Toward Intrusion Tolerant Cloud Infrastructure

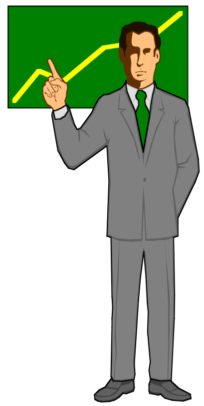
Daniel Obenshain, Tom Tantillo, Yair Amir

Department of Computer Science
Johns Hopkins University

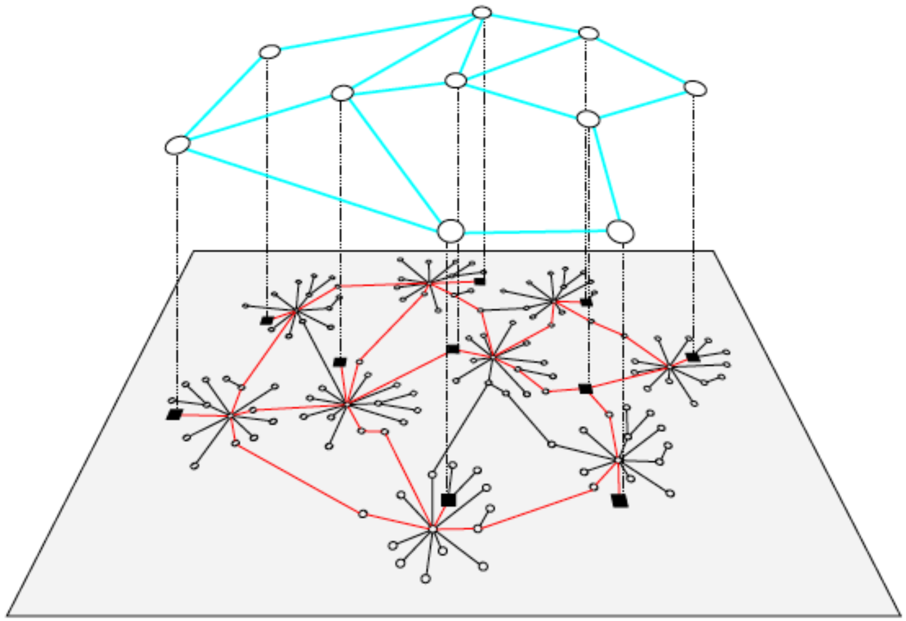
Andrew Newell, Cristina Nita-Rotaru

Department of Computer Science
Purdue University

http://www.dsn.jhu.edu



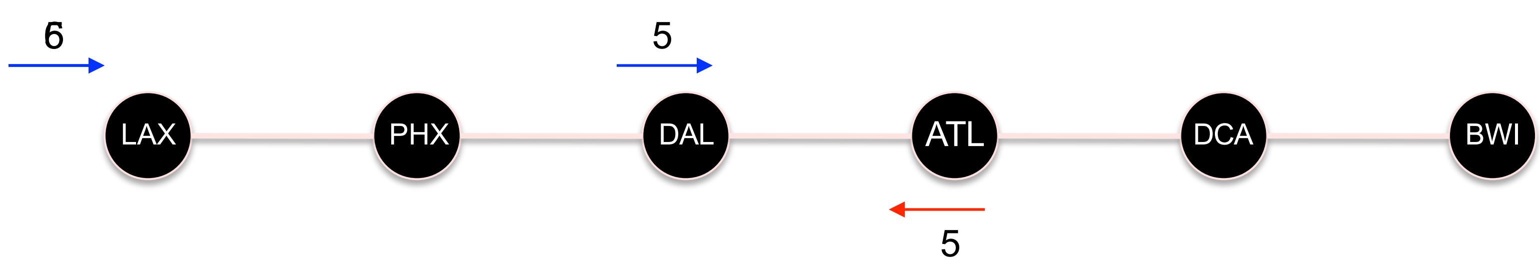
DSN03: The Overlay Paradigm



- Overlay paradigm:
  - In contrast to “keep it simple in the middle and smart at the edge”
  - Move intelligence and resources to the middle
    - Software-based overlay routers working on top of the internet
    - Overlay links translated to Internet paths
- Smaller overlay scale (# nodes) → smarter algorithms, better performance, and new services.

DSN03: Hop-by-Hop Reliability

- 50 millisecond network, five hops
  - 10 milliseconds to tell node DAL about the loss
  - 10 milliseconds to get the packet back from DAL
- Only 20 milliseconds to recover a lost packet
  - Lost packet sent twice only on link DAL – ATL
  - In contrast to at least 100 milliseconds on the Internet

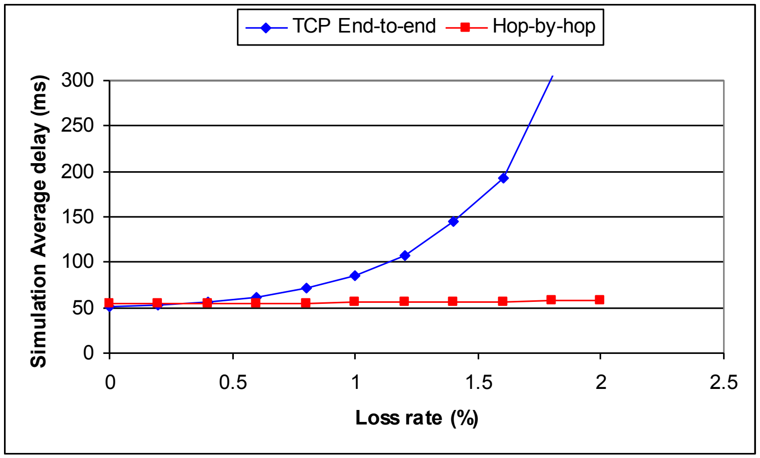


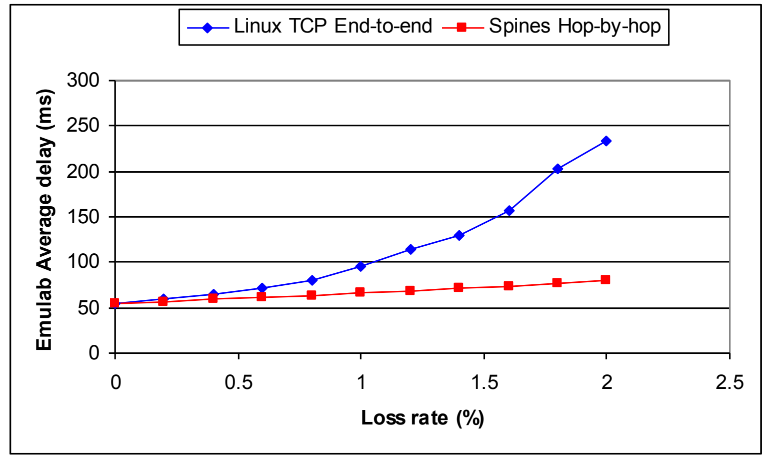
DSN03: Average Latency and Jitter

Simulation

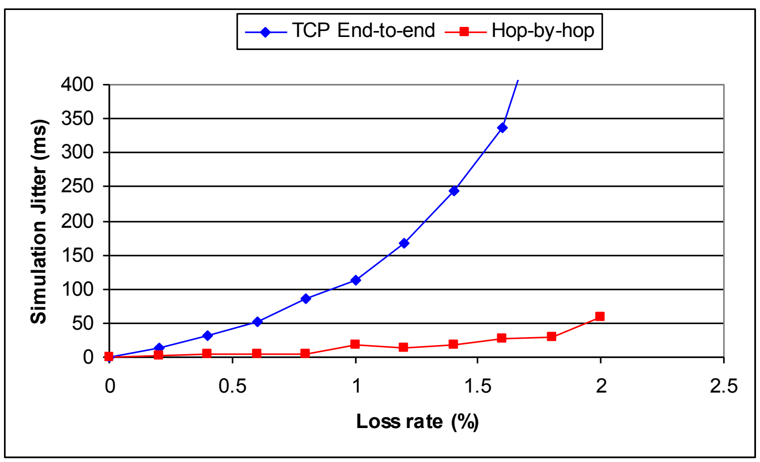
Spines on Emulab

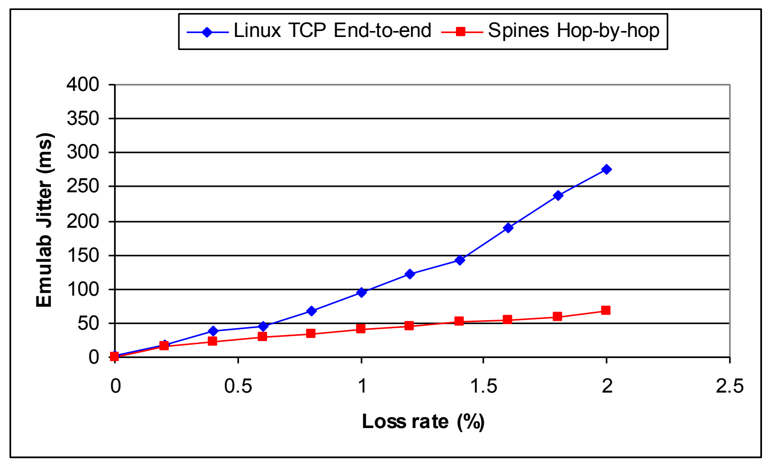
Latency





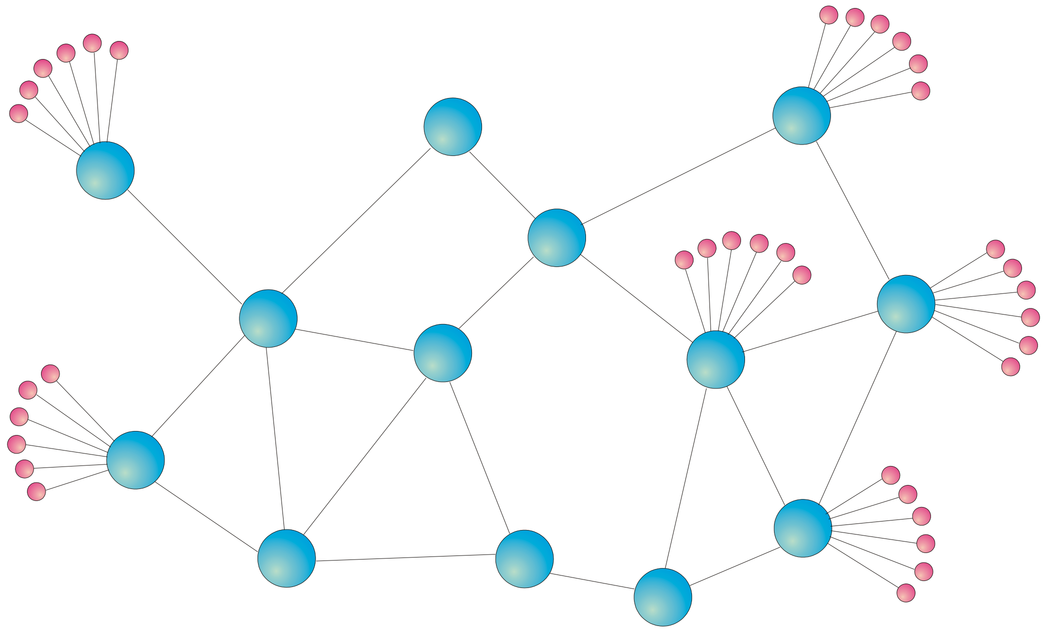
Jitter





The Spines Platform

www.spines.org

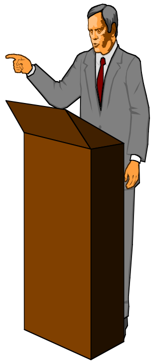


[DSN03, NOSSDAV05, TOM06, Mobisys06, TOCS10]

