Mitigating Attacks in Unstructured Multicast Overlay Networks

Cristina Nita-Rotaru, Aaron Walters, David Zage

Dependable and Secure Distributed Systems Lab ((DS)^2)

Department of Computer Science and CERIAS

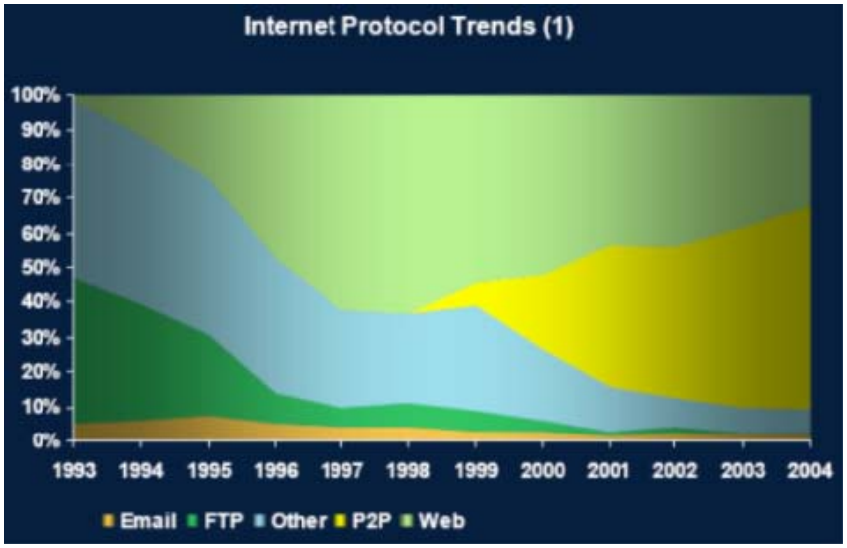
Purdue University

Dependable and Secure Distributed Systems Lab

- Collaborative services for wireless mesh networks
- Security of peer-to-peer multicast/streaming systems
- Byzantine-resilient replication
- Funding: NSF CyberTrust and DARPA

A Paradigm Shift

- Web traffic was dominant in the previous decade
- Explosion of p2p traffic, file sharing, Skype, streaming



M. Meeker D. Joseph *Web 2.0 2006*

Some reports claim about 60% of Internet traffic is P2P

Overlay Networks

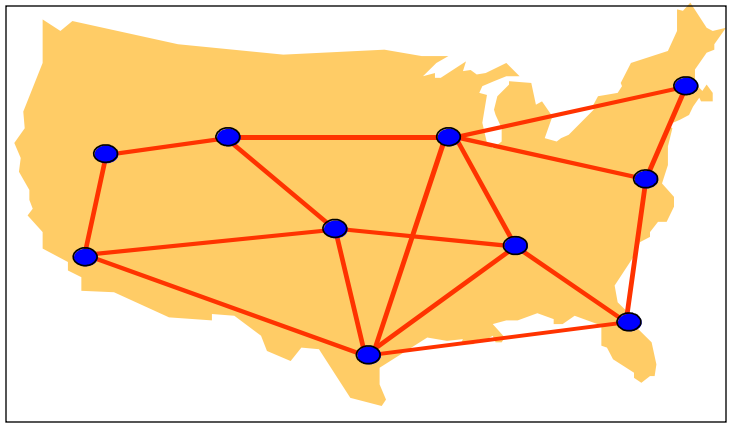
- Enable file sharing and multicast applications
- Provide performance and reliability
  - **Increased capacity**: nodes provide storage space, and computing power
  - **Increased performance**: nodes dynamically optimize application-centric metrics
  - **Increased reliability**: more resilient dissemination, data replicated over multiple peers, data lookup without relying on a centralized index server

Overlays Architectures

- **Structured overlay networks**:
  - **Neighbor set selection is constrained:** a small subset of nodes meeting prescribed conditions are eligible to become neighbors
- **Unstructured overlay networks**:
  - **Neighbor set selection is not constrained:** anybody can be a neighbor
- **Hybrid overlay networks**:
  - Combines characteristics of both

Zooming on Overlay Multicast

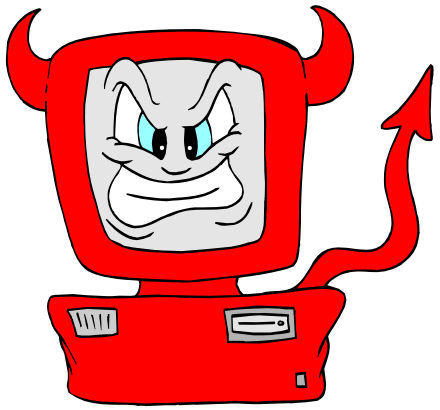
- Multicast tree(s) or a mesh that **adapts** to meet/improve application performance and resilience
  - structured overlays: Scribe, SplitStream
  - unstructured overlays: ESM, Nice, Overcast, ALMI, Chainsaw



Example of a mesh overlay

Security and Overlay Networks

- Deployment over public open networks
  - Vulnerable to malicious attacks coming from outside the overlay network
- Push trust to end-nodes: anybody can be part of the overlay
  - Vulnerable to malicious attacks coming from inside the overlay network (**Byzantine attacks**): attacker can use the overlay to attack the Internet, or attack the overlay itself



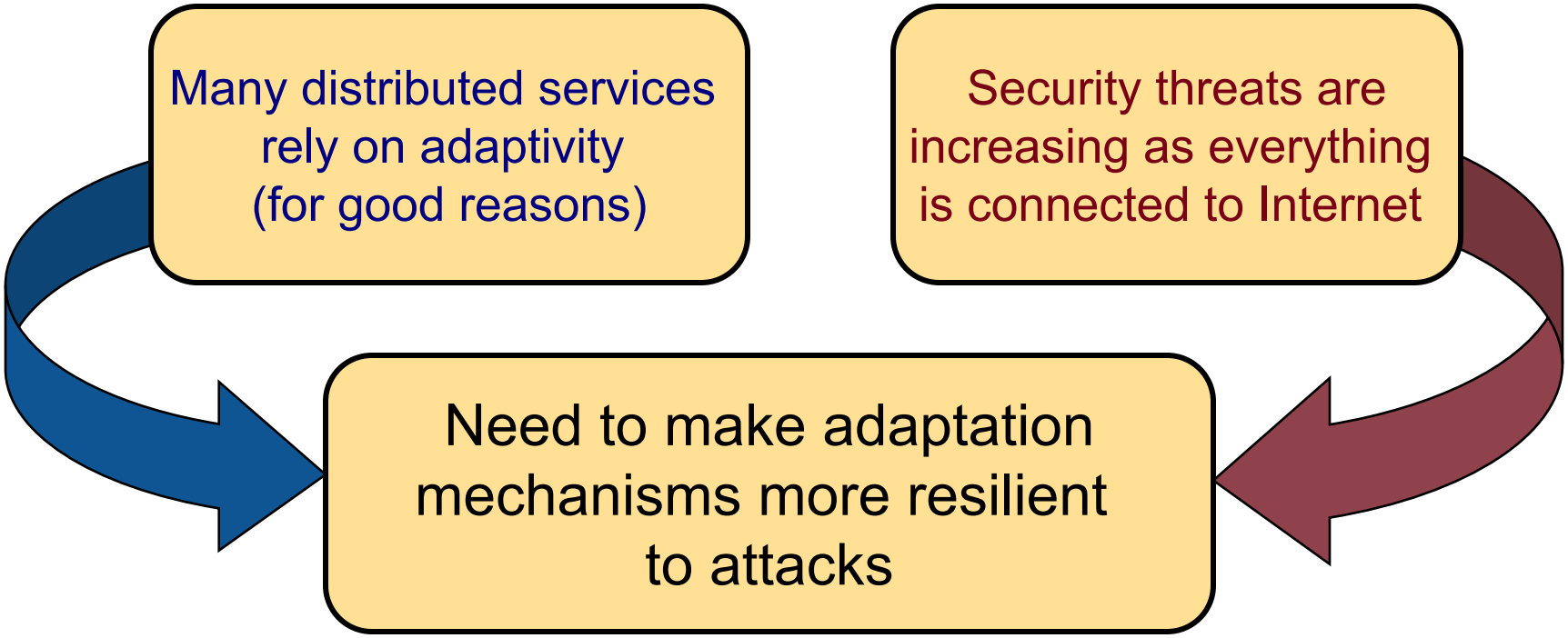
In Summary …

Explosion of p2p systems

Security is critical for these systems

Need to examine security of overlays and make them more resilient to attacks

Beyond Overlay Networks …



This Talk …

- Presents Byzantine attacks against **adaptation mechanisms** in *unstructured multicast overlays*
- Describes mechanisms to **prevent incorrect adaptation** decisions and **limit the impact of the attack**
- Shows how to apply the proposed solution to other services such as Internet virtual coordinate systems

