Wide-area Dissemination under Strict Timeliness, Reliability, and Cost Constraints

Amy Babay, Emily Wagner, Yasamin Nazari,
Michael Dinitz, and Yair Amir

Distributed Systems
and Networks Lab
www.dsn.jhu.edu

Problem: Combining Timeliness and Reliability over the Internet

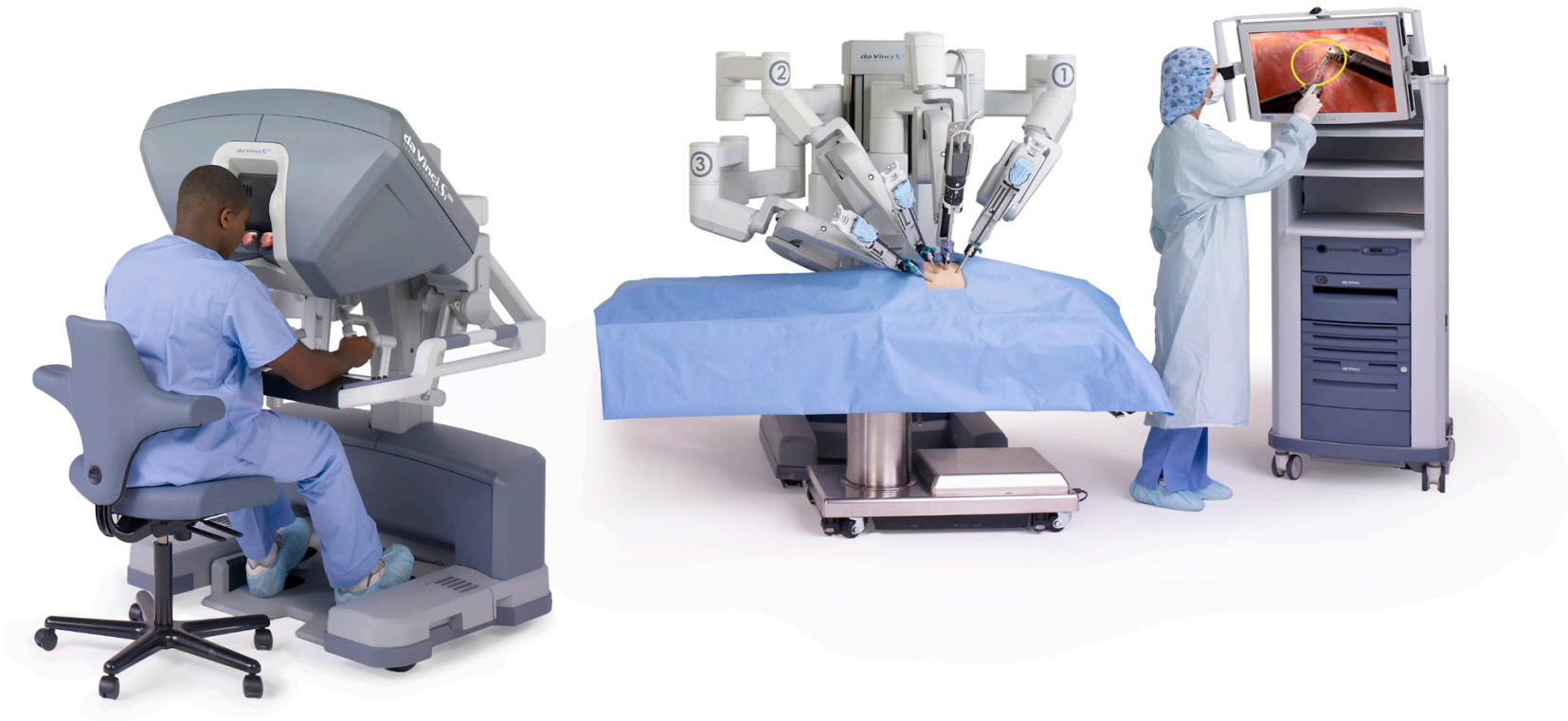
- Internet natively supports end-to-end reliable (e.g. TCP) or best-effort timely (e.g. UDP) communication
- Our goal: support applications with extremely demanding combinations of timeliness and reliability requirements in a cost-effective manner
- Applications have emerged over the past few years that require both timeliness guarantees and high reliability
  - e.g. VoIP, broadcast-quality live TV transport

State-of-the-art: Combining Timeliness and Reliability over the Internet

200ms one-way latency requirement, 99.999% reliability guarantee

40ms one-way propagation delay across North America

New Challenges: Combining Timeliness and Reliability



130ms **round-trip** latency requirement

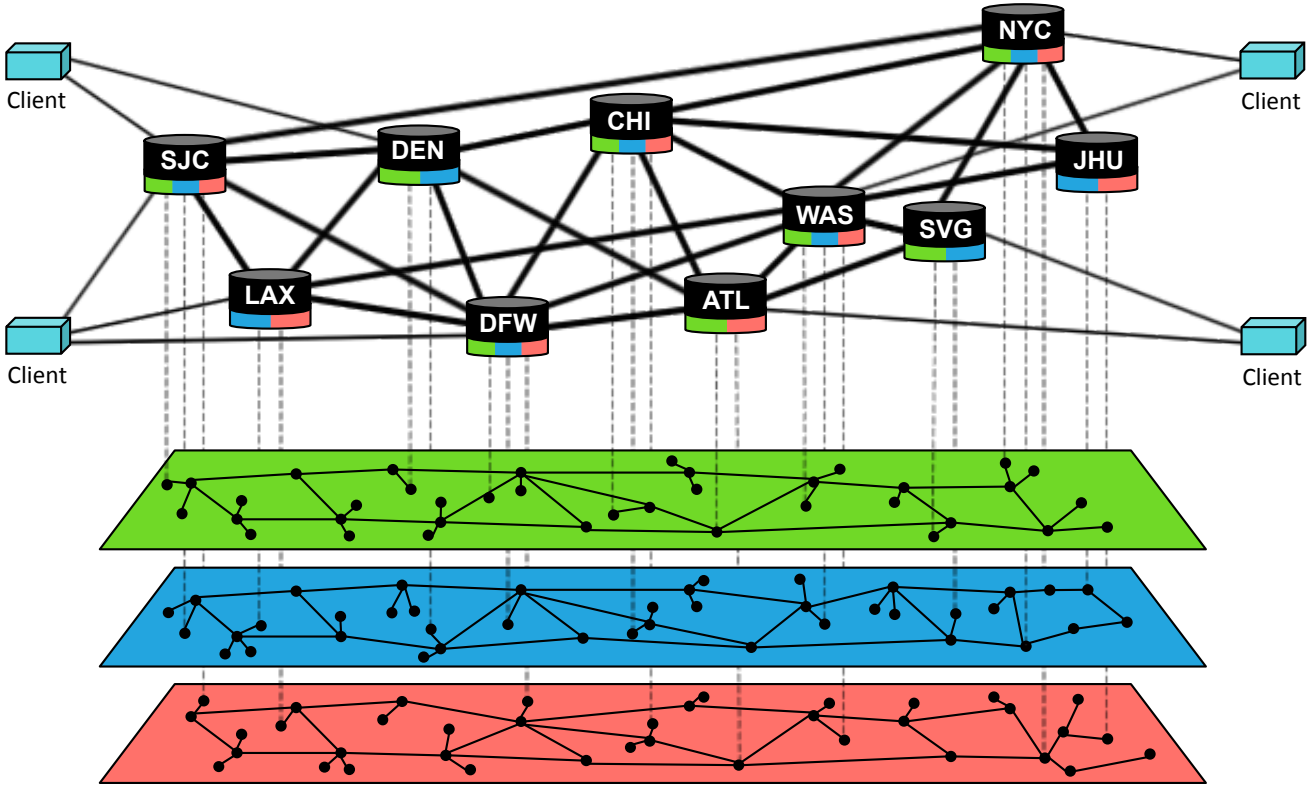
New Challenges: Combining Timeliness and Reliability

