MORE IS LESS - REDUCING LATENCY VIA REDUNDANCY

+ USING THE GENI MESOSCALE TESTBED FOR NETWORK EXPERIMENTS

Invited Talk - University of Colorado at Boulder
April 8th, 2013.

Ashish Vulimiri
University of Illinois at Urbana-Champaign

Oliver Michel
University of Vienna

P. Brighten Godfrey
University of Illinois at Urbana-Champaign

Scott Shenker
University of California at Berkeley

Latency

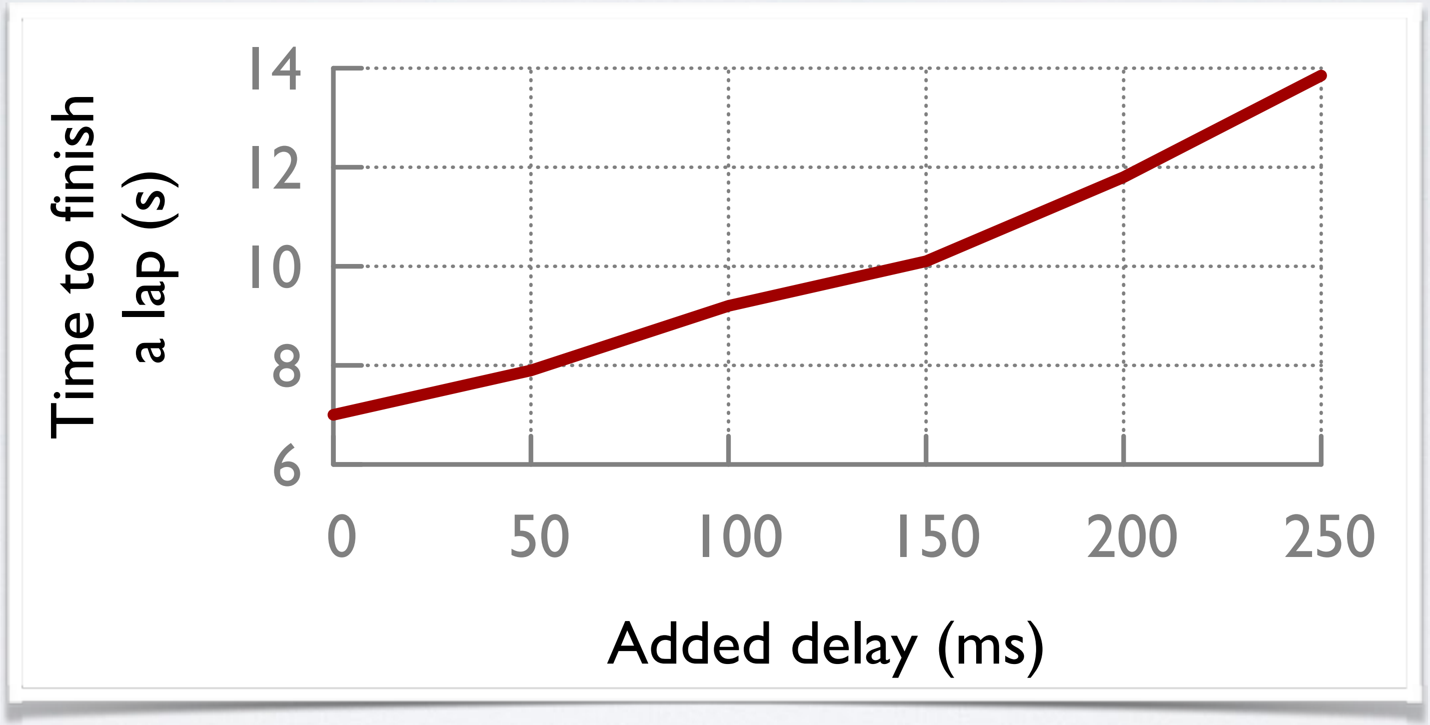
- Online services

	Delay	Result
Amazon	+100ms	-1% revenue
Bing	+500ms	-1.2% revenue
Google	+400ms	-0.6% searches

Latency

- HCI studies

User performance in a racing game



[3] L. Pantel, L.C. Wolf, "On the impact of delay on real-time multiplayer games", NOSSDAV '02

Latency

1. Pervasive uncertainty

Latency

1. Pervasive uncertainty

- Link congestion

Latency

1. Pervasive uncertainty

- Link congestion
- Cache miss

Latency

1. Pervasive uncertainty

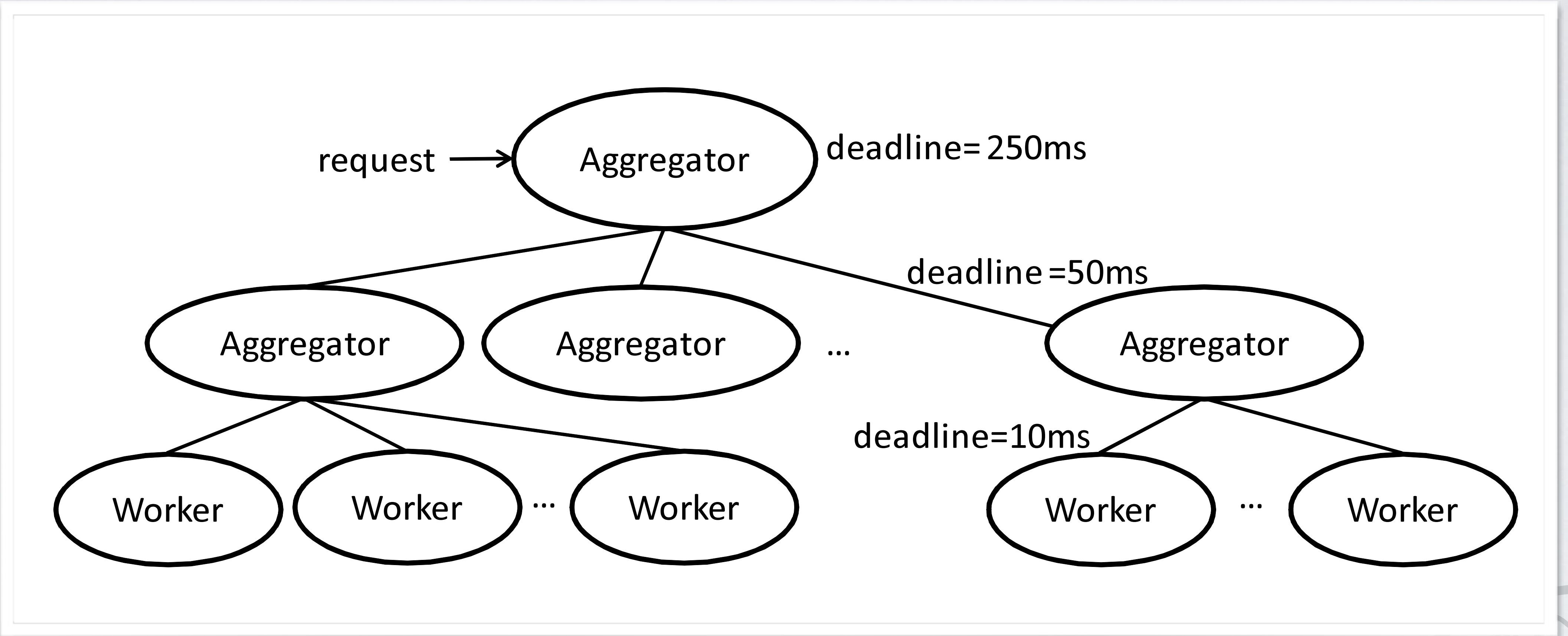
- Link congestion
- Cache miss
- Slow disk lookup