Outline

- Introduction
- **System and attacker model**
- Attacks classification and demonstration
- Discuss solution space
  - Prevent poor adaptations
  - Isolating malicious nodes
- Virtual coordinate systems
- Conclusion

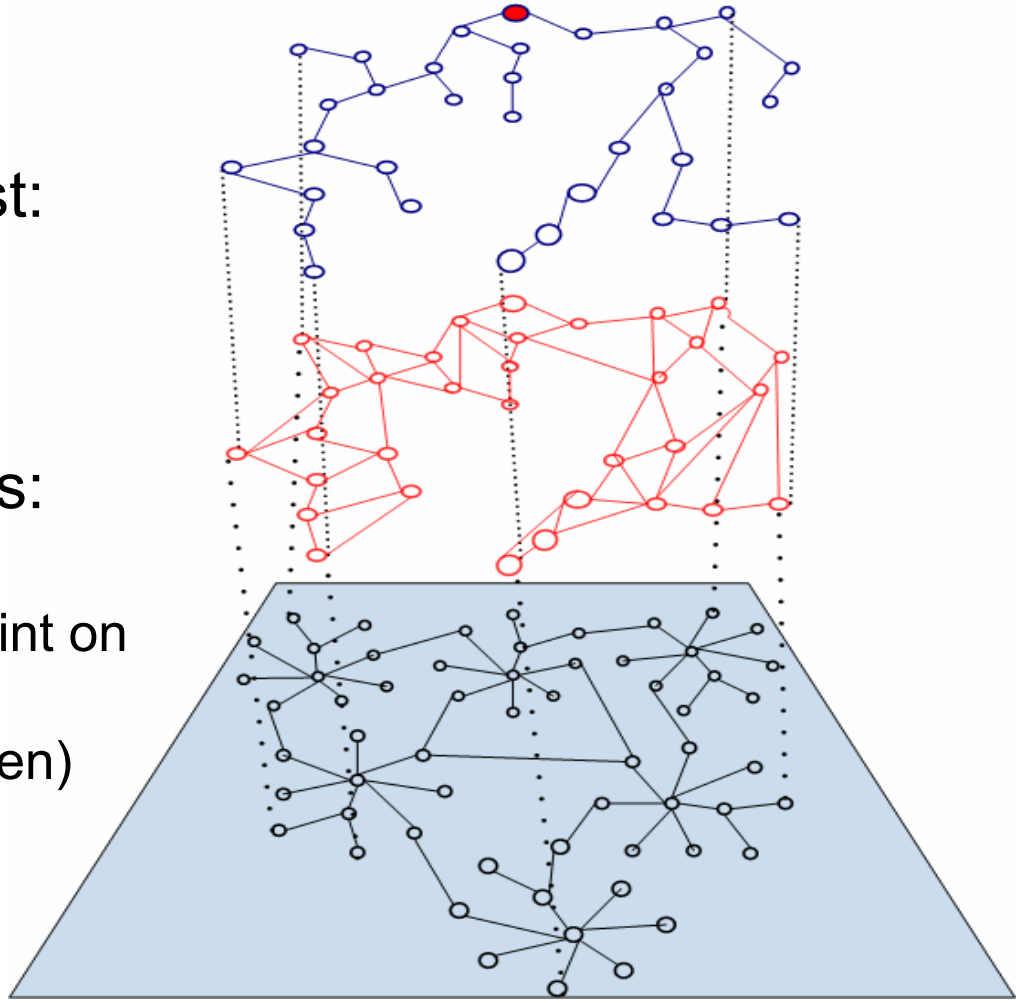
QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Related Work

- Adaptivity exploited by adversaries against TCP: [KK03], generalized in [GBM04,GBMZ05]
- Solutions against malicious attacks or mis-configurations of BGP [ZPW+02]
- Attacks against routing in structured overlay networks [CDGRW02,SNDW06]
- Attacks using p2p against Internet [NR06], [DKM07], [STR07]

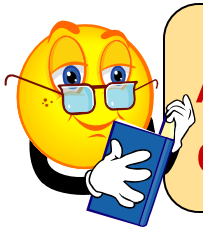
Unstructured Multicast Overlay

- Mesh control plane
- Tree-based multicast: adapts to maintain application specific performance
- Each node maintains:
  - Parent
  - Peer set: no constraint on neighbor selection
  - Routing table (children)



Adaptation

- Metrics are collected by nodes through
  - Passive observation of their own performance from the source
  - Periodic probing of peer nodes about their performance from the source
- Metrics are used to compute a utility function
- Based on the utility function, a node makes the decision to change its parent in the tree



Accurate interpretation of performance observations and the correctness of the responses from probed nodes are critical!

Example: ESM Adaptation

- Metrics considered: **available bandwidth, latency, RTT and saturation degree**
- Data quality:
  - Data sampling and smoothing are used to address variations in the metrics
  - Damping, randomization, hysteresis are used to address instabilities in the observed data
- Decision quality:
  - Utility functions based on bandwidth, and/or latency

Attacker Model

- Attacker is one of the nodes in the overlay (he compromised one or several nodes, or infiltrated in the overlay)
- Bounded percentage of malicious nodes **f** ($0 \leq f < 1$) out of total N nodes
- Attacker has access to all cryptographic keys stored on the compromised node.
- **Compromised nodes**
  - **can lie about the observation space (bandwidth, latency, degree)**
  - **can impose an artificial influence toward the observation space**

Outline

- Introduction.
- System and attacker model
- **Attacks classification and demonstration**
- Discuss solution space
  - Prevent poor adaptations
  - Isolating malicious nodes
- Virtual coordinate system
- Conclusion

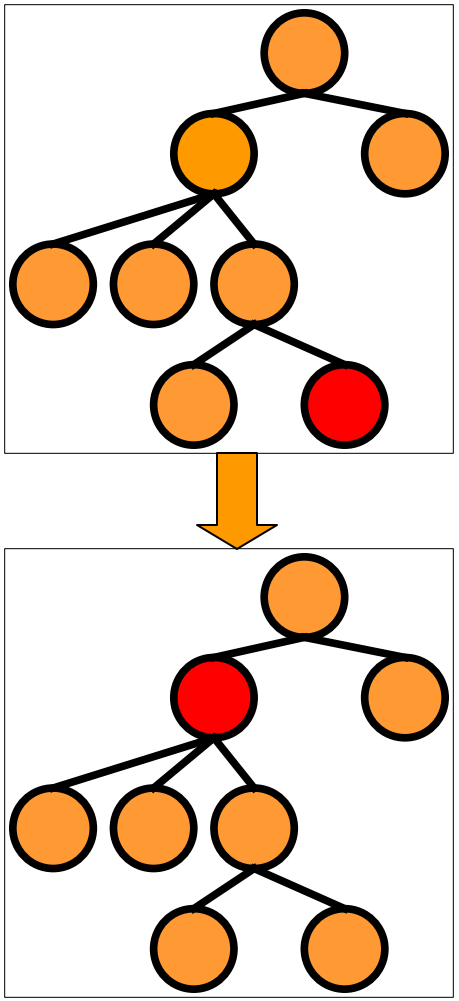
QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Attacks Exploiting Adaptation

- Classification of attacks based on their effect on the control of path:
  - Attraction attacks
  - Repulsion attacks
  - Disruption attacks
- Used to facilitate further attacks:
  - Selective data forwarding
  - Traffic analysis
  - Overlay partitioning
  - and more ….

