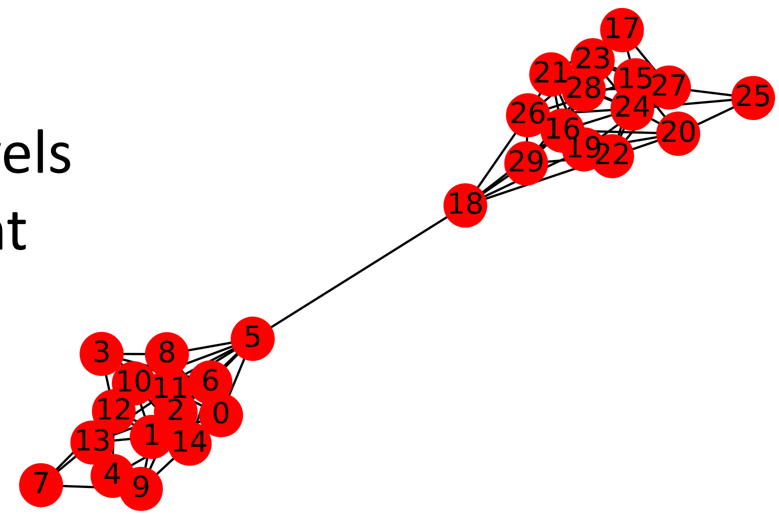
Networks

Networks of Networks

- Many networked systems rely on multiple interconnected networks; even the internet is an example of such a **network of networks (NoNs)**
- NoNs are characterized by higher levels of connectivity within the component networks than the connectivity between networks
- Links interconnecting subnetworks are called *structural bottlenecks*



UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

- Bottlenecks are easy targets for network disruption:
  - Denial-of-Service attacks and routing loops for adversary that can inject packets into network
  - Jamming for wireless links

- From defensive standpoint, if we know bottlenecks, we can:
  - Use traffic shaping to prevent adversary from detecting bottlenecks
  - Use physical formation control to create additional connections to avoid single points of failure

