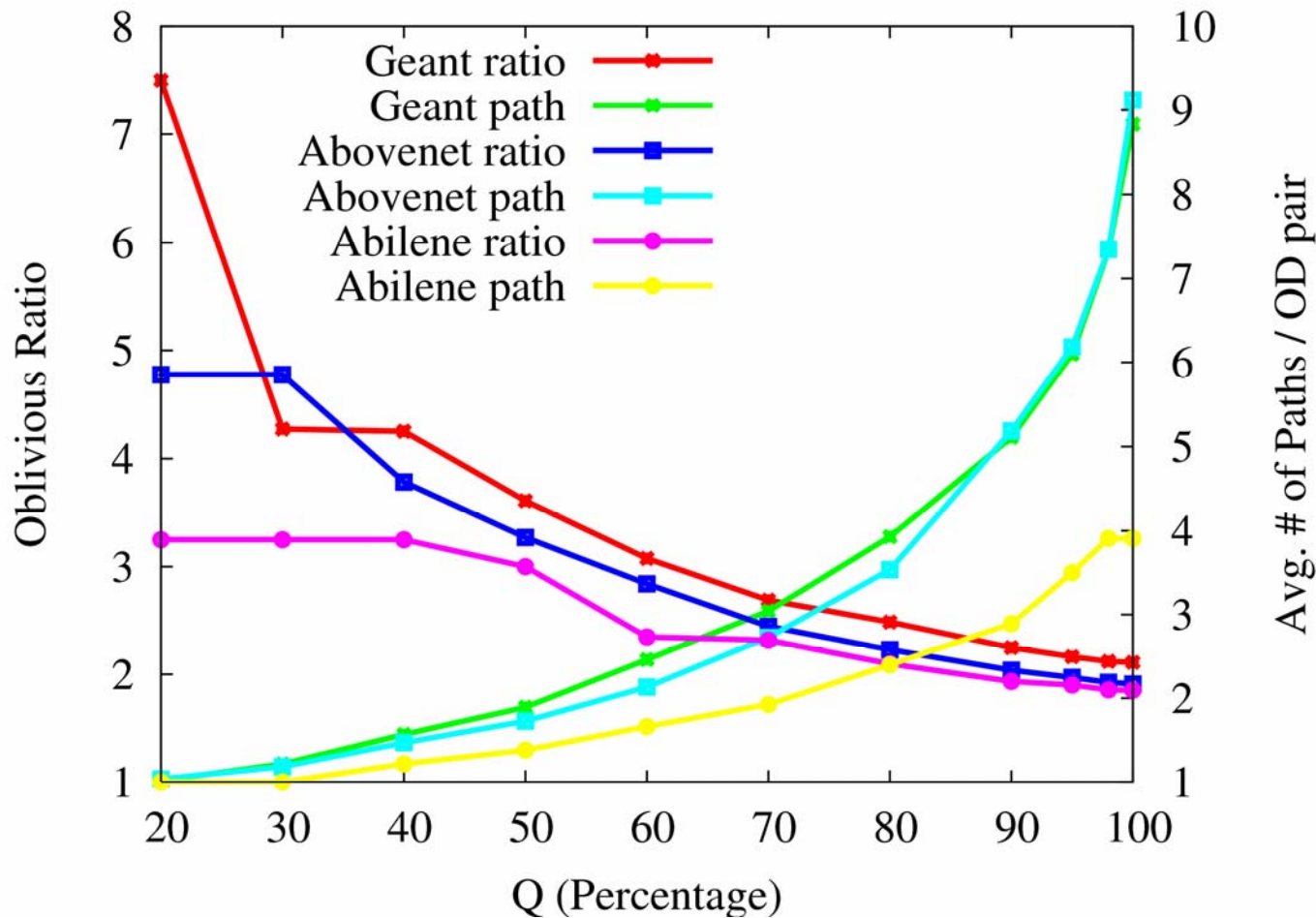


G'EANT trace



Unexpected cases: Morsel's performance is bounded and much better than dynamic
Predicted cases: Morsel is very close to optimal