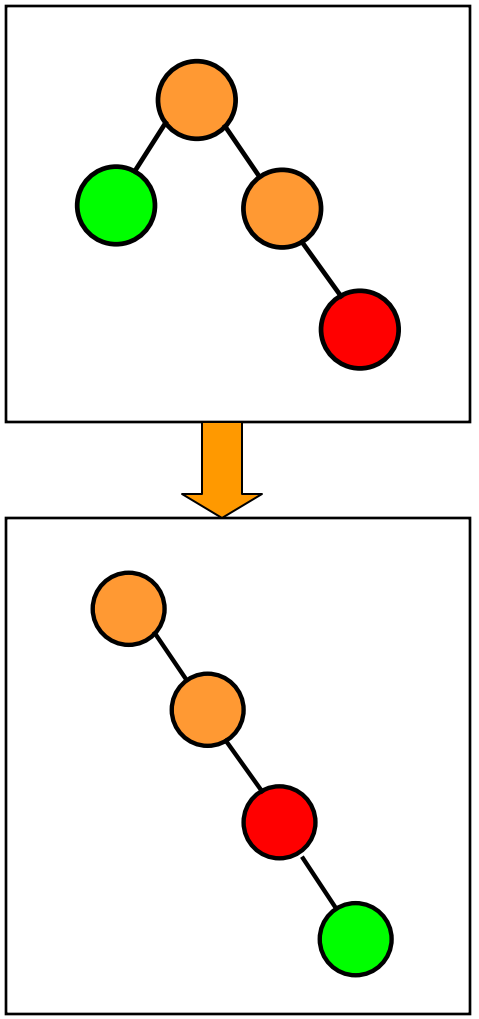
Attraction Attacks

- *The more children a node has or higher in the tree is, the higher the control of data traffic*
- **Attacker goal**: attract more nodes as children in the overlay structure
- **How does it work:** a node **makes things look better** by lying about its reported metrics
- **Result:** controlling significant traffic, further conduct traffic analysis or selective data forwarding

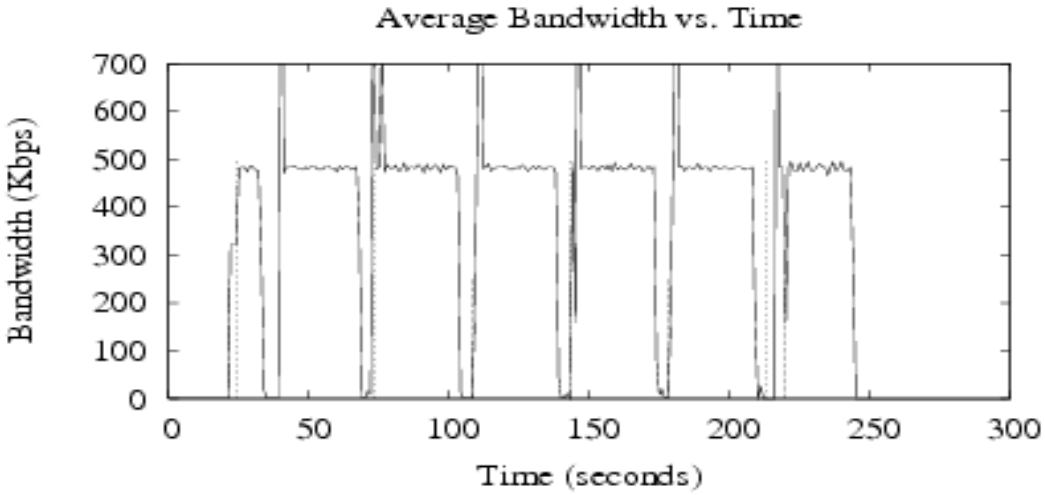


Repulsion Attacks

- *A node in the overlay may affect the perception of the performance from the source*
- **Attacker goal**: reduce the appealing of other nodes or its own
- **How does it work**:
  - a node lies in responses to probes
  - a node manipulates the physical or logical infrastructure to create the perception of lower utility of other nodes
- **Result**: freeloading, traffic pattern manipulation, augmenting attraction attacks, instability



Disruption Attacks



- *Frequent adaptations can create instability*
- **Attacker goal**: exploit the adaptation to turn the system against itself
- **How does it work:** attacker injects data to influence the observation space metric data to generate a series of unnecessary adaptations, similar with TCP attack
- **Result**: jitter, flapping, or partitioning the overlay

Experimental Setup

- Using ESM
- Planetlab and DETER
- Deployments of 100 nodes
- Experiment durations of 30 and 90 minutes.
- Saturation degree of 4-6 nodes
- Constant bit rate of 480 Kbps

Attraction Attacks

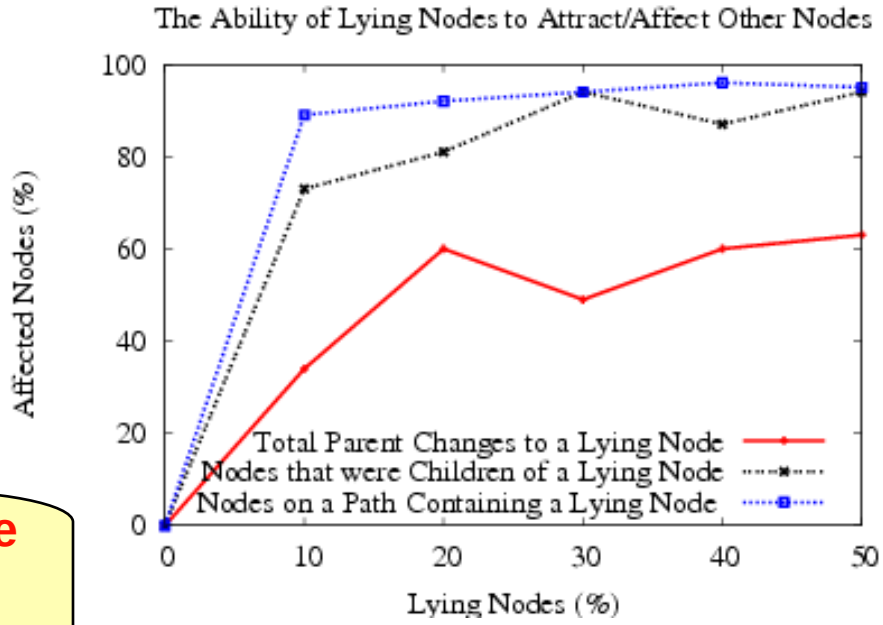
100 nodes, PlanetLab, 60 minutes, malicious nodes lie about bandwidth, latency, saturation

Impact of 1 malicious node

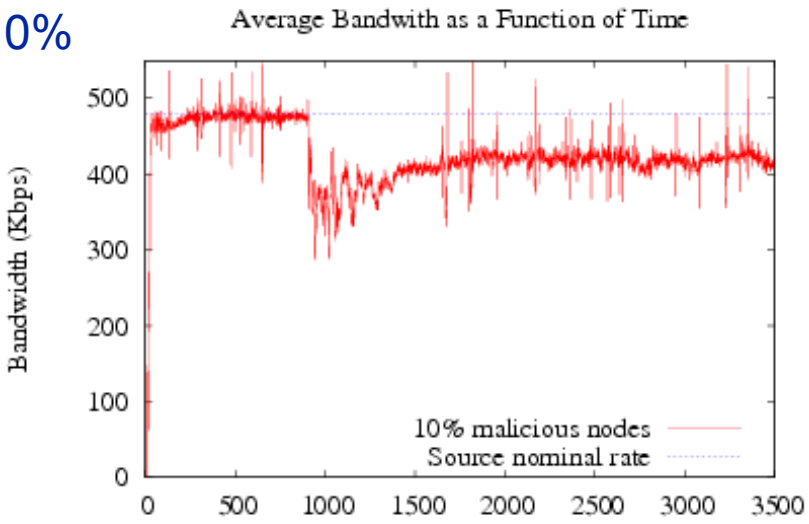
	Selected as parent	Parent Changes
Lying	72	369
Not Lying	15	216

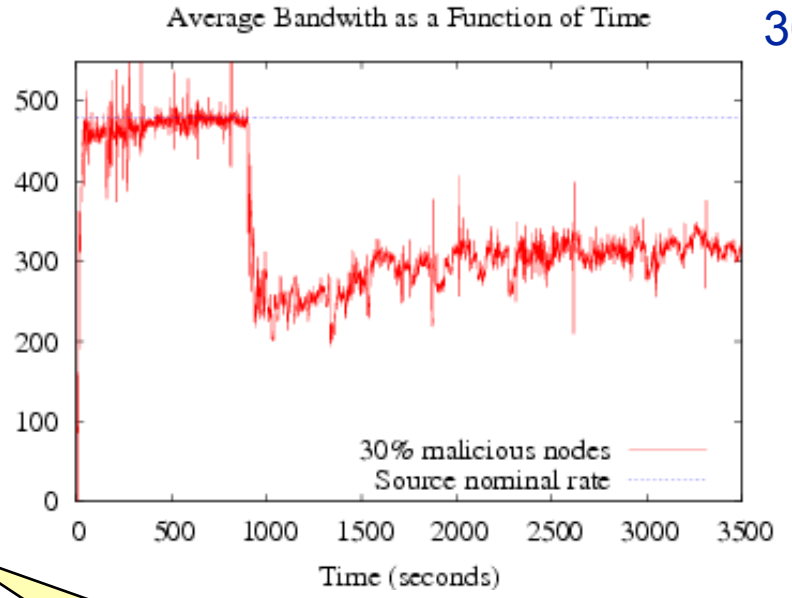
Lying increases the chance of a node being selected as parent almost 5 times

