UMassAmherst

A Scalable Distributed Routing Protocol for Networks with Data-Path Services

University of Massachusetts Amherst

Xin Huang, Sivakumar Ganapathy, and Tilman Wolf

{xhuang,sganapat,wolf}@ecs.umass.edu

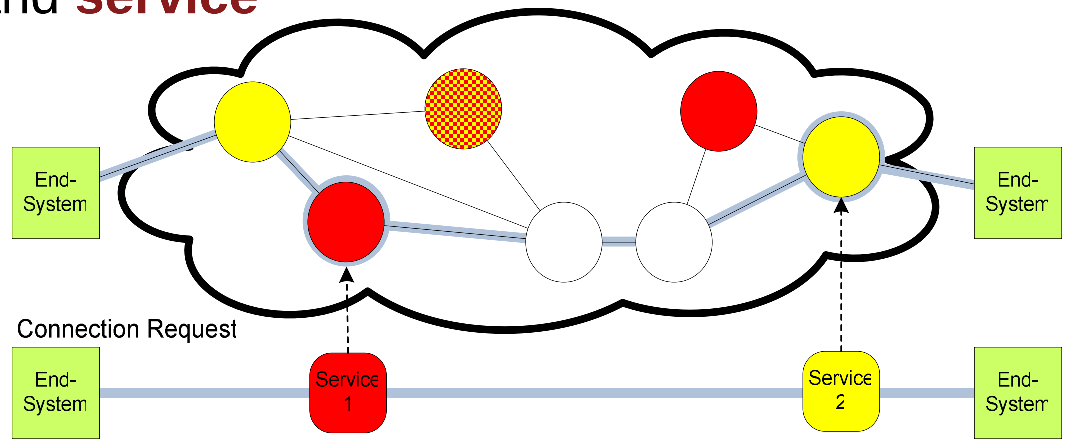
Next-Generation Internet

- Network community in search of **new network architecture**
  - Meet **new requirements:** security, manageability, etc.
  - Accommodate **heterogeneous** end systems: PDAs, sensors, etc.
  - Support emerging **networking paradigms:** P2P, CDN, etc.
- Next-generation Internet architecture design
  - "**Clean slate**": no backward compatibility required
- Network service architecture*
  - **Services** inside the network
  - **Decomposition** of communication into basic services
  - **Custom composition** of services along the data-path

**Tilman Wolf, "Service-centric end-to-end abstractions in next-generation networks," in Proc. of Fifteenth IEEE International Conference on Computer Communications and Networks (ICCCN), Arlington, VA, Oct. 2006, pp. 79-86.*

Service Routing Problem

- Connection request: **sequence of services,** $R = (s, t, \{S_1, S_2\})$
- New routing problem:
  - **Optimal** path:
    - Least-cost path: communication cost + processing cost
    - Single cost metric
  - Two dimensions: **cost** and **service**
- **Centralized solution***
  - Global view necessary
  - Limited scalability
- **Distributed solution ??**



*Sumi Y. Choi, Jonathan S. Turner, and Tilman Wolf, "Configuring sessions in programmable networks," Computer Network, vol. 41, no. 2, pp. 269-284, Feb. 2003.

Our Contributions

- A **distributed** algorithm for service routing problem
- A **scalable** routing protocol for networks with data-path services
- An **evaluation prototype** based on **Emulab**
  - Quality of the distributed routing algorithm
  - Performance of the routing protocol

Outline

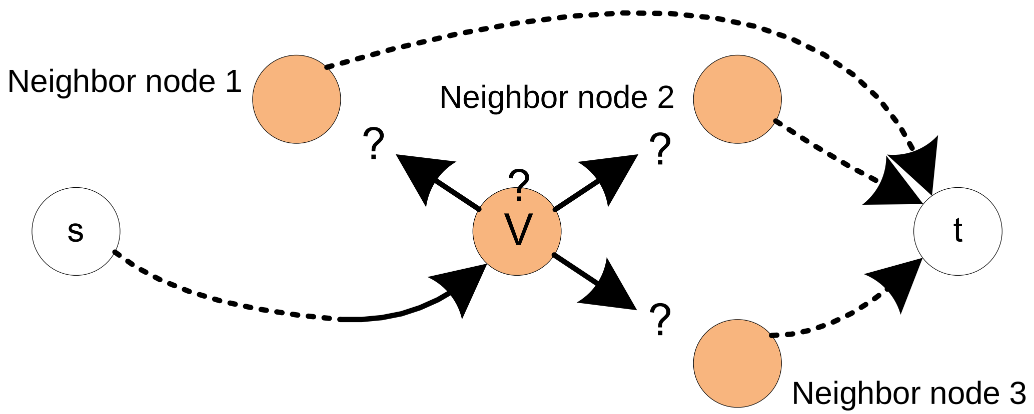
- Introduction
- Related Work
- Distributed Routing Algorithm
- Scalable Routing Protocol
- Evaluation and Results
- Summary