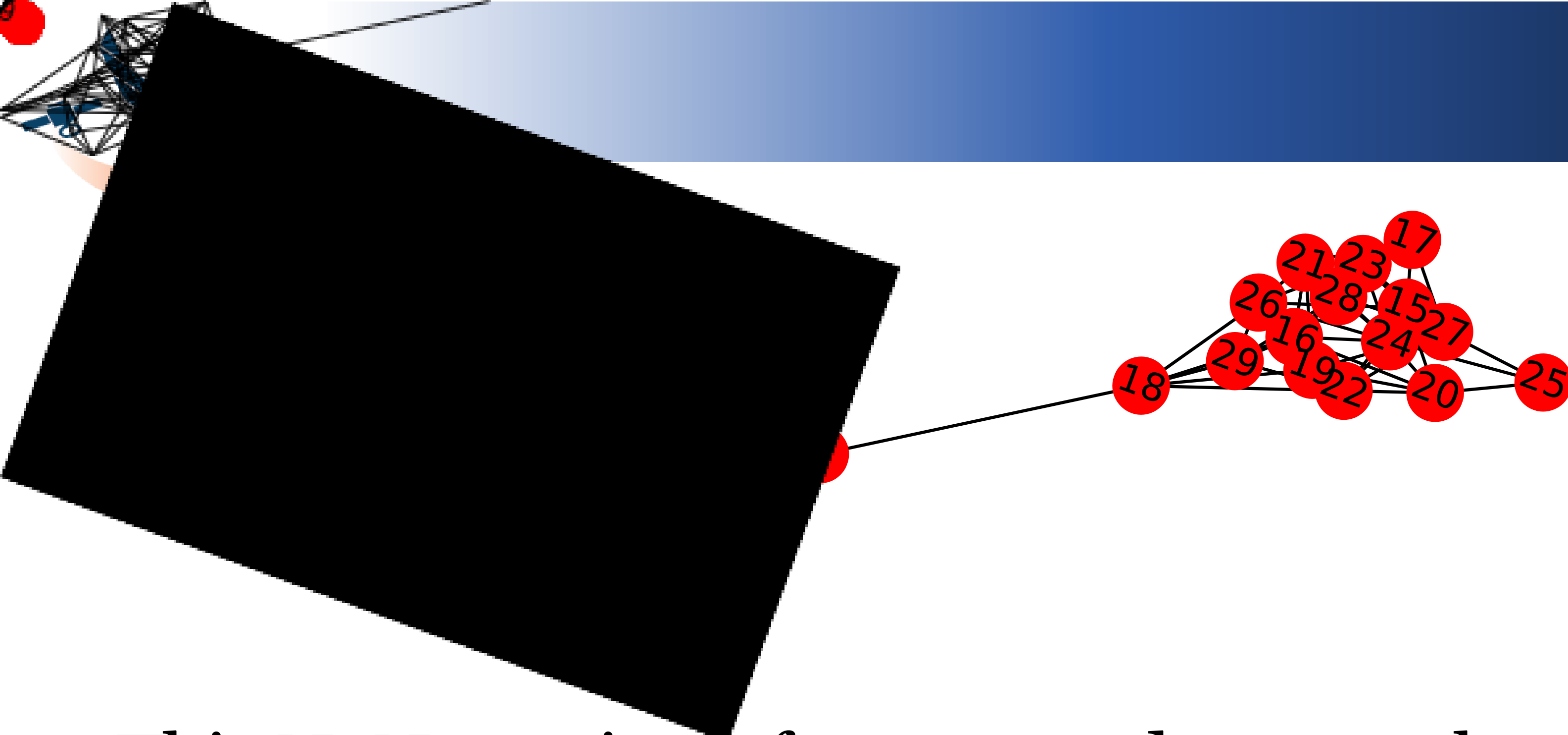
Bottleneck Detection

- Significant body of work on detecting bottlenecks in networks **when graph of network is known**
- Another large body of literature on **network tomography – requires lots of information**

- **Goal:** develop techniques to detect bottlenecks
  - without knowing network structure
  - without having to depend on complicated network protocols and extensive data collection to infer full network structure

System Model

- Consider a NoN with two subnetworks connected via a single link: the bottleneck
- We model the network as a graph $G = (V, E)$ that is a *single bottleneck graph* if:
  - *G is connected, and*
  - *G contains a (unique) bridge $b \in E$ that cuts G into two bridgeless, connected components.*
- If the bridge is $b = (u, v)$, then we call b the *bottleneck*, and u and v *bottleneck nodes*



- This NoN consists of two 15-node networks
- The bottleneck is (5, 18)
- We refer to the component networks as G_u and G_v
- In this case, G_u contains nodes $0 \rightarrow 14$ and G_v contains nodes $15 \rightarrow 29$ (or vice versa)

Observers

- We wish to construct an algorithm to detect the bottleneck by an agent that can monitor only **end-to-end delays** for traffic from one or more *observer source nodes (OSNs)*
- We assume:

1) The observer knows that G is a single-bottleneck graph. He also knows all the nodes in G, i.e., the set V. Other than these two piece of information, the observer has no further information about the topology of G. In particular, he does not know u, v, E, V_u, or V_v.

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

2) The observer is able to send packets from one or more nodes, called *observer source nodes* (OSNs), to any other node in G, and measure the round-trip delay of each sent packet reaching the destination node and then the acknowledgment arriving back at the OSN.

3) If the observer uses multiple OSNs, then all the OSNs are located within the same component network (either G_u or G_v) of G, but the observer does not know to which component the OSNs belongs.

4) The observer does not know the underlying traffic pattern in G but does know that the networking delay at a node is dependent on the amount of traffic passing through that node.