- Daemons create an overlay network on the fly
- Clients are identified by the IP address of their daemon and a port ID
- Clients feel they are working with UDP and TCP using their IP and port identifiers
- Protocols designed to support up to 1000 daemons (locations), each daemon can handle up to about 1000 clients

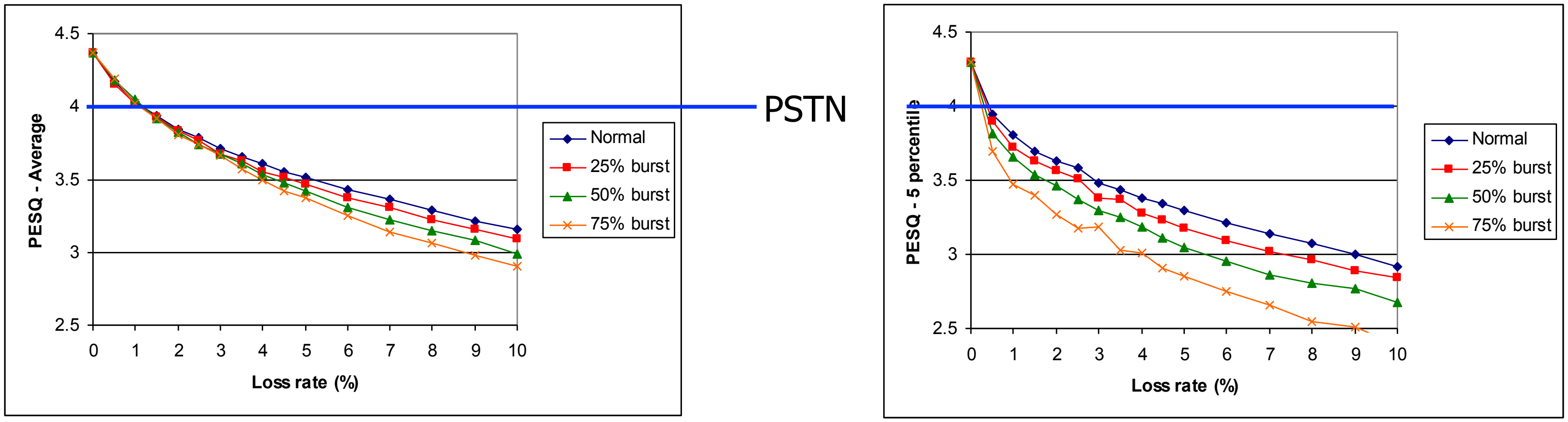
Outline

- From overlay networks to clouds
  - Going back in time – it all started with a DSN03 paper
  - From research to practice – lessons learned
- Toward intrusion-tolerant cloud infrastructure
- Intrusion-tolerant cloud monitoring and control
  - Monitoring: Priority-based flooding with source fairness
  - Control: Reliable flooding with source-destination fairness
- Intelligent use of diversity to increase resiliency
  - The Diversity Assignment Problem (DAP)
  - Optimal assignments on a cloud
  - Naïve assignments can hurt
  - Application patterns matter
- Summary



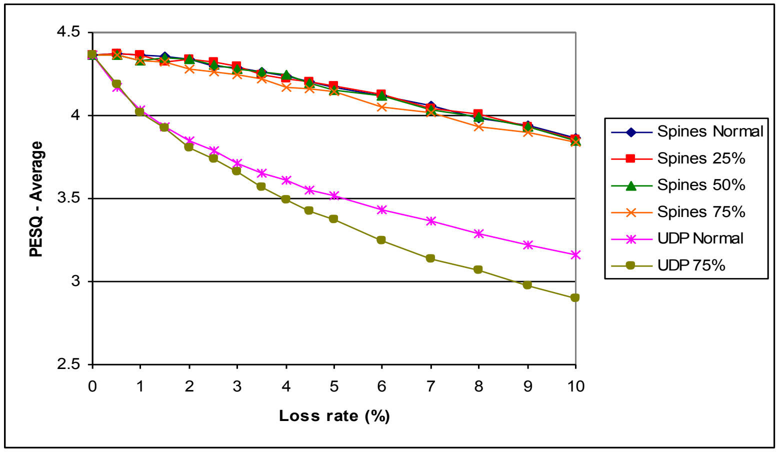
2004-2006: The Siemens VoIP Challenge

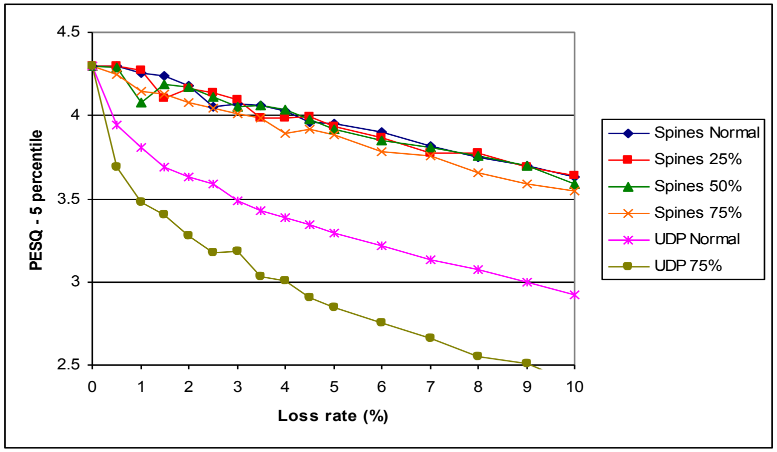
- Can we maintain a "good enough" phone call quality over the Internet?
- High quality calls demand predictable performance
  - VoIP is interactive. Humans perceive delays at 100ms
  - The best-effort service offered by the Internet was not designed to offer any quality guarantees
  - Communication subject to dynamic loss, delay, jitter, path failures



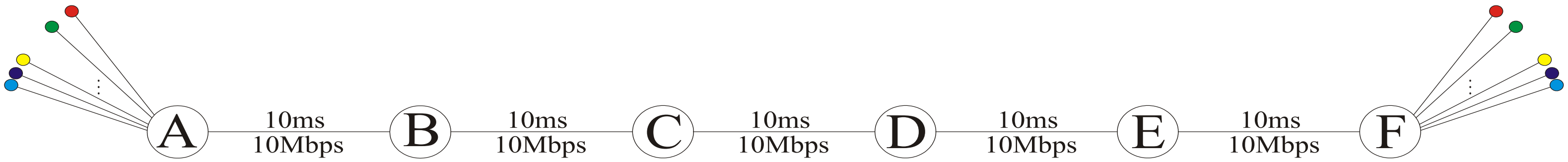
50ms network delay

VoIP Quality Improvement





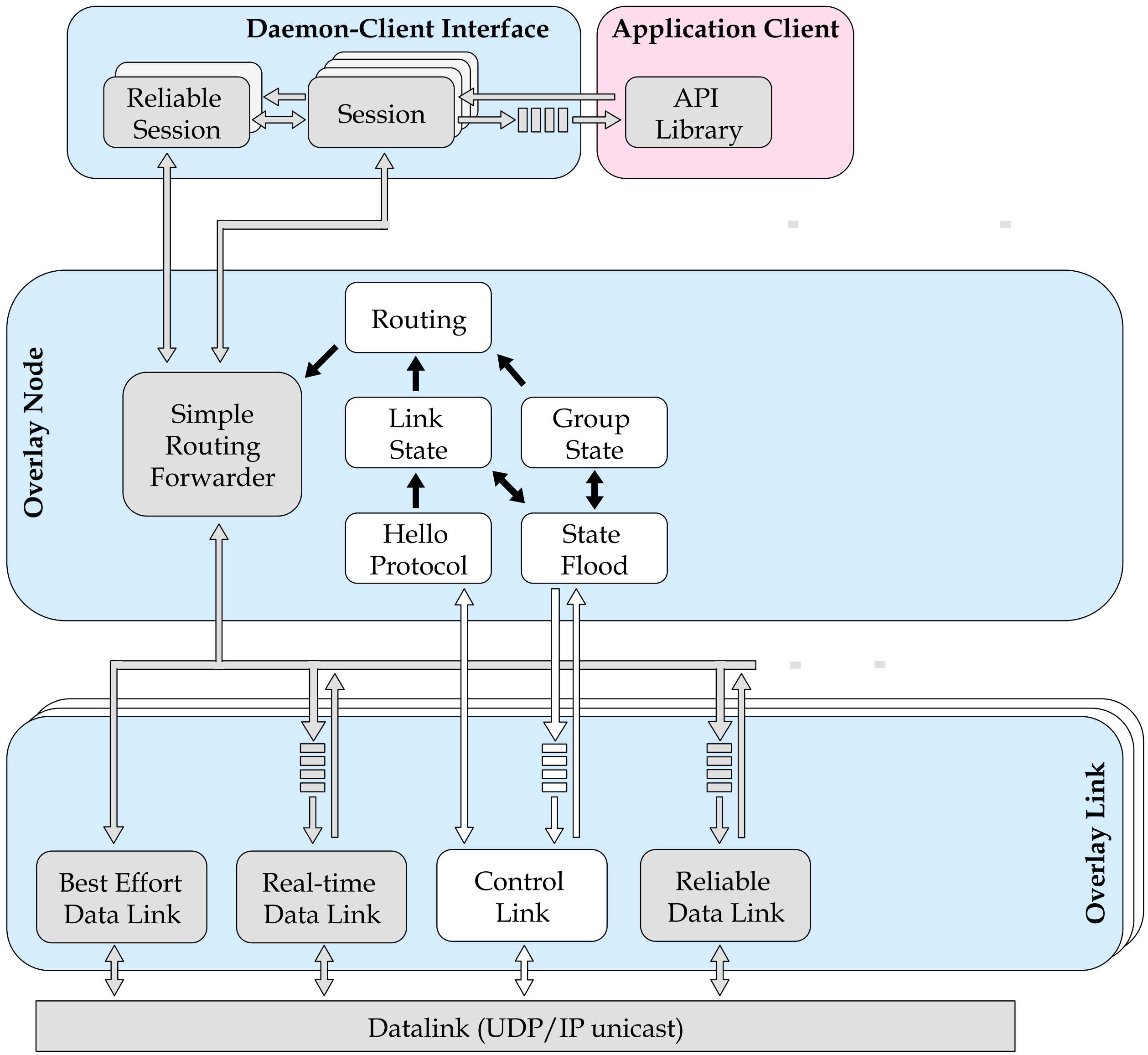
- Spines overlay – 5 links of 10ms each
- 10 VoIP streams sending in parallel
- Loss on middle link C-D