Related Work

- **Network services**
  - Cisco Service-Oriented Network Architecture [2]
  - **Network Service Architecture (Wolf and Ganapathy [3,23])**
  - IETF Open Pluggable Edge Services (OPES) [8]
  - Router systems with services (Ruf, Farkas, Hug and Plattner [4] and Hutchinson and Peterson [5])
- **Supporting platforms**
  - Router virtualization (Anderson, Peterson, Shenker and Turner [6])
  - Programmable routers (Ruf, Farkas, Hug and Plattner [4])
- **Routing Algorithms**
  - Traditional shortest path routing
    - Routing Information Protocol (RIP [24])
  - Service routing:
    - Centralized solution: layered graph (Choi, Turner and Wolf [12])
    - Capacity constraint: extension of layered graph (Choi and Turner [13])
    - Distributed solution: this is the first published work

Distributed Service Routing

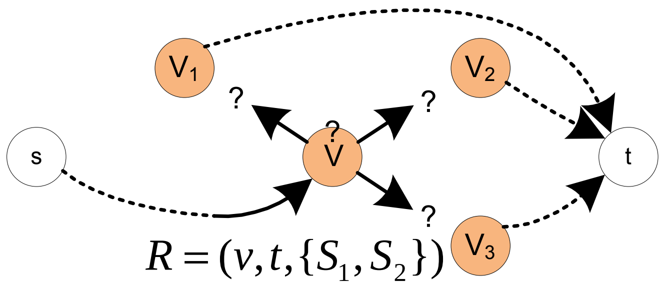
- A node needs to select **path** and decide the **services** to perform.
  - How can a node decide the optimal path?
    - Which direction (neighboring node) to route connection?
    - Better to perform services locally?
    - Better to defer to downstream nodes?
  - What routing information needs to be exchanged between nodes?



Distributed Service Matrix Routing (DSMR)

- Similar to Distance Vector routing (Bellman-Ford)
  - Each neighbor announces cost of the optimal path to each destination
  - Each node calculate the optimal path by: $c_v(t) = \min_{n_v}\{w_{v,n_v} + c_{n_v}(t)\}$
- Distributed Service Matrix Routing (DSMR)
  - Expand vector to include service: "**service matrix**"
    - Periodic service matrix exchange
    - Eventually converge to a stable stage
  - Each node can update the **routing table** by:

$$c_v^{\,j_1,..j_k}(t) = \min_{0\le i\le k}\{\sum_{l=1}^{i} c_{v,j_l} + \min_{n_v}\{w_{v,n_v} + c_{n_v}^{\,j_{i+1},..j_k}(t)\}\}$$



$$R = (v, t, \{S_1, S_2\})$$

no service | services

destinations

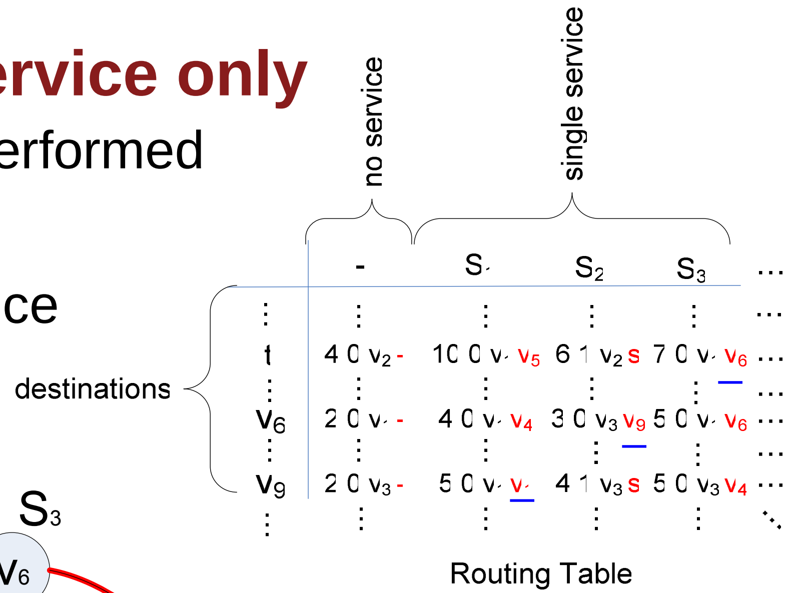
	-	S_1	S_2	S_1S_2	S_2S_1	...
s	3,0,v_1	6,0,v_1	9,0,v_1	11,1,v_1	10,0,v_1	⋯
t	4,0,v_2	7,1,v_2	5,0,v_3	12,1,v_3	13,0,v_1	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