Latency

1. Pervasive uncertainty

- Link congestion
- Cache miss
- Slow disk lookup
- Delay due to virtualization

2. Application structure



[4] Partition/Aggregate Pattern - Alizadeh et.al..
"Data Center TCP", SIGCOMM '10

Redundancy

Throughput

Latency

Redundancy

Throughput

Redundancy

Latency

Idea

- Make the Internet faster by converting extra bandwidth into reduced latency
- Send out multiple copies of a packet
- Use only the packet arriving first

Redundancy

- Past uses
  - Distributed jobs (speculative execution) [5]
  - DTNs [6]
  - DHT queries [7]

[5] Ananthanarayanan et al., "Why let resources idle? Aggressive cloning of jobs using Dolly", HotCloud '12
[6] Soljanin, "Reducing delay with coding in multi-agent information transfer", Allerton '10
[7] Li et al., "Bandwidth efficient management of DHT routing tables", NSDI '10

Argument

1. Overhead should be tolerable

2. When is cost < benefit?

3. Example applications

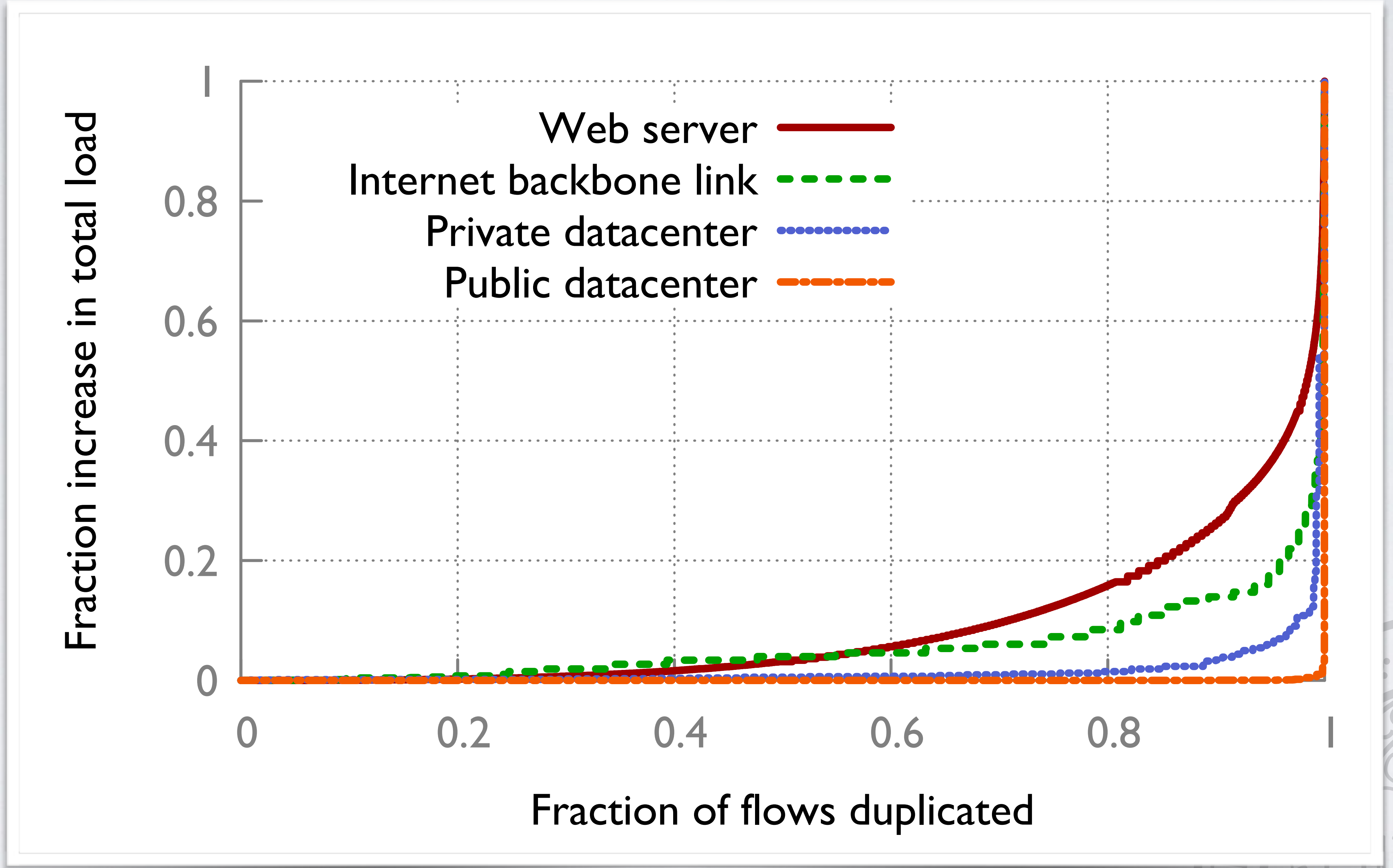
Overhead

Intuitively overhead should be low because

1. Latency-sensitive tasks likely to be small
2. Heavy tails are pervasive

Overhead

What is the overhead from replicating the x% smallest flows?