130ms **round-trip** latency requirement

80ms round-trip propagation delay across North America

State-of-the-art: Combining Timeliness and Reliability over the Internet

- Overlay networks enable specialized routing and recovery protocols



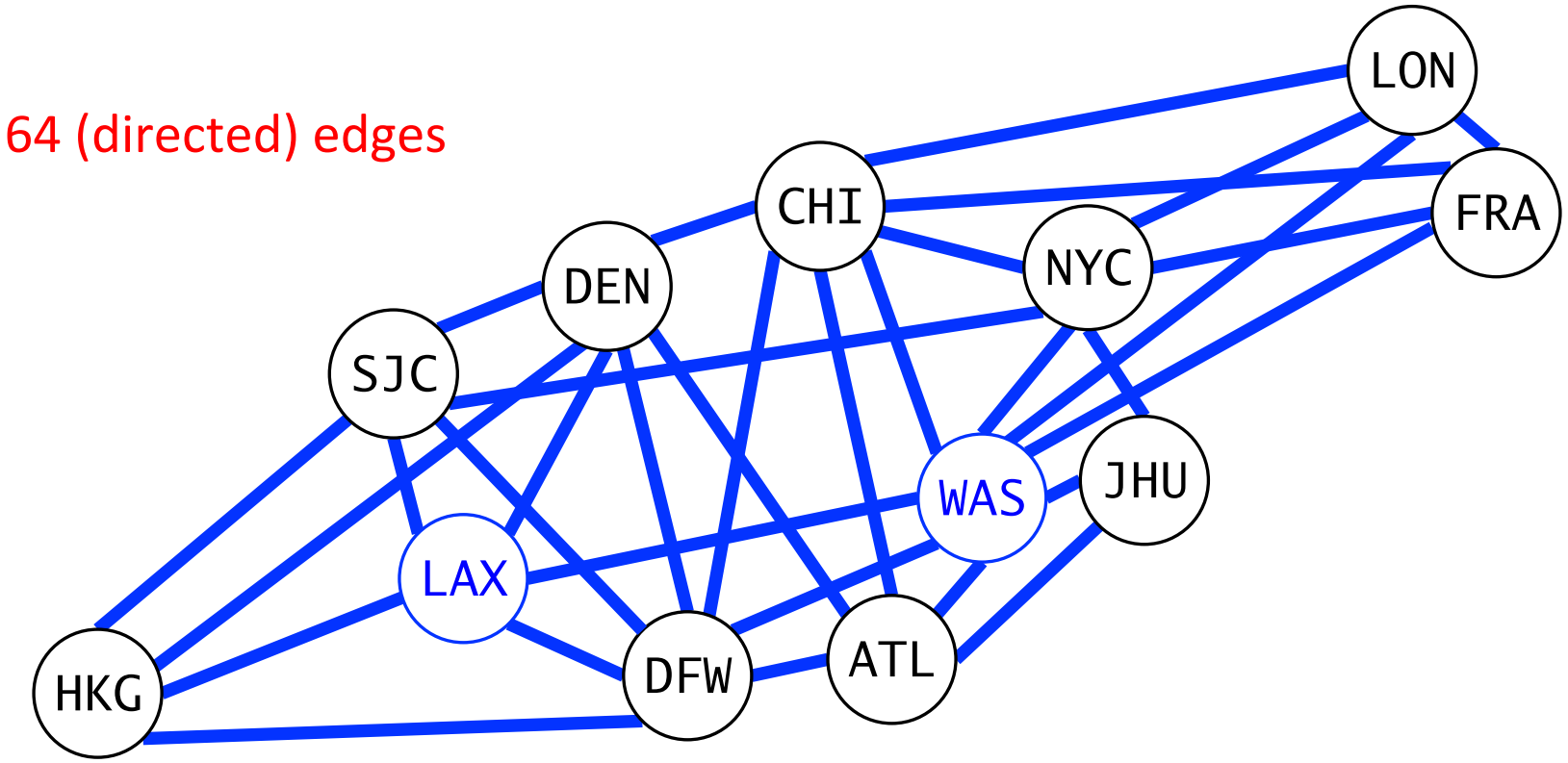
Addressing New Challenges: Dissemination Graph Approach

- Stringent latency requirements give less flexibility for buffering and recovery
- Core idea: Send packets redundantly over a subgraph of the network (a dissemination graph) to maximize the probability that at least one copy arrives on time

How do we select the subgraph (subset of overlay links) on which to send each packet?

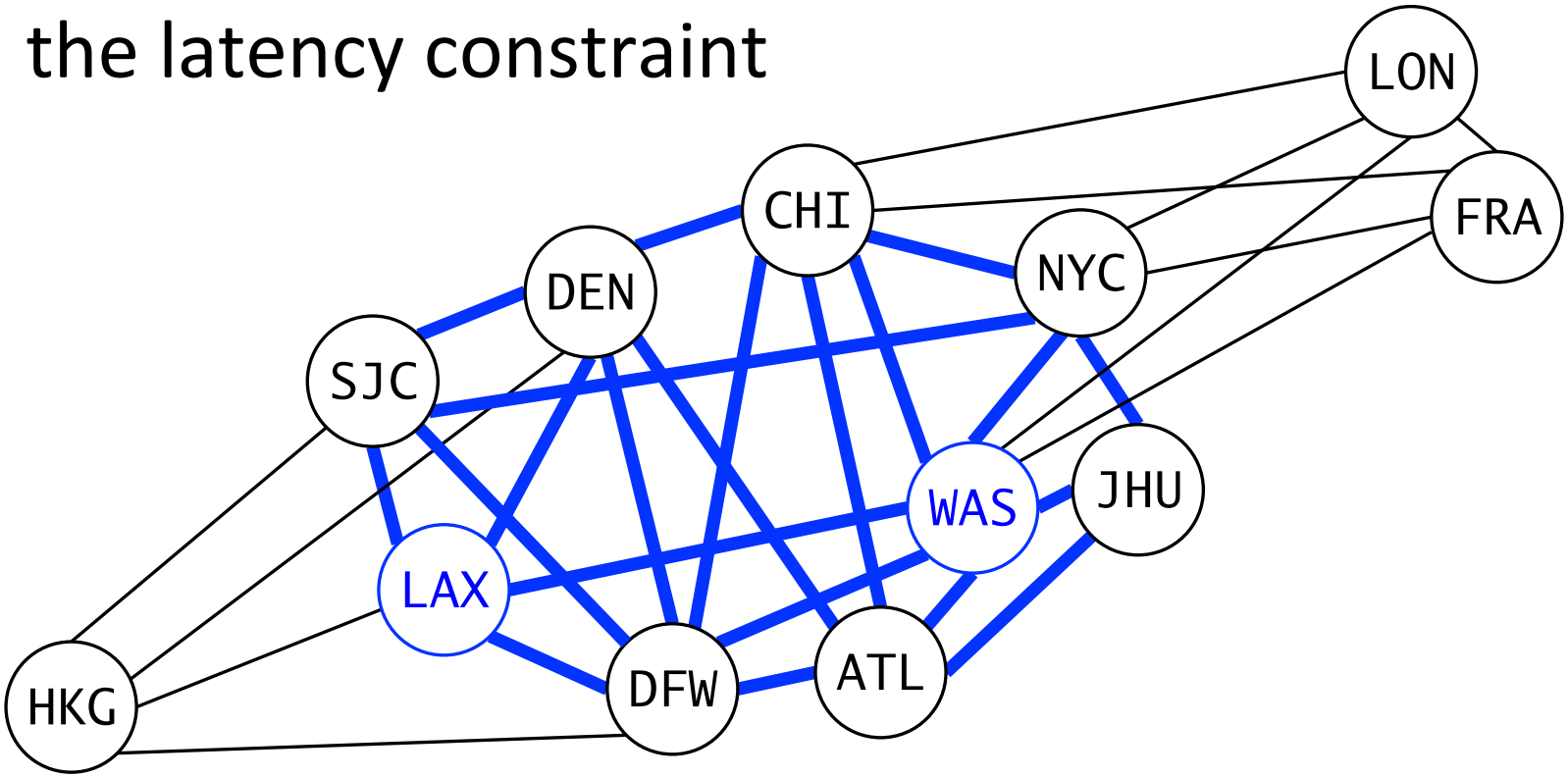
Initial Approaches to Selecting a Dissemination Graph