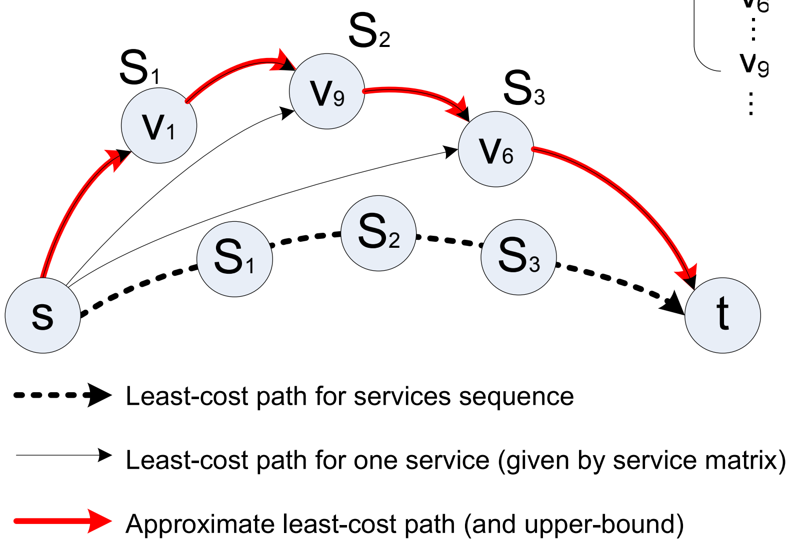
Routing Table

  - Challenge: each service combination requires columns in matrix
    - Matrix size grows fast with number of services

Approximate DSMR

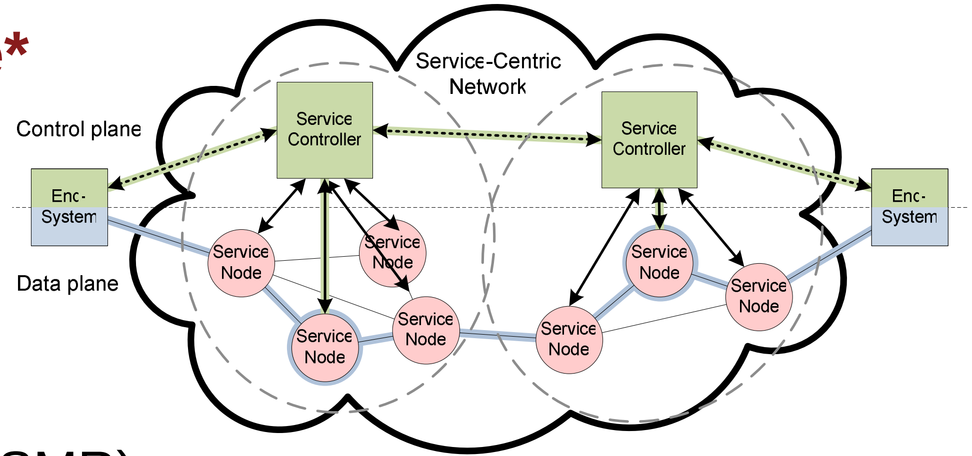
- Use information for **no/single service only**
  - Matrix lists node where service is performed
- Routing of multiple services
  - Allocate best node for the **last** service
  - Find optimal path for second-to-last service to that node
  - Repeat for all services
- **Upper bound** for the optimal path cost

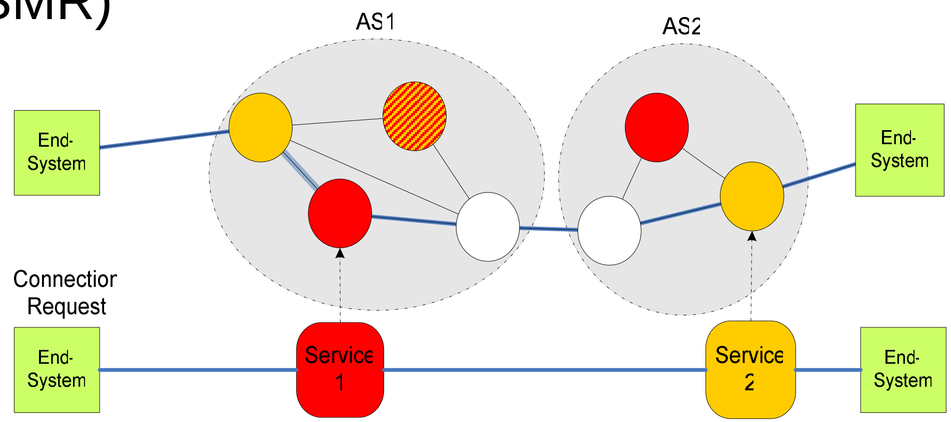




Distributed Service Routing Protocol (DSRP)