Overhead - cost calculation

Redundancy is only useful if

cost < benefit

Overhead - cost calculation

Redundancy is only useful if

cost < benefit

cost of overhead ($/KB)

Overhead - cost calculation

Redundancy is only useful if

cost	<	benefit
cost of overhead ($/KB)	<	

Overhead - cost calculation

Redundancy is only useful if

cost	<	benefit
cost of overhead ($/KB)	<	latency savings (ms/KB)

Overhead - cost calculation

Redundancy is only useful if

cost < benefit

cost of overhead ($/KB) < latency savings (ms/KB) x

Overhead - cost calculation

Redundancy is only useful if

cost < benefit

cost of overhead (\$/KB) < latency savings (ms/KB) × value of time (\$/ms)

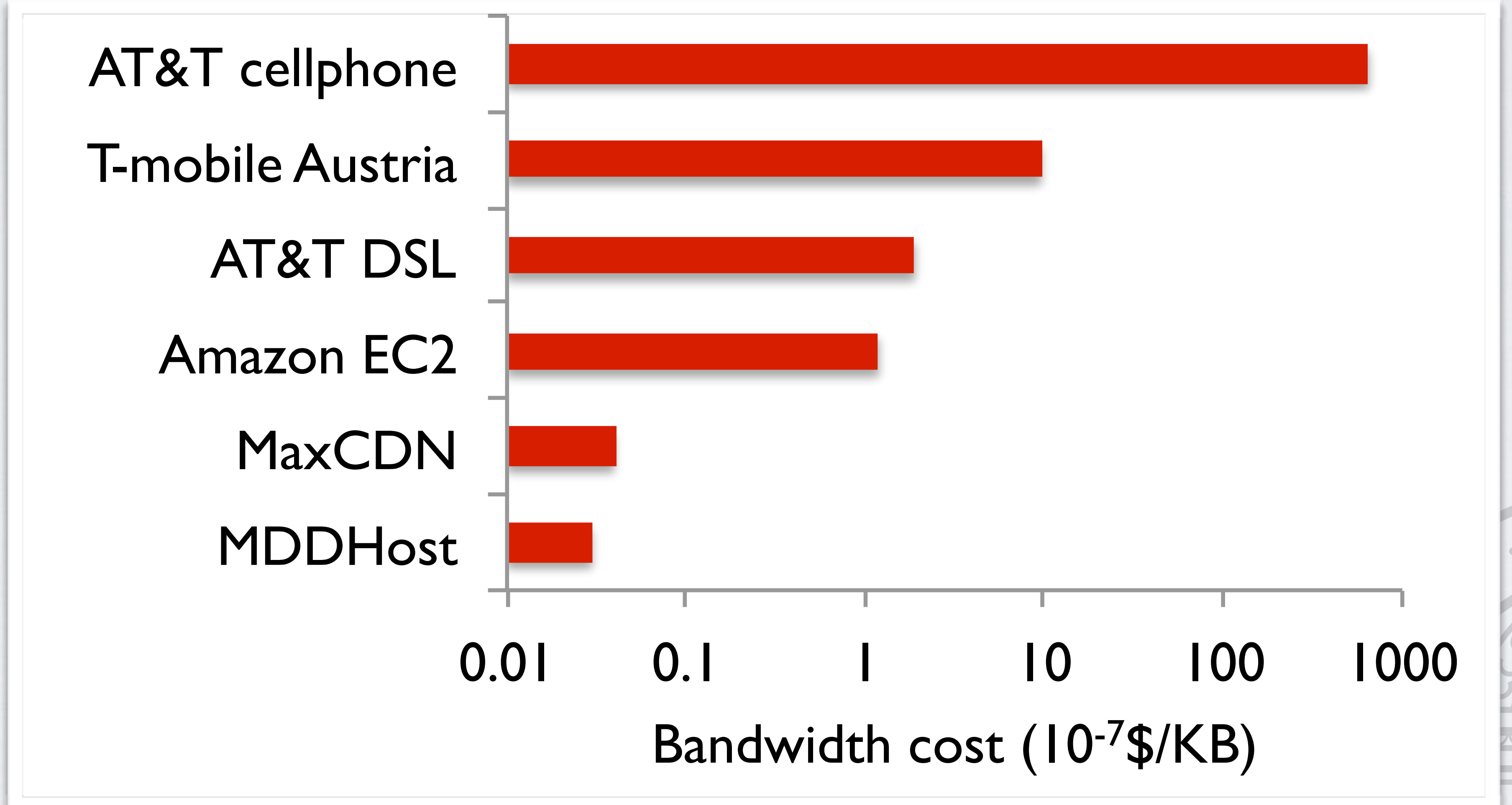
Overhead - cost calculation

$$\text{cost of overhead} < \text{latency savings} \times \textbf{value of time}$$

- Hard to estimate
- Approximation
  - U.S. median wage = 23.5$/h

Overhead - cost calculation

cost of overhead < latency savings × value of time



Overhead - cost calculation

Redundancy is even useful with the most expensive cell phone plan:

$$\text{cost of overhead } (\$/\text{KB}) < \text{latency savings } (\text{ms}/\text{KB}) \times \text{value of time } (\$/\text{ms})$$

Overhead - cost calculation

Redundancy is even useful with the most expensive cell phone plan:

$$6.5 \times 10^{-5} \ \$/\text{KB} < \text{latency savings (ms/KB)} \times \text{value of time (\$/ms)}$$

Overhead - cost calculation

Redundancy is even useful with the most expensive cell phone plan:

$$6.5\times10^{-5}\ \$/\text{KB} < \text{latency savings (ms/KB)} \times 6.5\times10^{-6}\ \$/\text{ms}$$