Delay model

- We assume min-hop routing is used between every pair of nodes
- We assume queueing delays >> physical transmission times, so the end-to-end delay $\approx$ sum of the queueing delays on the return route
- Average delay at each node is modeled as proportional to the number of min-hop routes that pass through that node —proportional to the **betweenness centrality**
- Assume delays are exponential random variables and independent among nodes and between forward and return paths

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

- Three Step Algorithm:
  1. *Preprocessing:*

     Let $S \subseteq V$ be the set of OSNs

     Let $D_{\{s,v\}}(i)$ be the ith measured roundtrip delay between OSN s and node $v \in V \setminus \{s\}$

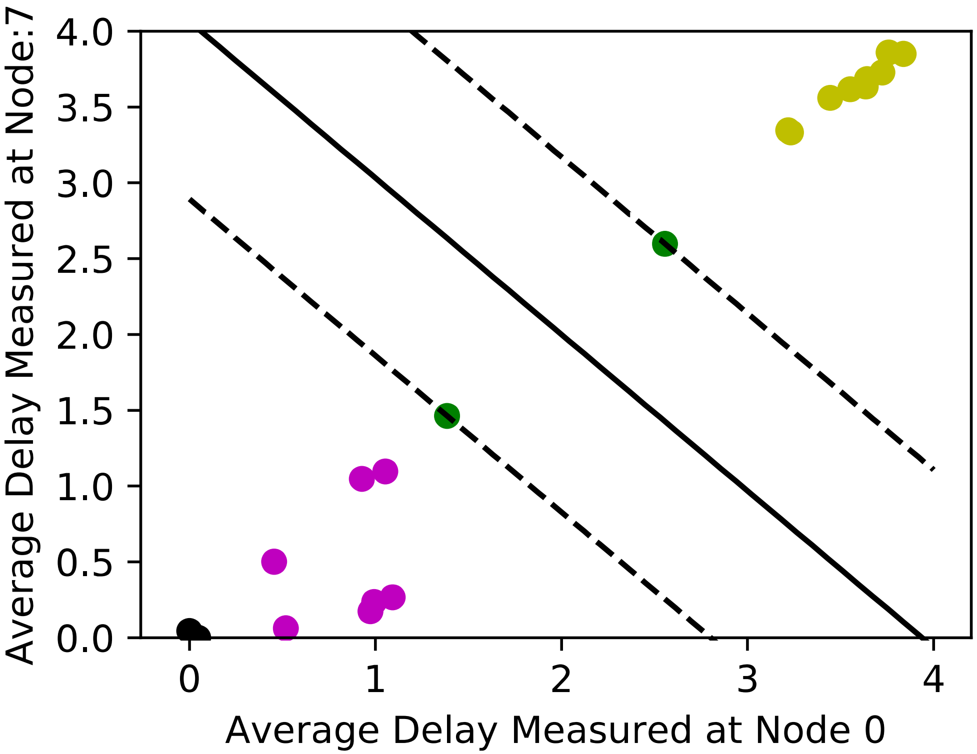
     Then calculate the average round-trip delays

$$D_{s,v} = \frac{1}{N_{s,v}} \sum_{i=1}^{N_{s,v}} D_{s,v}(i)$$

     For each $v \in V$, concatenate the $D_{\{s,v\}}$ to form a $|S|$ *average delay vector*