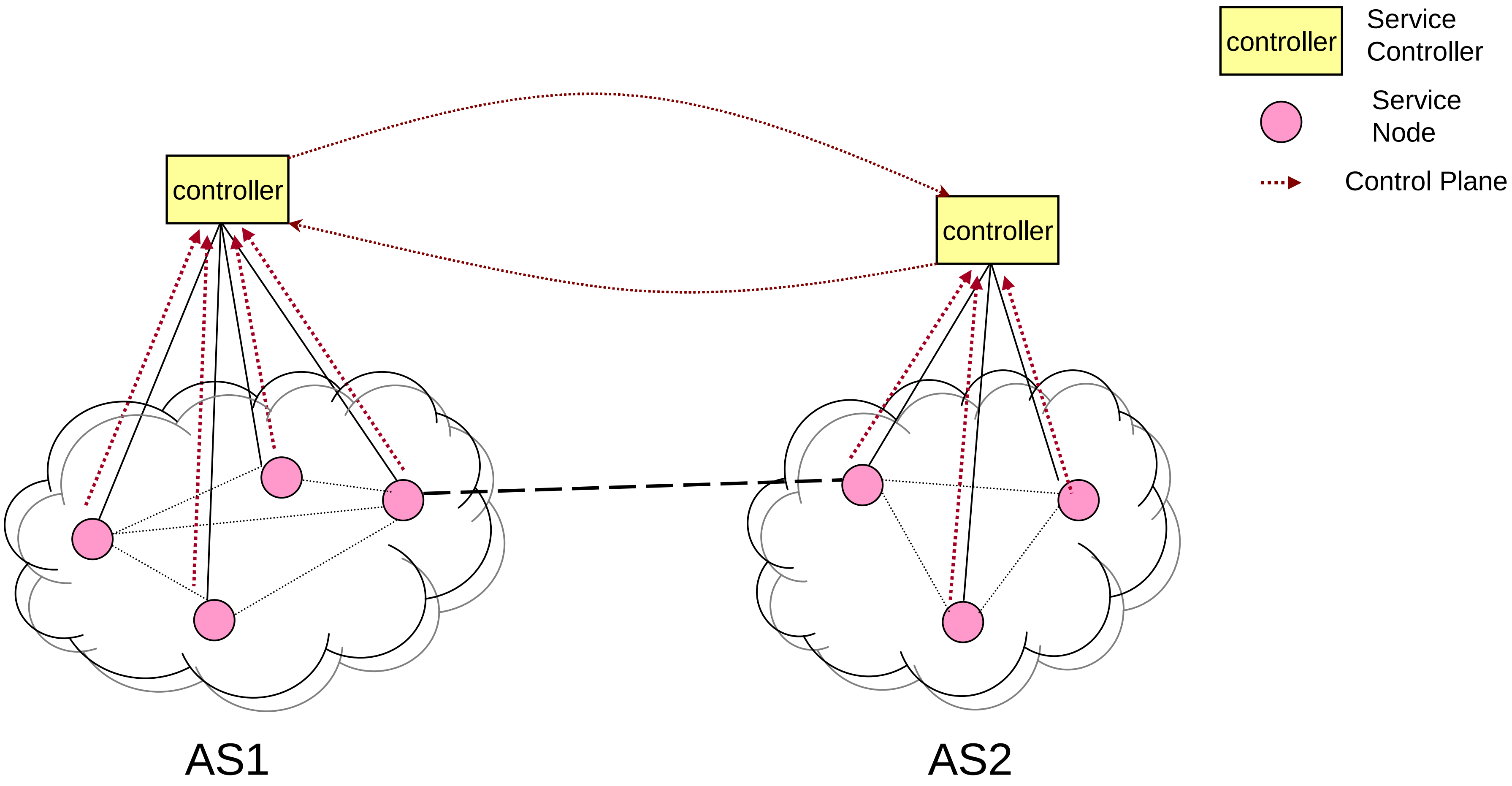
- **Hierarchical architecture***
  - Autonomous System (AS) abstraction
  - Controller & service nodes
- **Hierarchical routing**
  - Inter-AS routing: distributed algorithm (DSMR or appr. DSMR)
    - Services handled locally
    - Next-hop AS
  - Intra-AS routing: centralized algorithm (layered graph)
    - Nodes for services
    - Local path



