Distributed Systems 600.417

Intrusion-Tolerant Networks

Department of Computer Science
The Johns Hopkins University

First, Some Context…

- You've just heard about Intrusion-Tolerant State Machine Replication (e.g. Prime)
- So, now we know how to build systems that continue to work correctly, even if some of the replicas are compromised
- We can use diversity and proactive recovery to help the system survive for a long time
- But, those replicas still need to communicate!

Protecting Network Communication

- The Internet is becoming increasingly important to our society
  - Critical infrastructure, global clouds, financial systems, government, …
- People have been trying to prevent attacks for years
  - Firewalls, Intrusion Detection and Prevention Systems
- Security standards in different layers
  - IPsec, TLS/SSL, and others protect communication
  - BGPsec, DNSsec – These contain some good ideas, but aren't widely accepted (yet)
- But, none of these address the vulnerability to intrusions
  - Malicious attacks are becoming more prevalent and sophisticated
  - Therefore: constructing networks that are resilient to the point of intrusion tolerance is crucial – networks that work even if part of them is compromised – under the control of a sophisticated adversary

IP Networks Are Vulnerable

- IP networks are **efficient**, but based on trust
  - Internet routing is susceptible to routing attacks (BGP hijacking)
  - Compromises in the network can completely disrupt communication
- IP networks are **scalable**, but fragile
  - Single IP networks are susceptible to failures, attacks, and misconfigurations
  - Sophisticated DDoS attacks (Crossfire) can severely degrade QoS of targeted Internet flows

Intrusion-Tolerant Networks Goals

- Support critical infrastructure (power grid, clouds)
  - Requires strong data delivery semantics
    - Guaranteed Timeliness vs. Guaranteed Reliability
- Performance guarantees under attack
- Always available
  - No downtime incurred when detecting/finding intrusions
  - No hiccups when adversary launches an attack
  - No startup costs or high delay
- Optimal intrusion tolerance
- Willing to pay for these properties (for some important messages)

Intrusion-Tolerant Networks (more details)

- Any node can be a source
- Any node can be compromised
- Compromised nodes may be undetectable
  - Cannot prefer one node's traffic over another's
  - Risk of favoring compromised nodes and starving correct sources' traffic
- Different applications need different messaging semantics (e.g. timely vs. reliable)
- Requires cryptographic mechanisms for authentication and integrity

Intrusion-Tolerant Network Approaches

- On-Demand Secure Byzantine Routing
- Authenticated Adversarial Routing
- Network Layer Protocols with Byzantine Robustness (Perlman)
- SCION
- SCION/SIBRA
- Practical Intrusion-Tolerant Networks (Spines)

Intrusion-Tolerant Network Approaches

- On-Demand Secure Byzantine Routing
- Authenticated Adversarial Routing
- Network Layer Protocols with Byzantine Robustness (Perlman)
- SCION
- SCION/SIBRA
- Practical Intrusion-Tolerant Networks (Spines)

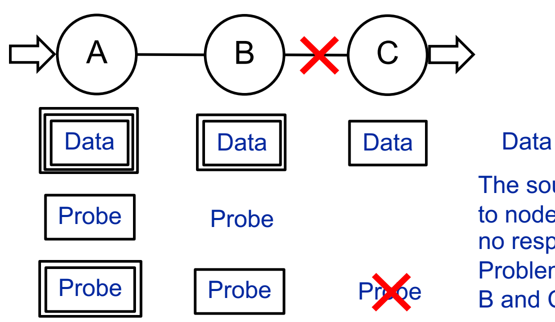
On-Demand Secure Byzantine Routing
(AHNR2002, ACHN+2008)

- Discovers potential paths by flooding a ping-type message across the network
- Uses source-based routing to specify that path on the data messages
- Uses layers of encryption to obfuscate messages
- If there is a problem, can probe along the path to find the problematic link, remove it, and try again
- Eventually, all bad links are removed and messages are sent along the shortest remaining path (optimal)

On-Demand Secure Byzantine Routing

The source sends data, which is unwrapped at each hop.

If data doesn't make it, the source probes to node B, gets a response.



The source probes to node C, gets no response. Problem is between B and C!

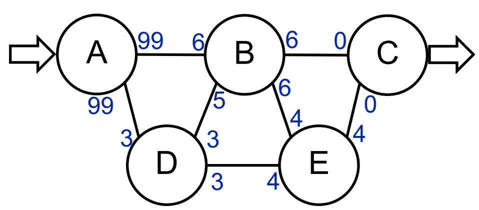
- Probing takes time, during which you may not get any messages
- An adversary can choose when you will experience this downtime

Intrusion-Tolerant Network Approaches

- On-Demand Secure Byzantine Routing
- Authenticated Adversarial Routing
- Network Layer Protocols with Byzantine Robustness (Perlman)
- SCION
- SCION/SIBRA
- Practical Intrusion-Tolerant Networks (Spines)

The Slide Protocol (building block)

(AGR1992)



- Also called gravitational flow
- Source “pumps” in messages, destination is a “sink”
- Messages flow across the network (like water), moving from high-pressure to low-pressure nodes
- On each link, a process sends on a link if the other side of that link has fewer messages (lower pressure)
- Once enough messages have been sent, some must arrive at the destination

Authenticated Adversarial Routing
(ABO2009)

- Uses the Slide protocol as a building block
- Adds cryptography
  - For every message sent, need a signed receipt
- If enough messages have been pumped in, but no messages arrive at destination, there is a problem
  - Stop system temporarily
  - Audit to detect bad node, tracking receipts for every message in the network
- Eventually optimal (one in, one out)
- Requires n^3 messages to start up! Auditing takes n^4!

Intrusion-Tolerant Network Approaches

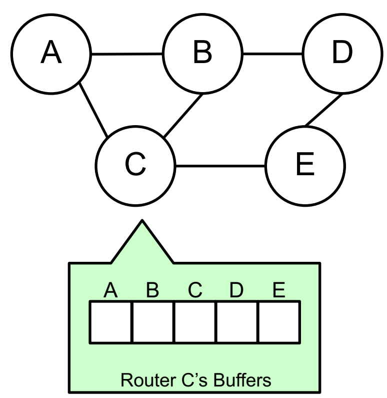
- On-Demand Secure Byzantine Routing
- Authenticated Adversarial Routing
- Network Layer Protocols with Byzantine Robustness (Perlman)
- SCION
- SCION/SIBRA
- Practical Intrusion-Tolerant Networks (Spines)

Network Layer Protocols with Byzantine Robustness

- Radia Perlman's Ph.D. Thesis – MIT 1989
- One of the first works to consider how to route packets in the presence of Byzantine faults
- Goal: disseminate link-state routing updates in a network with potentially compromised routers
  - Addresses Byzantine forwarding nodes
  - First to address Byzantine source nodes
- Requires changes to the network infrastructure

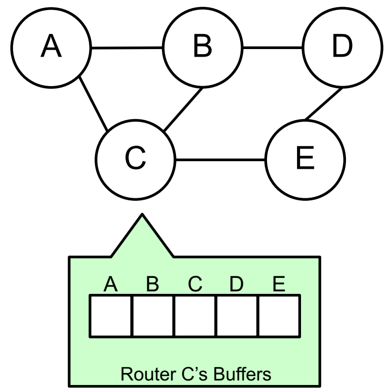
Network Layer Protocols with Byzantine Robustness

- All messages are signed and verified using public-key cryptography
  - Routers cannot impersonate other routers
- Routers maintain space for the most recent message from each router
- Messages are flooded across the network in round-robin fashion
  - Optimal resiliency for delivery
  - Network fairness
- Overtaken-by-event semantics
  - Data freshness



Network Layer Protocols with Byzantine Robustness

- Meant for routing updates, not data
- No way to provide data delivery semantics needed by applications
  - Reliable delivery only works if routers wait "long-enough" for messages to reach the destination before issuing the next message
  - Applications do not always want their most recent messages to be preferred
- Pre-allocated memory and bandwidth
  - Protects against Byzantine faults, but…
  - No router gets more than $\frac{1}{n}$ of the bandwidth on each link
  - We want better (optimal) network utilization
- Not practical - requires changes to network Infrastructure (IP)