2. Clustering of Delay Measurements:

The delay vectors are partitioned into two clusters using *K*-means algorithm



3. *Bottleneck Node Identification:*

The clustering provides binary class labels for the nodes

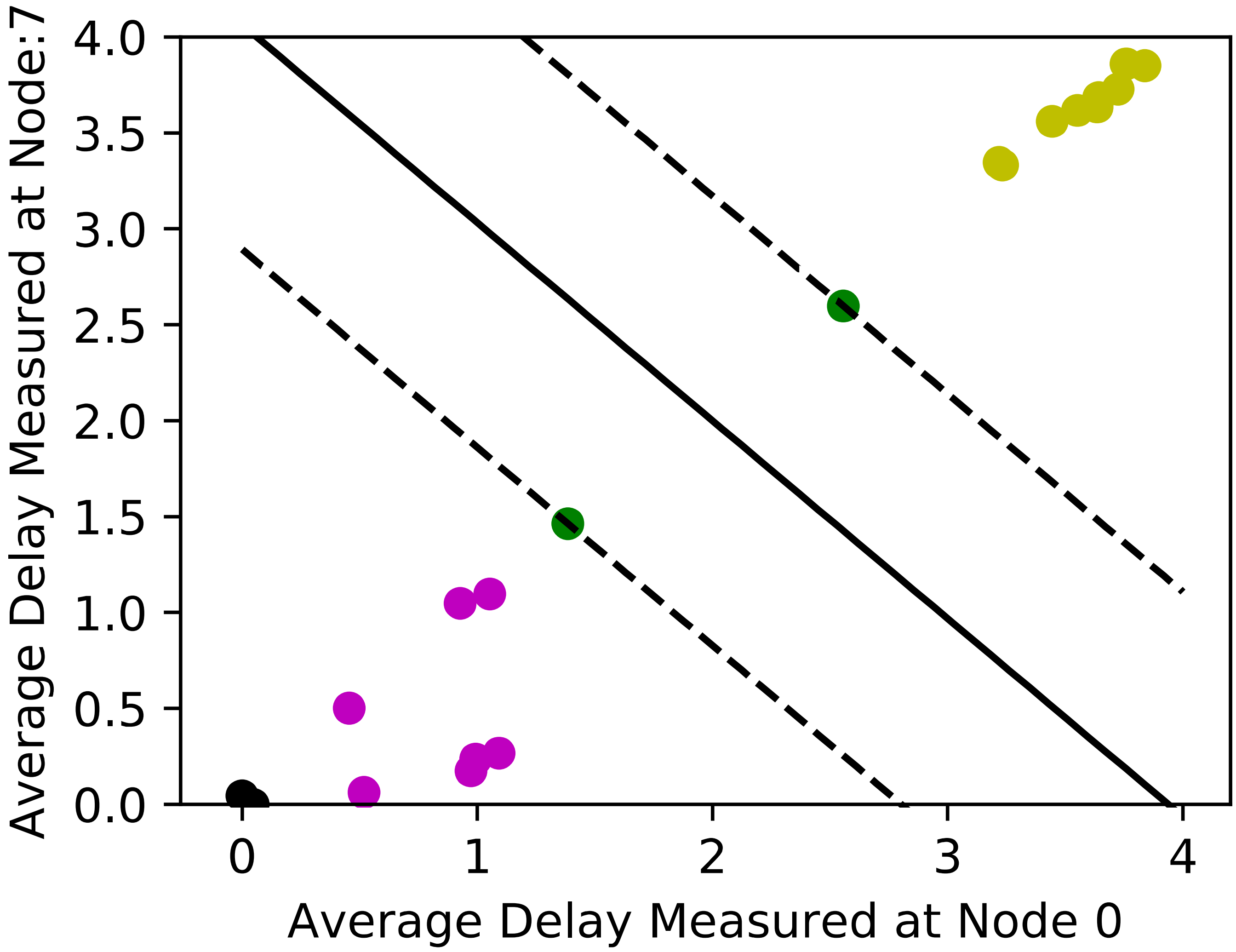
$$y_v = \begin{cases} 1 & \text{if } D_v \in \mathcal{D}_1 \\ -1 & \text{if } D_v \in \mathcal{D}_2 \end{cases}$$

We apply support vector machine (SVM) to find optimal margin hyperplanes to partition the labeled data, $\{(y_v, D_v)\}_{v \in V}$

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + c \sum_{v \in V} \xi_v$$

$$\text{subject to} \quad y_v \left(w^T \phi(D_v) + b \right) \geq 1 - \xi_v,$$

$$\xi_v \geq 0, \quad \text{for each } v \in V$$


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Average Delay Measured at Node:7
0
1
2
3
4
Average Delay Measured at Node 0