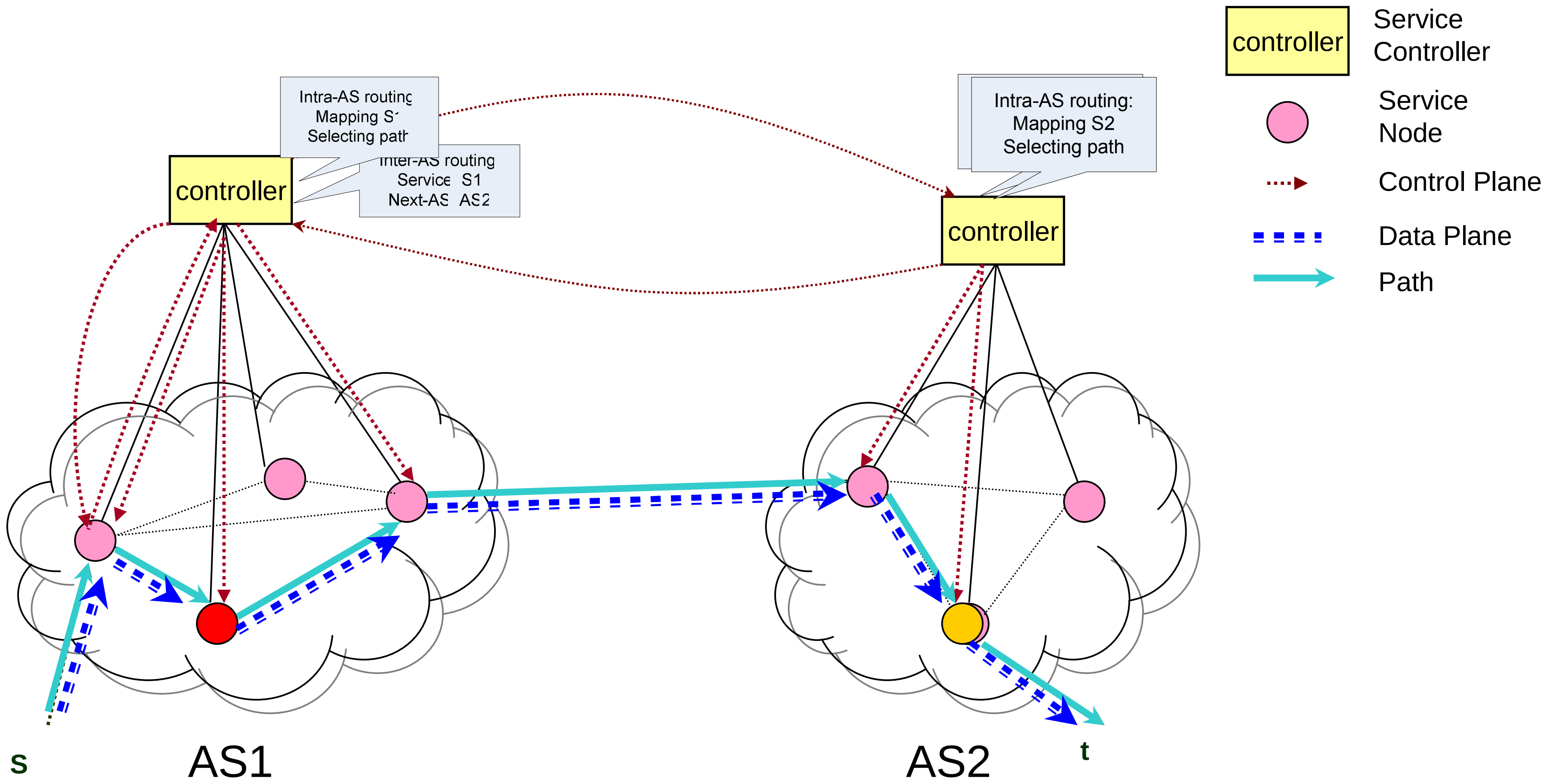


*Sivakumar Ganapathy and Tilman Wolf, "Design of a network service architecture," in Proc. of Sixteenth IEEE International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, Aug. 2007.

DSRP Stages (I): Setup & Routing Info. Exch.