- Overlay Flooding: send on all overlay links
  - Optimal in timeliness and reliability but expensive



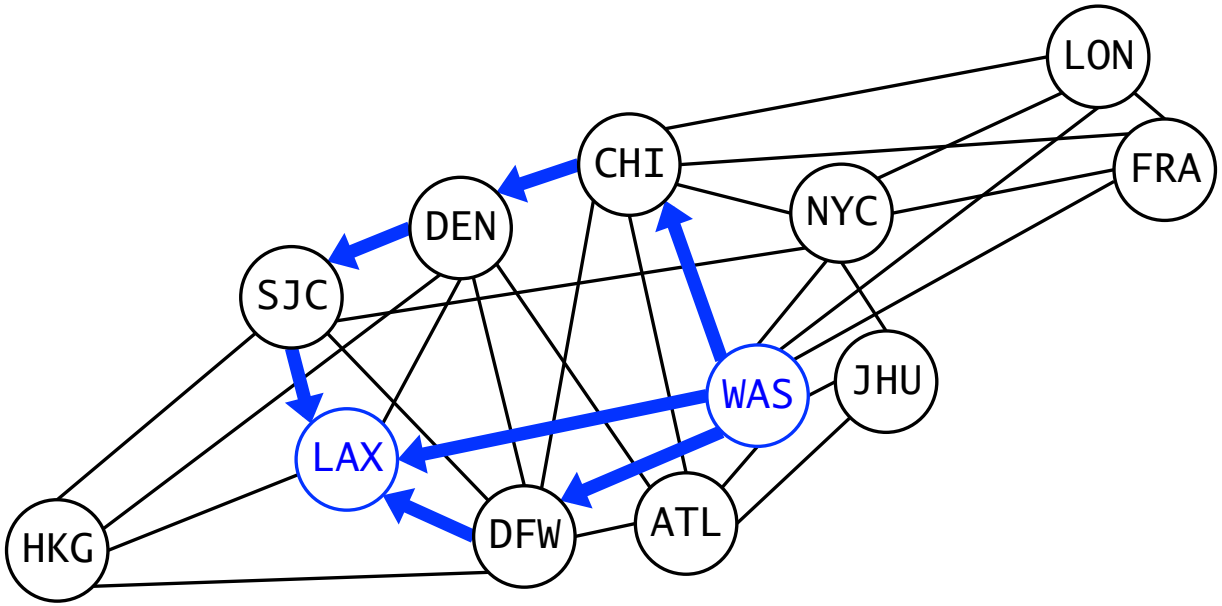
Initial Approaches to Selecting a Dissemination Graph

- Time-Constrained Flooding: flood only on edges that can reach the destination within the latency constraint



Initial Approaches to Selecting a Dissemination Graph

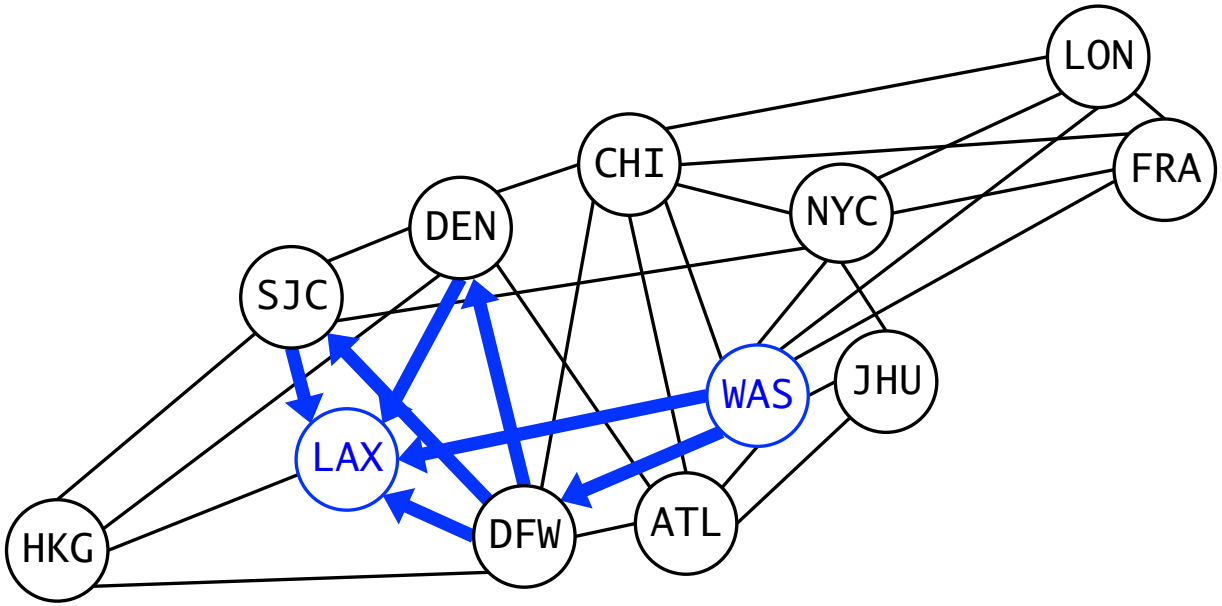
- Disjoint Paths: send on several paths that do not share any nodes (or edges)
  - Good trade-off between cost and timeliness/reliability
  - Uniformly invests resources across the network



Selecting an Optimal Dissemination Graph

Can we use knowledge of the network characteristics to do better?