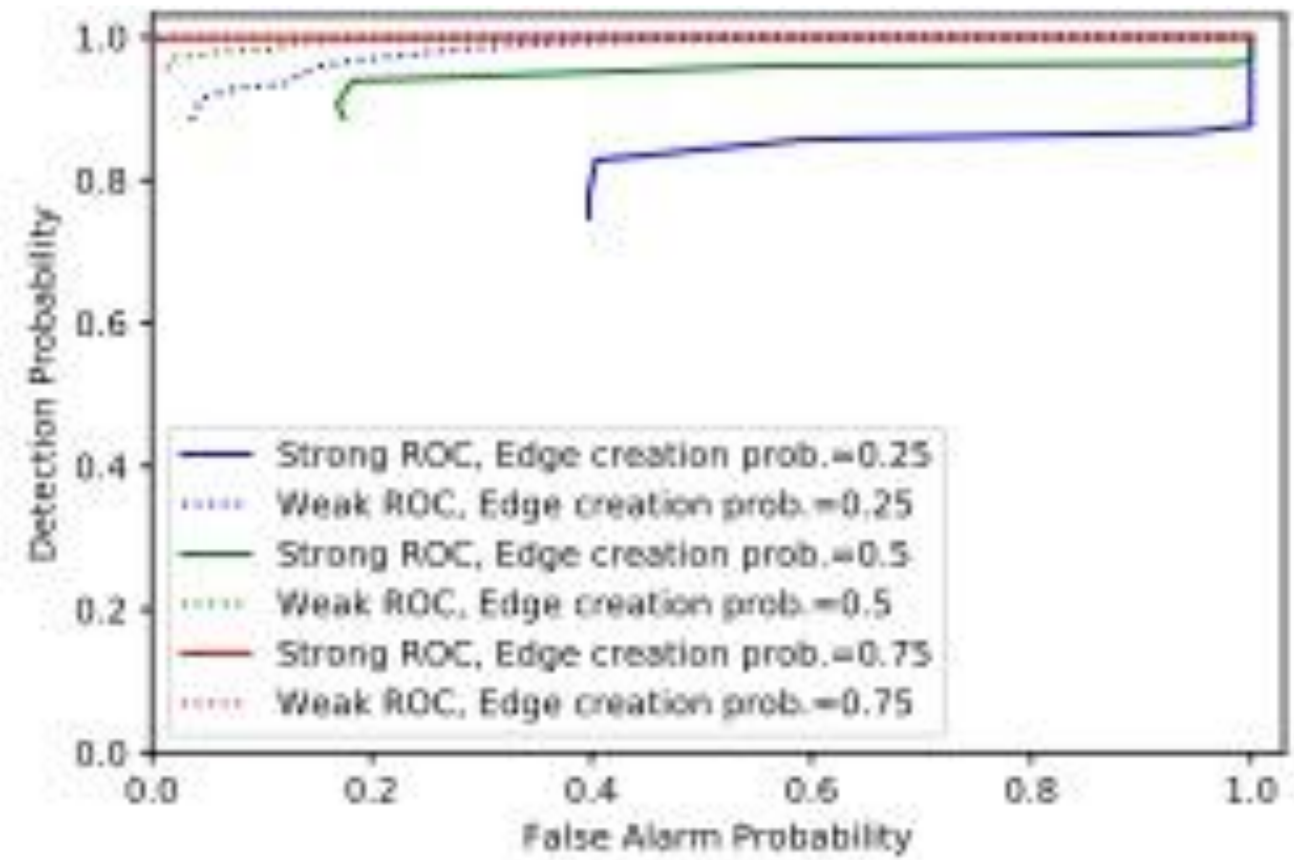
Simulation Results

- Formed random single-bottleneck networks by finding two connected, bridgeless Erdős-Rényi component graphs and adding a bottleneck node between random nodes selected from each component graph
- OSNs are chosen by randomly selecting a component network and randomly choosing nodes to act as the OSNs
- Delays are generated by randomly selecting destination nodes, finding a min-hop route, and generating independent exponential random variables along the return route