DSRP Stages (II): Connection Setup



Analytical Comparison of the Algorithms

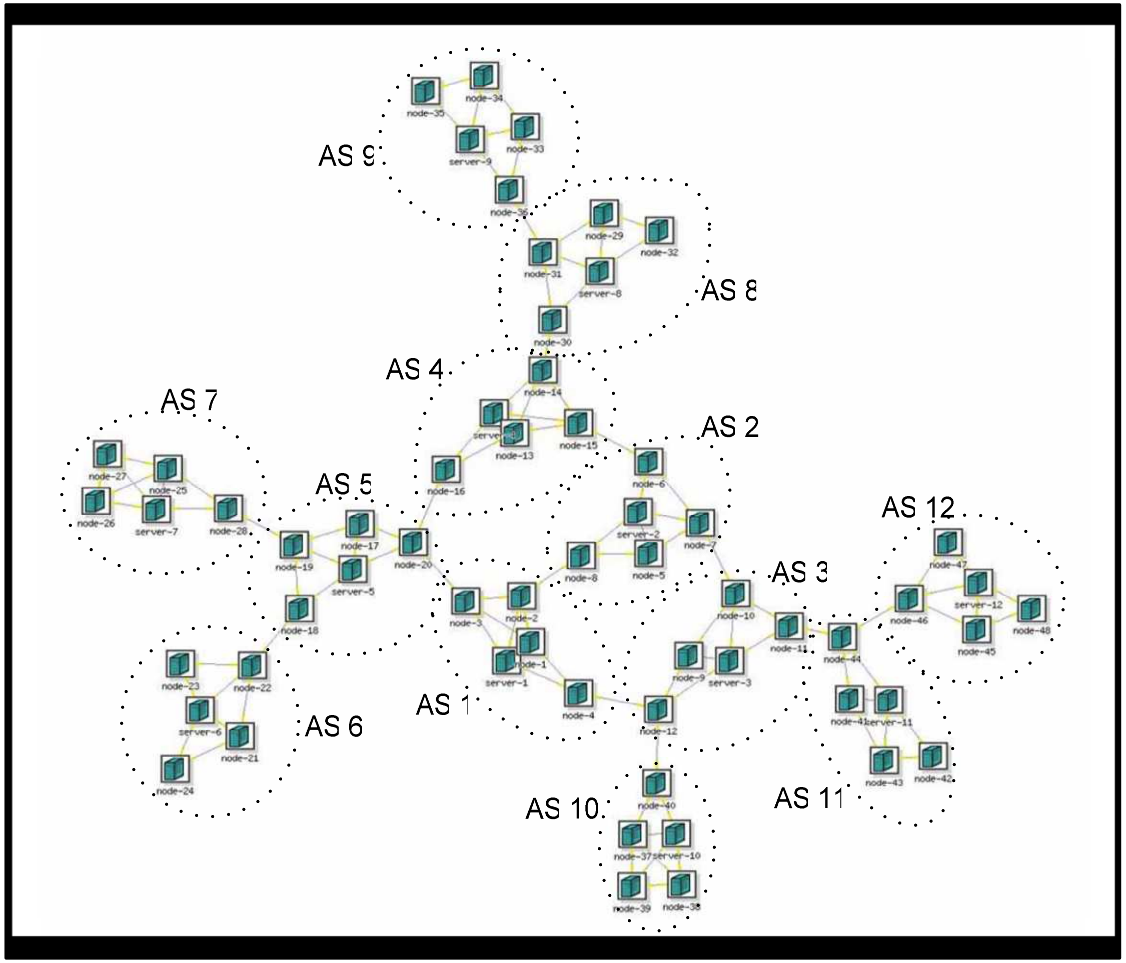
Time and Space Complexity of Algorithms

Algorithm	Time (route lookup)	Space
Layered graph	$\mathcal{O}(k(\lvert E\rvert + \lvert V\rvert)\log(k\lvert V\rvert))$	$\mathcal{O}(k \cdot (\lvert E\rvert + \lvert V\rvert))$
DSMR	$\mathcal{O}(1)$	$\mathcal{O}(\lvert S\rvert^{k_{max}} \cdot \lvert V\rvert)$
Approx. DSMR	$\mathcal{O}(k)$	$\mathcal{O}(\lvert S\rvert \cdot \lvert V\rvert)$

|E| : number of links in the network topology graph
|V| : number of nodes in the network topology graph
K: number of services in the connection request
K_{max}: maximum number of services in a connection request
|S|: number of services provided in the network

- Trade-off between accuracy and complexity
  - Centralized layered graph: optimal solution (+); significant computation time (-)
  - DSMR: optimal solution (+); prohibitive amount of space when |S| is large (-)
  - Approximate DSMR: approximate for optimal (-); good in terms of complexity (+)
- **DSMR and approximate DSMR have good scalability**
  - Time complexity is **independent** on network size (|V| and |E|).
  - When used for Inter-AS routing, space is **linear** to the **number of ASs**.

Emulab Prototype



Setup:

- 12 Autonomous Systems (AS)
- 1 controller & 4 service nodes per AS
- 4 services
- 10 ms for inter-AS links and 2 ms for intra-AS links
- 1 ms for service processing time
- Inter-AS: Appr. DSMR
- Intra-AS: Layered graph
- 149,760 connection requests

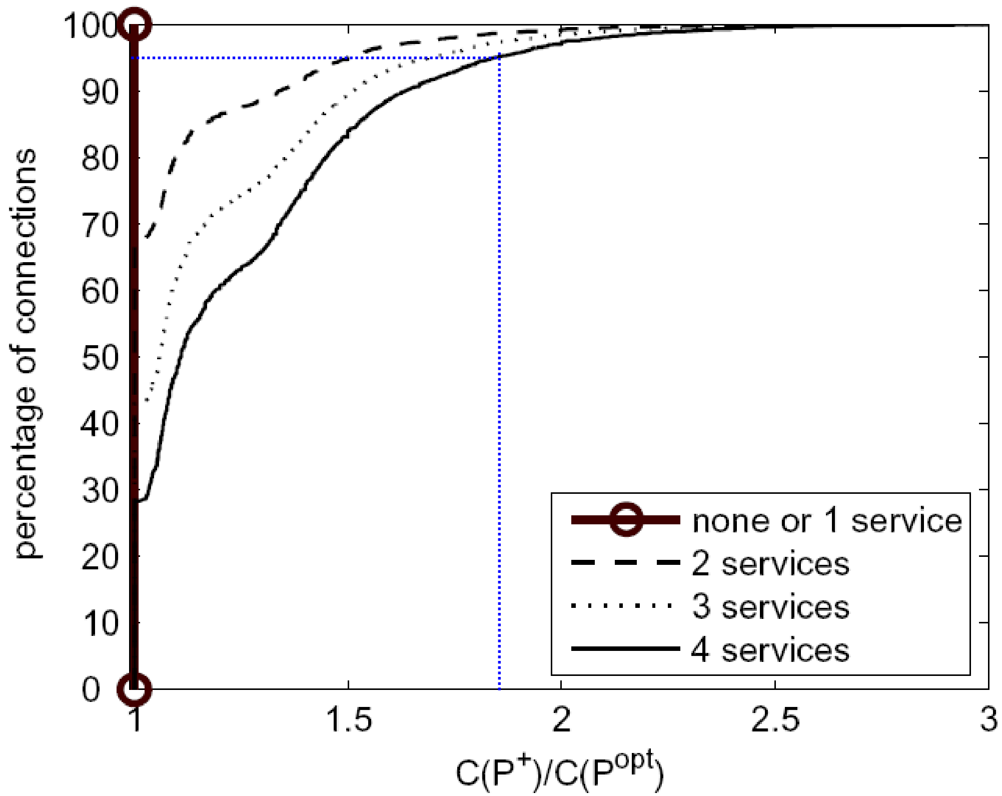
Metrics:

- Path cost
- Time for connection setup
- Control message size for connection setup

Quality of Approximate DSMR

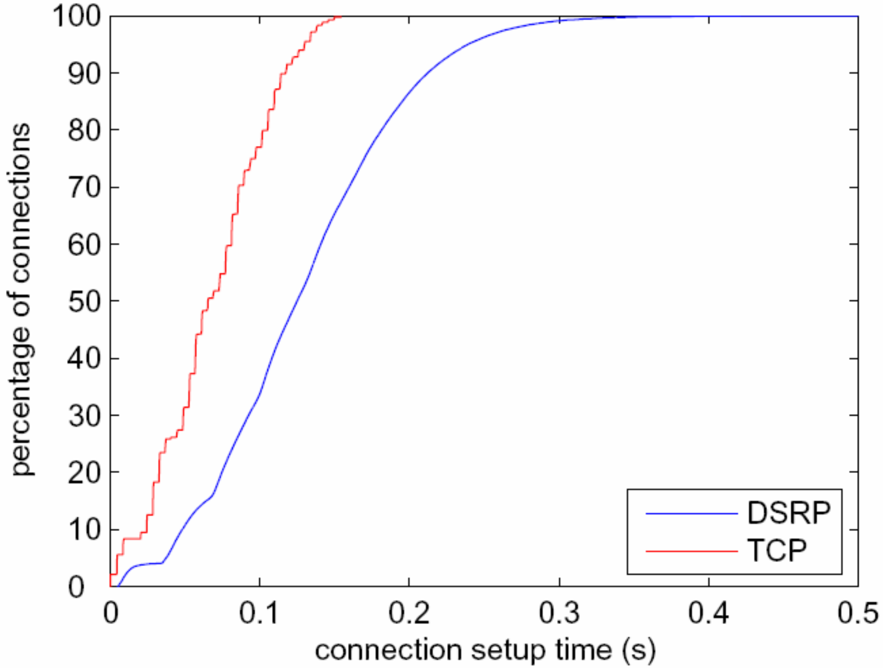
- Get optimal path for requests with zero or one service
- Approximation deteriorates with the increase of number of services required.
- Majority (⩾95%) of the connection requests has $C(P^{+})/C(P^{opt}) \leqslant 2.0$

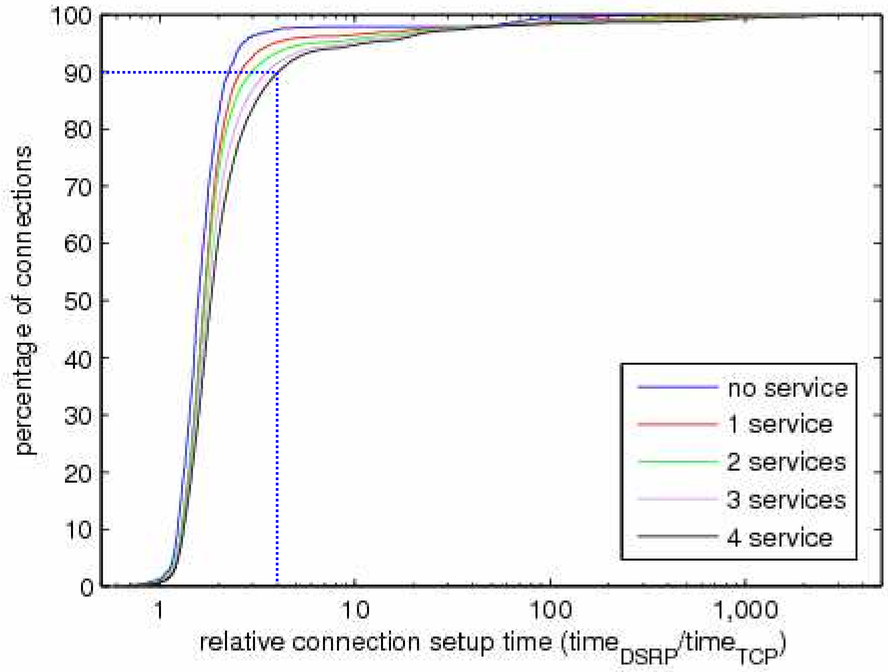
CDF of relative cost



$C(P^{+})$ is the cost of the approximate path, $C(P^{opt})$ is the cost of the optimal path