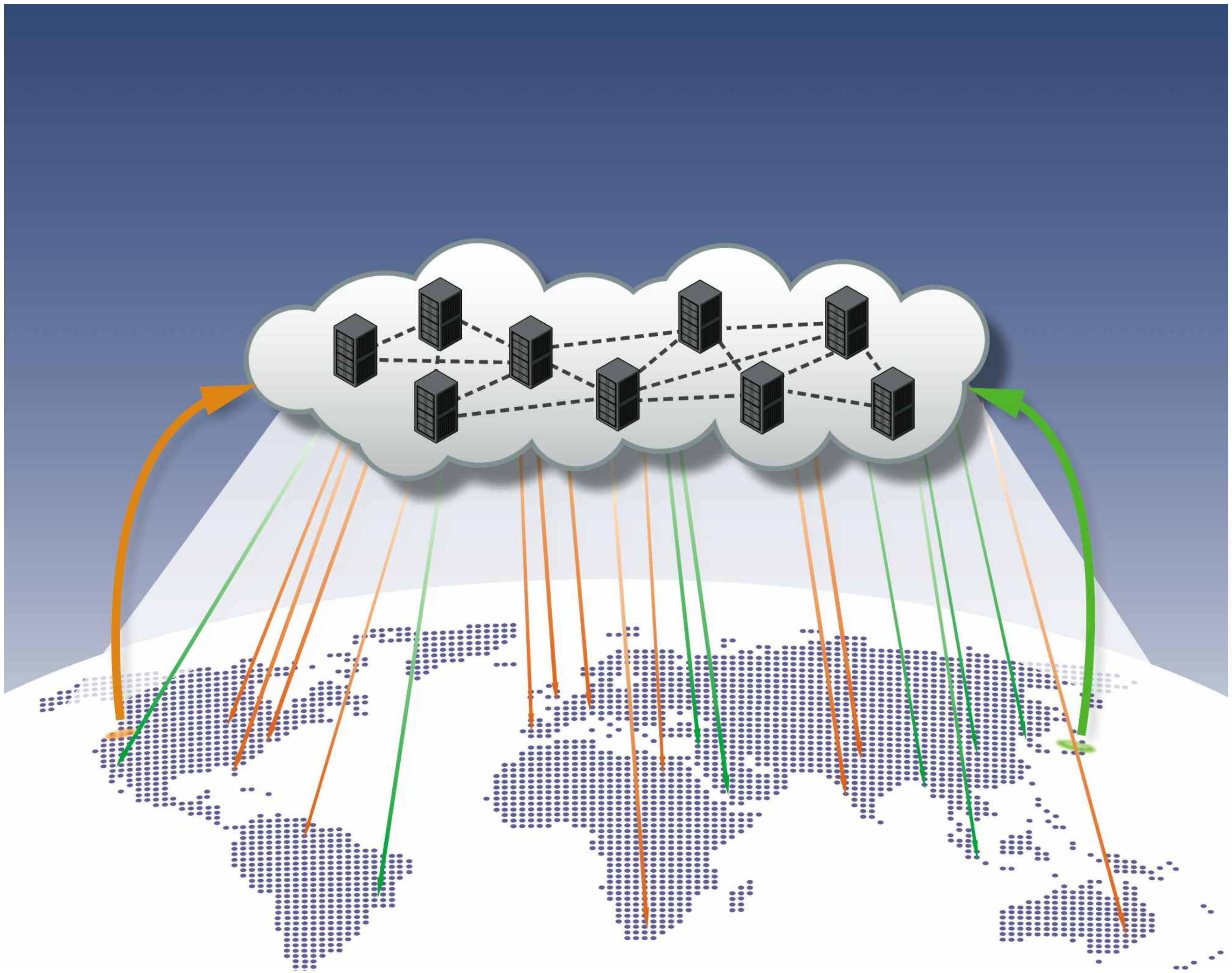
50ms network delay

The Spines Architecture (2006)

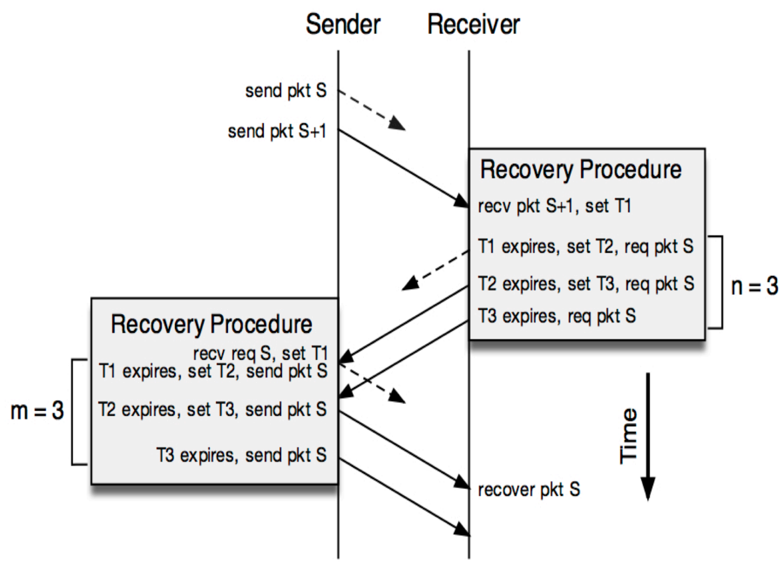
www.spines.org



2008-Present: The LiveTimeNet Cloud

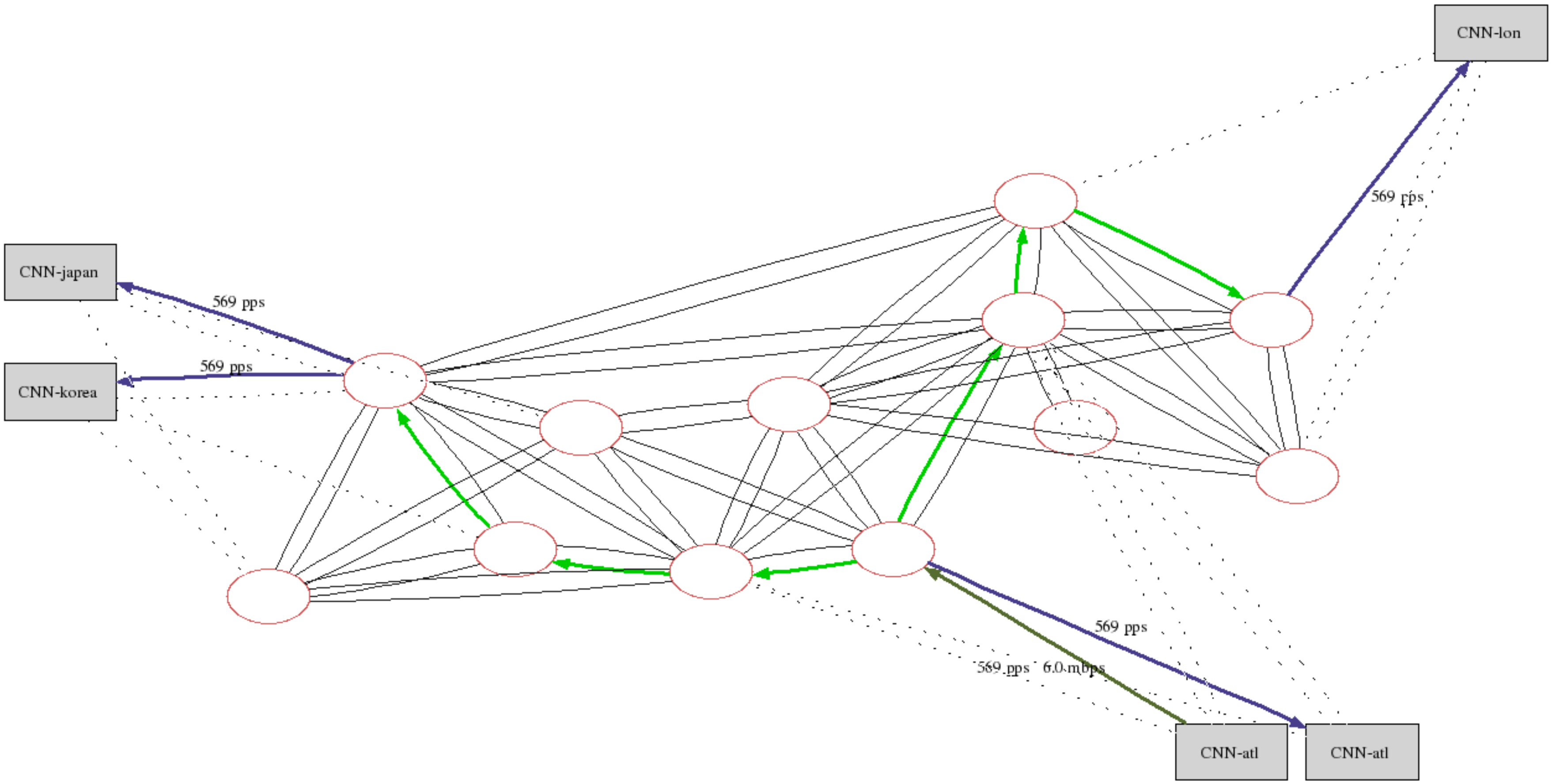


Almost Reliable, Real-time Protocol for Live TV



Network packet loss on one link (assuming 66% burstiness)	Loss experienced by flows on the LTN Network
2%	< 0.0003%
5%	< 0.003%
10%	< 0.03%

Zooming on a Single Channel



From Research to Practice

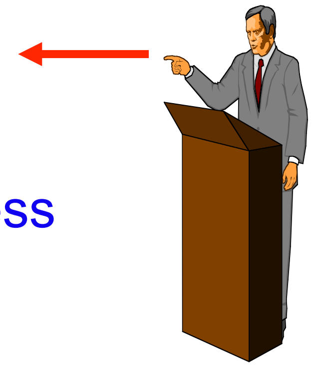
…

Toward Intrusion Tolerant Clouds

- Main premise:
  - The large gap in constructing resilient clouds is the vulnerability to intrusions
  - No good algorithms to construct distributed messaging systems and consistent global state at cloud scale, guaranteeing their integrity and performance even under intrusion attacks
- Main goal:
  - Invent, develop and transition the overlay messaging and replication tools necessary to make public and private clouds resilient to sophisticated intrusion attacks

Outline

- From overlay networks to clouds
  - Going back in time – it all started with a DSN03 paper
  - From research to practice – lessons learned
- Toward intrusion-tolerant cloud infrastructure
- Intrusion-tolerant cloud monitoring and control
  - Monitoring: Priority-based flooding with source fairness
  - Control: Reliable flooding with source-destination fairness
- Intelligent use of diversity to increase resiliency
  - The Diversity Assignment Problem (DAP)
  - Optimal assignments on a cloud
  - Naïve assignments can hurt
  - Application patterns matter
- Summary