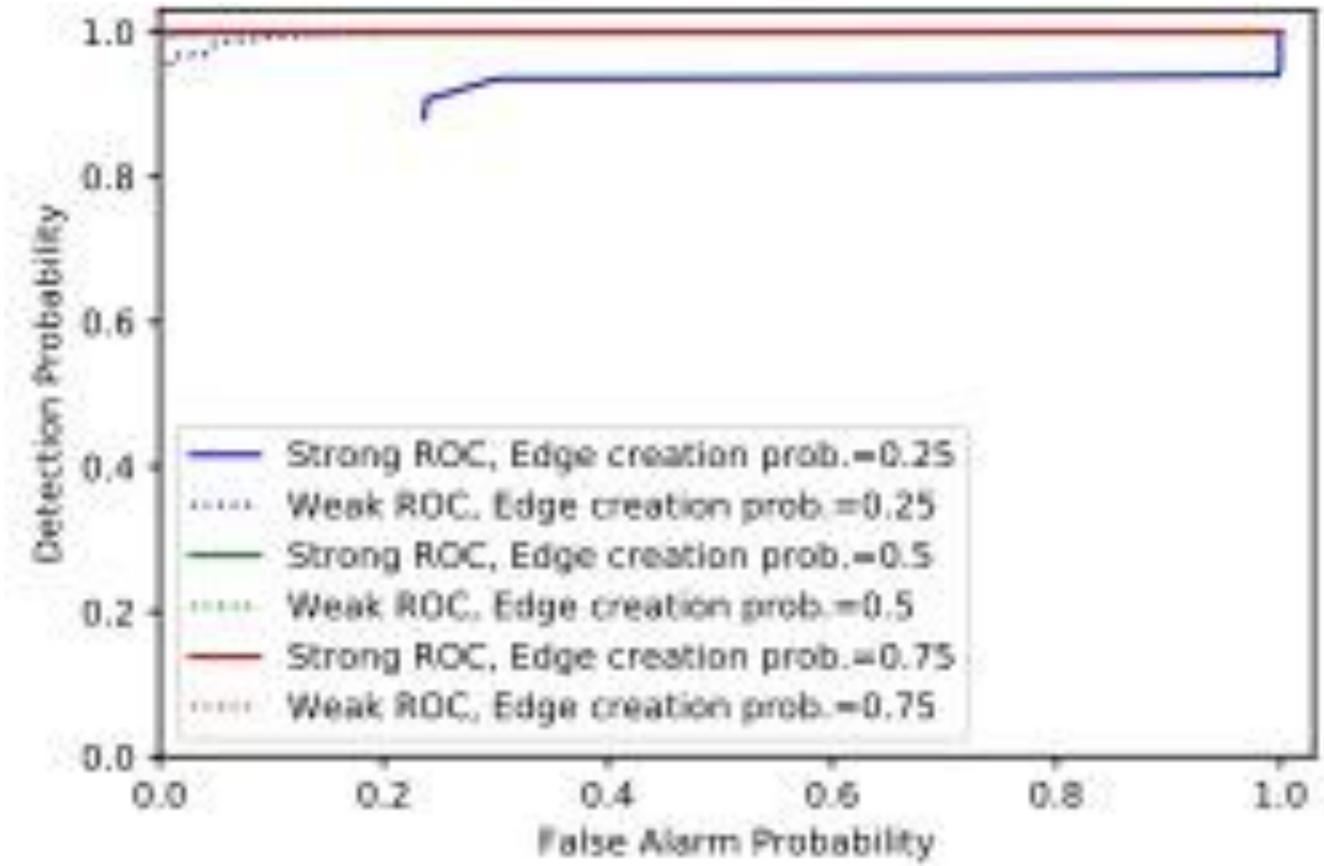
Metrics

- **Strong metric:**
  - *Detection* requires **both** bottleneck nodes be identified
- **Weak metric:**
  - *Detection* if **any** bottleneck node identified
- *False alarm* if any non-bottleneck node identified as bottleneck node