Intrusion-Tolerant Network Approaches

- On-Demand Secure Byzantine Routing
- Authenticated Adversarial Routing
- Network Layer Protocols with Byzantine Robustness (Perlman)
- SCION
- SCION/SIBRA
- Practical Intrusion-Tolerant Networks (Spines)

SCION

(ZHHC+2011)

- Clean-slate Internet architecture aiming to secure and protect Internet routing
  - Organize Autonomous Systems (ASes) into Isolation Domains (ISDs) based on policies (e.g., geographic boundaries)
  - Setup ISDs in hierarchical tree, with few trusted core ASes at the root that are common to all path selections (routing)
  - Source/destination jointly setup several end-to-end paths through the tree that only communicate along secure ISDs
- Requires coordination and cooperation of ISPs and ASes at the IP level, creating practical barriers to deployment
  - Incremental deployment is possible – can connect SCION-enabled ISPs with IP tunnels (similar to the MBone)
- Vulnerable to resource consumption attacks
  - Compromised end hosts and compromised ASes

SCION/SIBRA

(BRSP+2016)

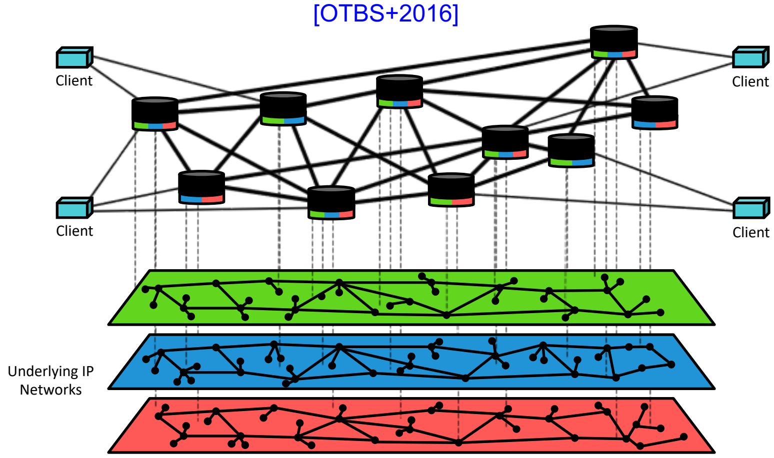
- Recent extension to SCION
- Designed to defeat resource consumption attacks
  - Contractual resource reservation scheme based on AS policies
  - Neighboring ASes establish bandwidth contracts between them, reserving bandwidth for long-term and short-term flows
  - Flows are continuously monitored, and flows violating their contracts are detected, reported, and throttled
- Scalable and efficient - almost no overhead imposed on routers for data plane traffic
- Significant practical barriers to deployment
  - ISPs require direct connections to setup and enforce contracts
  - Unlike SCION, incremental deployment is not feasible - need a contiguous end-to-end path of SIBRA-enabled ISPs

Intrusion-Tolerant Network Approaches

- On-Demand Secure Byzantine Routing
- Authenticated Adversarial Routing
- Network Layer Protocols with Byzantine Robustness (Perlman)
- SCION
- SCION/SIBRA
- Practical Intrusion-Tolerant Networks (Spines)

Overlay Approach: Resilient Network Architecture

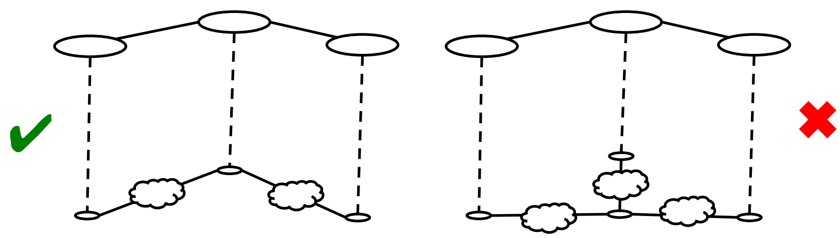
[OTBS+2016]



- Leverage existing IP network infrastructure
  - Sits on multiple IP networks
- Provide necessary resiliency and timeliness for intrusion tolerance
  - Programmability in the middle of the network

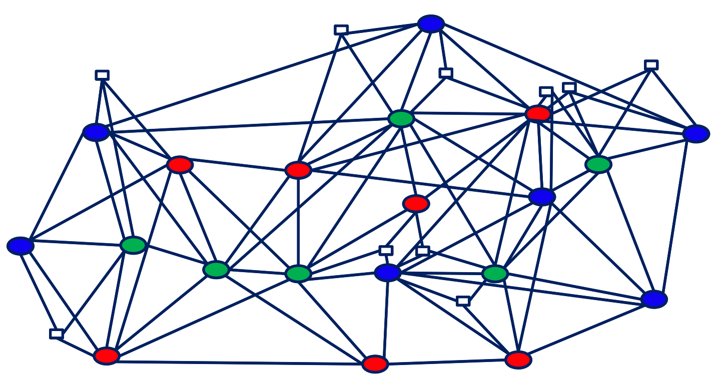
Resilient Overlay Construction

- Resiliency at the overlay level via redundancy
- Place overlay nodes in well-provisioned data centers
- Carefully create overlay edges between overlay nodes
  - Leverage available ISP backbone maps
  - Connect overlay nodes with predictable Internet routing between them to ensure high likelihood of disjoint overlay topology



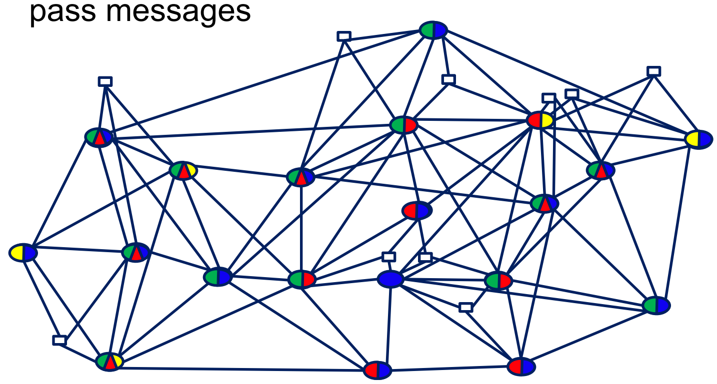
Diverse Network Providers

- With only one ISP under the overlay, a major problem can bring down the entire overlay
- Assigning diverse ISP variants is more resilient

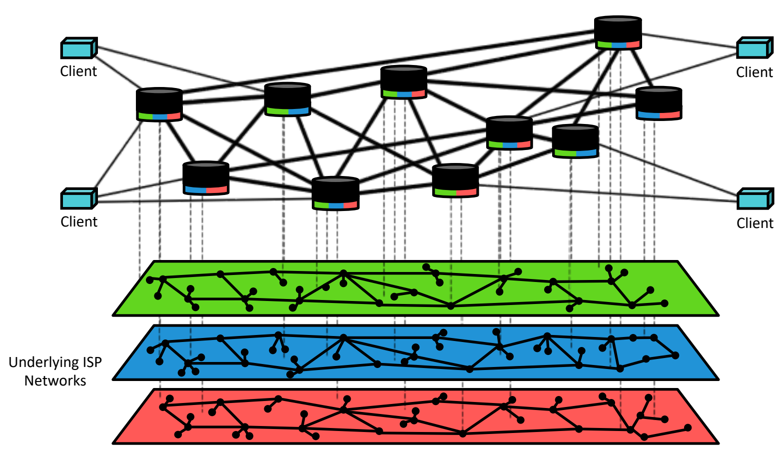


Multihoming

- Simultaneously get service from multiple ISPs at each overlay node
  - Overlay link is correct if at least one pair of ISPs can pass messages



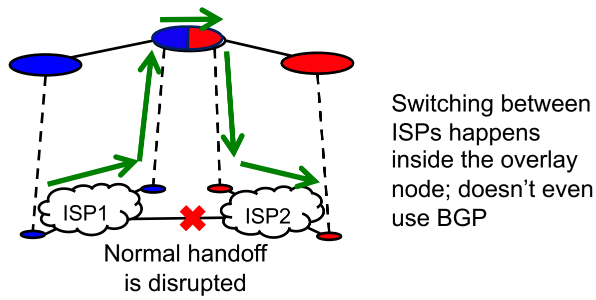
Resilient Network Architecture in Practice



- Place overlay nodes in well-provisioned data centers
- Multihoming at each overlay node
- Survive anything short of simultaneous meltdown of multiple underlying ISP backbones!