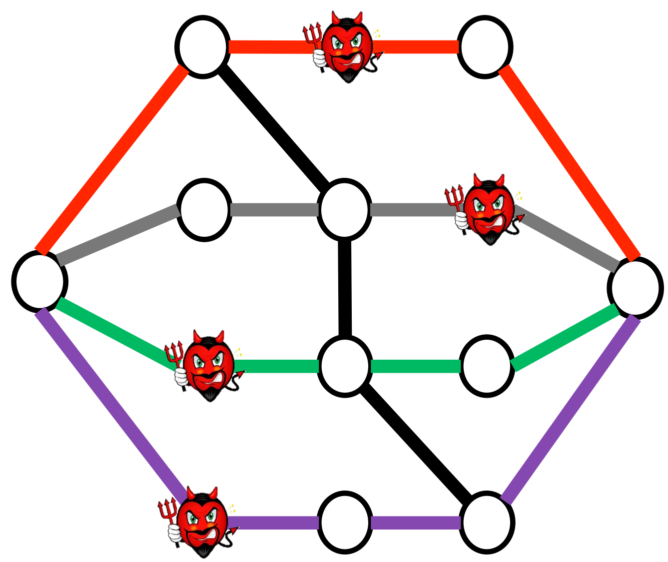


Intrusion-Tolerant Cloud Monitoring and Control

- Cloud infrastructure is remote to its administrators
  - Cloud management must be done through monitoring and control messaging
  - The chicken and the egg – at least part of the cloud software has to work to allow its administrators to make it work (e.g. react to attacks)
- Result: Monitoring and control messaging has to be intrusion-tolerant
  - Instances of the messaging service may reside on compromised nodes
- But: no practical intrusion-tolerant messaging that can perform well in cloud environments

Controlled Authenticated Flooding

- Uses overlay topology and authentication to limit the power of incorrect (compromised) nodes
- Floods messages at most twice on each overlay link
- Optimal intrusion tolerance
- Optimal latency
- High cost – up to 15-30 times higher than secure link-state overlay routing on relevant topologies
- Two protocols with differing properties and guarantees:
  - Monitoring: priority-based flooding with source fairness
  - Control: reliable flooding with source-destination fairness



Priority-based Flooding with Source Fairness

- The Need:
  - Motivated by the demands of a monitoring system in a cloud infrastructure
  - Distributing continuous streams of messages across a network that may contain compromised nodes
  - Any node in the network can be a source
  - Some messages are more critical than others
  - Delivery should be timely
  - If resources are constrained, critical information must pass

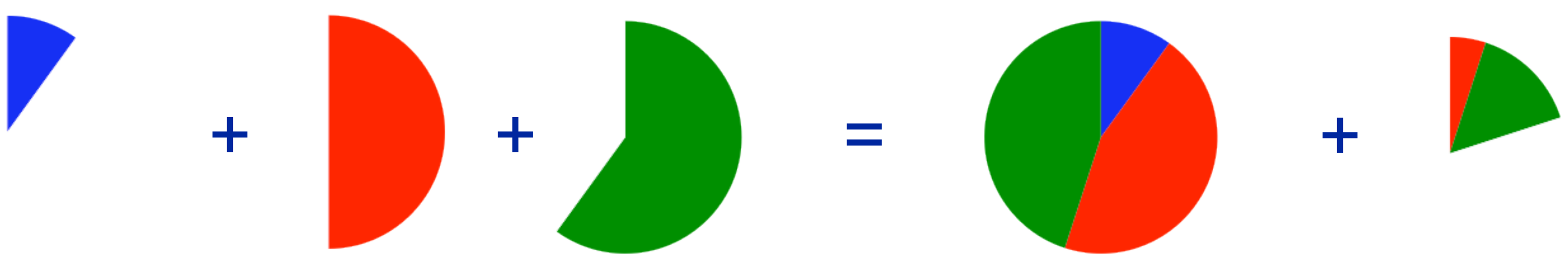
Priority-based Flooding with Source Fairness

- Main ideas:
  - Each overlay link schedules the next outgoing message by selecting a source with messages contending for this link based on a fair round-robin scheme
  - The highest priority message from the selected source is sent first
  - Memory buffer space and outgoing bandwidth is allocated on each overlay link based on a source-fair scheme
  - When memory is exhausted, the lowest priority message of the source with the most messages in the overlay link buffers is dropped first

Source Fairness for Malicious Environments Example

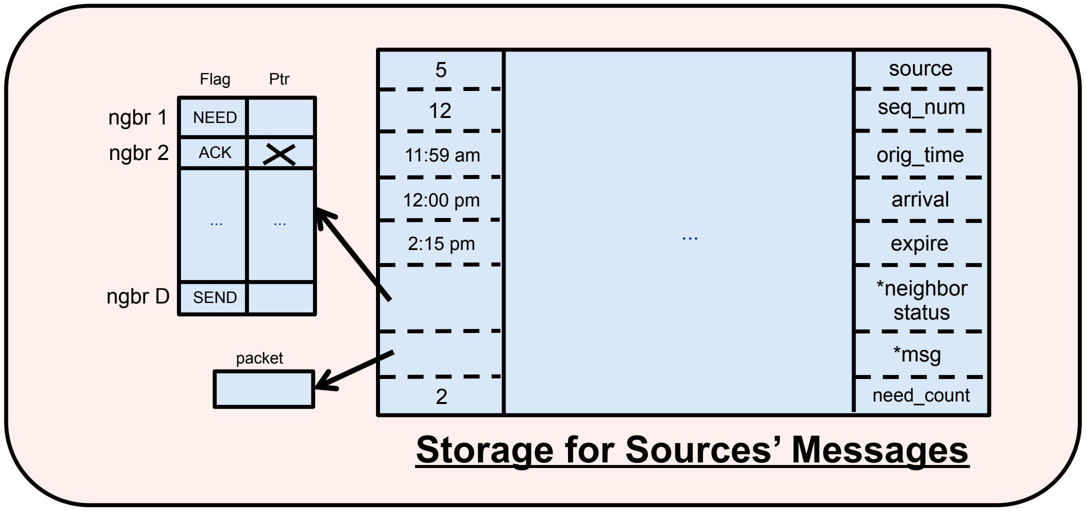
- Source A is sending at 10 Mbps, Source B at 50 Mbps, Source C at 60 Mbps, and link's capacity is 100 Mbps

- Source A gets all 10 Mbps
- Source B gets 45 out of the 50 Mbps it wants
- Source C gets 45 out of the 60 Mbps it wants