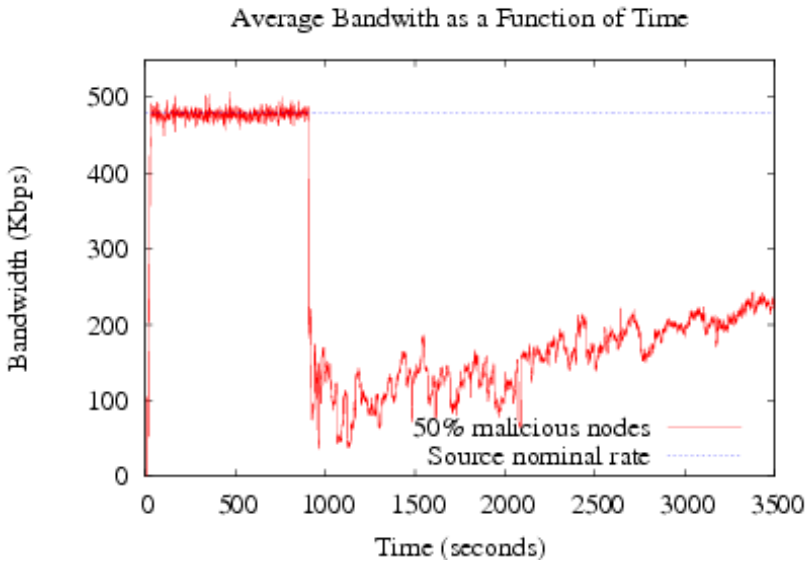
Impact of % of malicious nodes



Impact of Number of Adversaries



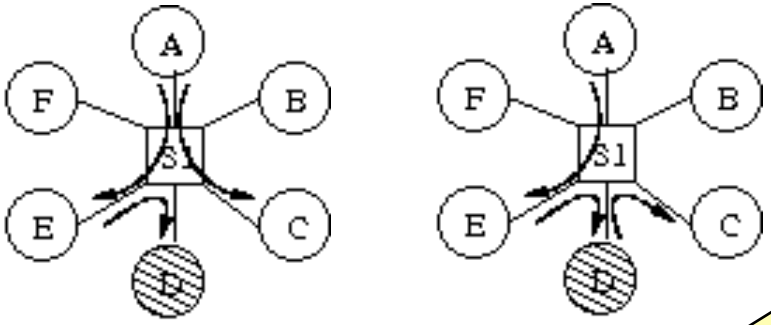




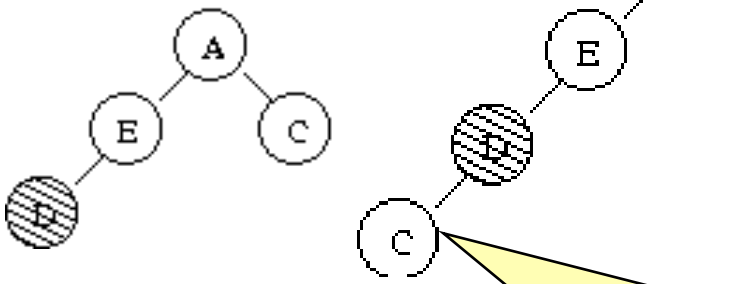
Nodes were randomly selected

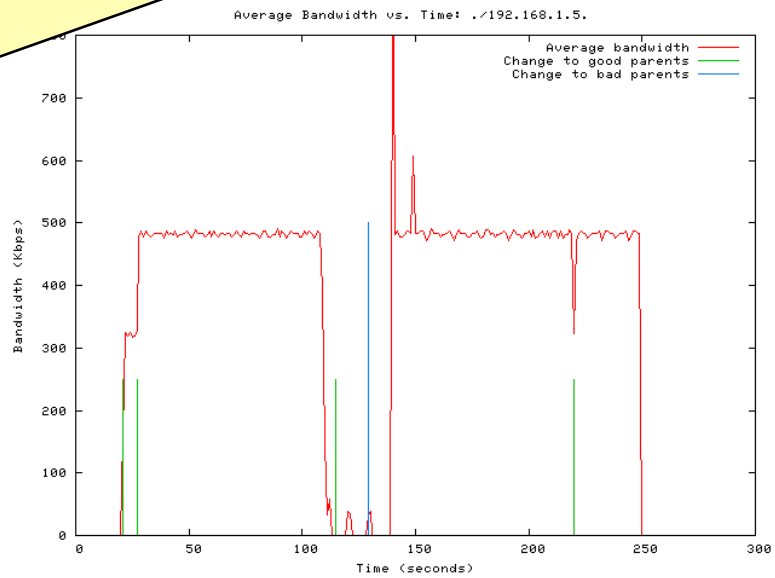
Tree is not resilient to malicious behavior, several malicious nodes can cause significant disturbance**!**

Repulsion Attacks



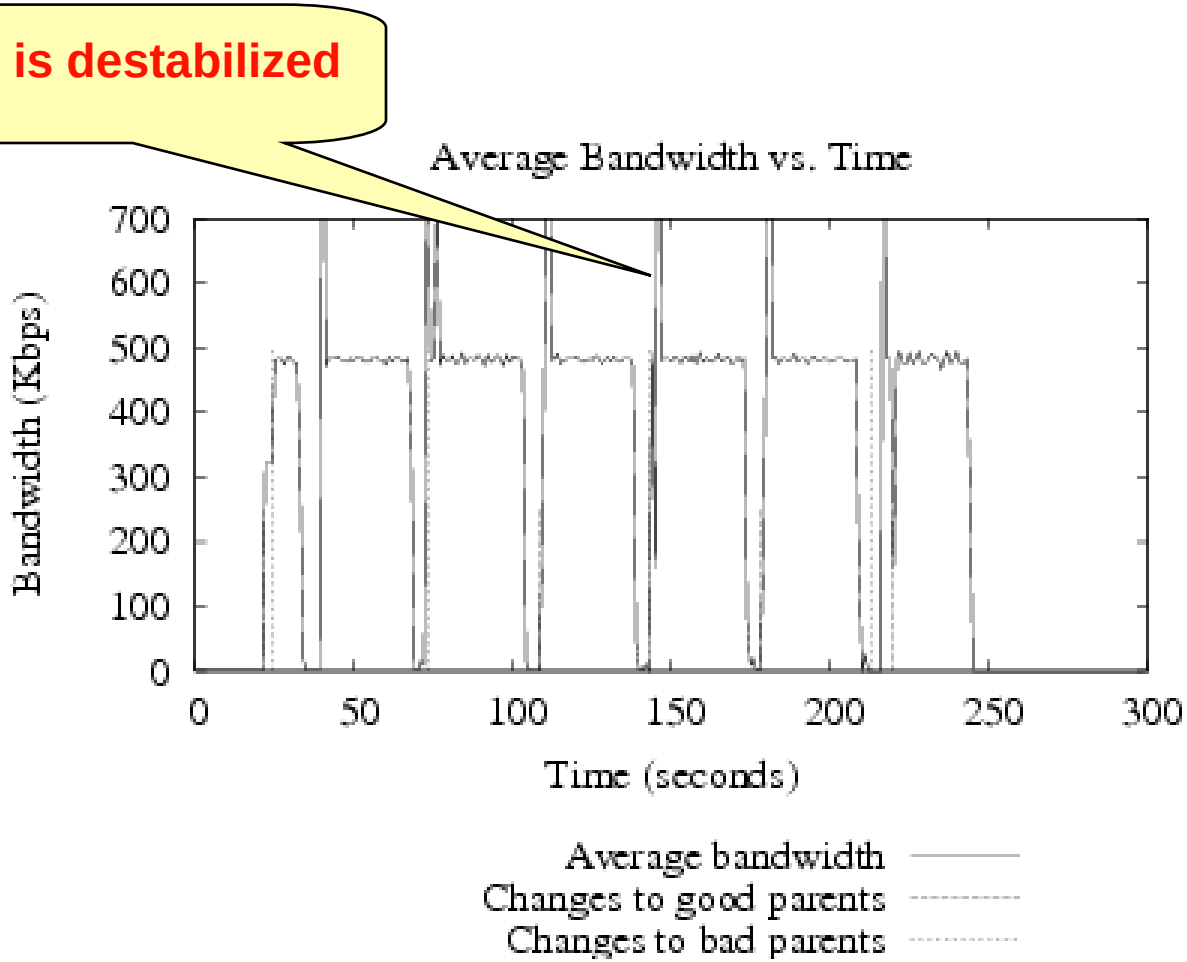
D exploits the physical topology to make C disconnect from the source