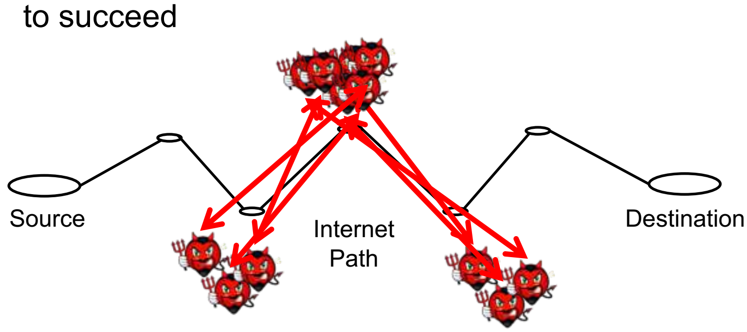
Attack Resilience: BGP Hijacking

- Malicious advertisements cause BGP to reroute
  - BGP Hijacking has occurred in the wild
- Overcome by Resilient Architecture
  - Traffic that is "on net" will be unaffected



Attack Resilience: Crossfire DDoS Attack

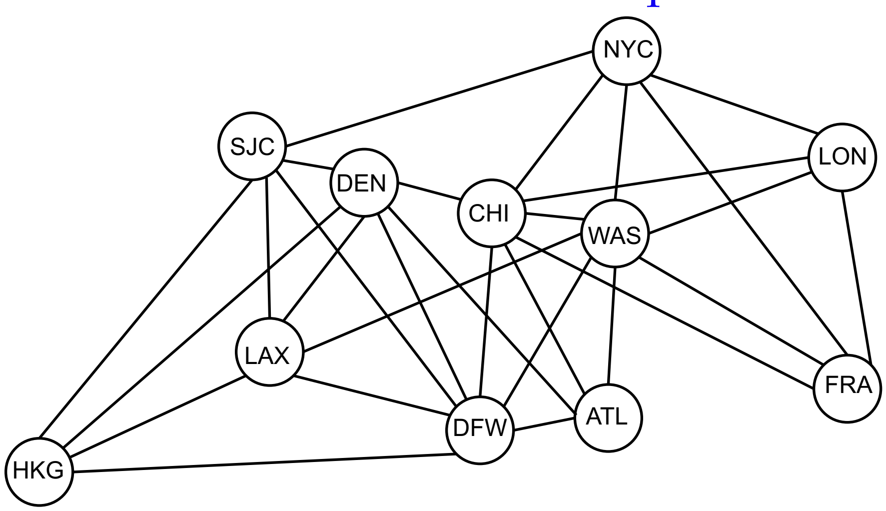
- Advanced, persistent resource-consumption attack in the underlying physical network
- Overcome by Resilient Architecture
  - Attack must affect many links on many different ISPs to succeed



Overlay is Susceptible to Compromises

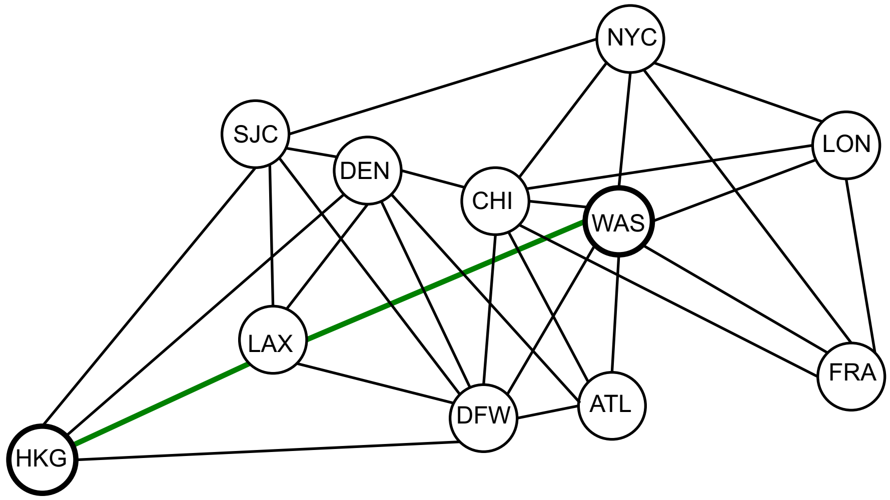
- Resilient Networking Architecture overcomes any attack or compromise in the underlying IP network infrastructure
- But, the overlay itself (just like all networks) is still susceptible to compromises

A Live Network Graph



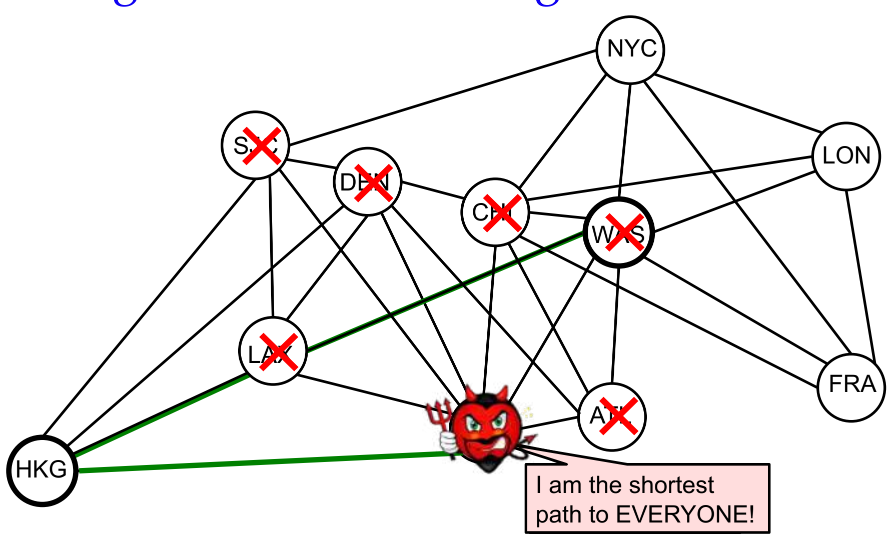
The connectivity graph of a commercial cloud network

Regular Secure Routing (e.g. IPsec)



Regular secure routing takes the shortest path from source (HKG) to destination (WAS).

Regular Secure Routing Under Attack



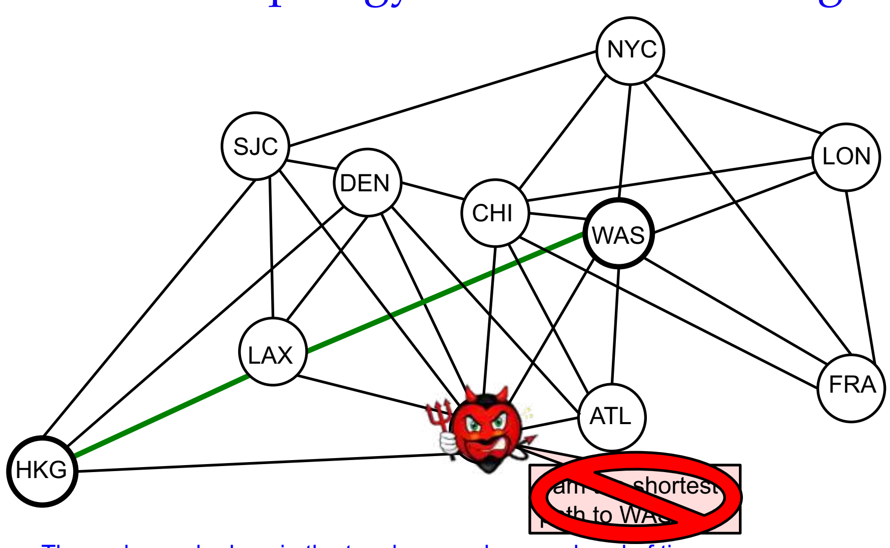
A compromised node can lie and attract traffic, which can then be dropped.