Invest more resources in more problematic regions:



Problem Definition: Selecting an Optimal Dissemination Graph

- We want to find the best trade-off between cost and reliability (subject to timeliness)
  - Cost: # of times a packet is sent (= # of edges used)
  - Reliability: probability that a packet reaches its destination within its application-specific latency constraint (e.g. 65ms)
- **Client perspective**: maximize reliability achieved for a fixed budget
- **Service provider perspective**: minimize cost of providing an agreed upon level of reliability (SLA)

Selecting an Optimal Dissemination Graph

- Solving the proposed problems is NP-hard
  - Without the latency constraint, computing reliability is the two-terminal reliability problem (which is #P-complete)
  - Computing optimal dissemination graphs in terms of cost and reliability is also NP-hard
- We expand on this later in the talk

Data-Informed Dissemination Graphs

- Goal: Learn about the types of problems that occur in the field and tailor dissemination graphs to address common problem types
- Collected data on a commercial overlay topology (www.ltnglobal.com) over 4 months
- Analyzed how different dissemination-graph-based routing approaches (time-constrained flooding, single path, two disjoint paths) would perform (Playback Network Simulator)

Data-Informed Dissemination Graphs

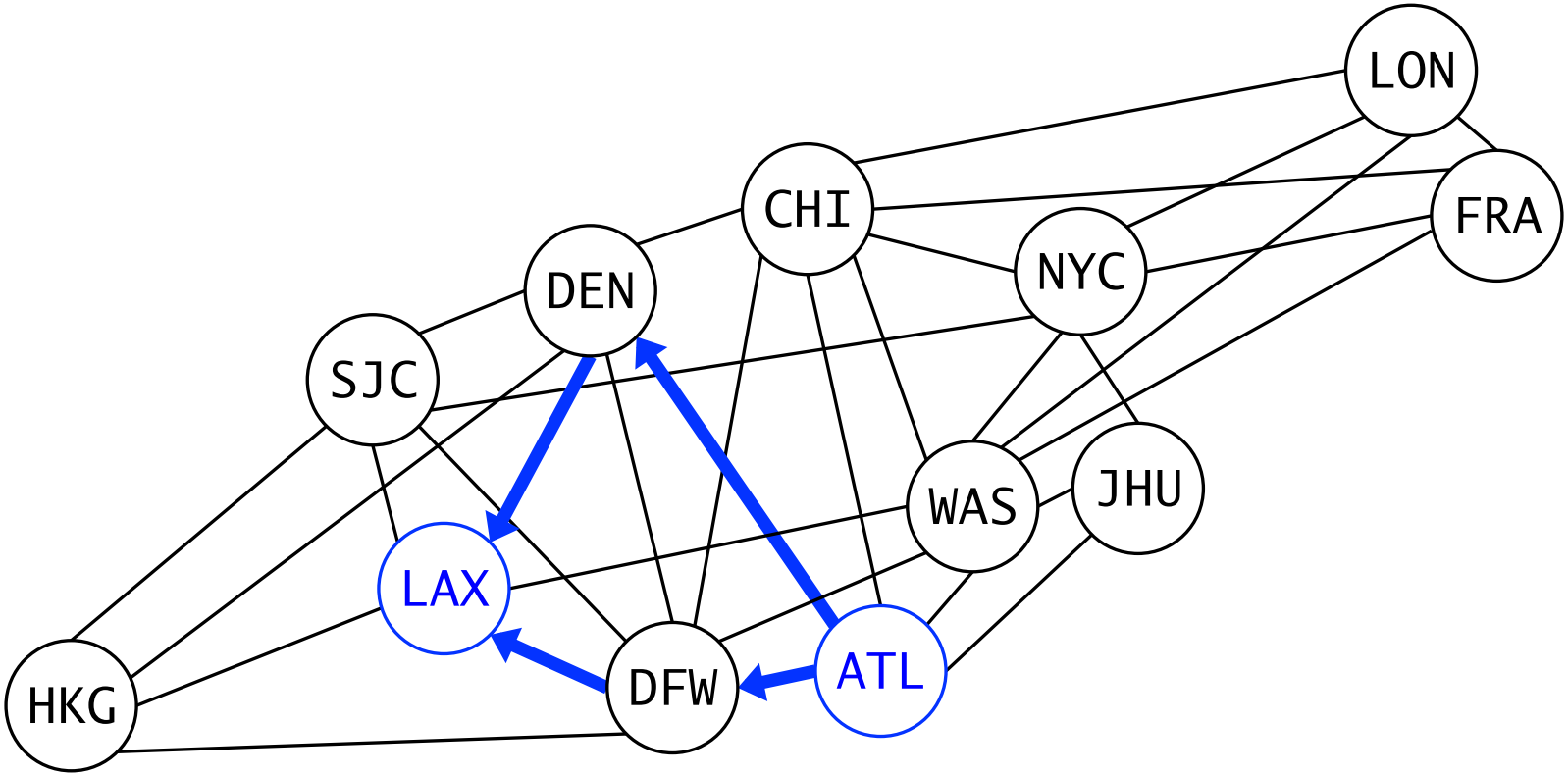
- Key findings:
  - Two disjoint paths provide relatively high reliability overall
    - Good building block for most cases
  - Almost all problems not addressed by two disjoint paths involve either:
    - A problem at the source
    - A problem at the destination
    - A problem at both the source and the destination

Dissemination Graphs with Targeted Redundancy

- Our approach:
  - Pre-compute four graphs per flow (more on this later):
    - Two disjoint paths (static)
    - Source-problem graph
    - Destination-problem graph
    - Robust source-destination problem graph
  - Use two disjoint paths graph in the normal case
  - If a problem is detected at the source and/or destination of a flow, switch to the appropriate pre-computed dissemination graph
  - Converts optimization problem to classification problem