C is now 3 hops away from the source

Disruption Attacks



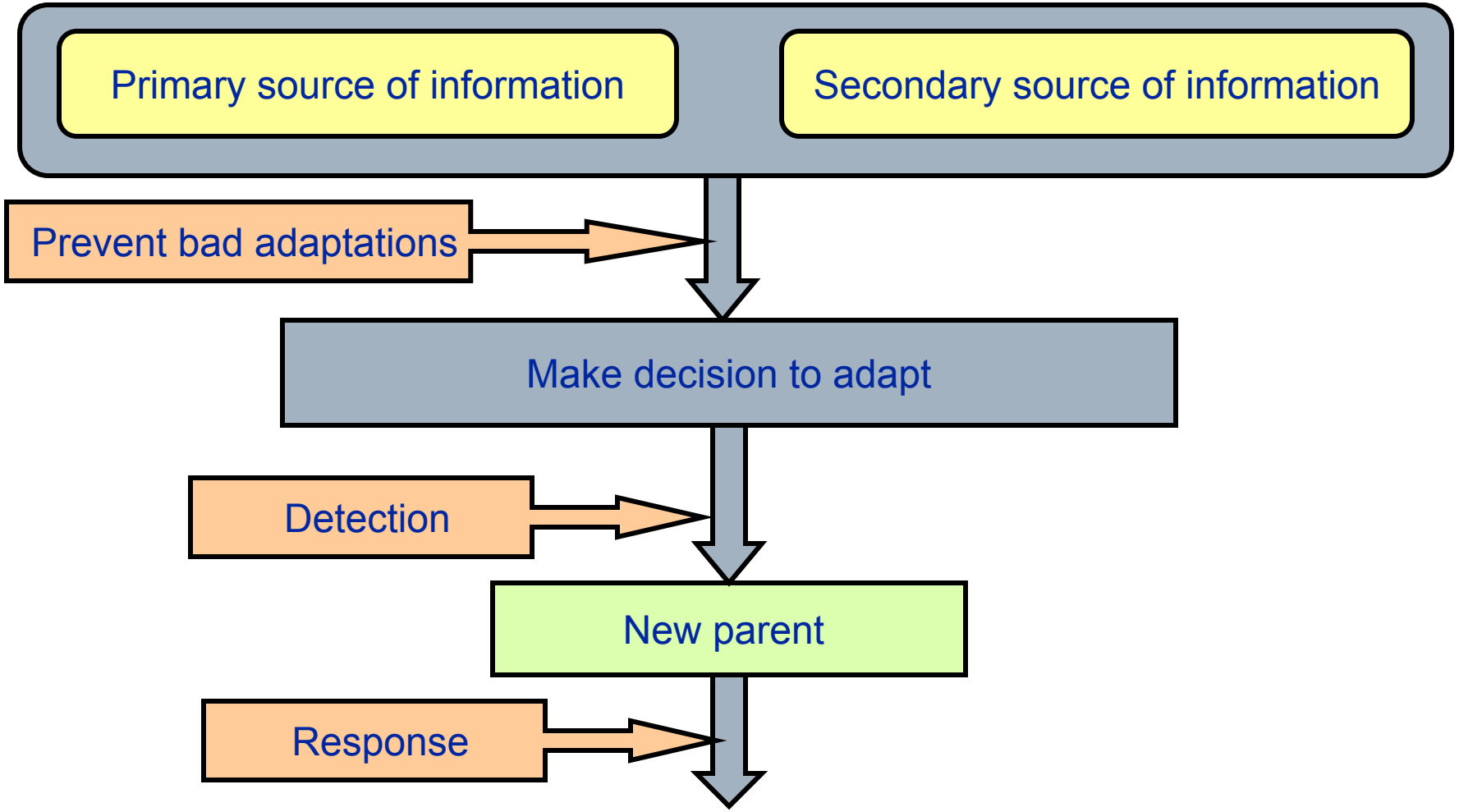
Outline

- Introduction.
- System and attacker model
- Attacks classification and demonstration
- **Discuss solution space**
  - Prevent poor adaptations
  - Isolating malicious nodes

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

- Virtual coordinate system
- Conclusion

Solution Framework



Prevention

- **Goal**: Reduce the likelihood of making poor adaptations, *before* they take place
- **Approach**: Use local context to filter out the outliers
- **How**: Using a combination of spatial and temporal correlations and estimation techniques based on statistical outlier detection
- **Challenges**: Analyzing the effect on overall performance, method will not completely eliminate bad adaptations

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Detection

- **Goal**: Detect adversary-controlled adaptations *after* they occurred
- **Approach**: target secondary attacks with observable effect (such as selective forwarding)
- **How**: Use global information available at the source (such as the full path from source to the node) to detect inconsistencies and the low bandwidth unicast channel to transmit the information
- **Challenges**: Effectiveness of method depends on the size of malicious coalition. May have high convergence and overhead

Response

- **Goal**: isolate malicious nodes once they have been suspected
- **Approach**: limit the amount of damage malicious nodes can create
- **How**: Use reputation systems, nodes can also rehabilitate themselves after some time
- **Challenges**: False positives may have an impact on the system.

Improve Stability

- **Goal**: Reduce the instability caused by unnecessary changes
- **Approach**: explicitly integrate stability metrics into the adaptation estimation function
- **How**: include the time a node was connected to his current parent, the frequency of changes, or the degree of variance in metrics
- **Challenges**: Use a stronger control theoretic solution and evaluating the benefit

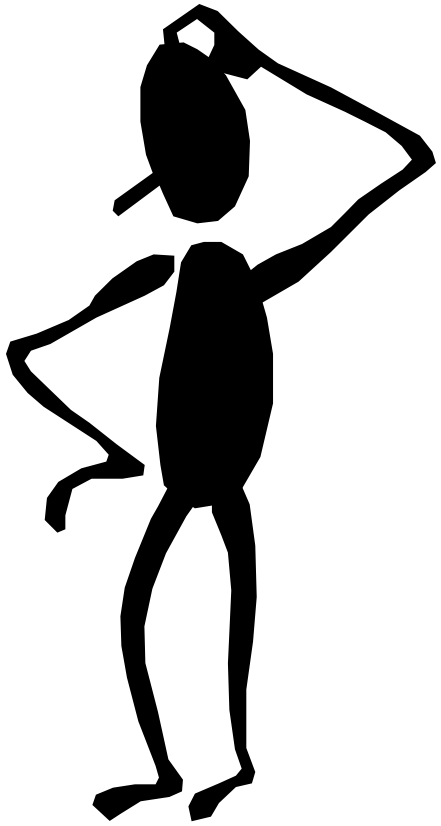
Outline

- Introduction.
- System and attacker model
- Demonstrate attacks
- Discuss solution space
  - **Prevent poor adaptations**
  - Isolating malicious nodes
- Virtual coordinate systems
- Conclusion

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

What We **Can Not Do**

- We can not delimitate correct behavior from an incorrect one in all cases
  - when many nodes collude
  - not enough history is available
- We can not corroborate the information if there is a single source of information



What We **Can Do**

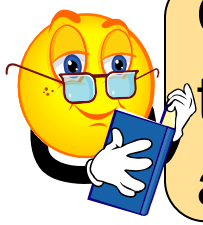
- Make the attack more difficult for the attacker; Take out the very bad decisions.
- Exploit the nature of the system and protocol semantic:
  - look for cases when a node does not lie consistently
- Reduce the number of bad adaptations without adding significant overhead to the system

To Avoid Detection A Node Must Lie:

- C1: consistently with what the other peers are reporting during a probe cycle about current conditions
- C2: consistently with the bandwidth, latency, and influence yielded towards the RTT
- C3: consistently with what it said in the past.

Using Outlier Detection

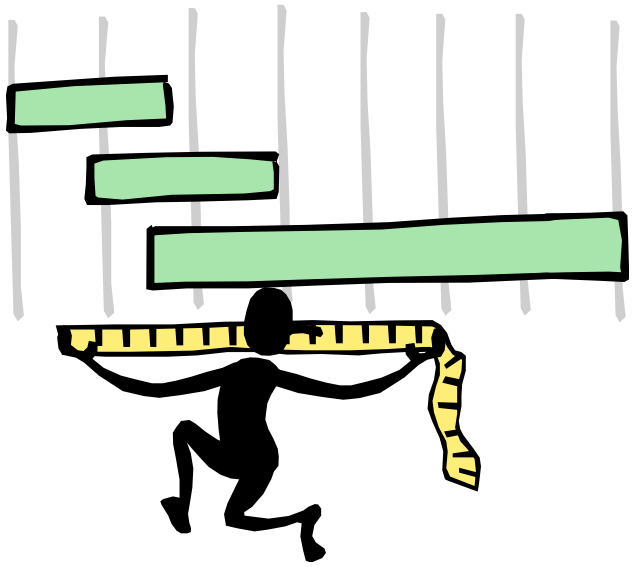
- Detection is performed locally by each node using spatial and temporal correlations.
- **Spatial outlier detection** compares the reported metrics received from each node in the set of probed nodes (C1 and C2).
- **Temporal outlier detection** examines the consistency in the metrics received from an individual probed node over time (C2 and C3).