Overhead - cost calculation

Redundancy is even useful with the most expensive cell phone plan if

10ms/KB < latency savings

Overhead - cost calculation

Redundancy is even useful with the most expensive cell phone plan if

10ms/KB < latency savings

Redundancy is useful with a DSL plan if

0.03ms/KB < latency savings

Experiments

1. DNS

2. Multipath overlay

Experiments

1. DNS

2. Multipath overlay

Targets

10ms/KB (cell)

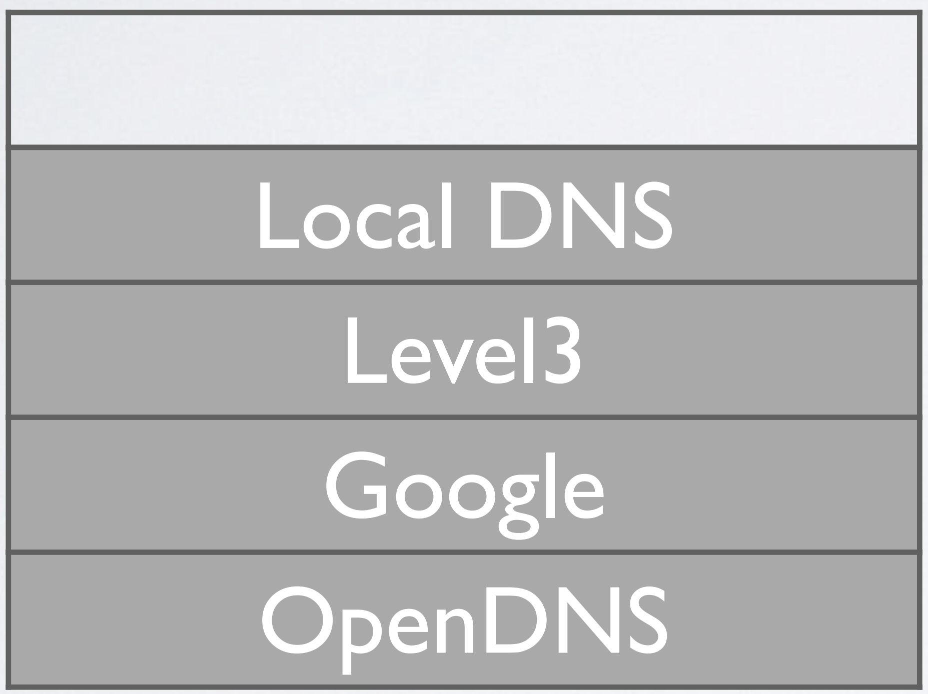
0.3ms/KB (DSL)

DNS

- Replicate DNS queries to multiple servers in parallel
- Evaluation: PlanetLab experiments

DNS

- Replicate DNS queries to multiple servers in parallel
- Evaluation: PlanetLab experiments



DNS

- Replicate DNS queries to multiple servers in parallel
- Evaluation: PlanetLab experiments

	avg. response time (s)
Local DNS	0,27
Level3	0,61
Google	0,16
OpenDNS	0,37

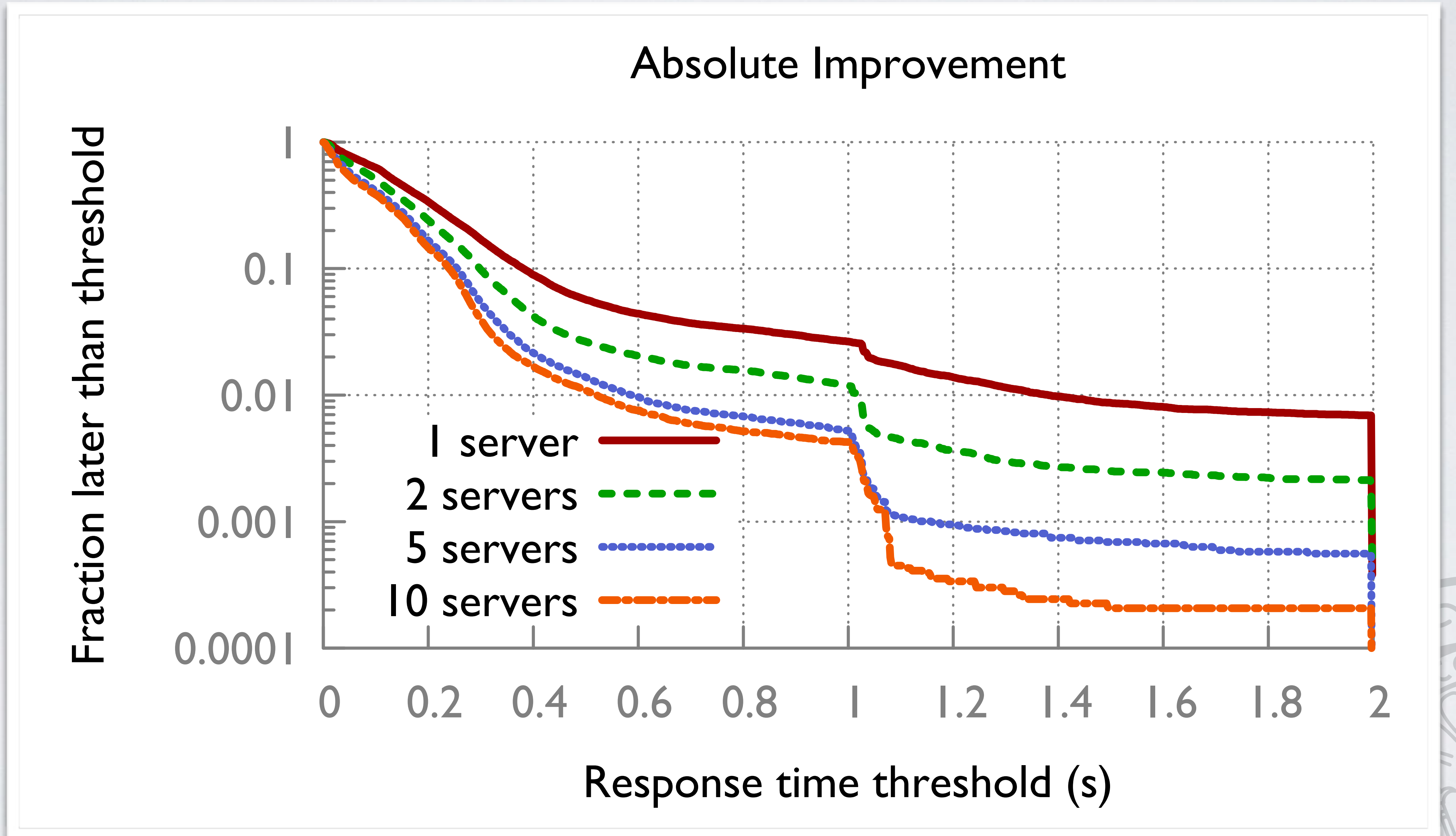
DNS

- Replicate DNS queries to multiple servers in parallel
- Evaluation: PlanetLab experiments