Dissemination Graphs with Targeted Redundancy: Case Study

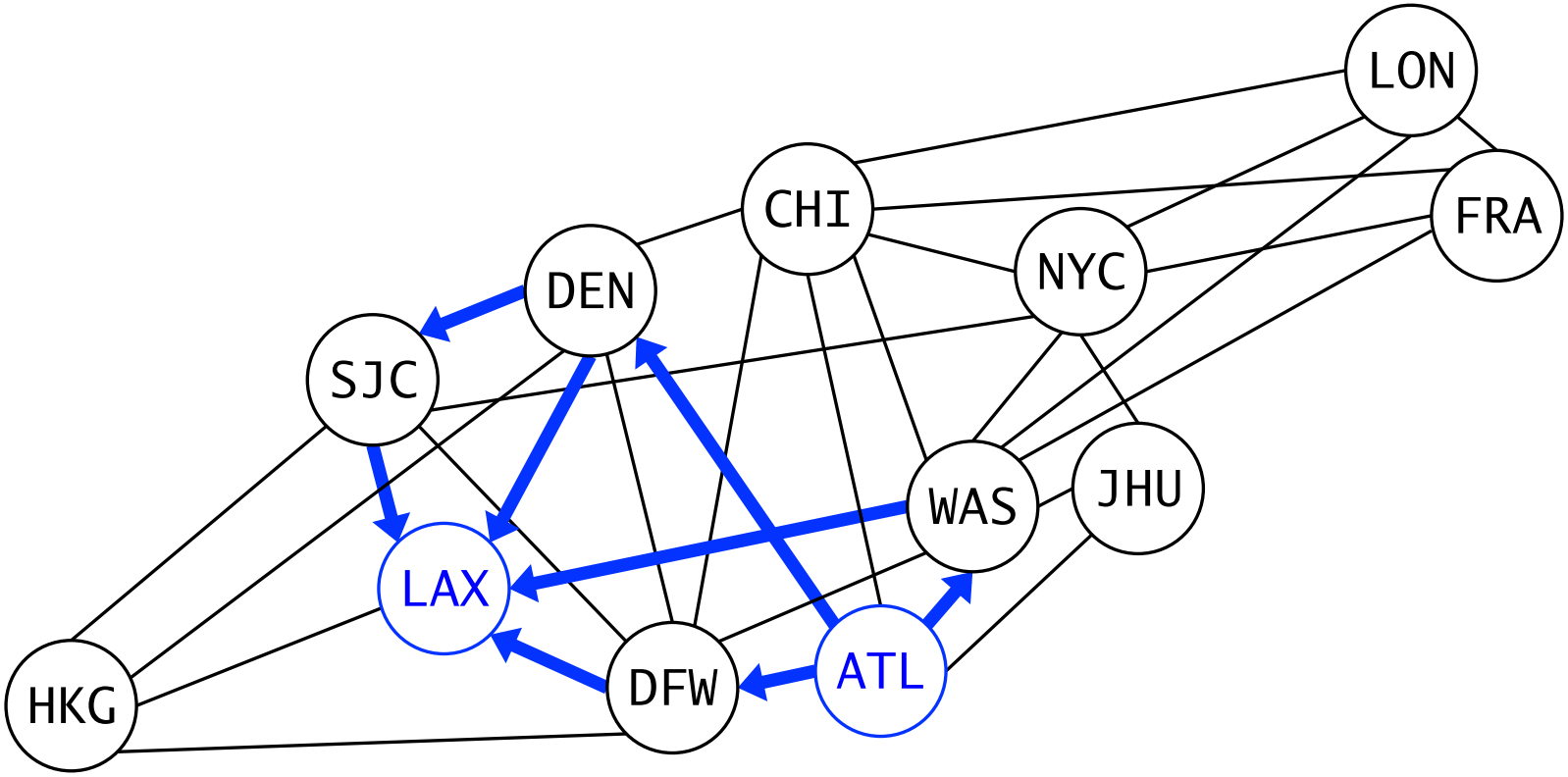
- Case study: Atlanta -> Los Angeles



Two node-disjoint paths dissemination graph (4 edges)

Dissemination Graphs with Targeted Redundancy: Case Study

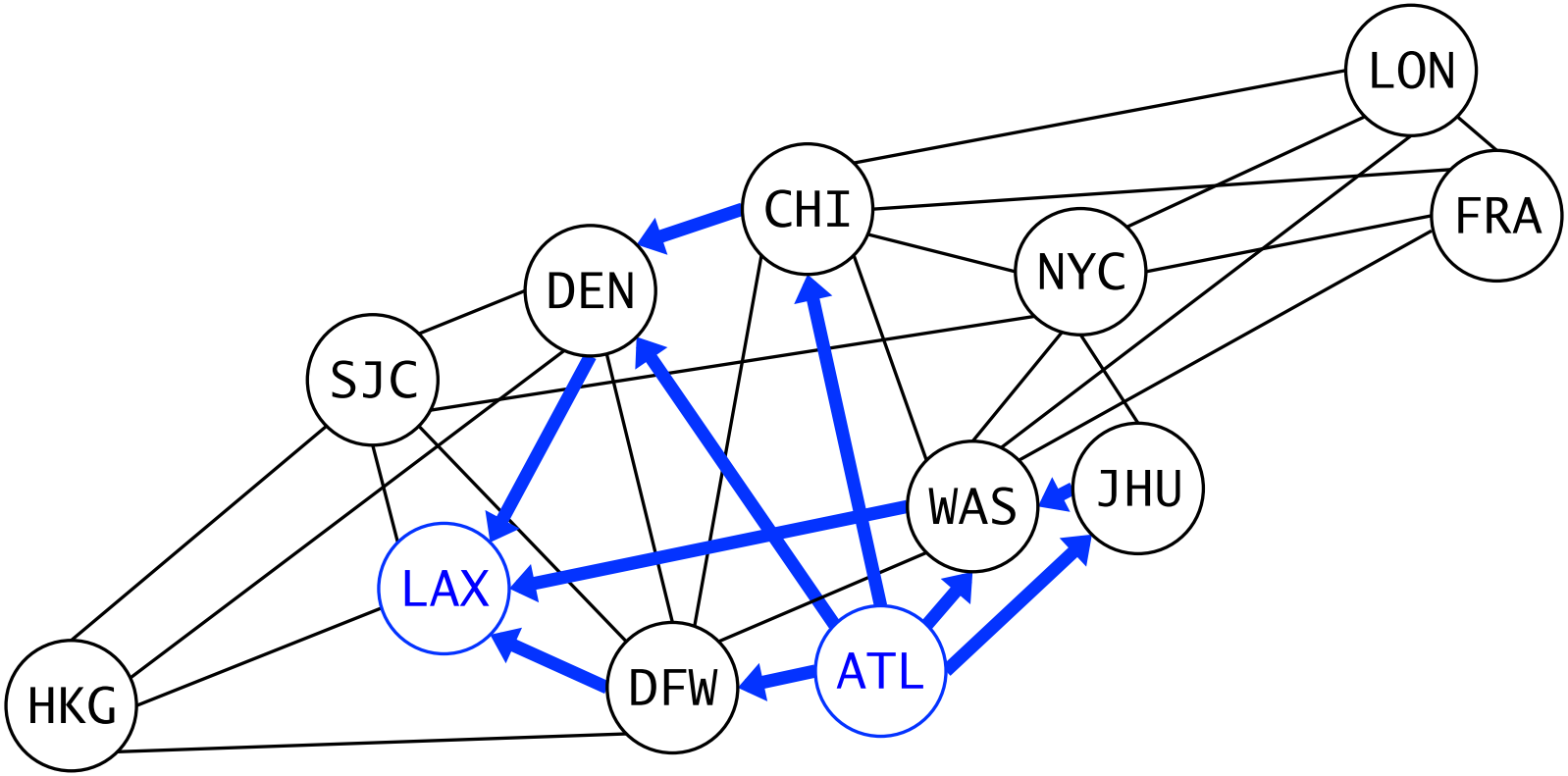
- Case study: Atlanta -> Los Angeles



Destination-problem dissemination graph (8 edges)