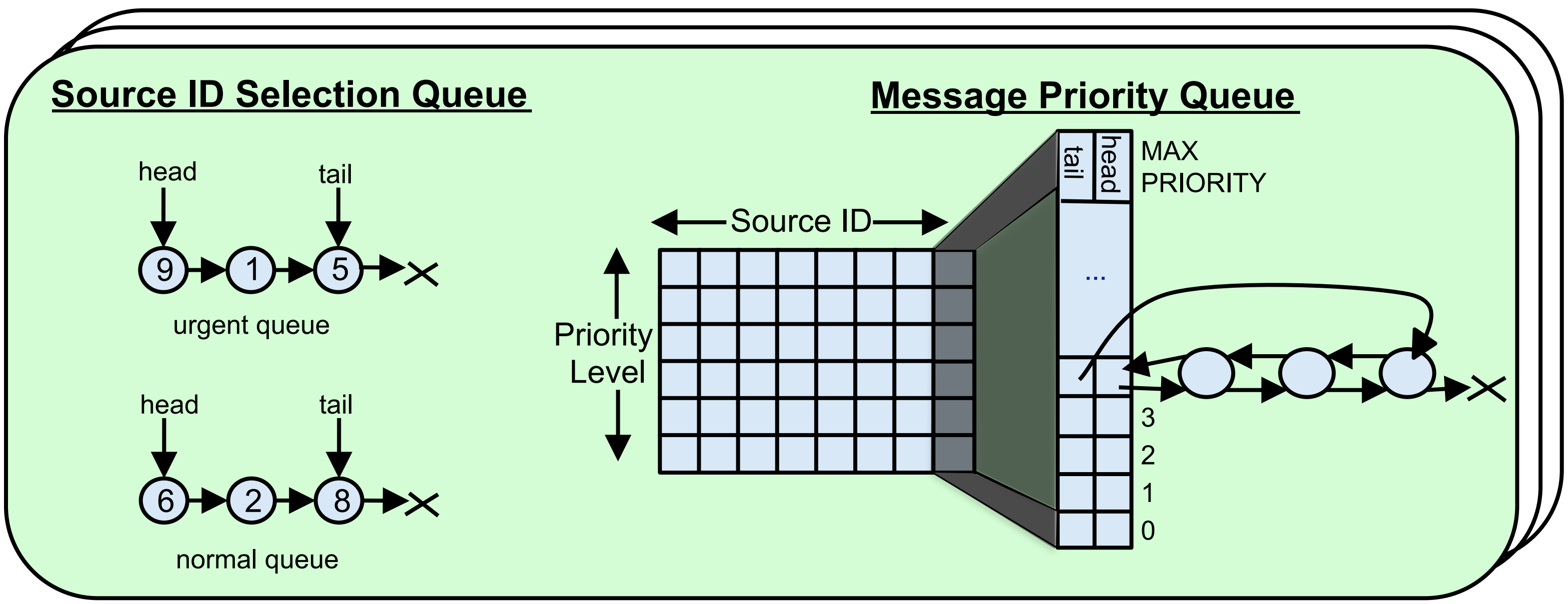


Priority-based Flooding with Source Fairness

One at each Node

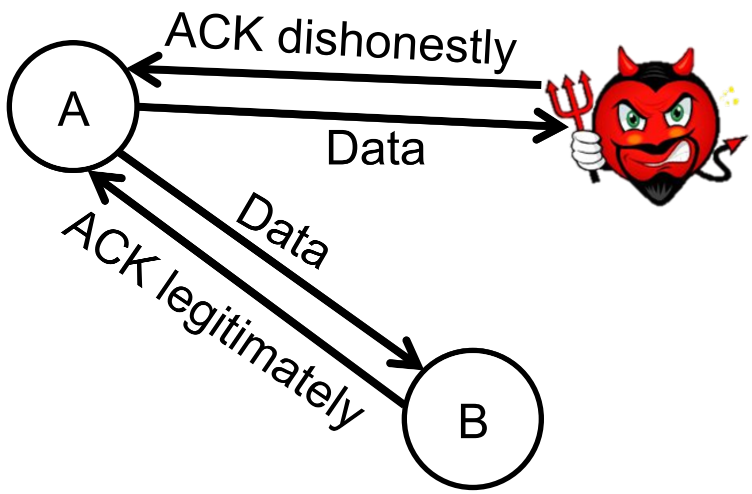


One for each Neighboring Link



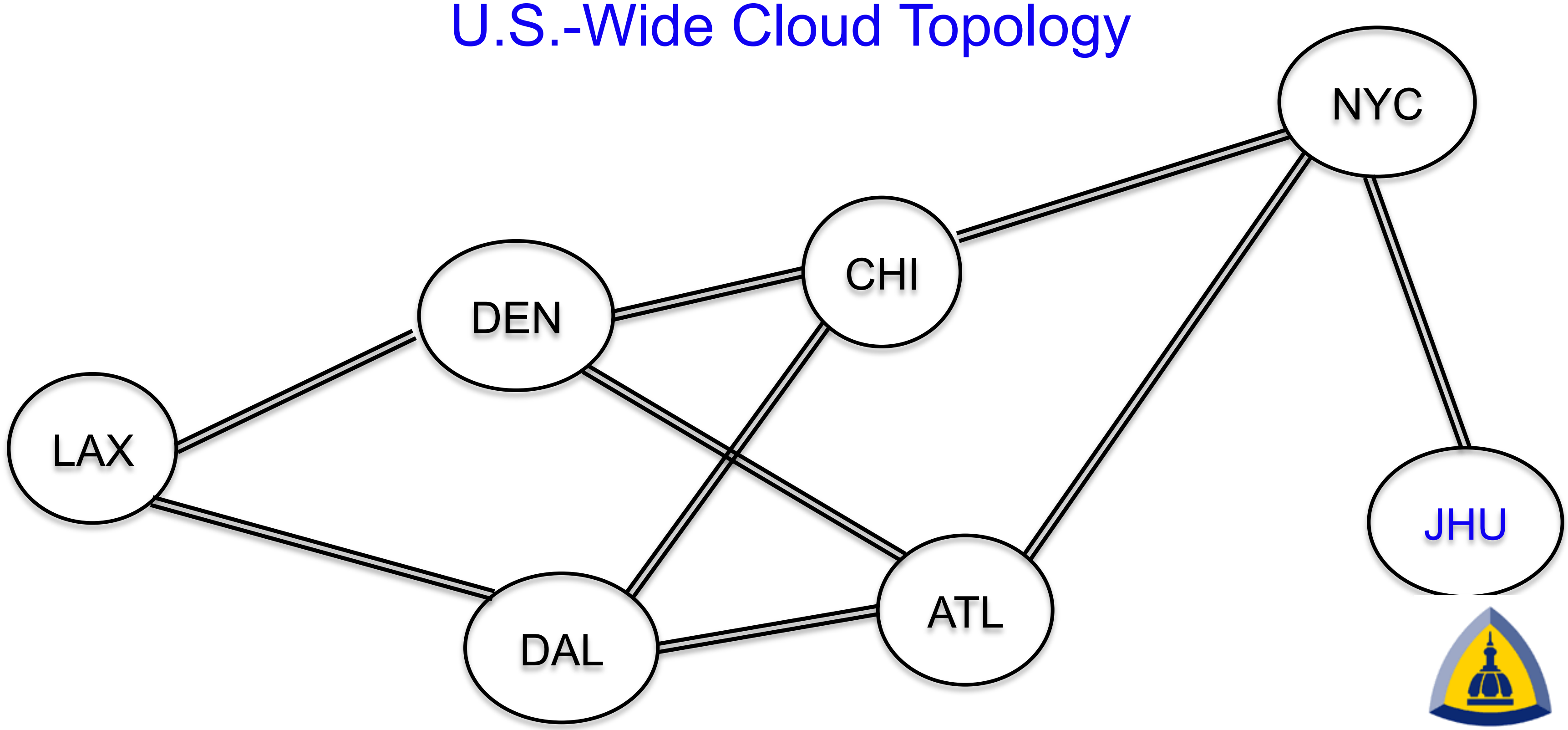
Priority-based Flooding with Source Fairness

- Intrusion-Tolerant Point-to-Point Reliable Link Protocol
  - Compromised nodes can lie to cause good nodes to "run fast"
  - Use up good system resources (processing, bandwidth)
  - Intrusion-Tolerant Link Protocol ensures nodes can't ACK packets they never received
  - We include a nonce on each packet that must be included in the corresponding ACK



Priority-based Flooding Validation on a Cloud

U.S.-Wide Cloud Topology

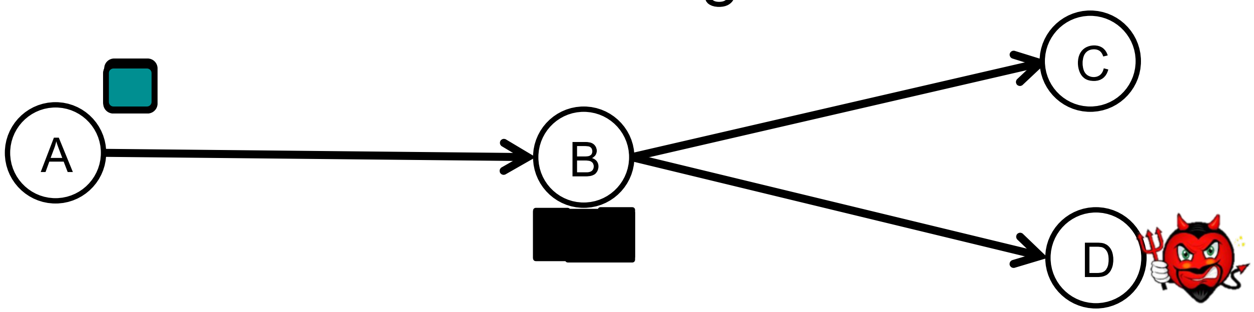


Reliable Flooding with Source-Destination Fairness

- The Need:
  - Motivated by the demands of a control system in a cloud infrastructure
  - Any node in the network can be a source
  - All messages are critical because they change the state of the cloud
  - Messages must be delivered reliably
  - Delivery should be timely

The Problem of Source-based Fairness in Reliable Communication

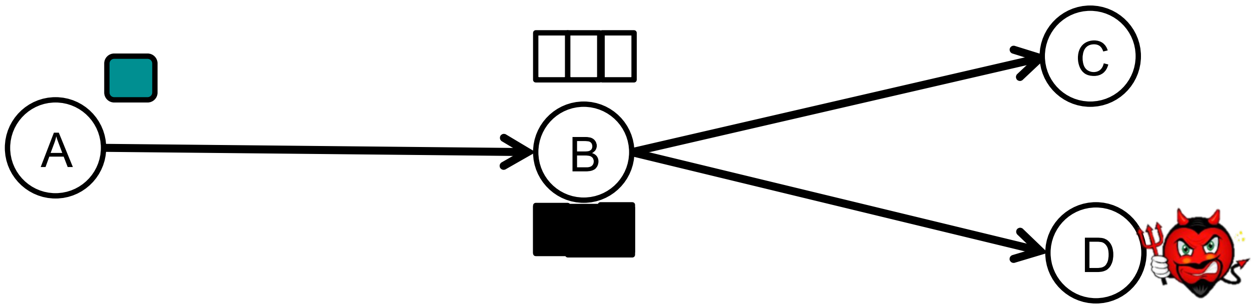
- If we used source based fairness, a malicious destination could block a good source



- A sends to C and D, via B
- D is malicious and refuses to acknowledge packets
- A cannot make progress with either C or D (because it's a reliable protocol)

Source-Destination Fairness

- Instead, treat each source-destination flow separately.



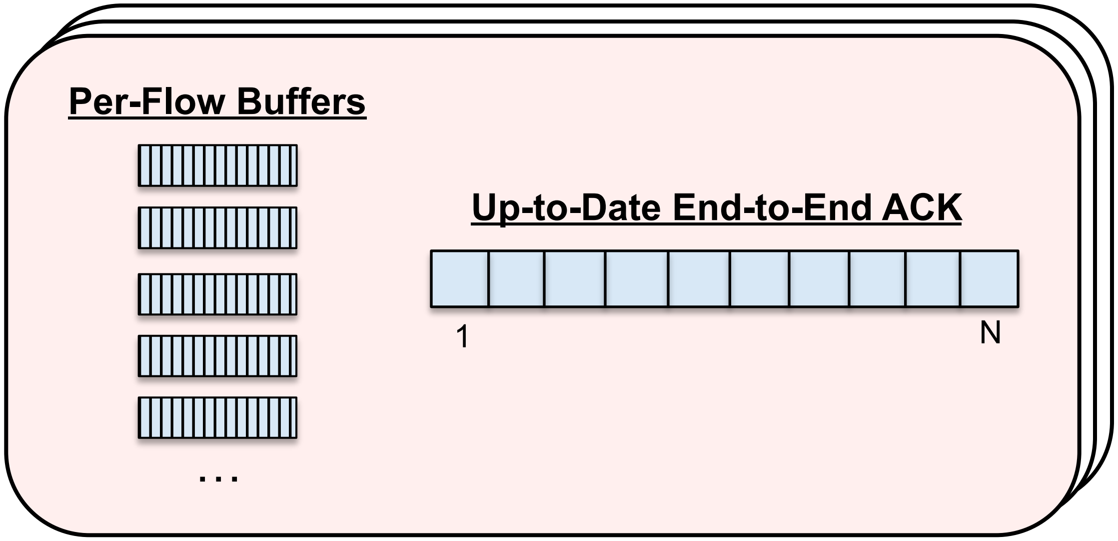
- The A-D flow becomes blocked
- The A-C flow does not

Reliable Flooding with Source-Destination Fairness

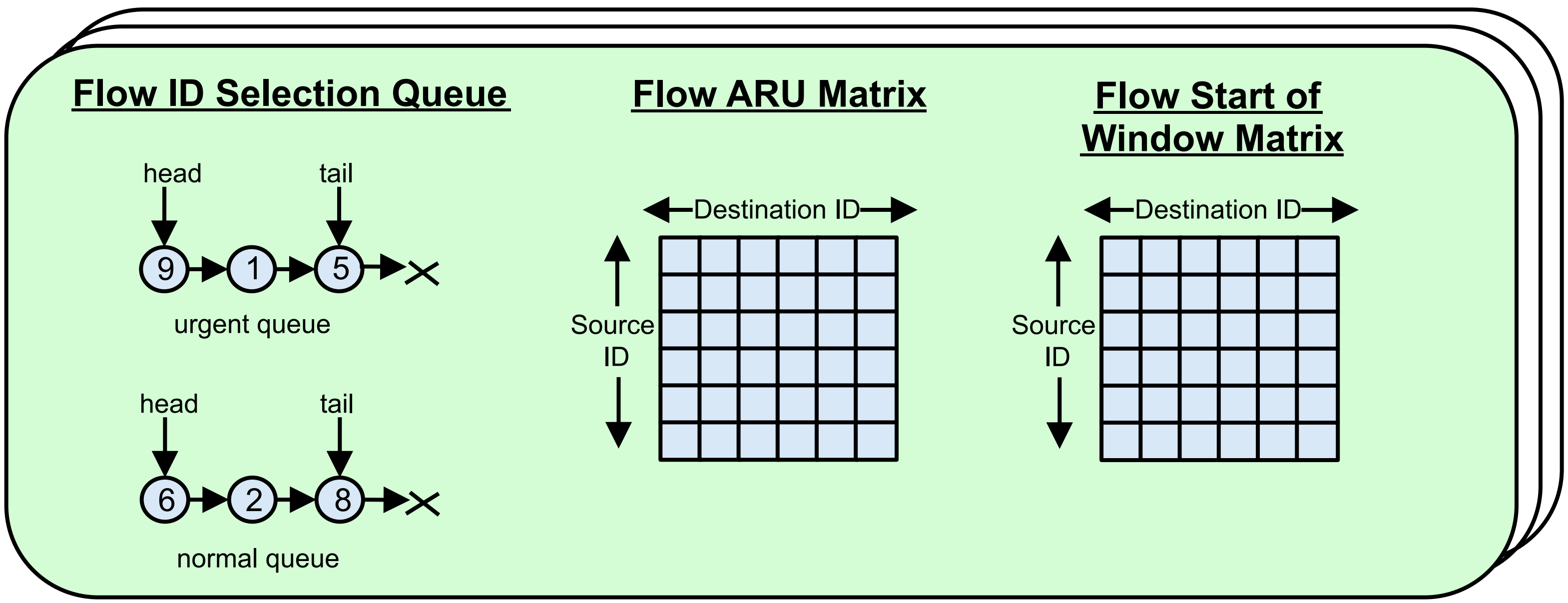
- Main ideas:
  - A correct node maintains a message until either of the following conditions is met:
    1) All of its direct neighbors have it
    2) An end-to-end acknowledgement is received
  - Memory buffer space and outgoing bandwidth are allocated on each overlay link based on a source-destination pair fair scheme, so that bad destinations cannot block other flows by not acknowledging messages
  - Back pressure is employed all the way to the source if a source-destination pair exhausts its memory buffer