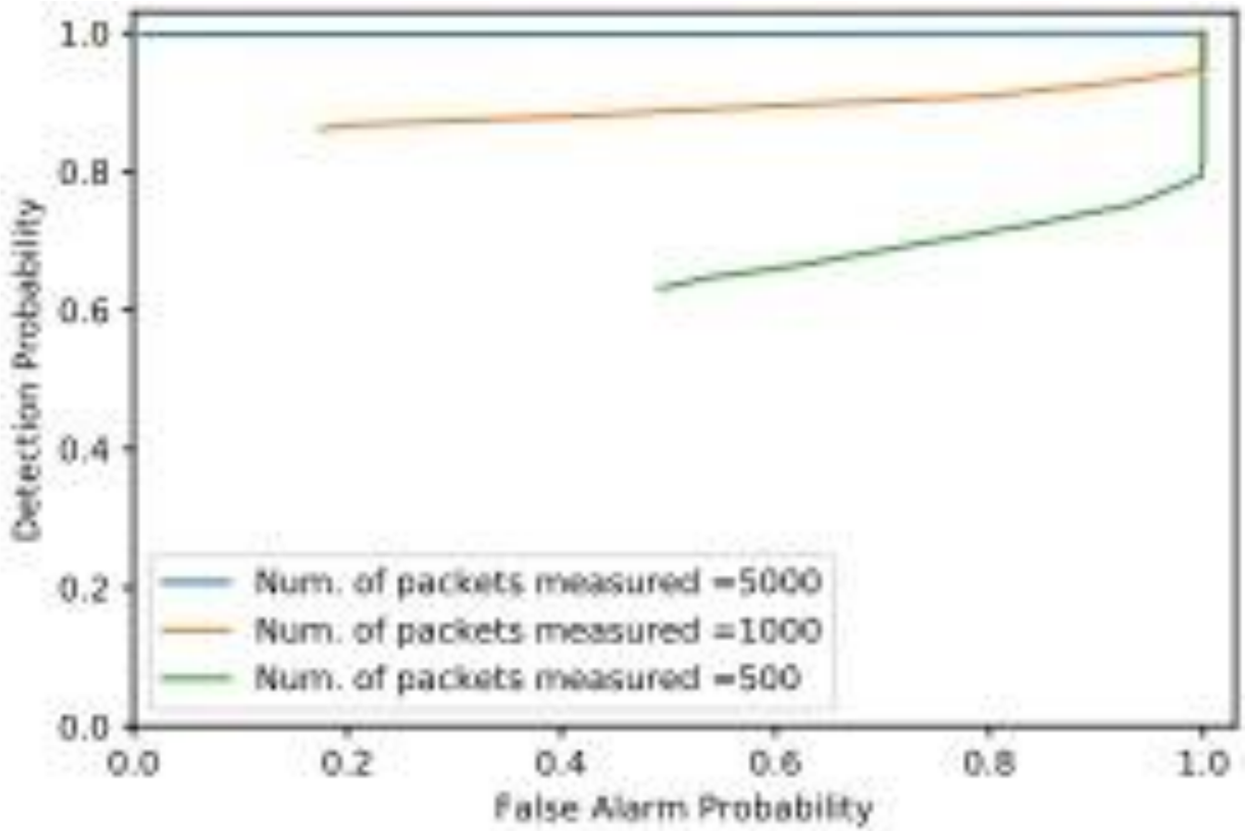
- 10-node component networks with 2 OSNs, 2000 end-to-end delay measurements