Dissemination Graphs with Targeted Redundancy: Case Study

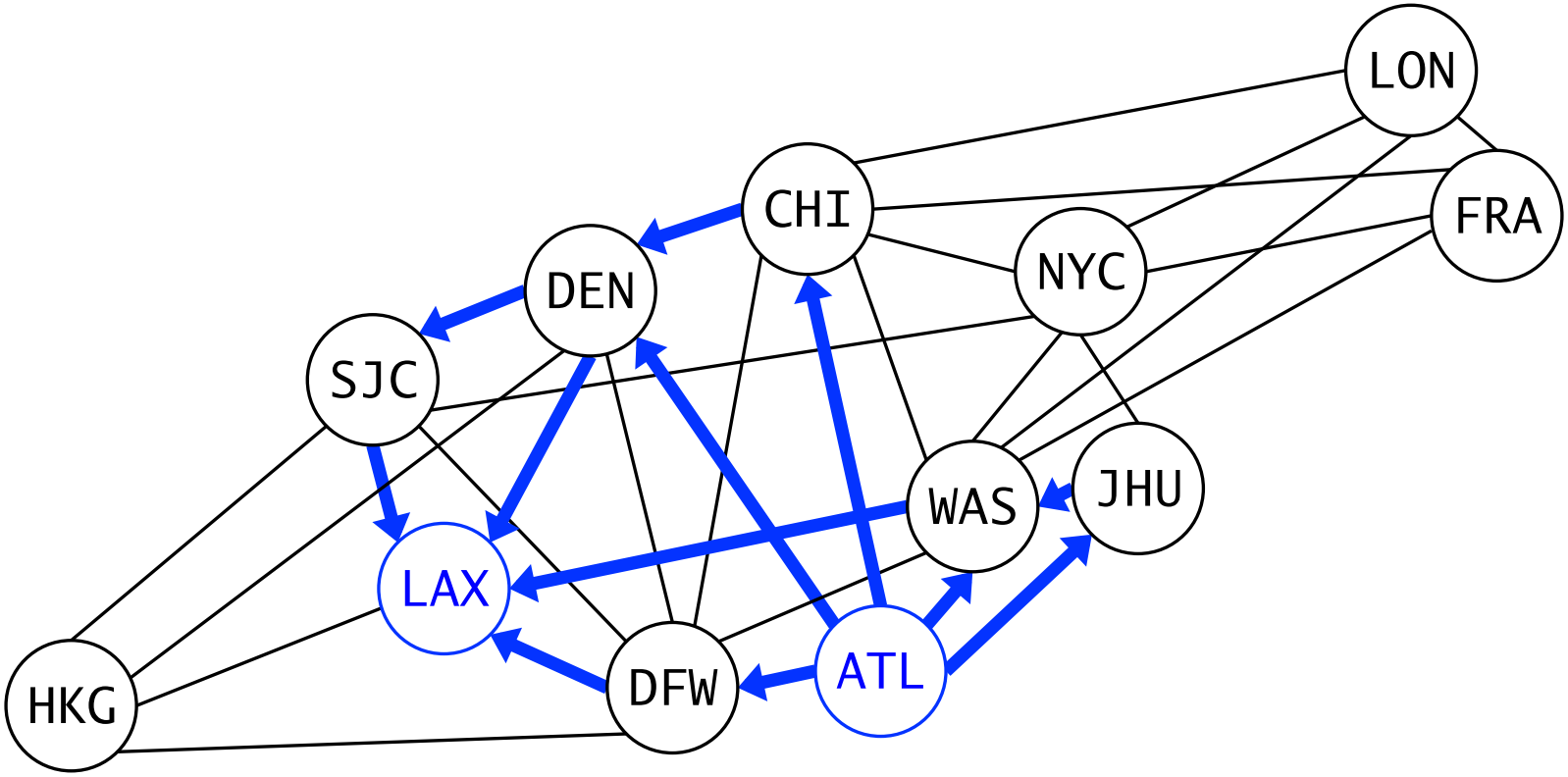
- Case study: Atlanta -> Los Angeles



Source-problem dissemination graph (10 edges)

Dissemination Graphs with Targeted Redundancy: Case Study

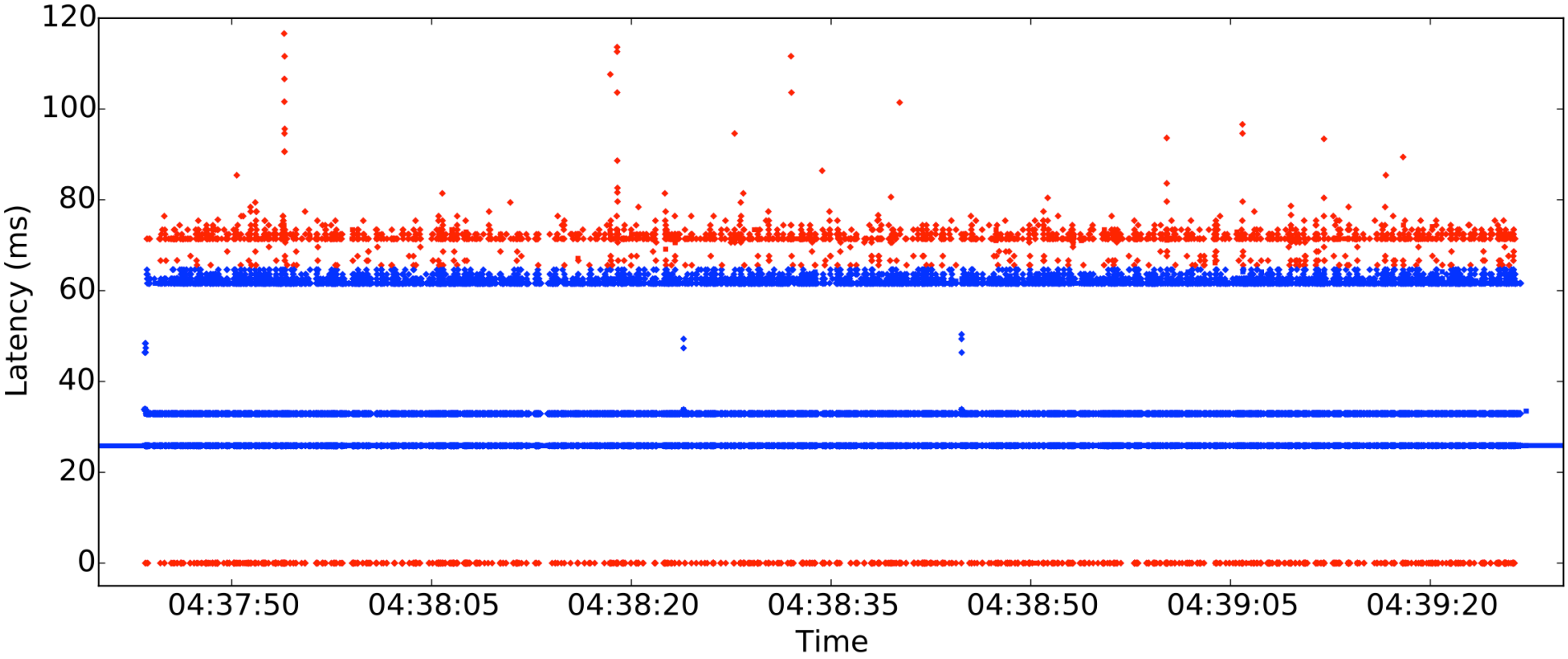
- Case study: Atlanta -> Los Angeles



Robust source-destination-problem dissemination graph (12 edges)