This attack would succeed even if IPsec is used!

Intrusion-Tolerant Overlay Network

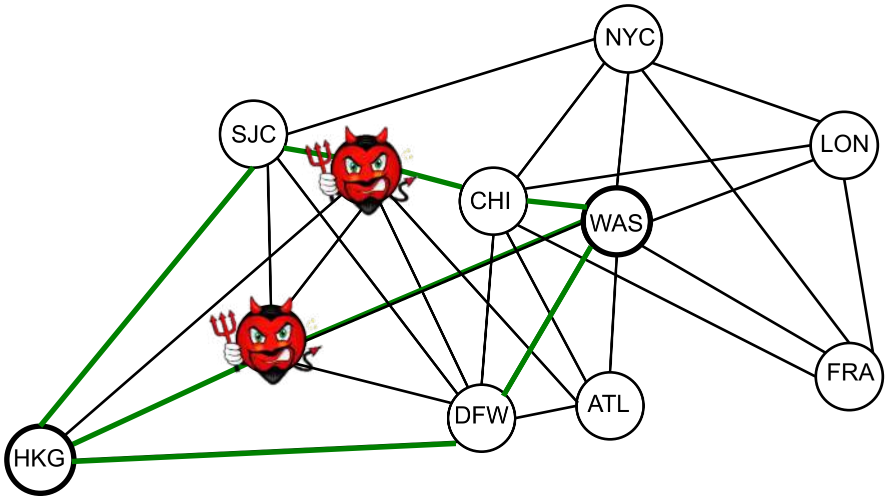
- Resilient architecture reduces problem to **single** (albeit hard) issue of tolerating compromises at the overlay level
- Overlays enable new **practical** solutions that were previously infeasible
  - Programmability
  - Single administrative domain
- Complete solution requires resilient networking architecture combined with intrusion-tolerant overlay

Maximal Topology with Minimal Weights



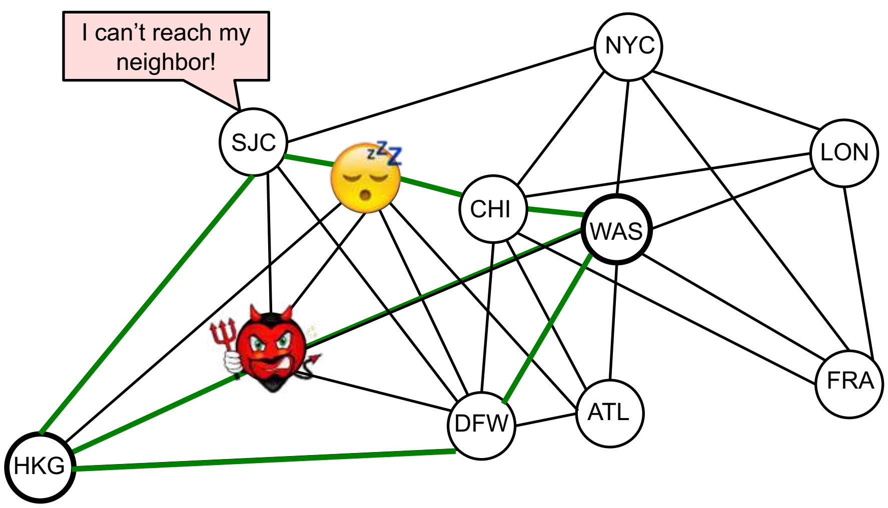
- The nodes and edges in the topology are known ahead of time
- No node can advertise weights below the minimal weights – attack defeated

K Node-Disjoint Paths



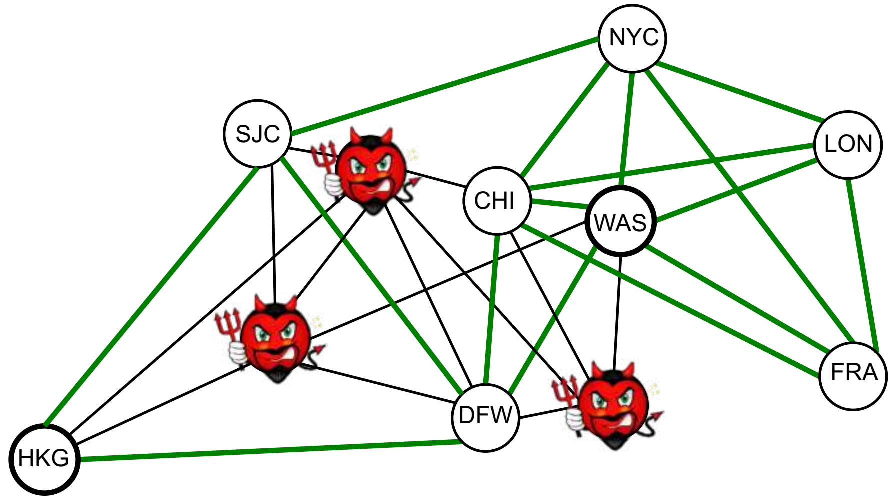
K node-disjoint paths defends against K-1 compromised nodes.

K-Paths Reroutes



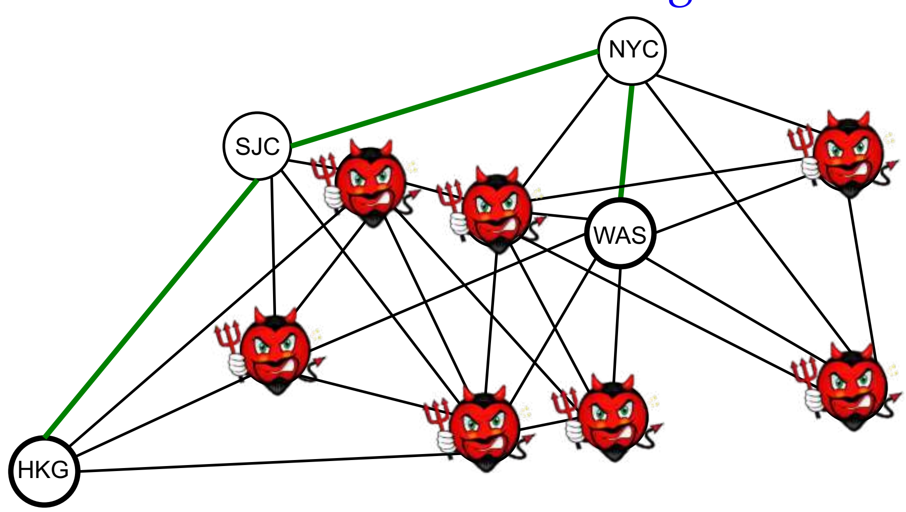
K-paths is resilient to K-1 intrusions, plus any number of benign faults, as long as the network minus the benign faults can still support K paths.

Constrained Flooding



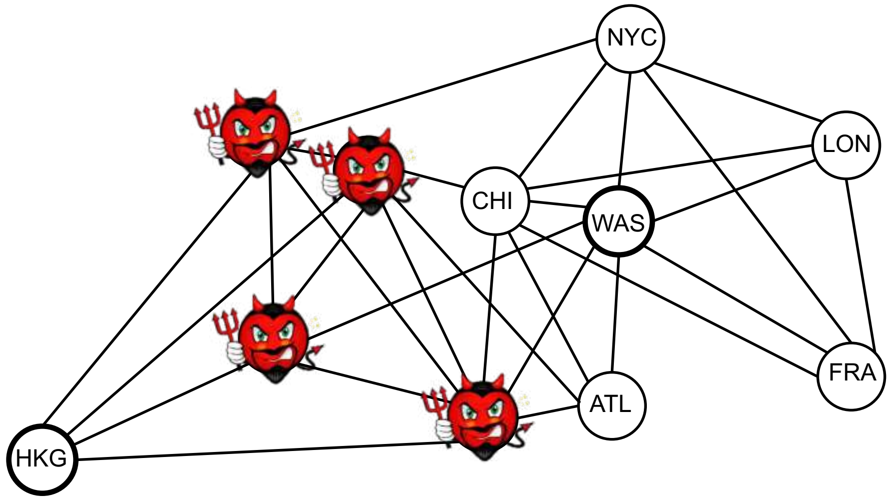
Flooding across the **overlay** network provides optimal resiliency.
Costs more, but we're willing to pay for the most important messages.