	avg. response time (s)
Google	0,16
Local DNS	0,27
OpenDNS	0,37
Level3	0,61

Experiments

- Try different levels of replication, using servers in the ranked order



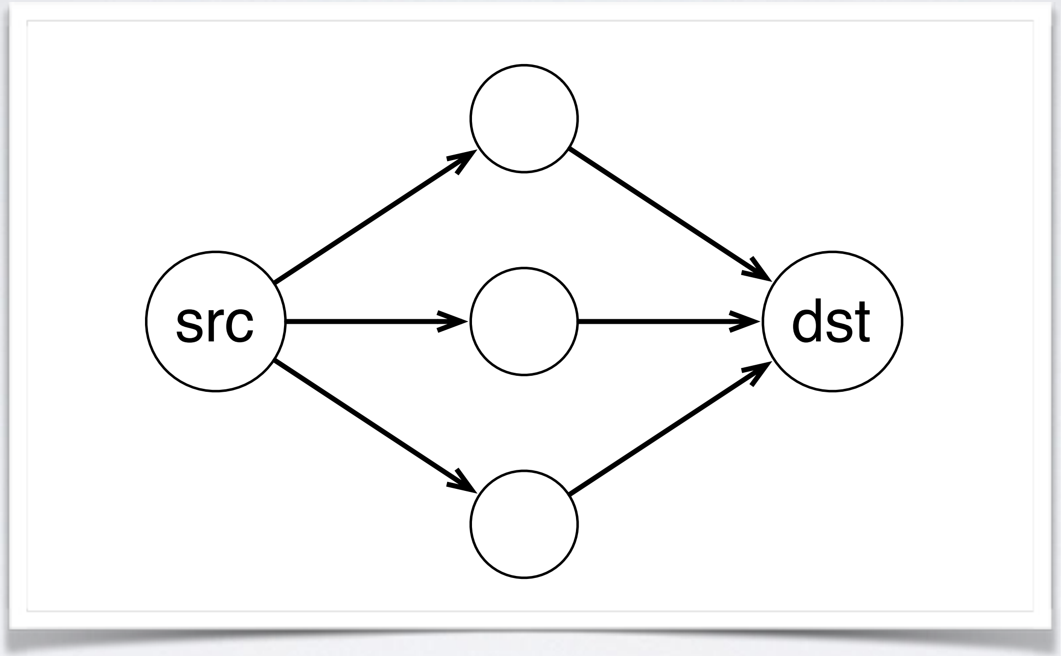
Experiments

Experiments

	Optimal number of servers per query	Average latency improvement
Cell phone	5	90ms
DSL	10	100ms

Multipath Overlay

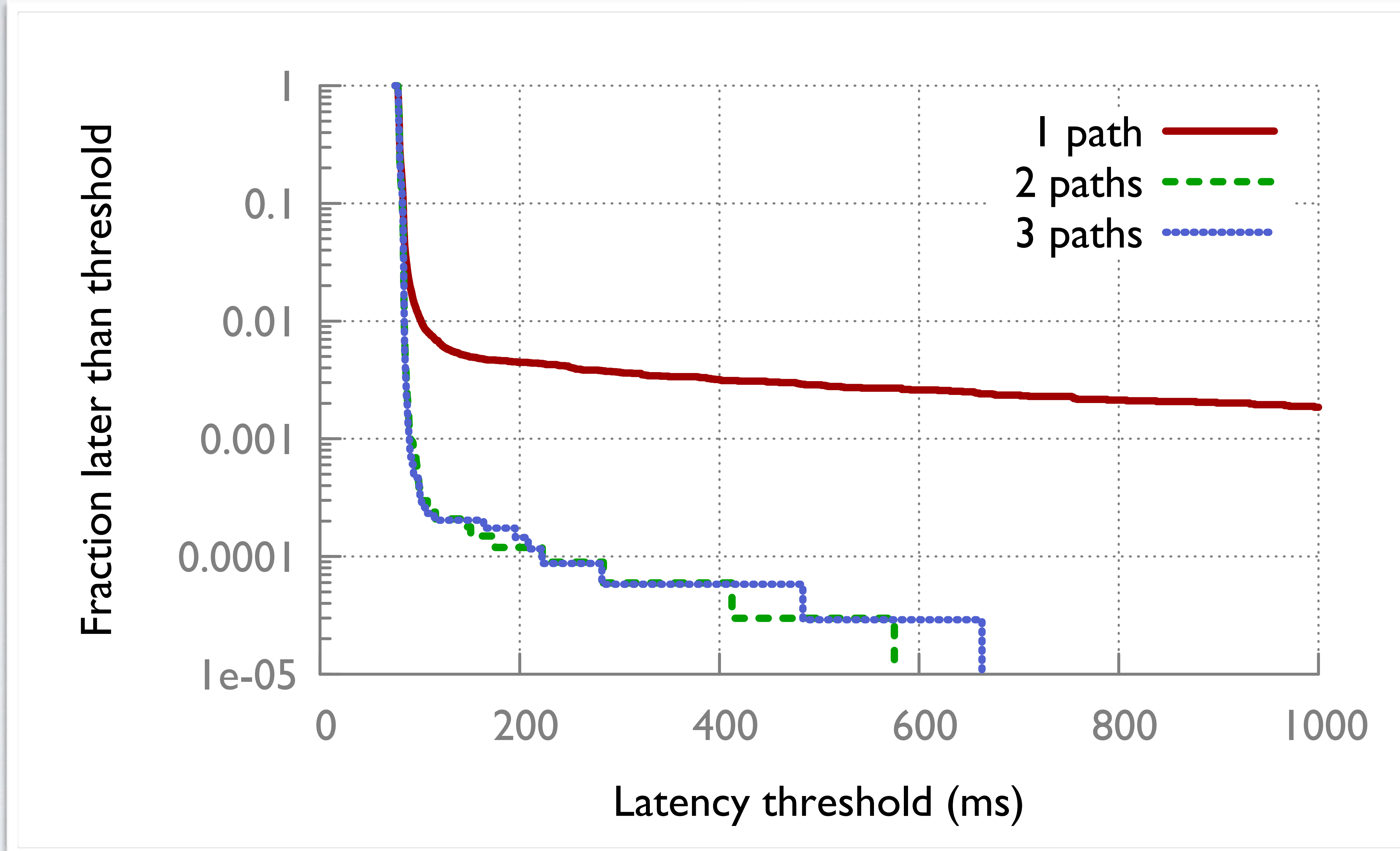
- Send copies of packets on different overlay paths