Dissemination Graphs with Targeted Redundancy: Case Study

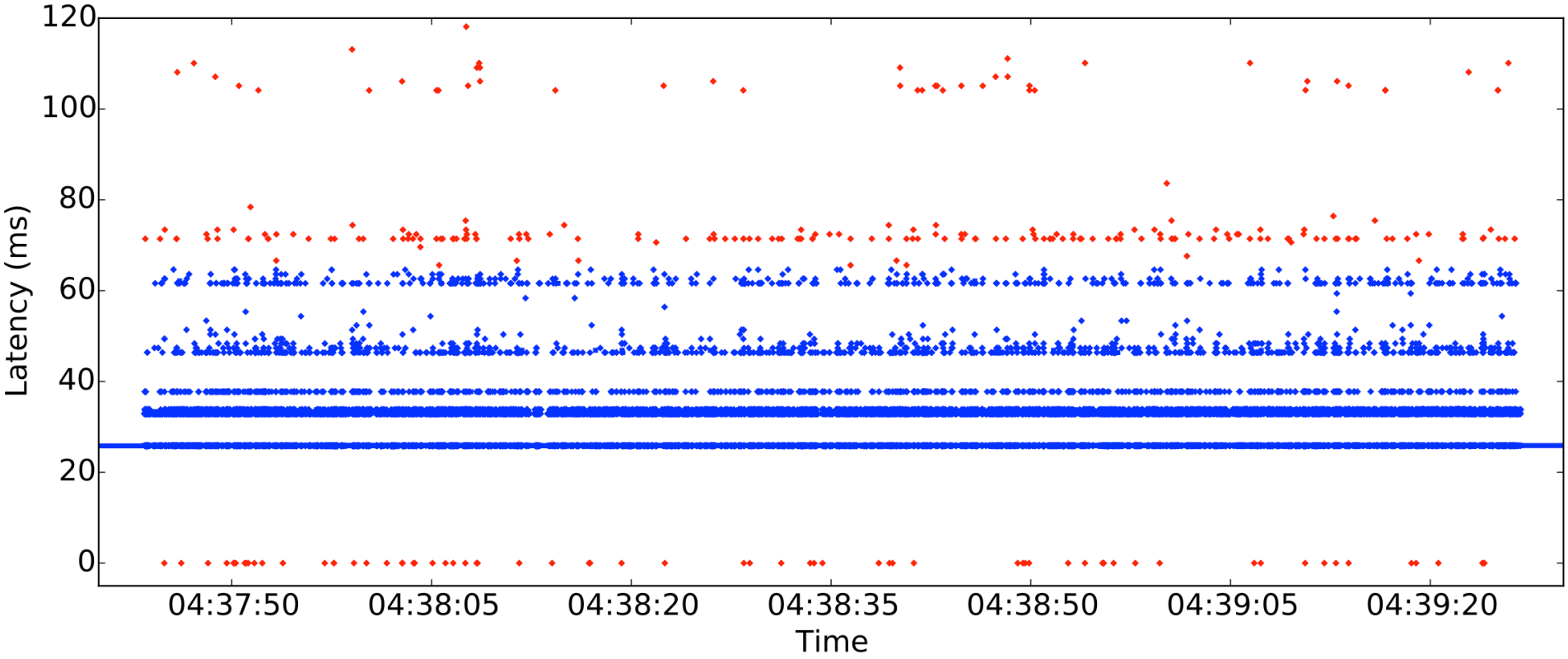
- Case study: Atlanta -> Los Angeles; August 15, 2016



Packets received and dropped over a 110-second interval using (adaptive) two disjoint paths (3982 lost/late packets, 20 packets with latency over 120ms not shown)

Dissemination Graphs with Targeted Redundancy: Case Study

- Case study: Atlanta -> Los Angeles; August 15, 2016



Packets received and dropped over a 110-second interval using our dissemination-graph-based approach to add targeted redundancy at the destination (299 lost/late packets)

Dissemination Graphs with Targeted Redundancy: Results

- 4 weeks of data collected over 4 months
  - Packets sent on each link in the overlay topology every 10ms
- Analyzed 16 transcontinental flows
  - All combinations of 4 cities on the East Coast of the US (NYC, JHU, WAS, ATL) and 2 cities on the West Coast of the US (SJC, LAX)
  - 1 packet/ms simulated sending rate

Dissemination Graphs with Targeted Redundancy: Results

Routing Approach	Availability (%)	Unavailability (seconds per flow per week)	Reliability (%)	Reliability (packets lost/ late per million)
Time-Constrained Flooding	99.995887%	24.88	99.999854%	1.46
Dissemination Graphs with Targeted Redundancy	99.995886%	24.88	99.999848%	1.52
Dynamic Two Disjoint Paths	99.995866%	25.00	99.998913%	10.87
Static Two Disjoint Paths	99.995521%	27.09	99.998453%	15.47
Redundant Single Path	99.995764%	25.62	99.998535%	14.65
Single Path	99.994206%	35.04	99.997605%	23.95

Results: % of Performance Gap Covered (between TCF and Single Path)

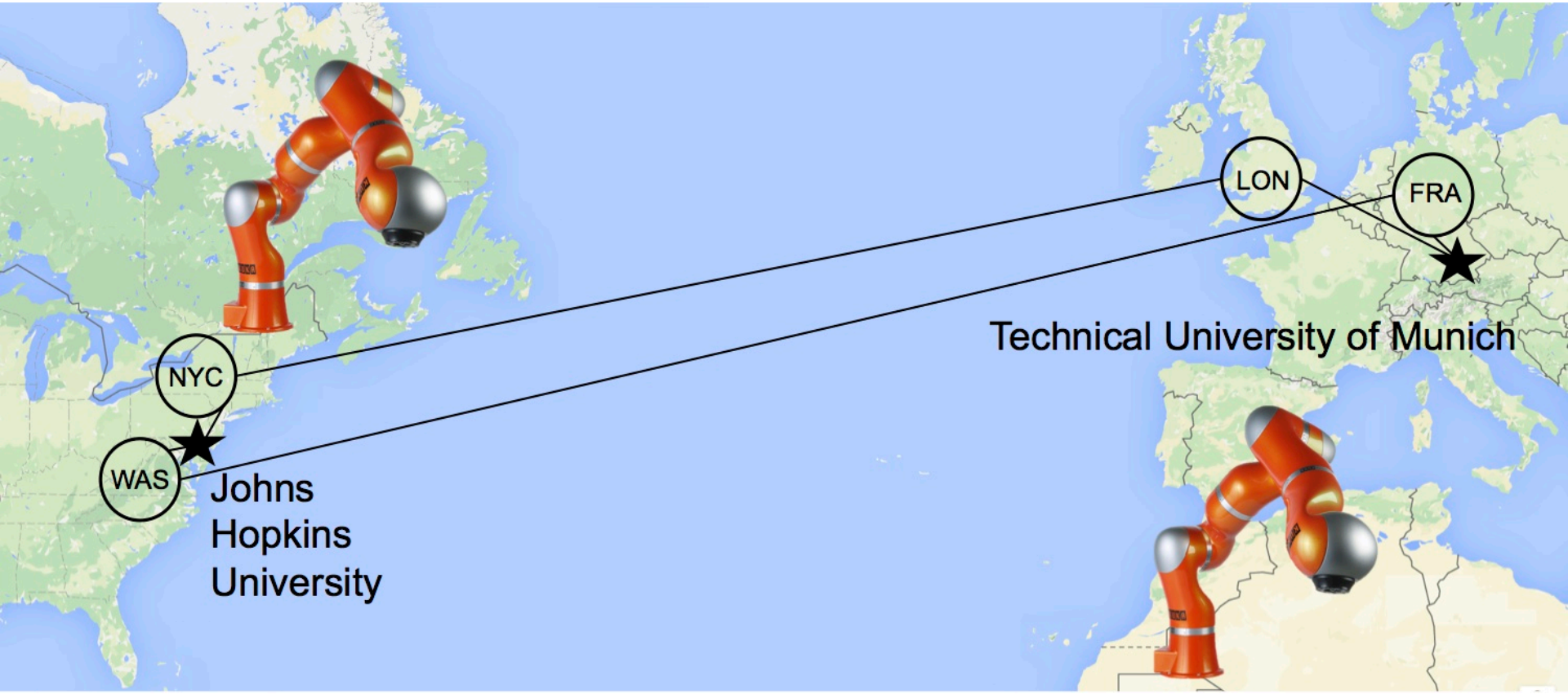
Routing Approach	Week 1 2016-07-19	Week 2 2016-08-08	Week 3 2016-09-01	Week 4 2016-10-13	Overall	Scaled Cost
Time-Constrained Flooding	100.00%	100.00%	100.00%	100.00%	**100.00%**	**15.75**
Dissem. Graphs with Targeted Redundancy	99.05%	99.73%	98.53%	99.94%	**99.81%**	**2.098**
Dynamic Two Disjoint Paths	73.63%	67.73%	94.75%	69.69%	**69.65%**	**2.059**
Static Two Disjoint Paths	37.89%	43.18%	-175.13%	51.63%	**44.58%**	**2.059**
Redundant Single Path	67.06%	47.72%	43.12%	58.00%	**54.59%**	**2.000**
Single Path	0.00%	0.00%	0.00%	0.00%	**0.00%**	**1.000**

Applications: Remote Manipulation

Video demonstration: www.dsn.jhu.edu/~babay/Robot_video.mp4

Applications: Remote Robotic Ultrasound

- Collaboration with JHU/TUM CAMP lab (https://camp.lcsr.jhu.edu/)



Part II: Theory

- Computing optimal dissemination graphs:
  - Formalization of problem
  - Hardness
  - Limited Progress
- Targeted redundancy:
  - Problem at source or destination
  - Problem at both
  - Which graphs to compute?