Outlier: data point that is significantly different from the rest of the data in the observation space based on a measure of **distance**

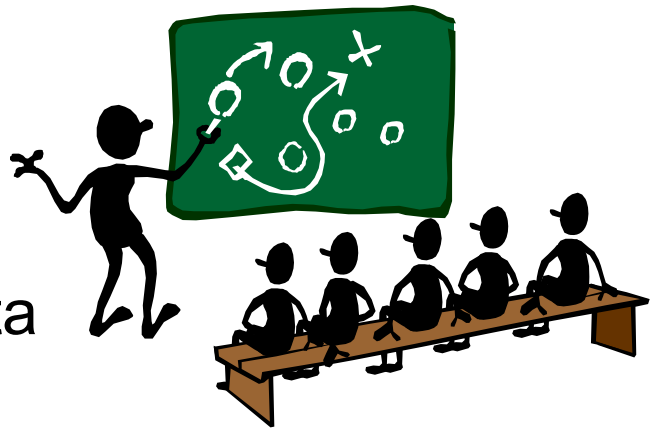
Mahalanobis Distance

- Good at detecting outliers with multiple attributes [LU04].
- Attributes with high variance receive less weight than components with low variance
- Good for applications where there is a dependency between the attributes



Spatial Outlier Detection

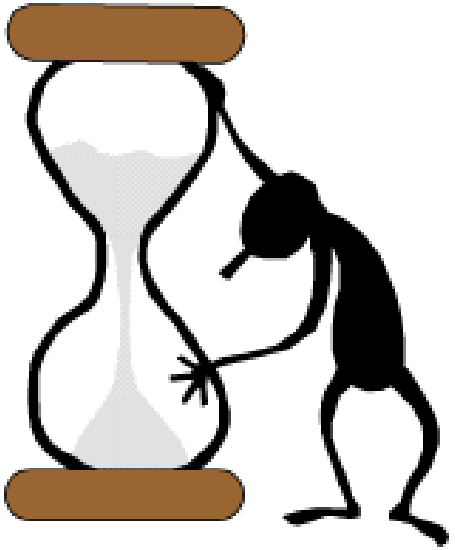
- Feature vector: bandwidth, latency, and RTT
- Performed during each probing period
- Observation tuples are used to compute the centroid of the data set
- Compare how far the observation tuple for each node is away from the centroid.



Spatial outlier detection compares the reported metrics received from each node in the set of probed nodes

Temporal Outlier Detection

- Temporal centroid: mean, standard deviation, and sample count associated with the observation tuple for each of the peers.
- Nodes do not need to maintain all history, centroid is incrementally updated with observations received during each probe cycle.



Temporal outlier detection compares the metrics received from an individual probed node over time

Putting It All Together

- Check the historical centroid:
  - if there are many temporal outliers then NO adaptation occurs during this probe cycle
  - If not, continue.

- Rank the peer nodes according to their spatial outlier distance from the centroid. Traverse nodes from closest to farthest from the centroid.
- Node closest to the centroid and has passed the utility function is chosen as the new parent.

Threshold Selection

- Spatial outlier detection: mathematically derived as in [spatialthreshold86], adjusted then experimentally
- Temporal outlier detection: set to 3.0 for to allow each of the three features to vary within one standard deviation from their temporally developed mean [kewang2004payl]

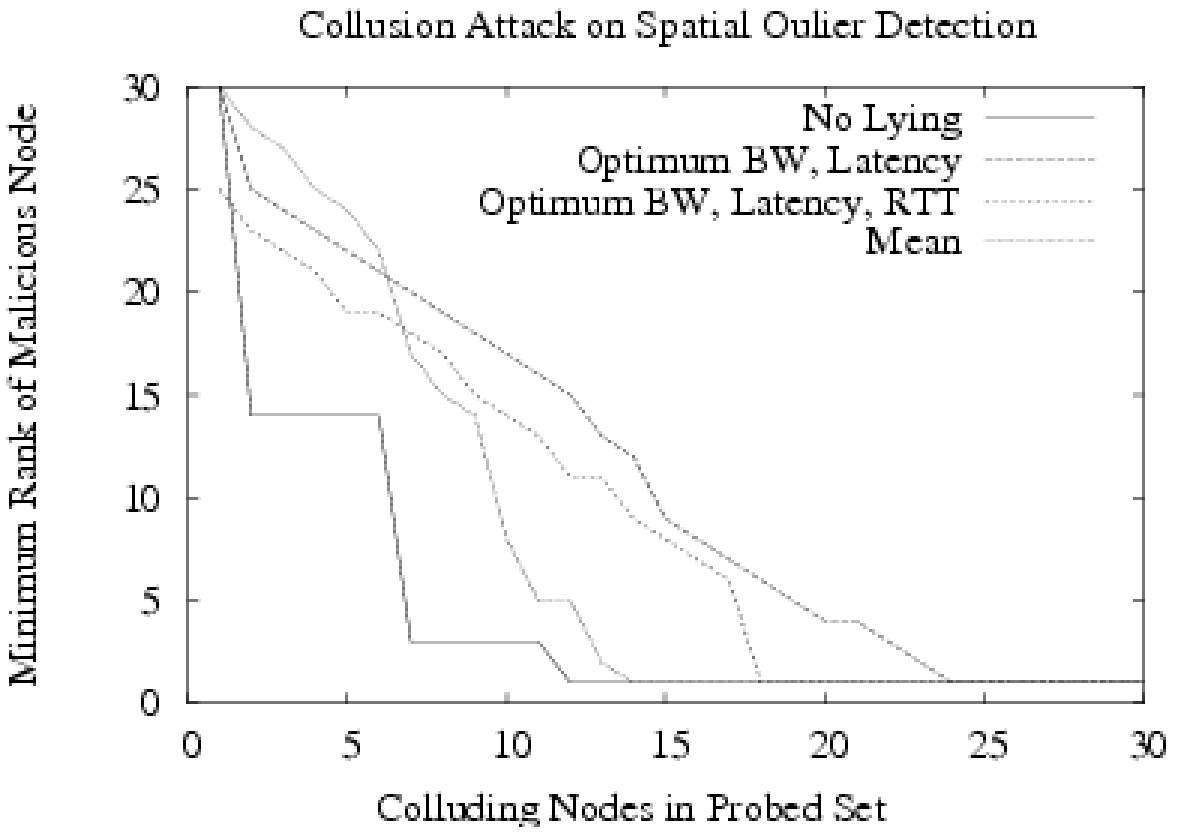
Effectiveness of Outlier Detection

- 100 nodes, over 60 minutes, 30% malicious nodes

Experiment	Changes to Malicious Parent	Total Parent Changes
No Lying	5	833
Lying	172	1032
Spatial	70	800
Spatial/Temporal	35	604

Improves stability and reduces the number of malicious changes (bandwidth did not change, with less changes)

Resilience to Coalition of Attackers



Overhead

- No additional communication introduced by the outlier detection: uses the same observation space as the utility function.
- Storage: each node will additionally maintain the mean, standard deviation, and sample count associated with the observation tuple within the routing table entry for each of the peers.

Isolating Malicious Nodes

- Neutralize malicious nodes once detected
  - Improves performance
  - Outlier detection does not “learn” malicious behavior
- Two-pronged approach
  - Local suspects list for quick response
  - Global black list created from shared information

Local Response

- Every node creates a suspicion value based for each neighbor based on how far it was from the spatial and temporal centroids
- Suspect list is gossiped to local neighbors
- Nodes are biased against choosing suspect list members as parents
- Once a node reaches a threshold suspicion value, it is reported to the source
- Good behavior rewarded (compensates also for transient network conditions)