Reliable Flooding with Source-Destination Fairness

One at each Node for each Destination

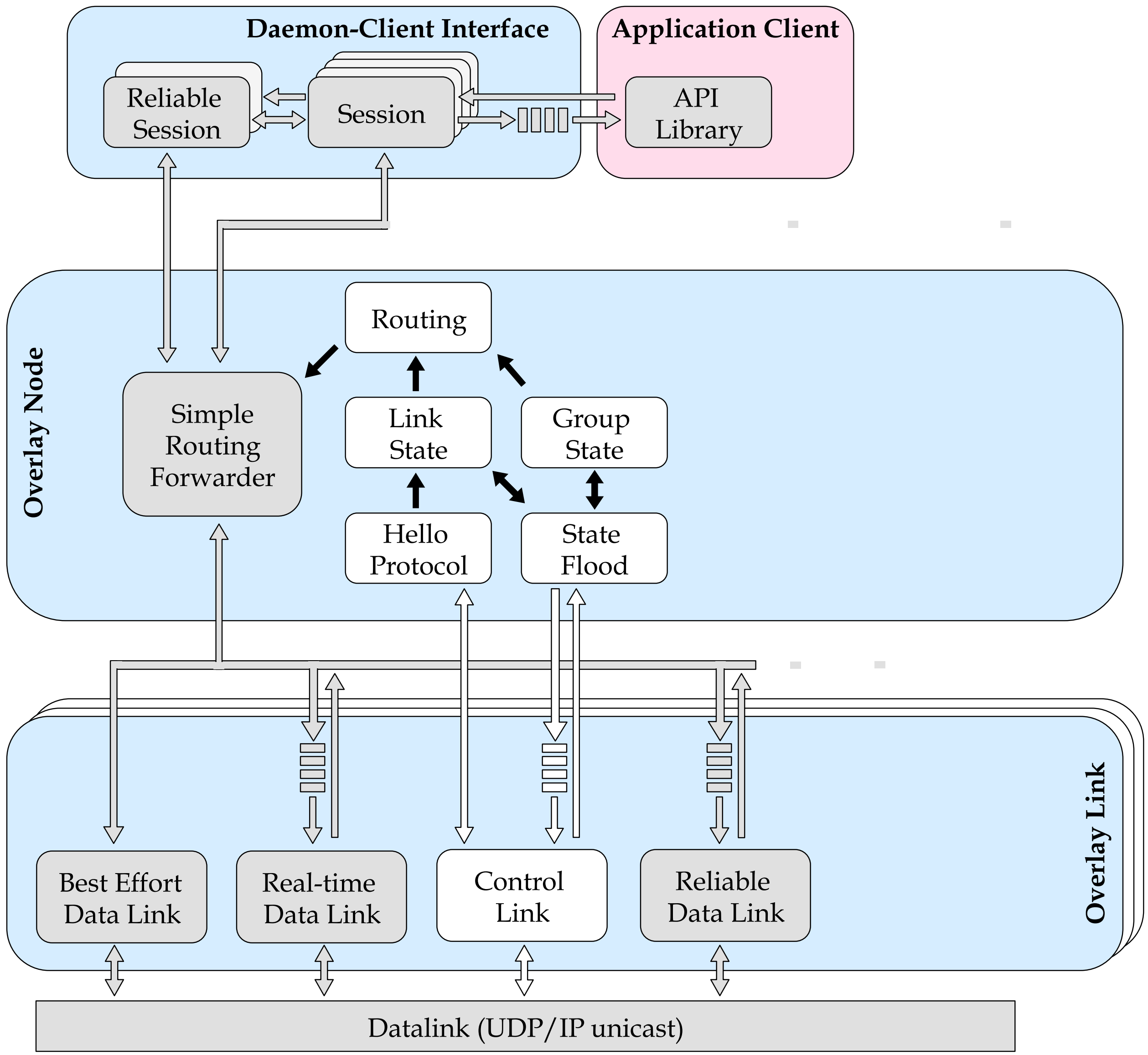


One for each Neighboring Link



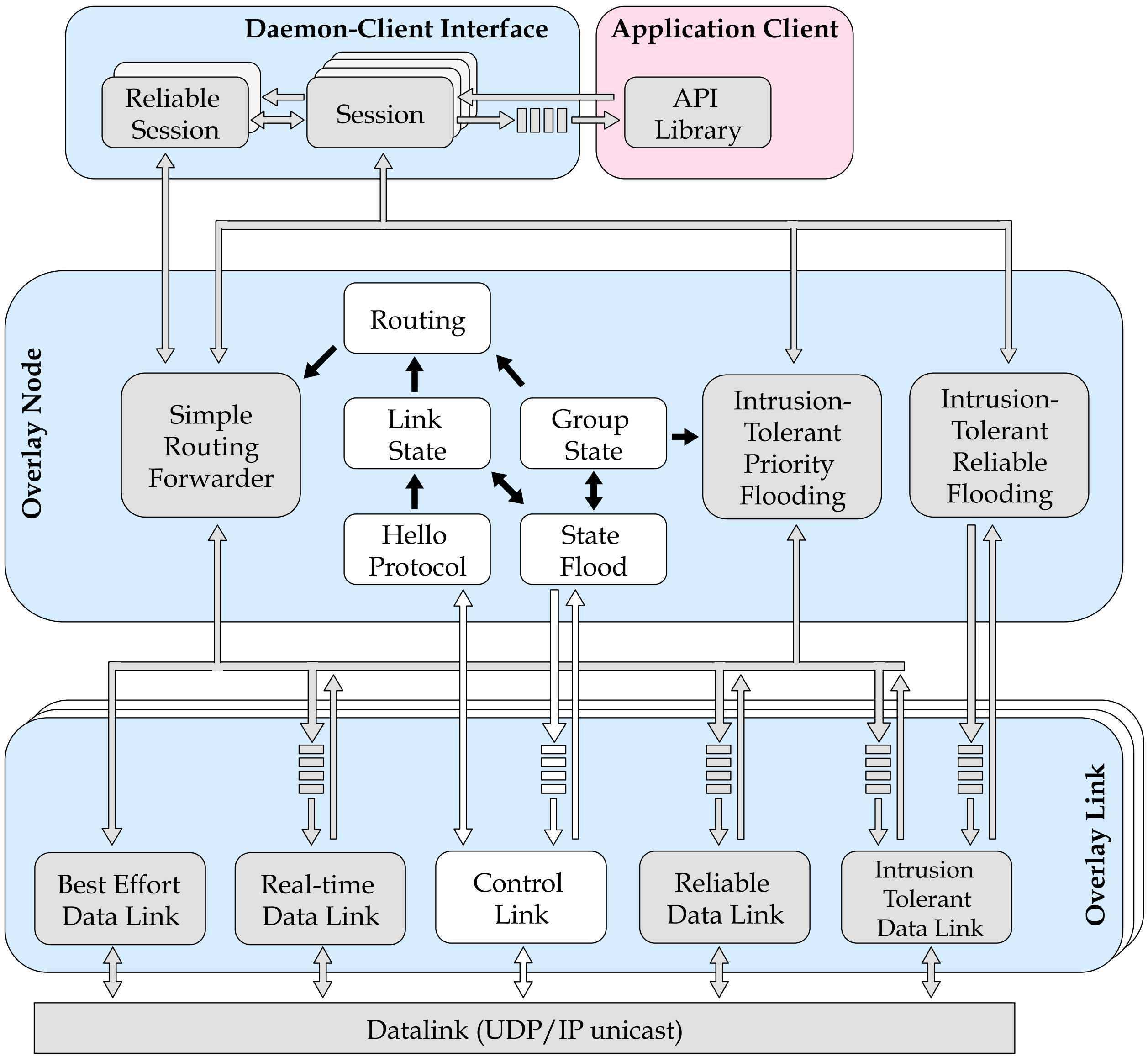
The Spines Architecture

www.spines.org



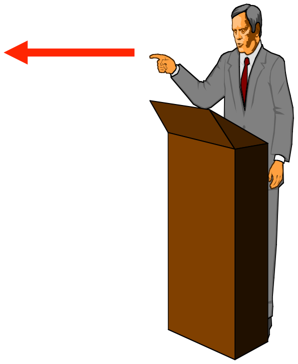
The New Spines Architecture

www.spines.org



Outline

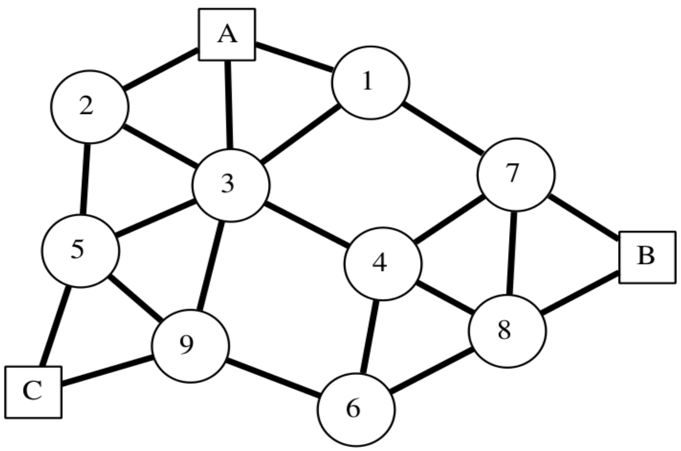
- From overlay networks to clouds
  - Going back in time – it all started with a DSN03 paper
  - From research to practice – lessons learned
- Toward intrusion-tolerant cloud infrastructure
- Intrusion-tolerant cloud monitoring and control
  - Monitoring: Priority-based flooding with source fairness
  - Control: Reliable flooding with source-destination fairness
- Intelligent use of diversity to increase resiliency