Constrained Flooding



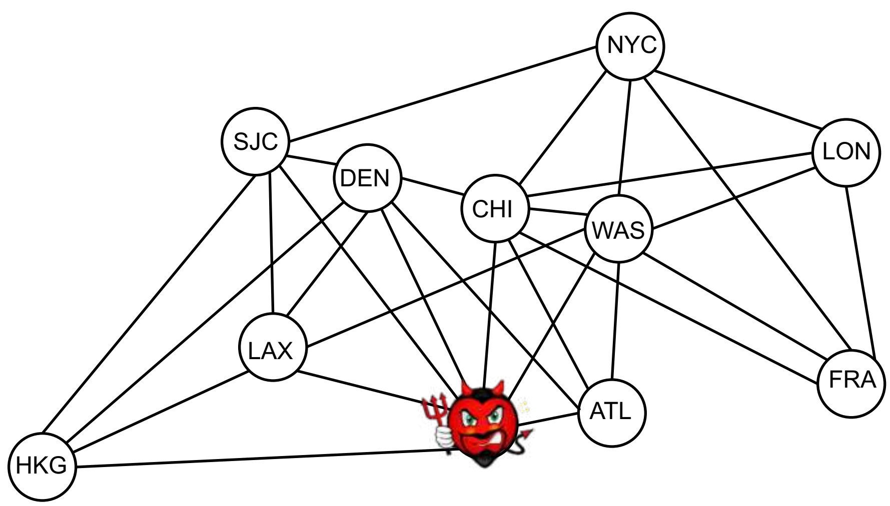
If even a single good path exists, constrained flooding will pass messages from source to destination in a timely manner.

Cutting the Network



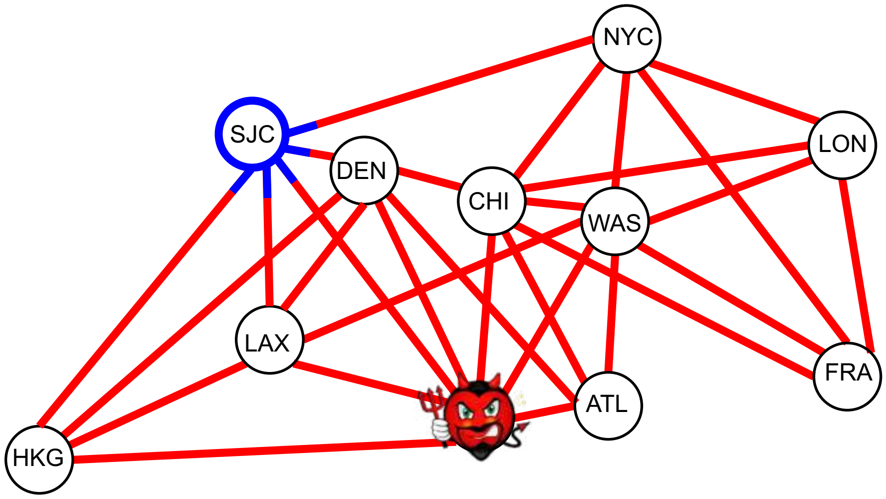
If the compromised nodes cut the network, no protocol can succeed.

What about Compromised Sources?



Misconception that compromises are limited to malicious forwarding.

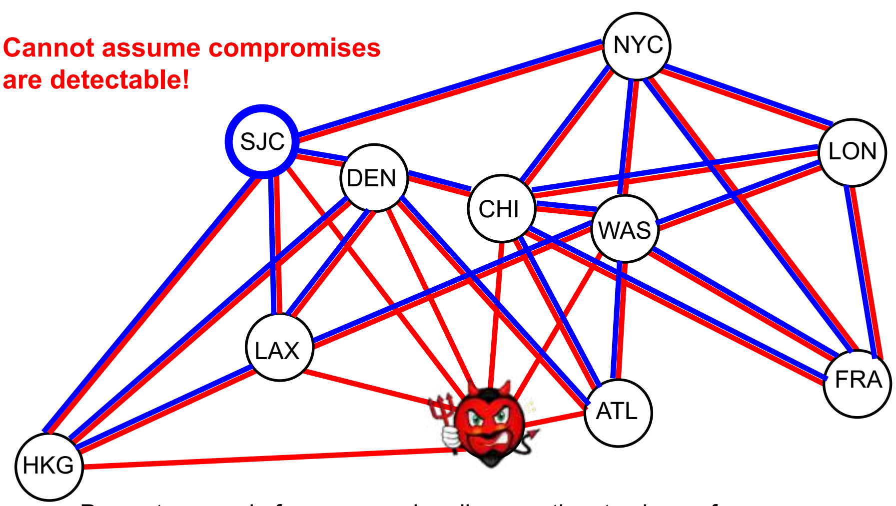
Compromises can Exhaust Resources



Compromised sources can inject spurious messages into the network, exhausting resources from other sources.

Enforce Fair Resource Allocation

Cannot assume compromises are detectable!

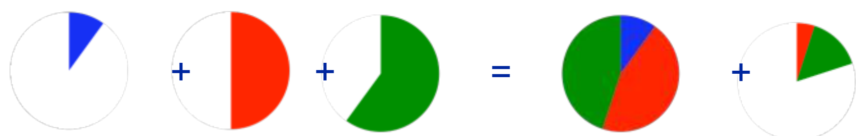


- Prevent any node from consuming disproportionate share of resources
- Each **active** source receives what they request, limited by fair allocation among contending sources

Fairness 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



High-Value Applications Require Semantics

- So far, the intrusion-tolerant overlay only provides best-effort message forwarding
- Critical applications require strong messaging semantics
  - Cloud monitoring: real-time stream of updates
  - Cloud control: reliability and consistency
  - SCADA for power grid: 100-200 ms updates
- We provide strong messaging semantics in the presence of compromises

Intrusion-Tolerant Messaging

	Priority	Reliable
K-Paths Routing		
Constrained Flooding		

Intrusion-Tolerant Messaging

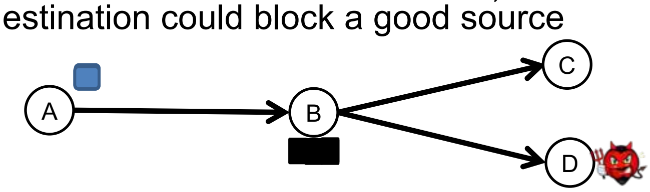
	Priority	Reliable
K-Paths Routing	Source-based routing on *K* node-disjoint paths Overcomes *K*-1 Compromises	
Constrained Flooding	Constrained authenticated flooding on a specified subset of the network topology Optimal Resiliency	

Intrusion-Tolerant Messaging

	Priority	Reliable
K-Paths Routing	Motivated by the real-time demands of cloud monitoring messages	Motivated by the reliability demands of cloud control messages
	Source fairness	Source-Destination fairness
	Source-defined priority for each message	Back pressure employed all the way back to the source
Constrained Flooding	Select a source in round-robin order, send its oldest highest priority message	Keep message until all neighbors have it (option) or end-to-end ACK is received
	Low-latency guarantees	Eventual-path reliability

The Problem of Source-Based Fairness in Reliable Communication

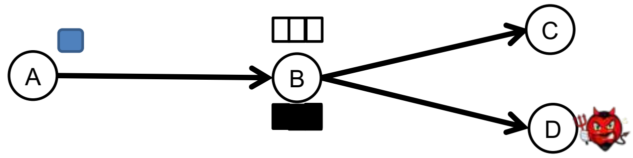
- If we used source based fairness, a malicious destination could block a good source