Tradeoff between performance and scalability

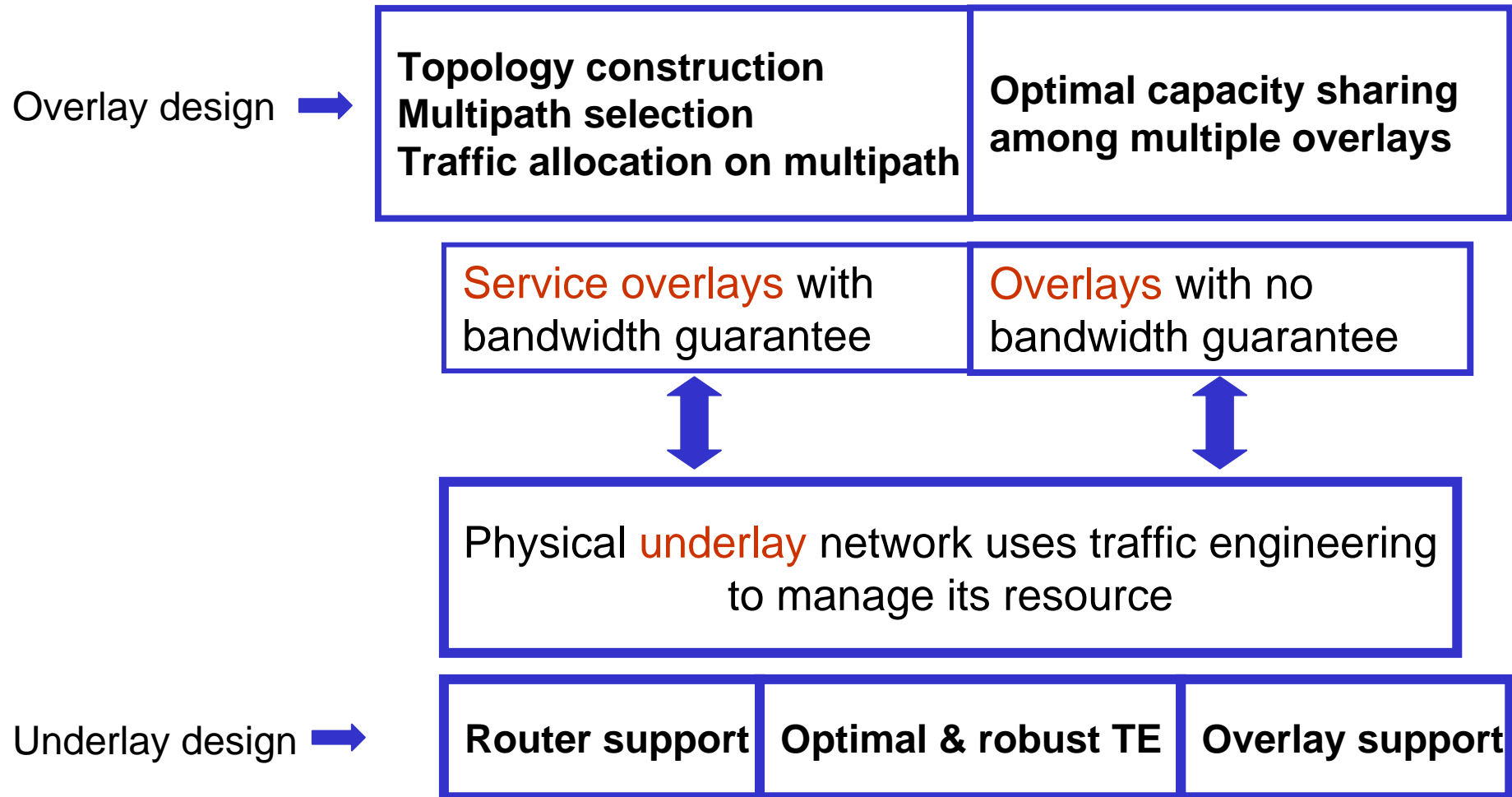


Our path selection algorithm gives explicit trade off between performance and scalability

Outline

- Background and motivation of using overlay networks
- Challenges of using overlays and our solutions
 - Resource management in a single service overlay
 - Resource allocation for multiple overlays
 - Integrated traffic engineering in underlay with overlay support
- Conclusions and future work

Summary



Conclusion and future work

- Overlay networks are considered key to an evolvable architecture to deploy next-generation Internet services [Peterson *et al* '02]
- Resource management in overlays is a fundamental problem that needs to be solved
- Next:
 - Underlay's support for overlay's resource management
 - Save the cost of overlay probing by providing the underlay info
 - Change the underlay routers to have better support of overlays
 - Consider more objectives in underlay such as charging in overlay networks to have more integrated design

Our work in the field

- **“Overlay Mesh Construction Using Interleaved Spanning Trees”** - [INFOCOM2004]
 - Overlay mesh construction using k-mst algorithms
- **“A New Multipath Selection Scheme for Video Streaming on Overlay Networks”** - [ICC 2004]
 - Propose the correlation between paths as a new QoS metric for overlay multipath selection
 - Propose the heuristic algorithm to select multipath in overlay for video streaming
- **“Managing a Portfolio of Overlay Paths”** - [NOSSDAV 2004]
 - Find the analogy between resource allocation in overlay multipath with portfolio theory
- **“Optimal Capacity Sharing of Networks with Multiple Overlays”** - [IWQoS 2006]
 - Propose the problem of capacity sharing among multiple overlays
 - Show the drawback of previous flow level rate control and design a new overlay level flow control
- **“Traffic Engineering in MPLS and VPN Networks”** - [Yale CS Tech Report 2007]
 - MPLS based optimal and robust TE with path selection, integrated with VPN support
- **Patent application:**
 - **“Selecting Multiple Paths in Overlay Networks for Streaming Data”**, *US2005083848(A1)*
 - **“Robust and Optimal Traffic Engineering in Internet”**, Yale OCR #4438 application filed

Acknowledgements

- I would like to thank all of my mentors and collaborators, without whom this work would not have been possible
- Daria Antonova
- Jiang Chen
- Arvind Krishnamurthy
- Joan Feigenbaum
- Larry Peterson
- Jennifer Rexford
- Huai-Rong Shao
- Chia Shen
- Avi Silberschatz
- Ravi Sundaram
- Hao Wang
- Randolph Y. Wang
- Anthony Young
- Y. Richard Yang

**THANK YOU FOR
LISTENING
ANY QUESTIONS?**