- Data rate: 32kbps-56kbps

- Topology, data rate both match Skype

Multipath Overlay



Multipath Overlay

	2 paths	3 paths
Mean latency savings (ms/KB)	0,8	0,4
99.9th %ile latency savings (ms/KB)	260	130

Multipath Overlay

	2 paths	3 paths
Mean latency savings (ms/KB)	0,8	0,4
99.9th %ile latency savings (ms/KB)	260	130

Targets

10ms/KB (cell)

0.3ms/KB (DSL)

Experiments

- Experiments on both the GENI Mesoscale/OpenFlow and the PlanetLab and ProtoGENI testbeds

Introduction to GENI

[8] cp. Riga et.al. Introduction to GENI, tutorial NSDI 2013, Lombard, IL

Introduction to GENI

- Science issues
  - cannot currently understand or predict the behavior of complex, large-scale networks

[8] cp. Riga et.al. Introduction to GENI, tutorial NSDI 2013, Lombard, IL

Introduction to GENI

- Science issues
    - cannot currently understand or predict the behavior of complex, large-scale networks
- Innovation issues
    - substantial barriers to at-scale experimentation with new architectures, services and technologies

[8] cp. Riga et.al. Introduction to GENI, tutorial NSDI 2013, Lombard, IL

Introduction to GENI

- Science issues
  - cannot currently understand or predict the behavior of complex, large-scale networks
- Innovation issues
  - substantial barriers to at-scale experimentation with new architectures, services and technologies
- Society issues
  - increasingly rely on the Internet but are unsure we can trust its security, privacy or resilience

[8] cp. Riga et.al. Introduction to GENI, tutorial NSDI 2013, Lombard, IL

Introduction to GENI

- Science issues
  - cannot currently understand or predict the behavior of complex, large-scale networks
- Innovation issues
  - substantial barriers to at-scale experimentation with new architectures, services and technologies
- Society issues
  - increasingly rely on the Internet but are unsure we can trust its security, privacy or resilience

relatively little innovation in the core of the network

[8] cp. Riga et.al. Introduction to GENI, tutorial NSDI 2013, Lombard, IL

Introduction to GENI

Introduction to GENI

- GENI is a nationwide suite of infrastructure for "at scale" experiments in networking, distributed systems, security, and novel applications
  - Federation of existing testbeds including Emulab, ProtoGENI, PlanetLab accessible via a common API

Introduction to GENI

- GENI is a nationwide suite of infrastructure for "at scale" experiments in networking, distributed systems, security, and novel applications
    - Federation of existing testbeds including Emulab, ProtoGENI, PlanetLab accessible via a common API
- GENI opens up huge new opportunities
    - Leading-edge research in next-generation networks
    - Rapid innovation in novel, large-scale applications

Introduction to GENI

Introduction to GENI

- Key GENI concept: slices & deep programmability
  - Openflow: FlowVisor
  - Network: VLANs
  - Hosts: different types of virtualisation or exclusive resource