  - The Diversity Assignment Problem (DAP)
  - Optimal assignments on a cloud
  - Naïve assignments can hurt
  - Application patterns matter
- Summary

The Diversity Assignment Problem (DAP)

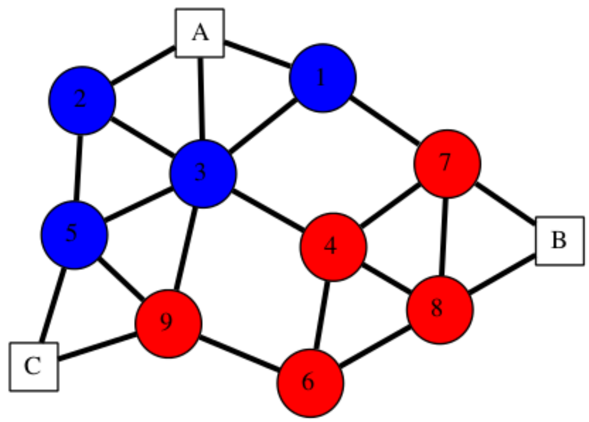
- Undirected graph of client and network nodes
- Variants are compromised independently with a certain probability over a period of time
- If a variant is compromised, all nodes of that variant are assumed compromised
- Goodness is generally measured by overall client-to-client expected connectivity
- Find the diversity assignment that maximizes goodness

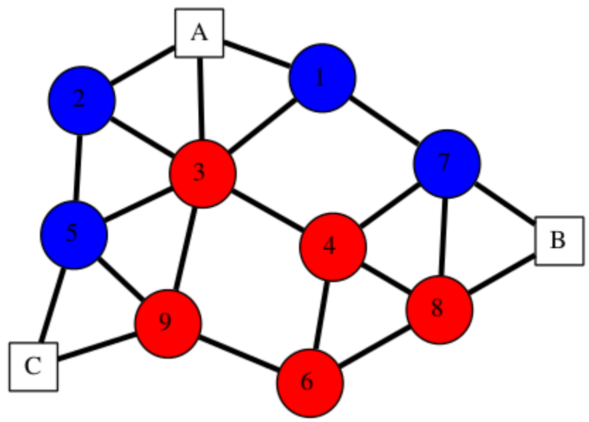


NP-Hard problem

Diversity Assignment Example

- Red: 90% resilience
- Blue: 85% resilience





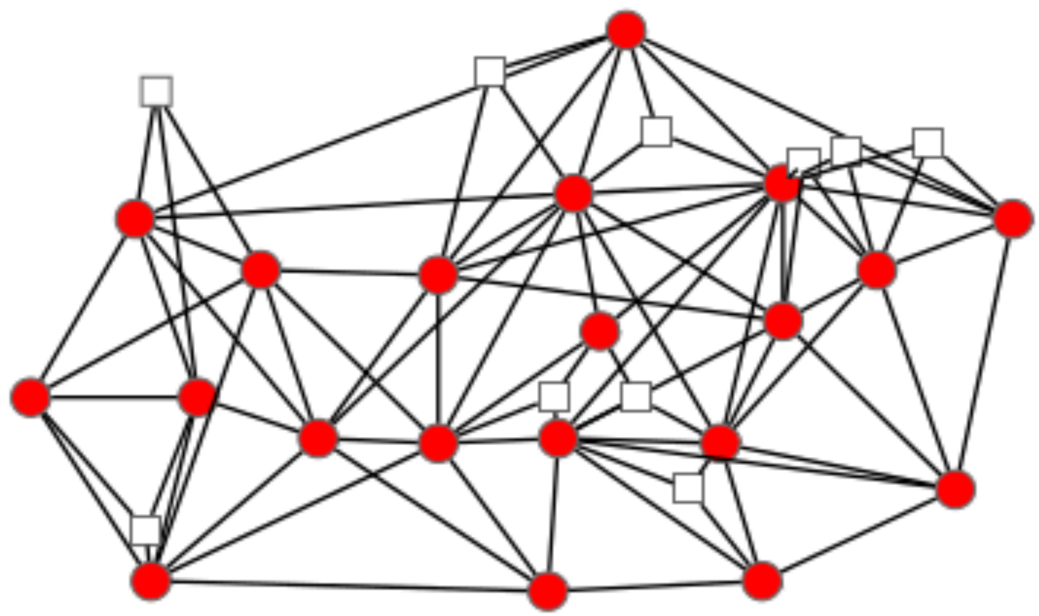
- Expected connectivity: 83.8%
- Expected connectivity: 95.7%

How one places variants in the network matters

Diversity Assignment on a Cloud Topology

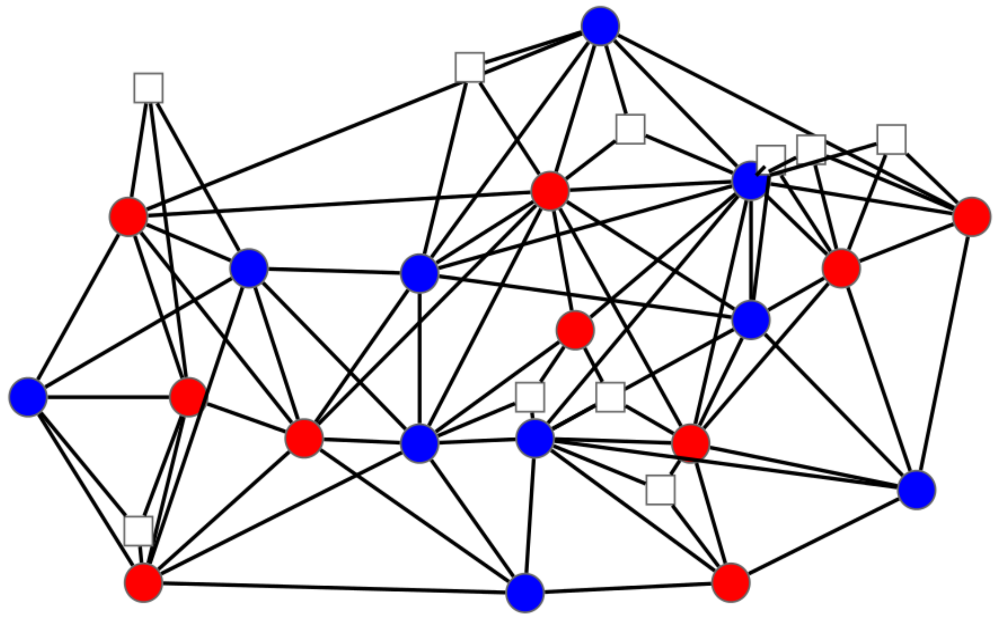
- Topology based on how the LiveTimeNet guys build their global cloud
  - 20 overlay nodes (datacenters)
  - 10 clients randomly placed
  - Up to 4 variants with differing resilient properties
    - (Red 90%) (Blue 85%) (Green 80%) (Yellow 75%)
- Model
  - Variants are compromised independently with a certain probability over a period of time
  - If a variant is compromised, all nodes of that variant are assumed compromised
  - Goodness is generally measured by overall client-to-client expected connectivity

Baseline: No Diversity



- Without diversity – just pick the most resilient variant
- Variant resilience: (Red 90%)
- Expected connectivity: 90%

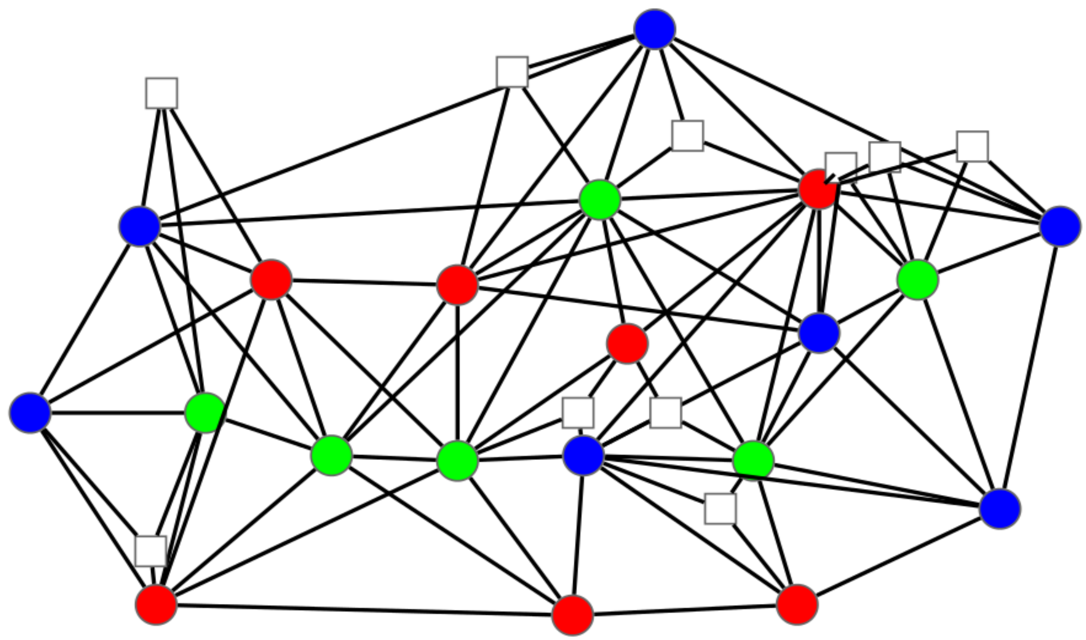
Diversity with Two Variants



- Variant resilience: (Red 90%) (Blue 85%)
- Expected connectivity: 98.5%

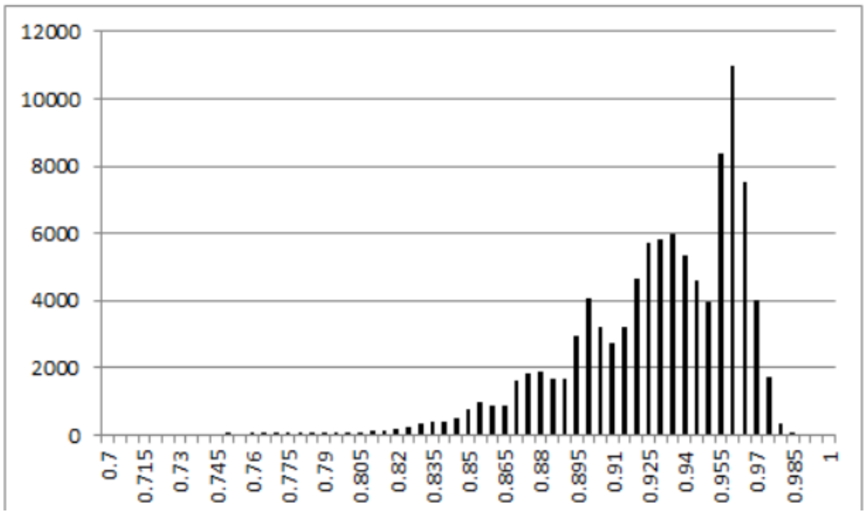
Inclusion of a more vulnerable variant results in higher resilience