- 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)

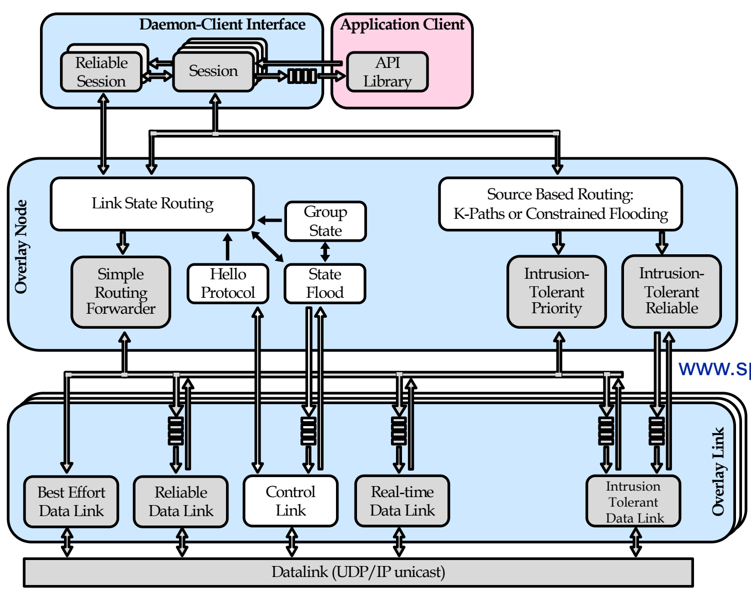
Flow-based Fairness

- Instead, treat each flow separately.



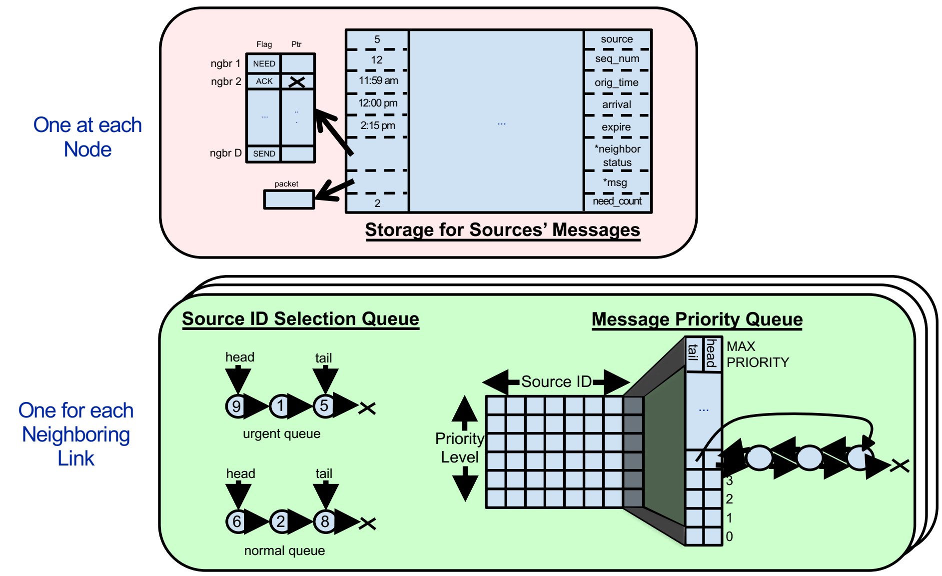
- The A-D flow becomes blocked
- The A-C flow does not

Intrusion Tolerant Spines

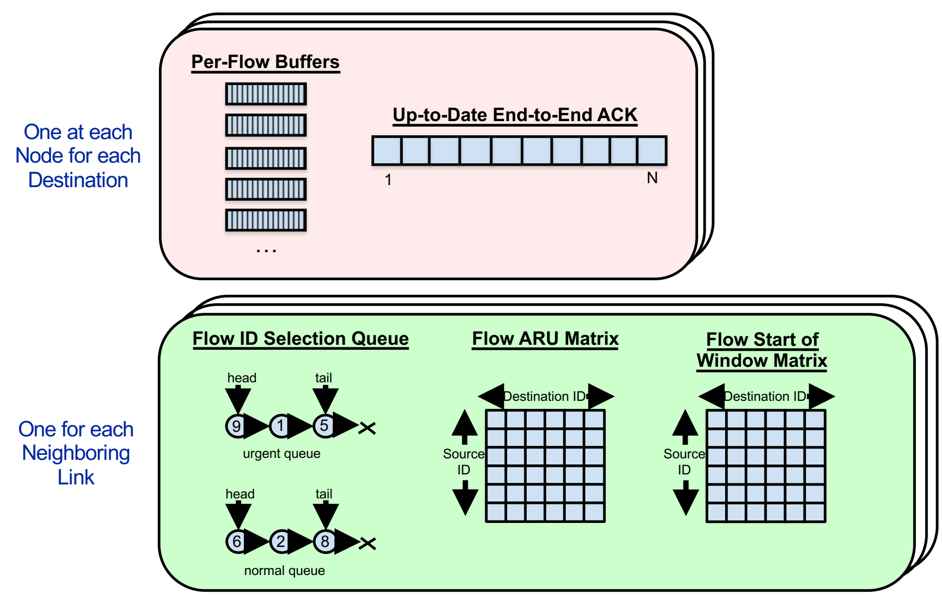


www.spines.org

Priority Semantics



Reliable Semantics



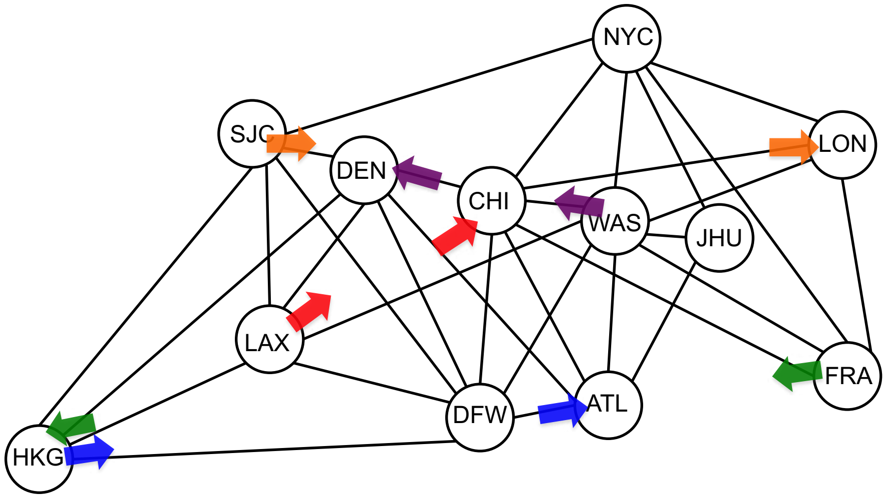
Cryptographic Protocols

- Network-Wide Authentication
  - Public/Private key pair for each overlay node
  - Each overlay node knows all public keys
  - Source nodes put RSA signature on each message
  - RSA verification of messages at each forwarding node
  - Alternative: EC crypto for low-bandwidth environments
- Hop-by-Hop Authentication
  - Authenticated Diffie-Hellman Key Exchange to establish a shared secret key
  - HMAC using SHA256 on all subsequent messages
- Implemented in Spines using OpenSSL