Introduction to GENI

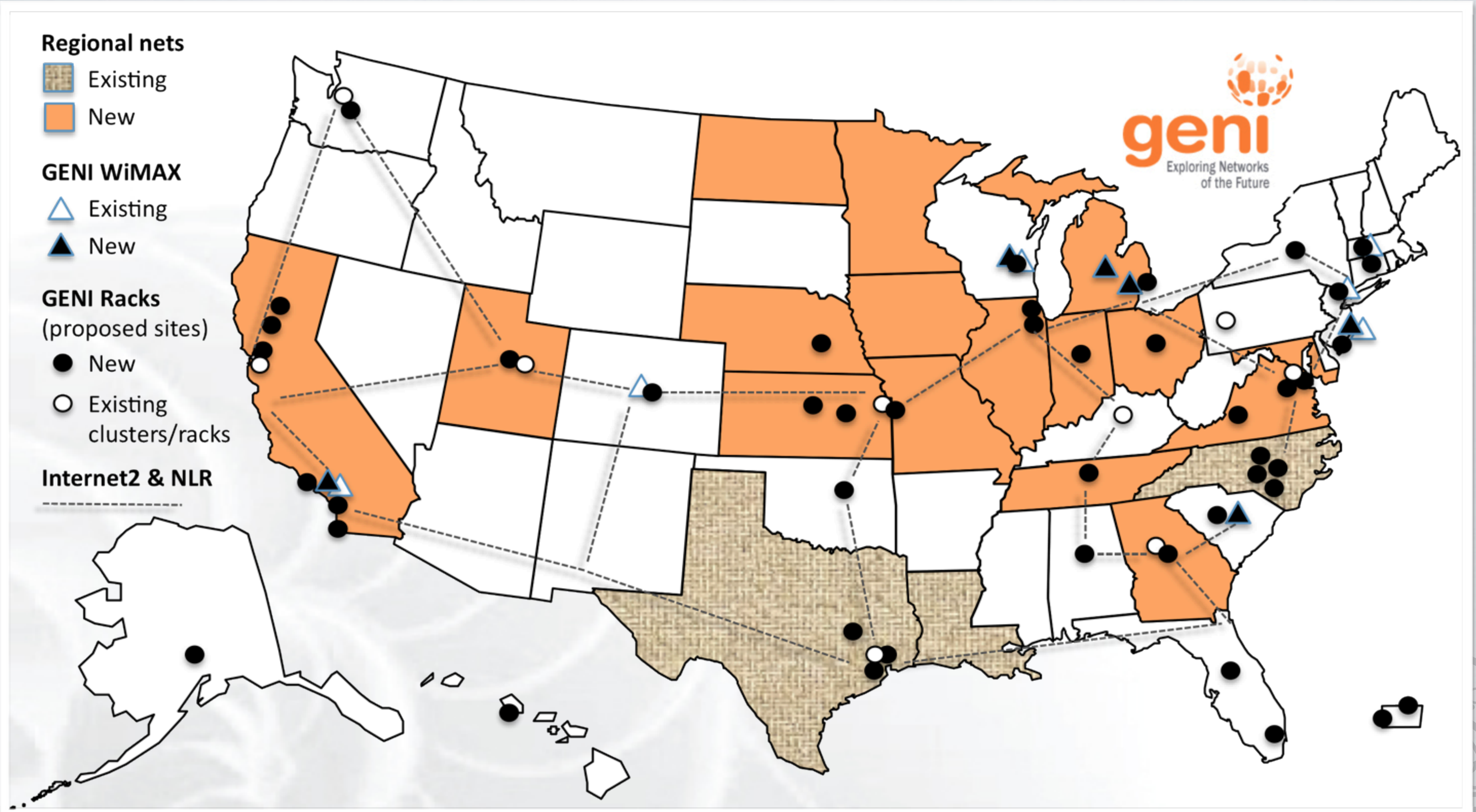
- Key GENI concept: slices & deep programmability
  - Openflow: FlowVisor
  - Network: VLANs
  - Hosts: different types of virtualisation or exclusive resource
- Efforts for connecting overseas (e.G. GLab, Deutsche Telekom)

Introduction to GENI

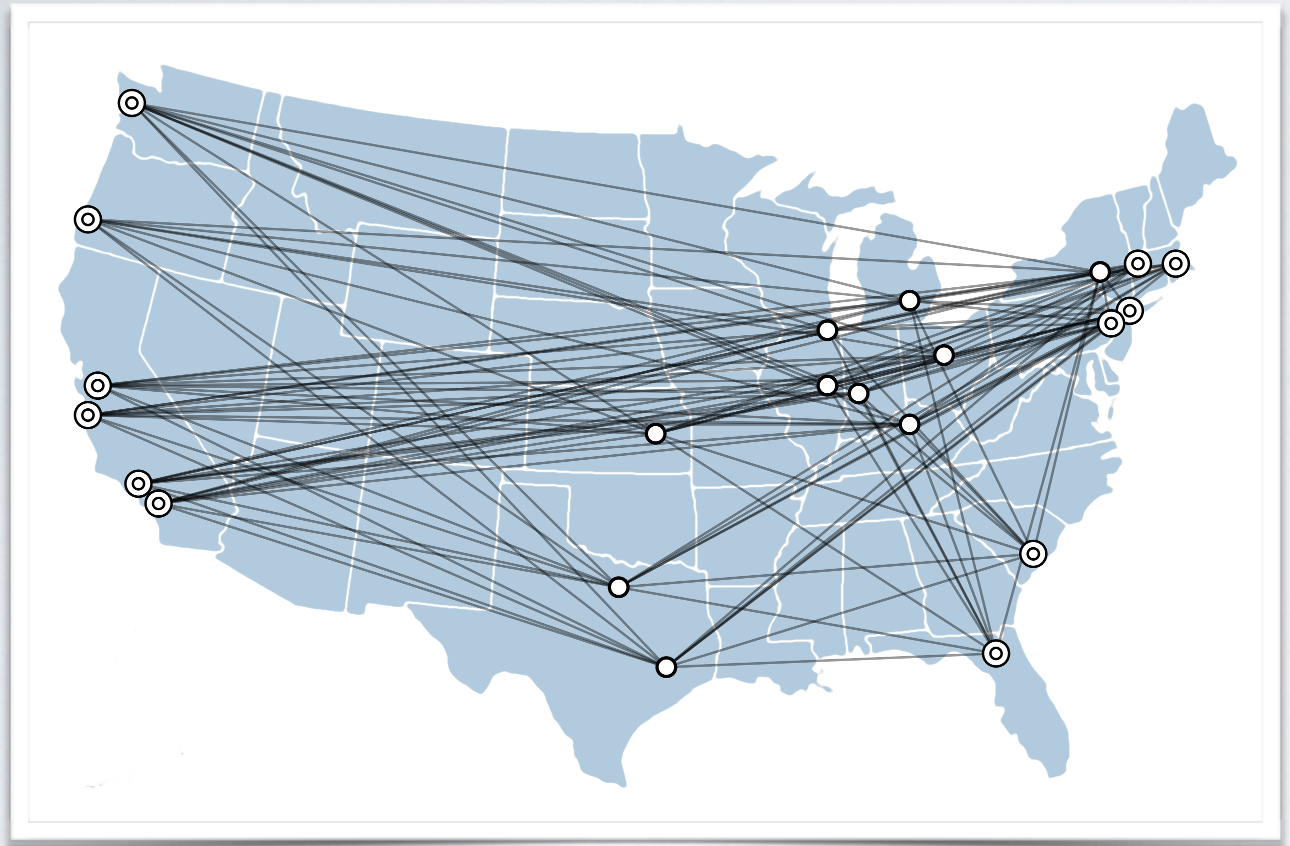
Introduction to GENI

- Two types of wide-area setups possible
  - Overlay: connect host resources through L3 via Internet
  - Mesoscale:
    - L2 data plane, L3 control plane
    - Internet2, NLR physical fiber
    - programmable OpenFlow switches at campuses and I2/NRL POPs
    - wide-area L2 broadcast-domain with hosts directly connected to OF switches