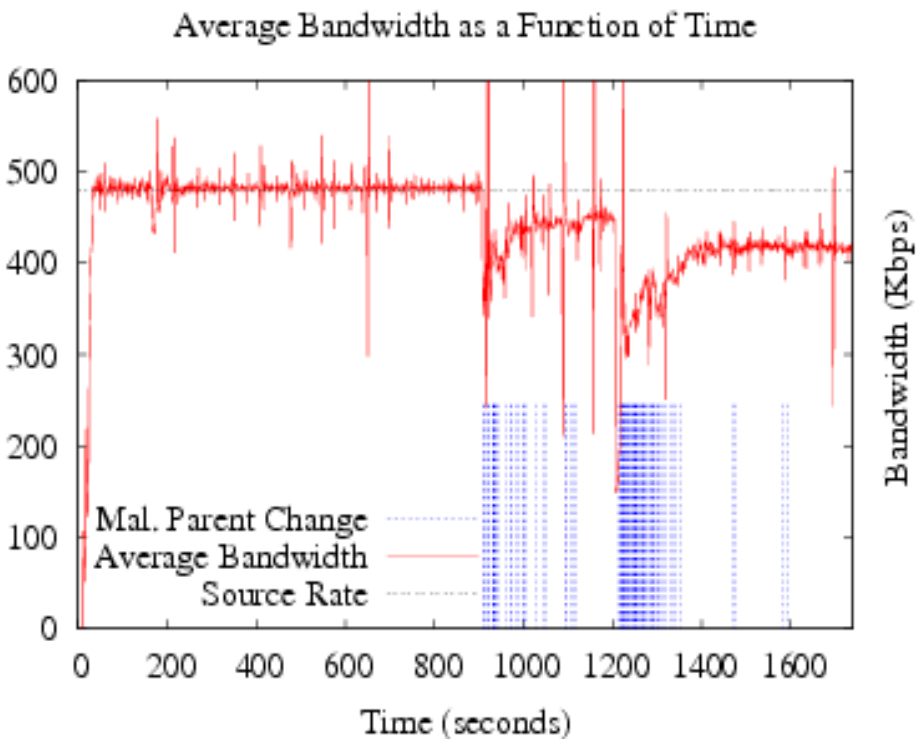
Global Response

- Source aggregates local suspect list into global view of trust
- Adaptation of the EigenTrust [KSG03] reputation system
- Trusted source allows for quick convergence and minimal computation
- Nodes falling below a threshold are placed on a global black list which is periodically disseminated to all nodes

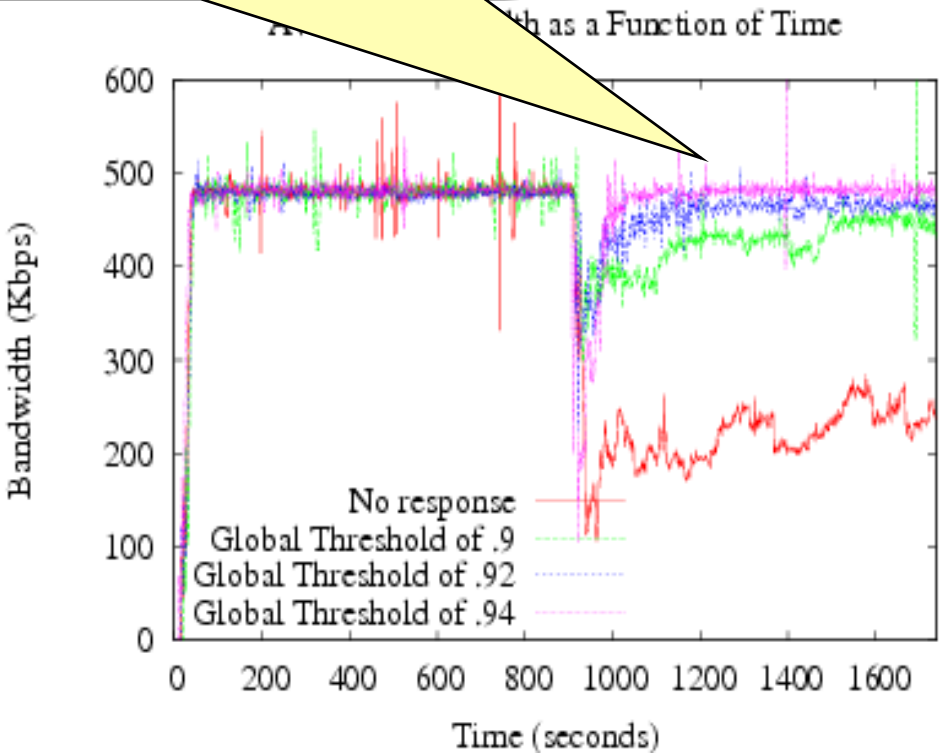
Effectiveness of Response Mechanism

- 100 nodes over 60 minutes, 30% malicious nodes

Bandwidth returns to value before attack

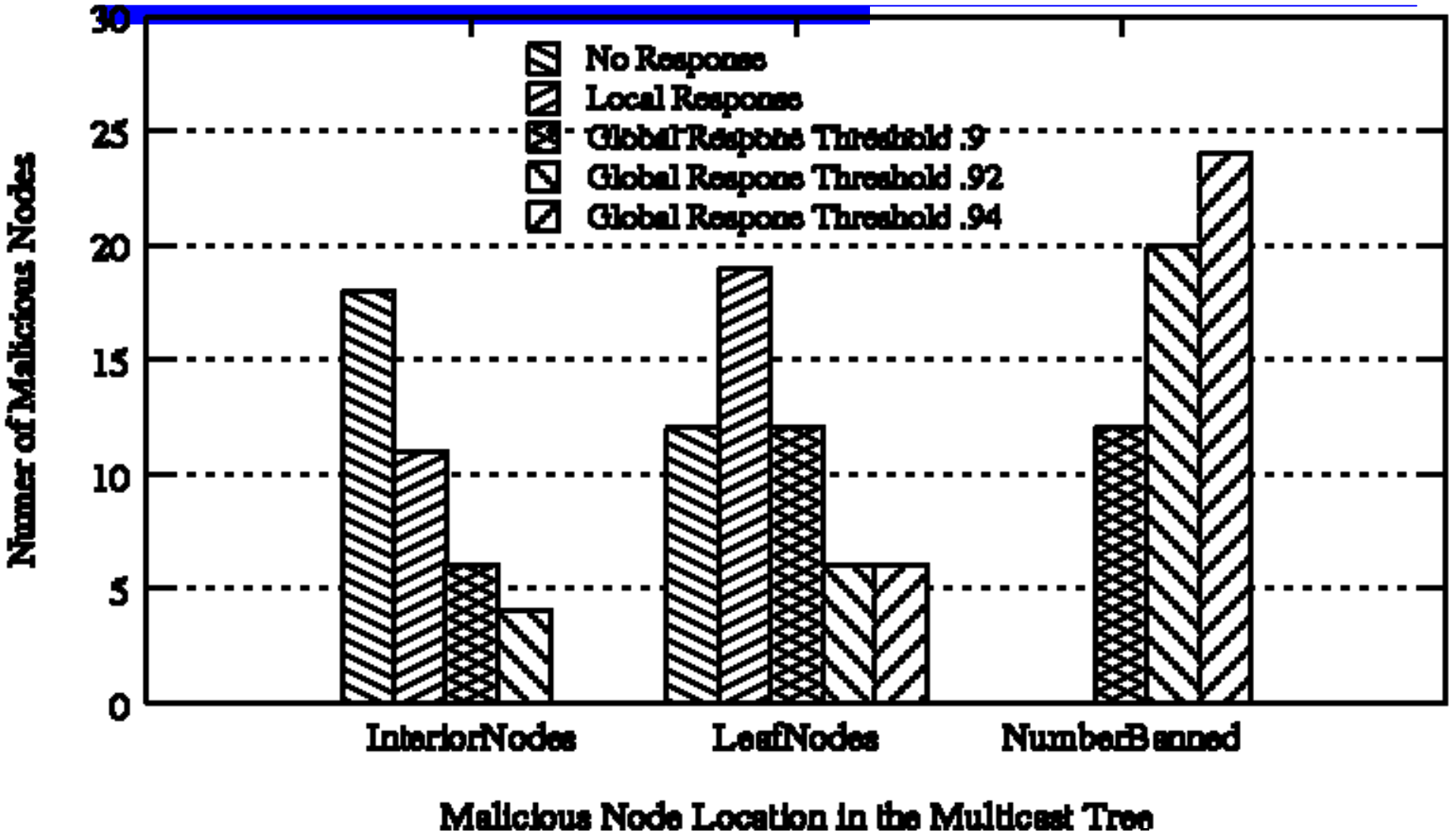


Local response only



Local and Global response

Malicious Nodes Pushed as Leaves or Banned



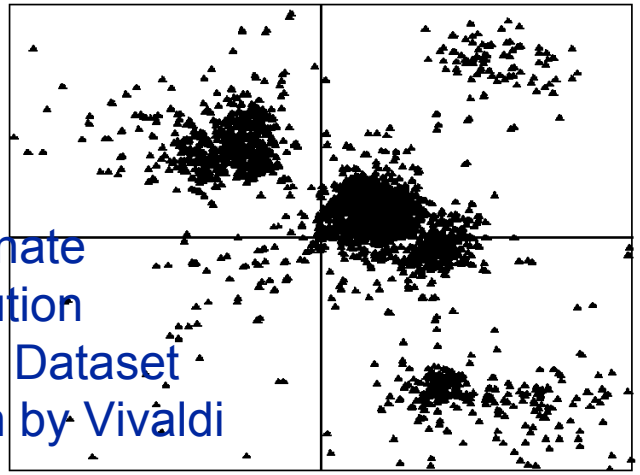
Outline

- Introduction.
- System and attacker model
- Demonstrate attacks
- Discuss solution space
  - Prevent poor adaptations
  - Isolating malicious nodes
- **Virtual coordinate systems**
- Conclusion