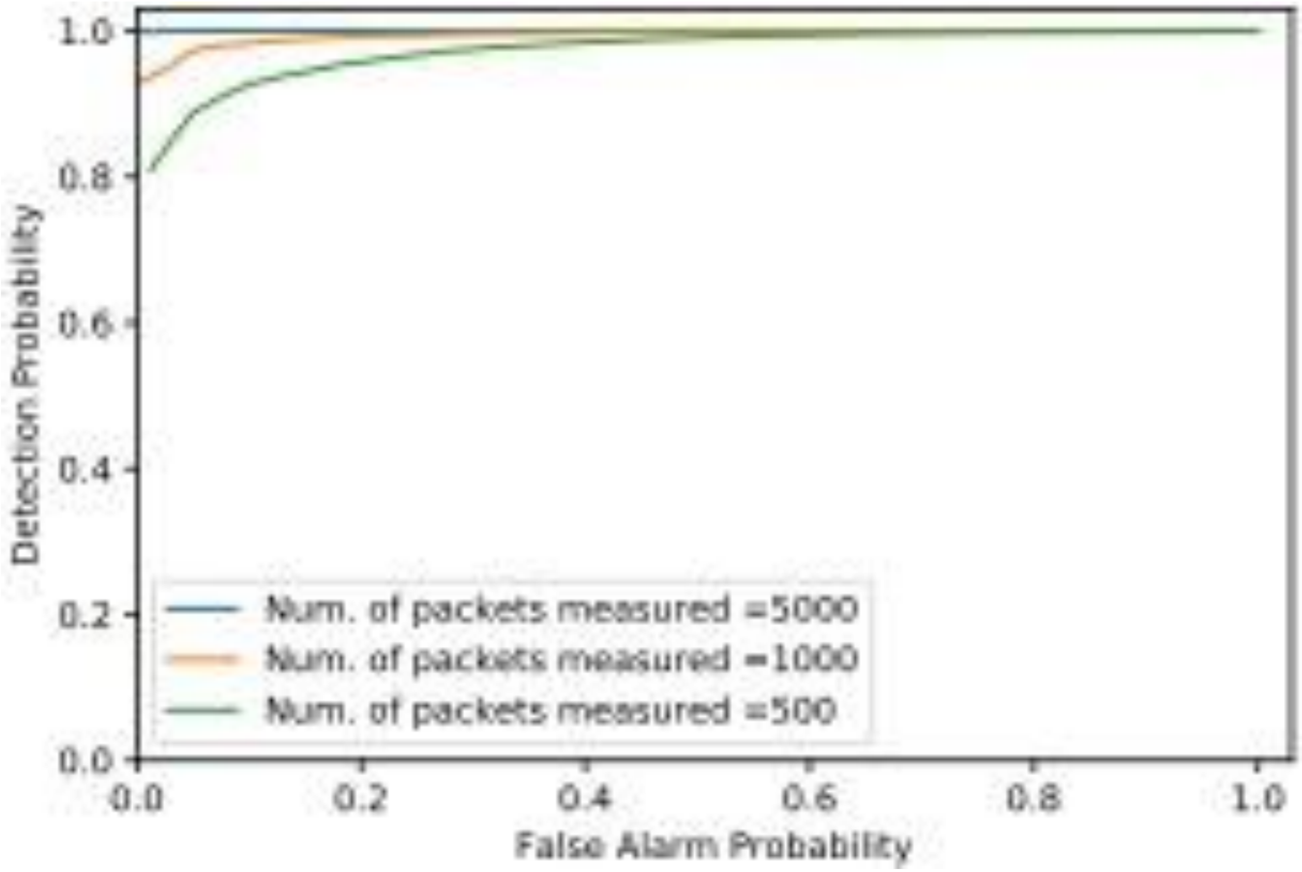
- 25-node component networks with 2 OSNs, 5000 end-to-end delay measurements



- Effect of number of delay measurements – **strong metric**, 25-node components, $p = 0.5$



- Effect of number of delay measurements – **weak metric**, 25-node components, $p = 0.5$



Future Work

- Multiple bottlenecks
- Unknown number of bottlenecks
- > 2 component networks
- Unknown number of component networks
- Better utilizing full delay measurement data instead of averaging it
- Better exploiting spatial information in delay measurements: bottleneck nodes are more likely to lie close to line connecting centroids of clusters

Invitation

- DARPA Spectrum Collaboration Challenge Championship Event
- October 23 at Mobile World Congress in LA
- Free to public

SPECTRUM
COLLABORATION
CHALLENGE

UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