Introduction to GENI

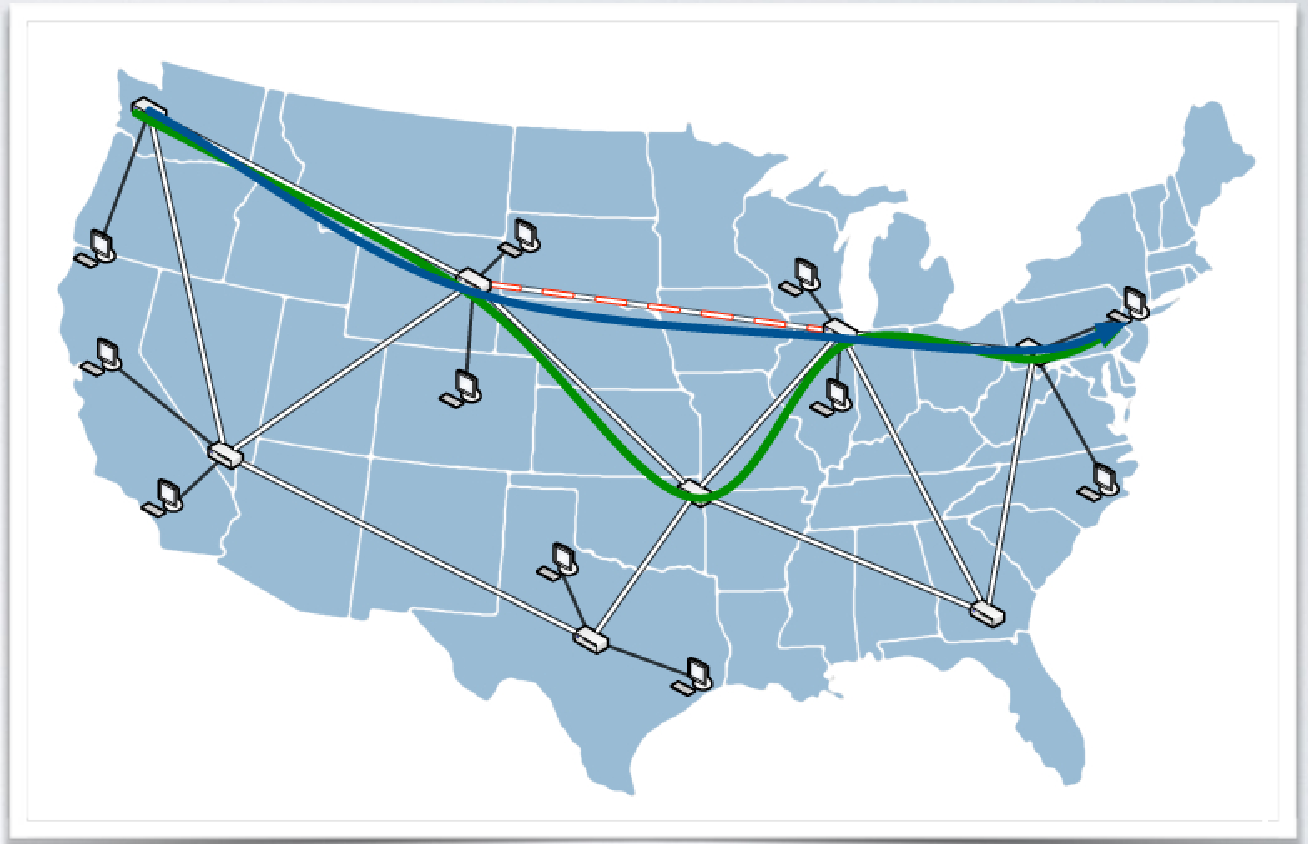


- **Experiments at UIUC**



Overlay

Vulimiri et. al. More is Less - Reducing Latency via Redundancy. HotNets-XI, Redmond, WA



Mesoscale

Michel et. al. Adaptive Source Routing. GEC13, Los Angeles, CA

Conclusion

Bandwidth

↓ Redundancy

Latency

Conclusion

Bandwidth

Redundancy

Latency

- Argument for trading bandwidth for reduced latency in certain scenarios

Conclusion

Bandwidth

Redundancy

Latency

- Argument for trading bandwidth for reduced latency in certain scenarios
- cost-benefit analysis

Conclusion

Bandwidth

$\downarrow$ Redundancy

Latency

- Argument for trading bandwidth for reduced latency in certain scenarios
- cost-benefit analysis
- DNS, wide-area multipath overlay experiments

Conclusion

Bandwidth

$\downarrow$ Redundancy

Latency

- Argument for trading bandwidth for reduced latency in certain scenarios
- cost-benefit analysis
- DNS, wide-area multipath overlay experiments
- Overview over used resources within the GENI federation

Conclusion

Bandwidth

$\downarrow$ Redundancy

Latency

Vulimiri, A., Michel, O., Godfrey, P. B., Shenker, S.

"More is Less - Reducing Latency via Redundancy"

11th ACM Workshop on Hot Topics in Networks (HotNets-XI)

October 2012, Redmond, WA

References

[1] Vulimiri, A., Michel, O. Godfrey, P. B., Shenker, S. More is Less - Reducing Latency via Redundancy. HotNets '12, Redmond, WA

[2] Michel, O., Vulimiri, A., Godfrey, P. B. Adaptive Source Routing. GEC13, Los Angeles, CA.

[3] Partition/Aggregate Pattern - Alizadeh et.al.. "Data Center TCP", SIGCOMM '10

[4] L. Pantel, L.C. Wolf, "On the impact of delay on real-time multiplayer games", NOSSDAV '02

[5] Ananthanarayanan et al., "Why let resources idle? Aggressive cloning of jobs using Dolly", HotCloud '12

[6] Soljanin, "Reducing delay with coding in multi-agent information transfer", Allerton '10

[7] Li et al., "Bandwidth efficient management of DHT routing tables", NSDI '10

[8] Riga et.al. Introduction to GENI, tutorial NSDI 2013, Lombard, IL

Thanks

Backup slides

How can you mitigate overhead?

- Strict prioritization
- Redundancy elimination[*]
- Network coding (fractional replication)

[*] Han et al., "RPT: re-architecting loss protection for content-aware networks", NSDI '12