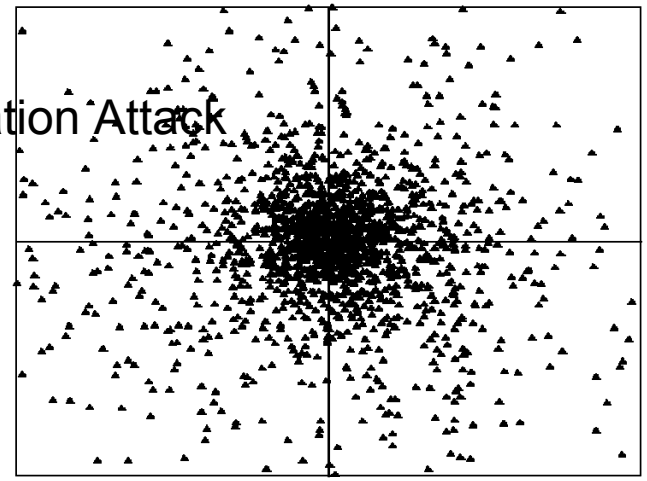
QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

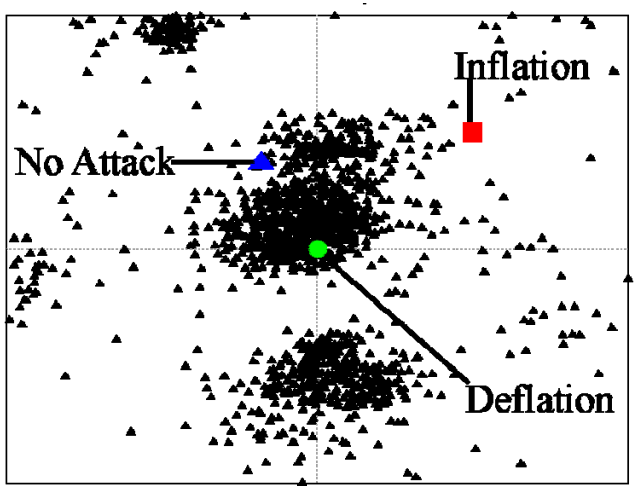
Decentralized Virtual Coordinates

Coordinate distribution of King Dataset chosen by Vivaldi



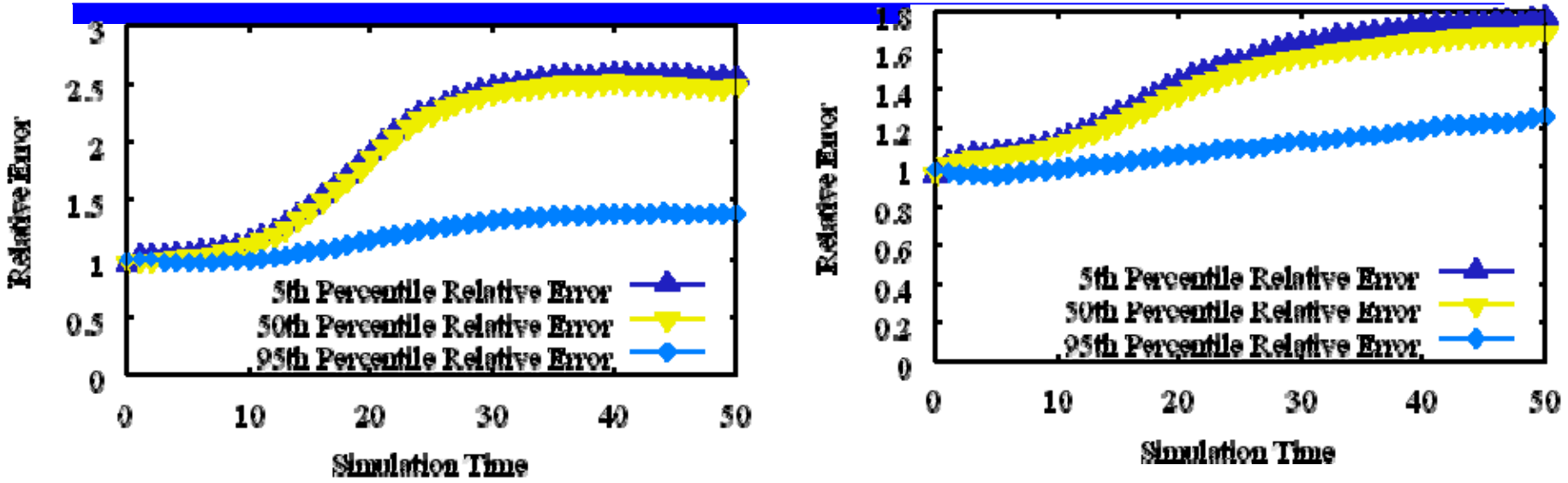
Oscillation Attack





Attack	Pred. Error
None	10
Inflation	60
Deflation	70
W/defense	11

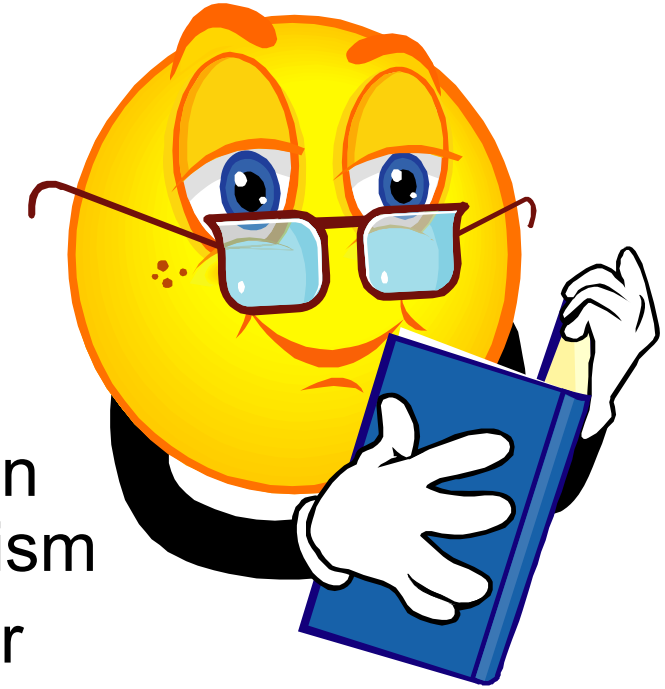
False Positive Rate and Relative Error



Mal. Nodes/Spatial Outlier Threshold	1.25	1.5	1.75	2.00
0%	28%, 16ms	21%, 16ms	17%,16ms	13%, 16ms
10%	17%, 17ms	13%,18ms	10%,19ms	5%, 20ms
20%	21%, 18ms	15%, 21ms	7%, 23ms	6%, 26ms
30%	27%, 20ms	11%,22ms	10%, 33ms	9%, 36ms

Summary

- Vulnerability of adaptation mechanisms in unstructured multicast overlay networks
- Comprehensive solution for addressing the attacks
- Spatial-temporal outlier detection an effective prevention mechanism
- Similar approach works for other distributed services such as decentralized virtual coordinate systems



What Next?

- Study mesh-based peer-to-peer streaming
- Solutions like the ones we describe don’t work because many of them are not based on measurements
- Key observation: attacker presence changes network topology

Contact Information and References

EMAIL: crisn@cs.purdue.edu

URL: http://www.cerias.purdue.edu/homes/crisn

- *A Framework for Mitigating Attacks Against Measurement-Based Adaptation Mechanisms in Unstructured Multicast Overlay Networks.* A. Walters, D. Zage and C. Nita-Rotaru. To appear in IEEE/ACM Transactions on Networking, 2007 (Feb. 2009).
- *Mitigating Attacks Against Measurement-Based Adaptation Mechanisms in Unstructured Multicast Overlay Networks.* A. Walters, D. Zage and C. Nita-Rotaru, ICNP 2006.
- *On the Accuracy of Decentralized Network Coordinate Systems in Adversarial Networks*. D. Zage and C. Nita-Rotaru. CCS 2007.
- Won't You Be My Neighbor: Neighbor Selection Attacks in Mesh-Based Peer-to-Peer Systems. J. Siebert, D. Zage and C. Nita-Rotaru, Under Submission.