DSRP Connection Setup Time

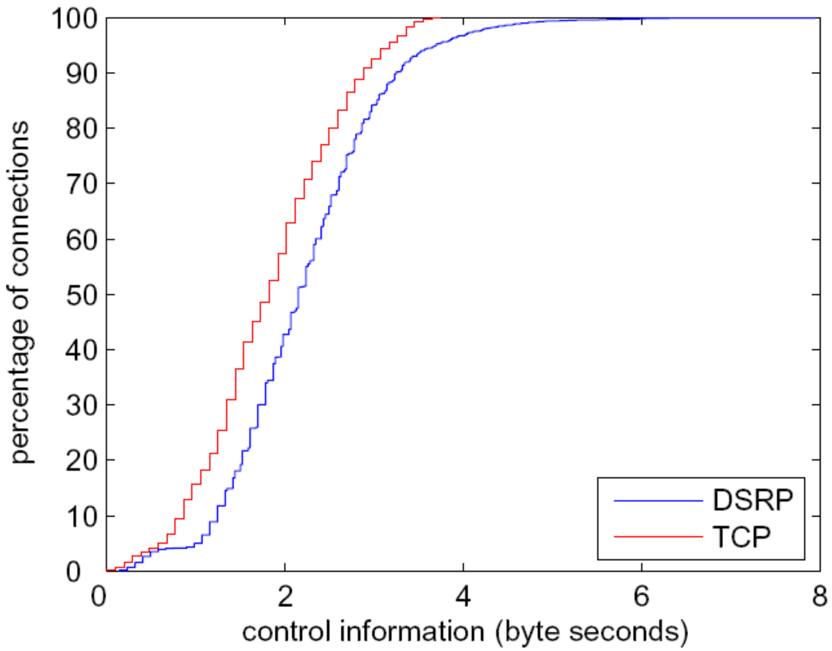


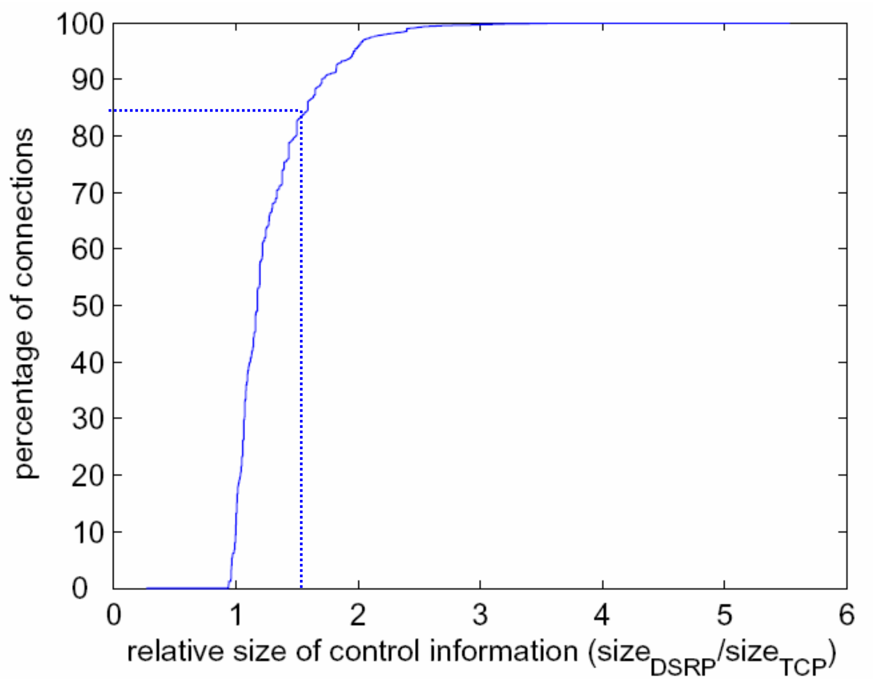


- TCP connection within 150 ms; 95% of DSRP connection within 250 ms.
- Relative setup time increases with increase in the number of required services.
- More than 90% of the requests have $time_{DSRP}/time_{TCP} \leqslant 5$.

Conclusion: DSRP is efficient in terms of connection setup time.

DSRP Control Information Size





- DSRP uses more control information, but difference are small
- For around 85% of the connections, $size_{DSRP}/size_{TCP} \leq 1.5$

Conclusion: DSRP has low protocol overhead when setting up connections.

Summary

- Next-generation Internet
  - **"Clean slate"** network architecture design
  - **Service routing problem** in networks with data-path services
- **Distributed Service Matrix Routing (DSMR) Algorithm**
  - Overcomes limitations of centralized solution
  - Approximation path to limit space requirement
- **Distributed Service Routing Protocol (DSRP)**
  - Hierarchical, scalable, and efficient
- Evaluation
  - **Prototype** Implementation on **Emulab**
  - Results shows **DSRP is scalable and efficient**

Thank You !!

Questions ?