Intrusion-Tolerant Network (Spines) Demonstration

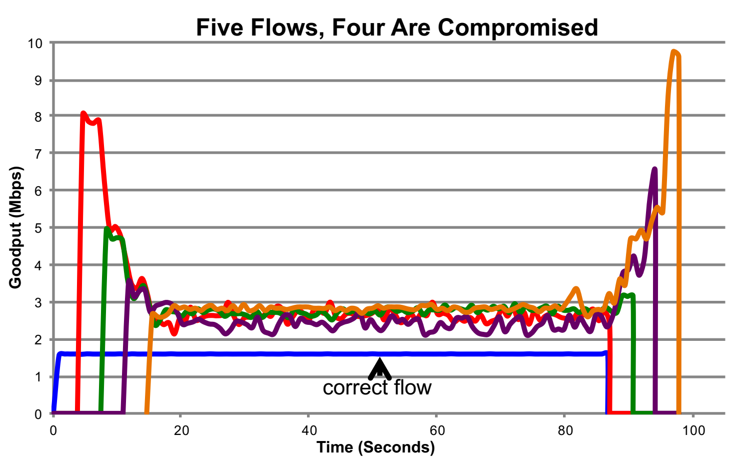
- Real time comparison of video channels
  - Local vs. Cross-country and back
  - (ATL to WAS) vs. (ATL to LAX to WAS)
- Compromise at DFW
  - Maliciously injected loss
  - Node goes dark at a point of its choosing
  - Malicious increased delay over time
- Left video: conventional shortest-path routing
- Right video: intrusion-tolerant protocols

Evaluation: LTN Global Cloud



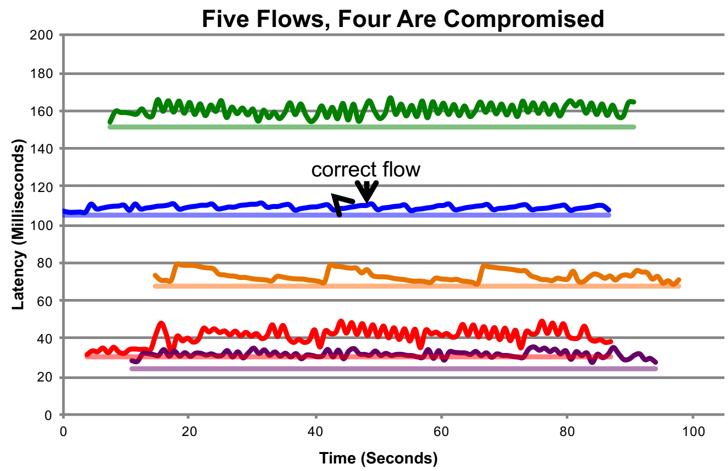
- All experiments run on the real cloud – no emulation
- Measured: communication cost, protocols under attack

Priority Flooding – Goodput



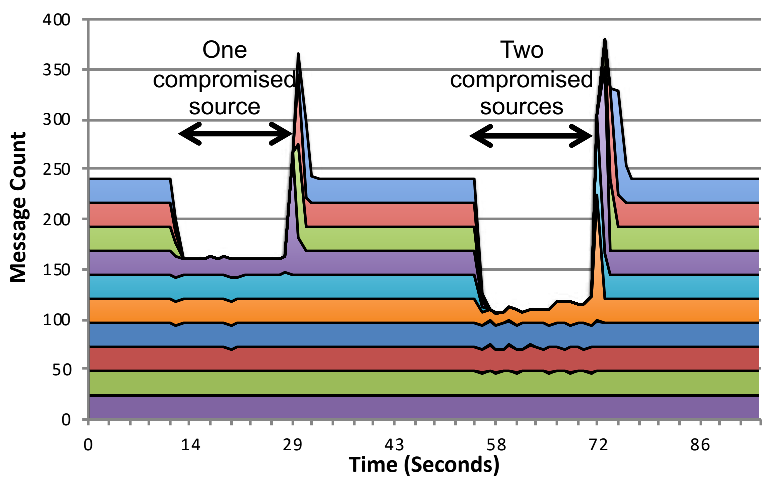
- Correct flow sending at fair share is unaffected by compromised flows that send at maximum capacity

Priority Flooding – Latency



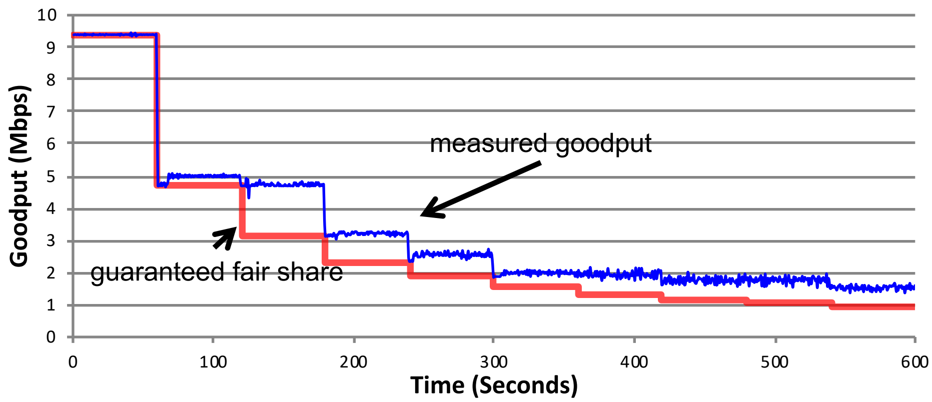
- All flows experience latency (jagged) close to propagation delay (flat)
- Correct flow is very close to propagation delay because it sends less than its fair share

Priority Flooding Under Attack



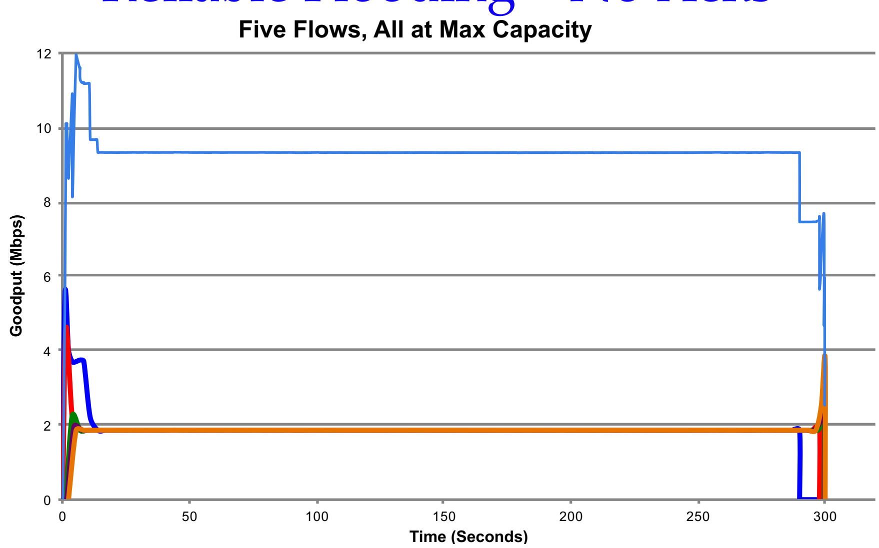
- Timely delivery of highest priority messages within correct flow's fair share is guaranteed

Priority Flooding with Many Compromised Sources



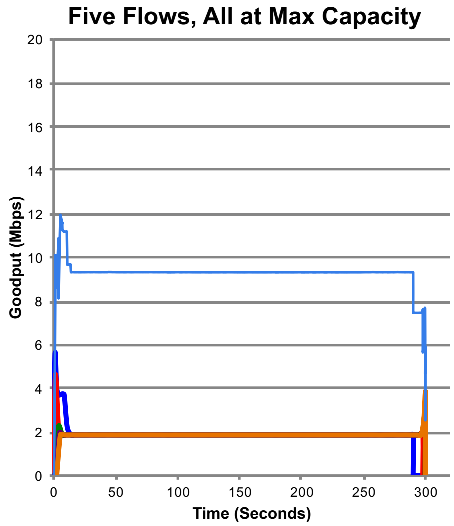
- Correct source (blue) is always guaranteed at least its fair share (pink) of goodput based on number of actively sending sources
- All other sources are compromised, sending at maximum link capacity