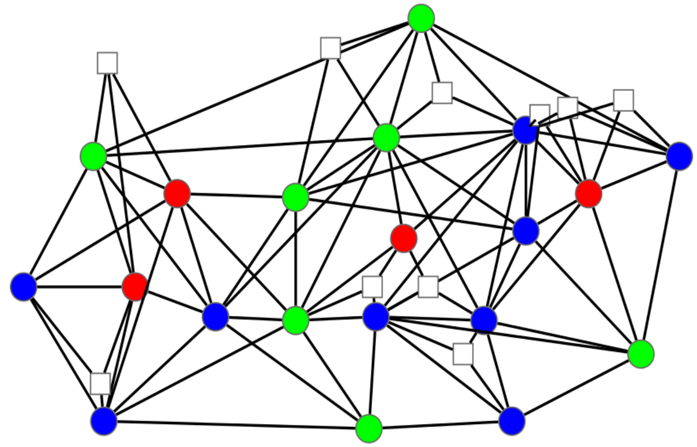
Diversity with Three Variants



- Variant resilience: (Red 90%) (Blue 85%) (Green 80%)
- Expected connectivity: 99.7%
- Even higher expected connectivity with another, even weaker variant

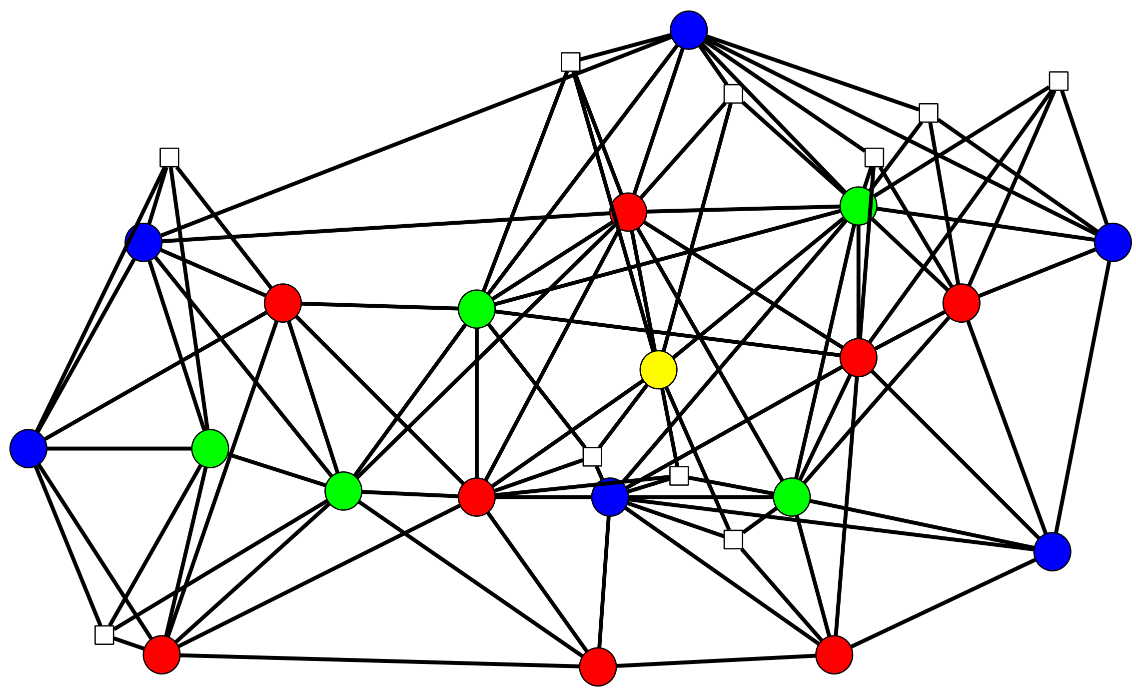
Naïve Assignments can Hurt





- 100,000 random assignments with 3 variants
  - Many random assignments have less than 90% expected connectivity (worse than without diversity)
  - Best value: 98.8%
  - *Expected dis-connectivity* = (1 – expected connectivity)
  - Best random is 4x worse in terms of expected dis-connectivity:
    - 1.2% vs 0.3%

Diversity with Four Variants



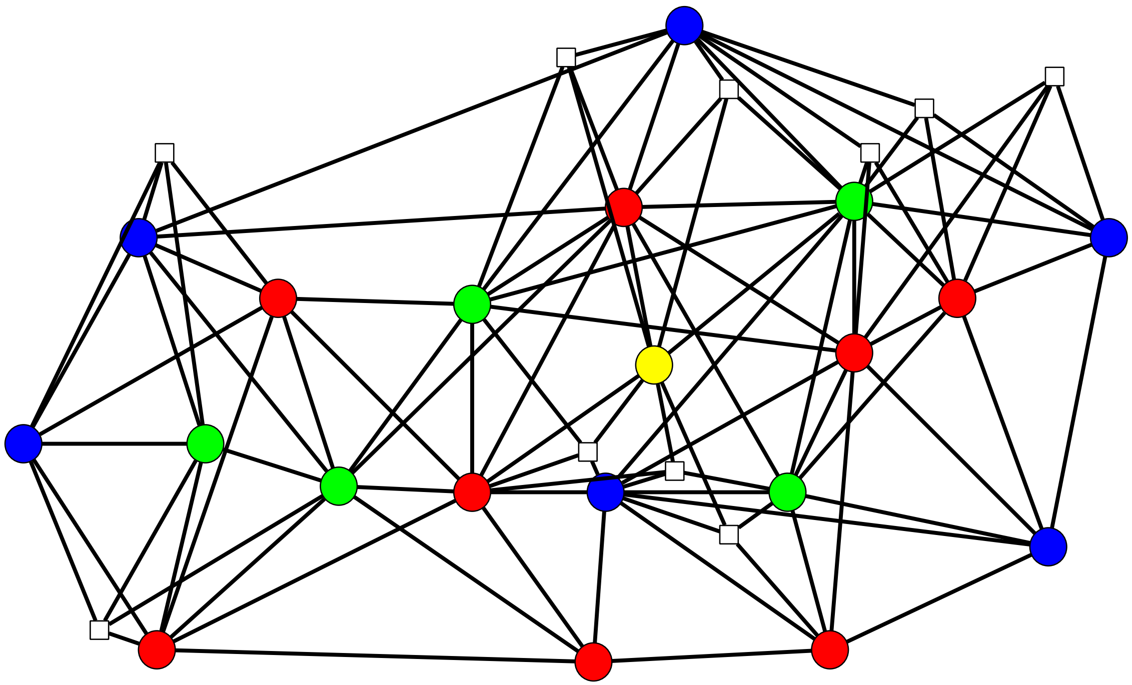
- Resilience: (Red 90%) (Blue 85%) (Green 80%) (Yellow 75%)
- Expected client connectivity: 99.75%

Application-Specific Diversity

A BFT Test Case

- Run BFT on top of this diversified network
  - The state machine replicas are the clients from the point of view of the network
  - Rather than maximize any-to-any connectivity, we maximize the chance that BFT can make progress
  - BFT needs a connected component of at least 2f+1 good replicas out of a total of 3f+1 replicas
- Maximize the probability that a connected component of correct replicas exists

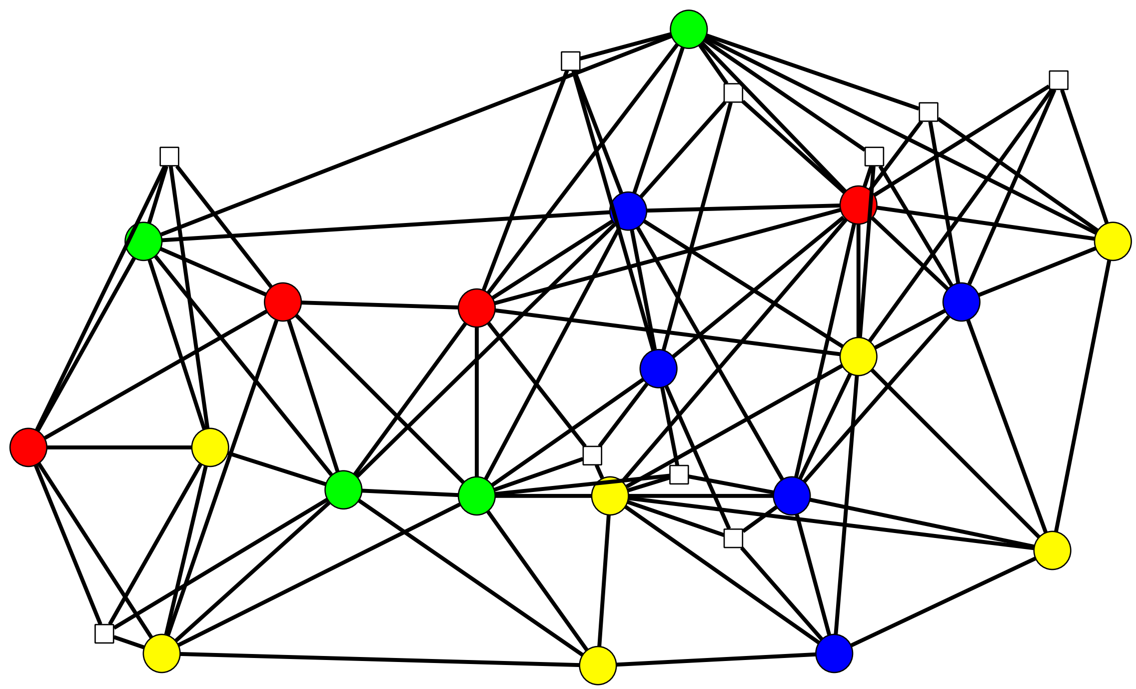
Diversity with Four Variants



Fault model:
1 Byzantine replica
2 partitioned replicas

- Resilience: (Red 90%) (Blue 85%) (Green 80%) (Yellow 75%)
- Expected client connectivity: 99.75%
- Probability of BFT progress: 99.7%

BFT-Specific Diversity



Fault model:
1 Byzantine replica
2 partitioned replicas

- Resilience: (Red 90%) (Blue 85%) (Green 80%) (Yellow 75%)
- Expected client connectivity: 98.06%
- Probability of BFT progress: 99.925%

Outline

- From overlay networks to clouds
  - Going back in time – it all started with a DSN03 paper
  - From research to practice – lessons learned
- Toward intrusion-tolerant cloud infrastructure
- Intrusion-tolerant cloud monitoring and control
  - Monitoring: Priority-based flooding with source fairness
  - Control: Reliable flooding with source-destination fairness
- Intelligent use of diversity to increase resiliency
  - The Diversity Assignment Problem (DAP)
  - Optimal assignments on a cloud
  - Naïve assignments can hurt
  - Application patterns matter
- Summary