Optimizing Dissemination Graphs

- Input:
  - $G = (V, E)$, $s,t \in V$
  - $p : E \rightarrow [0,1]$
  - $d: E \rightarrow R^+$; $c : E \rightarrow R^+$
  - $L \in R$; $B \in R$

- G_p: subgraph where each e fails w.p. p(e)

- Find subgraph H with minimum # edges s.t. *$Pr[s,t$ at distance at most L in $H_p] \geq B$*

Optimizing Dissemination Graphs

- Bad news: computing Pr[s,t connected in G_p] is *#P*-hard [Valiant]
  - So can't even tell if purported solution is feasible

Reliability

- But how hard is it *really*?
  - If reliability not incredibly close to 0, Monte Carlo sampling + Chernoff bound give $(1+\varepsilon)$-approx
  - For us, need reliability to be very large: maybe can still approximate optimal dissemination graph?

Ideas & Results

- Want practical, fast algorithms, so try greedy, local search, etc.
  - Counterexamples to everything

- Try 2: write (exponential-size) LP
  - Impractical , fine in theory
  - Can only approximately separate
  - How to round??

Ideas & Results

- Sample Average Approximation (SAA): sample scenarios, optimize just for sampled scenarios
  - Bad news: arbitrary samples is *Label Cover*-hard!

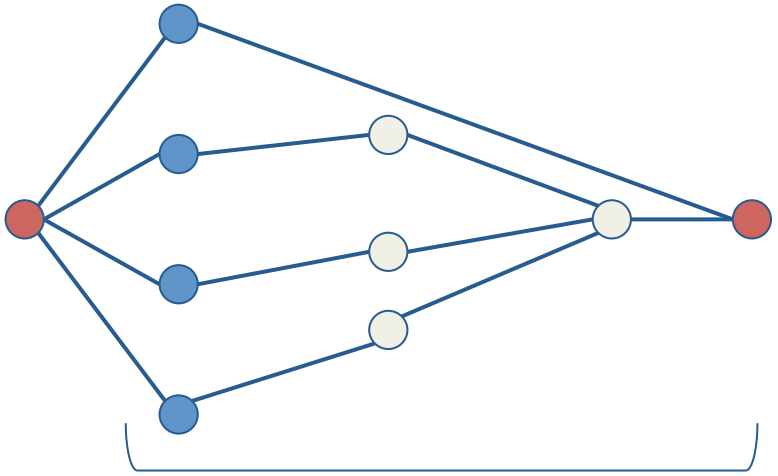
- If all samples trees:
  - Use *Minimum p-Union* approximation [D-Chlamtac-Makarychev '17] (generalization of Densest k-Subgraph) ⇒ $O(n^{1/2})$-approx
  - Still as hard as Densest k-Subgraph!

Targeted Redundancy

- Wanted to precompute dissemination graphs for problem at source, at sink, and at both
- What should these graphs be? Can we find the best?
  - Ongoing work...

Source or Sink Problem

- Send to all neighbors of source
- Cheapest tree where all neighbors have short enough path to sink

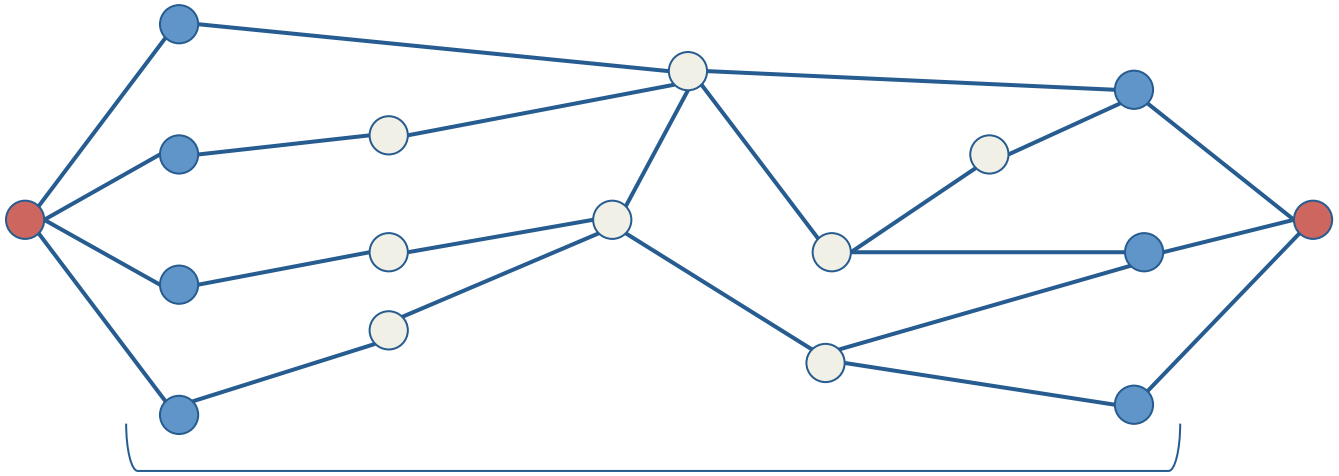


≤ L

- *Shallow-Light Steiner Tree*
- Known approximations, bicriteria approximations
- Currently: brute-force optimal solution

Source *and* Sink Problem

- Graph should be:
  - All neighbors S of source
  - All neighbors T of sink
  - Path of length at most L between all $s \in S$, $t \in T$



$\leq L$

Bipartite Shallow-Light Steiner Network

- *Label Cover*-hard (unlike shallow-light tree)
- $O(n^{3/5})$-approx using pairwise spanner approx [Chlamtac-D-Kortsarz-Laekhanukit '17]
- *(polylog, polylog)*-bicriteria

Next Steps: Theory

- Spinoffs of optimal dissemination graphs
  - Stochastic vaccination problems, with Aravind Srinivasan (UMD) & Anil Vullikanti (Va Tech)

- Bipartite Shallow-Light Network
  - Better approximations?
  - Exact algorithm, exponential in # terminals?
  - Generally: effect of *demand graph* on shallow-light / spanner problems?

Next Steps: Practice

- Deploying the full system and validating the simulation
  - Implementing dissemination-graph-based routing in the Spines Overlay Messaging Framework (www.spines.org)
  - Collecting data in parallel with the system deployment and comparing experimental and simulation results
- Integrating and experimenting with applications (e.g. remote ultrasound)

Thanks!

www.dsn.jhu.edu/funding/aitf/