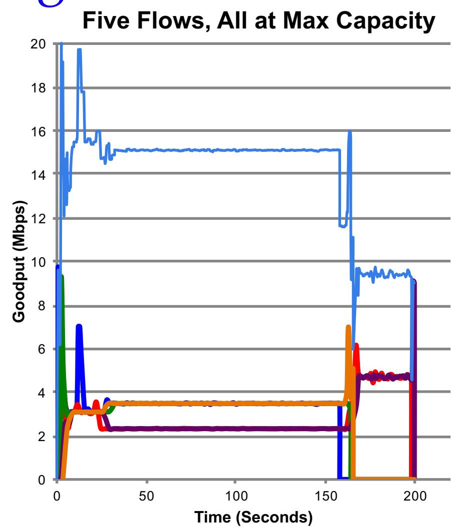
Reliable Flooding – No Acks



- All flows' messages contend equally on all network links, resulting in each flow getting exactly its fair share of bandwidth

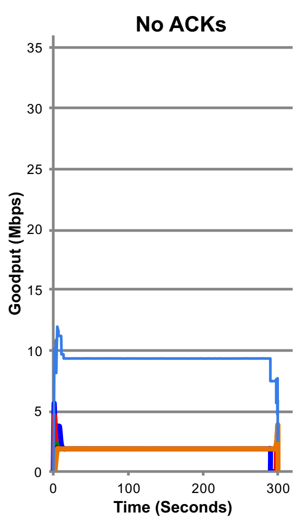
Reliable Flooding – HBH Acks

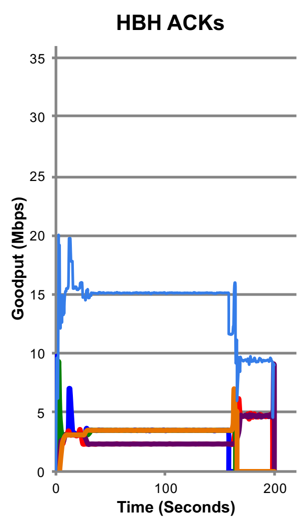


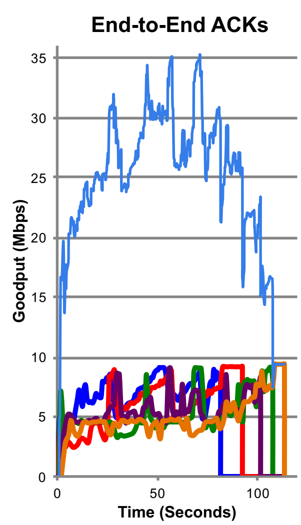


- Adding hop-by-hop neighbor acknowledgements increase per-flow bandwidth by not needing to send each message on every link

Reliable Flooding – HBH & E2E Acks

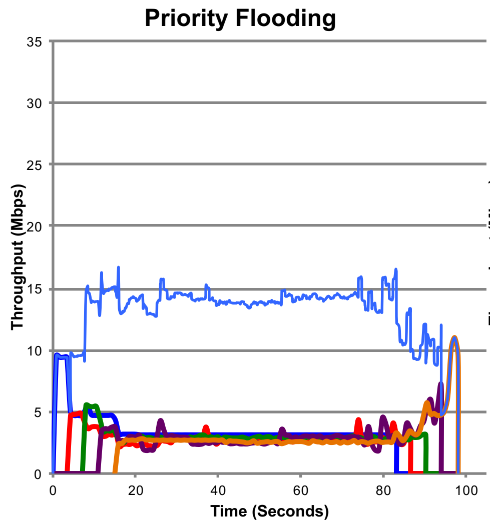


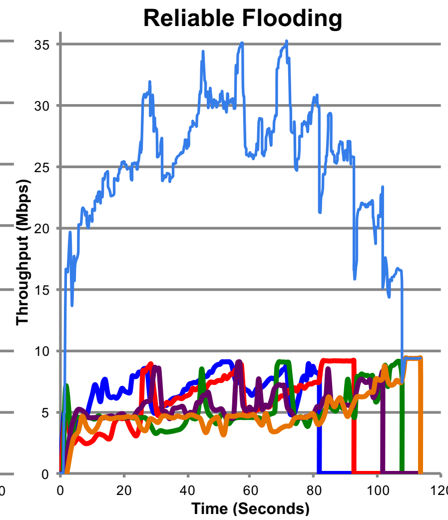




- Required E2E feedback actually enables even higher per-flow bandwidth utilization, as messages no longer need to be sent to all parts of the network

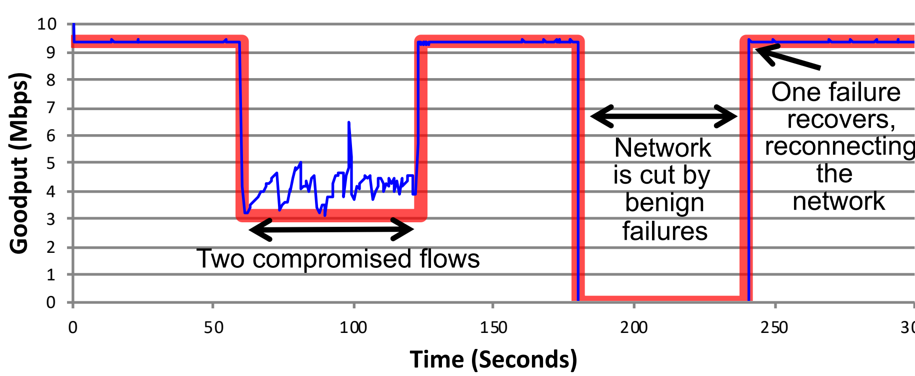
Acknowledgements Improve Throughput





- Reliable Messaging feedback allows it to better utilize bandwidth, but only Priority Messaging gives timeliness guarantees

Reliable Flooding Under Attack



- Goodput always exceeds guaranteed fair share
- All messages delivered end-to-end reliably and in order

Shadow Monitoring System

- Used deployment to carry copy of monitoring messages of the global cloud
  - Status of data centers, network characteristics (e.g. latency, loss), status of cloud clients, etc.
- 10 month deployment
- Used Priority K-Paths and Priority Flooding
- Validates intrusion-tolerant network
  - Messages were equally as timely with intrusion-tolerant guarantees (for a higher tunable cost)

Summary

- An overlay-based practical solution for intrusion-tolerant networking
- Expensive, but complete solution for high-value applications
- Validated on a global scale
- Open-Source implementation available in Spines overlay messaging framework – www.spines.org

References

- [AHNR2002] - B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. “An on-demand secure routing protocol resilient to Byzantine failures,” in Proc. 1st ACM Workshop on Wireless Security, 2002, pp. 21-30.
- [ACHN+2008] - B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, “ODSBR: An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks,” ACM Trans. Information and Syst. Security, vol. 10, no. 4, pp. 6:1–6:35, Jan. 2008.
- [AGR1992] - Y. Afek, E. Gafni, and A. Rosén. "The slide mechanism with applications in dynamic networks." In Proc. 11th ACM symposium on Principles of Distributed Computing (PODC), 1992.
- [ABO2009] - Y. Amir, P. Bunn, and R. Ostrovsky, “Authenticated adversarial routing,” in Proc. 6th Theory of Cryptography Conf (TCC), 2009, pp. 163–182.
- R. Perlman, “Network layer protocols with Byzantine robustness,” Ph.D. dissertation, Massachusetts Institute of Technology, 1989.
- [ZHHC+2011] - X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. Andersen, “SCION: Scalability, control, and isolation on next-generation networks,” in IEEE Symp. Security and Privacy (SP), May 2011, pp. 212–227.
- [BRSP+2016] - C. Basescu, R. Reischuk, P. Szalachowski, A. Perrig, Y. Zhang, H Hsiao, A. Kubota, and J. Urakawa. “SIBRA: Scalable Internet Bandwidth Reservation Architecture,” To appear in Proc. of NDSS 2016.
- [OTBS+2016] – D. Obenshain, T. Tantillo, A. Babay, J. Schultz, A. Newell, M. Hoque, Y. Amir, C. Nita-Rotaru. “Practical Intrusion-Tolerant Networks,” In Proc. of IEEE International Conference on Distributed Computing Systems (ICDCS), June 2016, pp. 45-56.