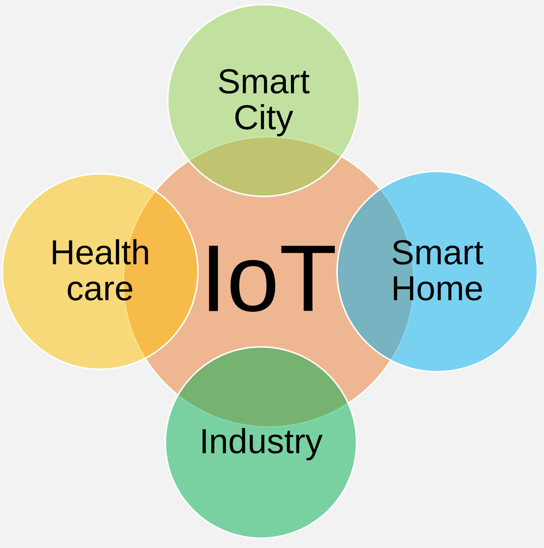
A Distributed & Lightweight Framework to Secure IoT Networks Against Network Layer Attacks

NC STATE UNIVERSITY

Raghav H.V, Prasesh A, Shakir M, M.Shahzad
Dept. of Computer Science, NC State University, Raleigh, NC, USA

INTRODUCTION

It is critical to secure the rapidly proliferating IoT Networks (IoTNs)



- IoTNs expose OSI layer-specific attack surfaces
- Need attack mitigation strategies customized to attack anatomies in each layer
- In this work, we focus on attacks at the network layer(NL)

RESEARCH QUESTION

Can we develop a distributed, light-weight, NL protocol independent defense framework?

Currently, there is no work that proposes an attack mitigation approach that can concurrently perform:

- Distributed NL attack detection and mitigation
- Generalized attack mitigation,
- Topology independent attack mitigation, and
- Simultaneous attack detection, localization & mitigation

APPROACH

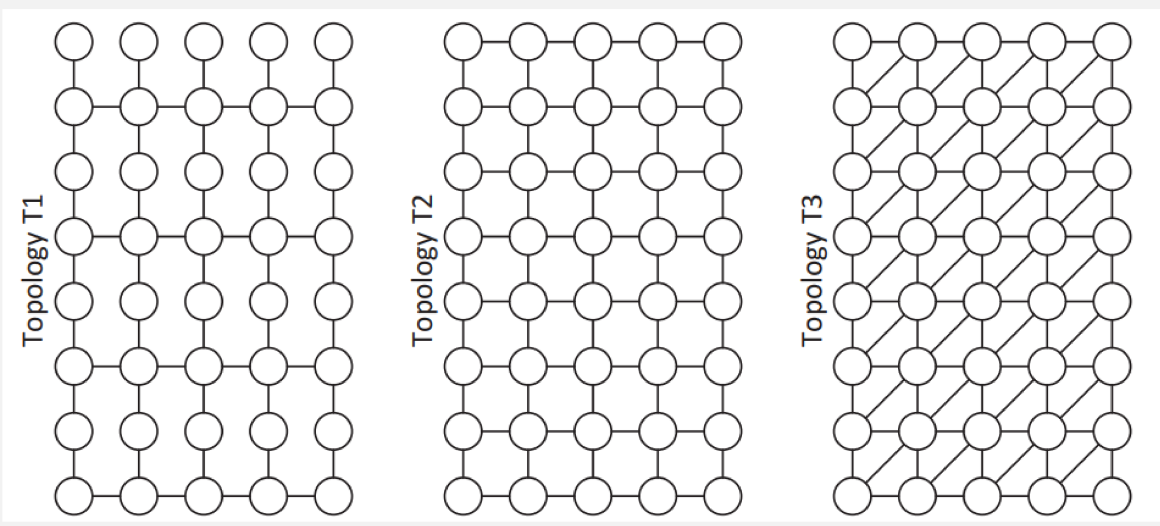
A load-balanced distributed attack monitoring and response algorithm based on performance metrics

1. Develop an exploratory study to derive key insights across NL attack types and topologies
2. Assume a threat model:
   i. Attacker can compromise nodes i.e take control of nodes in an IoTN
   ii. Attacker can forge performance metrics on nodes to evade detection
3. Create a dynamic, self-elected distributed network of monitoring nodes that:
   i. Detects arbitrary NL attacks
   ii. Locates compromised nodes
   iii. Mitigates attacks by automatic isolation of compromised nodes

EXPLORATORY STUDY

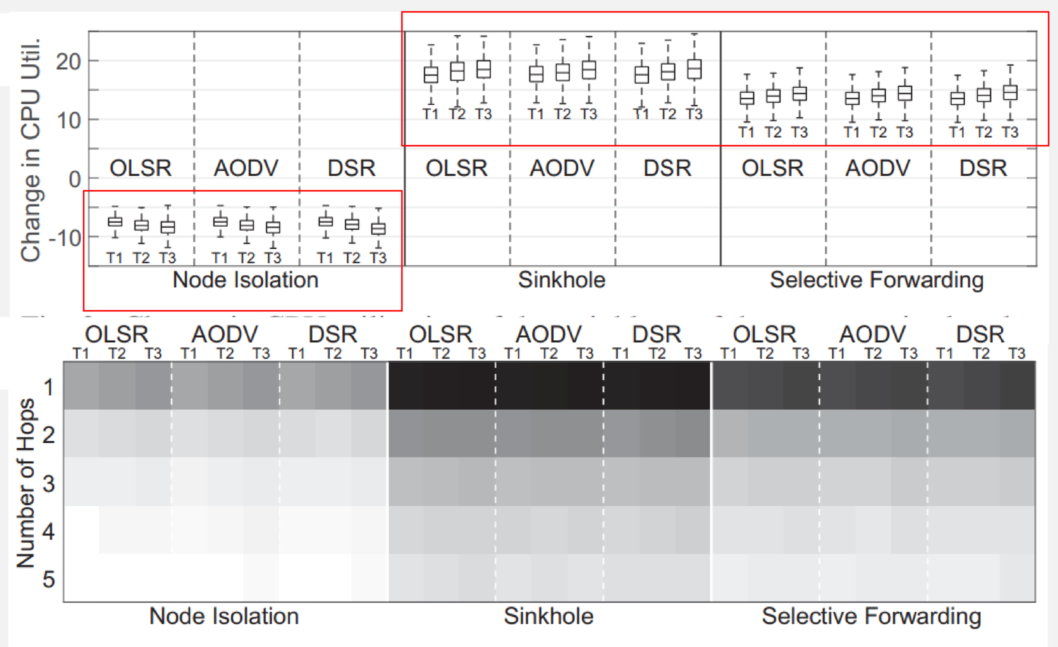
Study the performance metrics before and during NL attacks to observe patterns that help detect and simultaneously locate compromised nodes

1. Test Topology (T):
   40 Raspberry Pi's in three topologies T1, T2, T3 Connected by Ad-Hoc WiFi
2. NL Protocols (P):
   OLSR, AODV, and DSR
3. NL Attacks (A) :
   Sinkhole, Selective Forwarding, Node Isolation
4. Performance Metrics:
   Avg CPU Utilization, No. of Pkts Forwarded, No. of packets sent and recieved, Routing overhead
5. Study 27 Combinations (T,P,A), 5-min attacks, 5 nodes



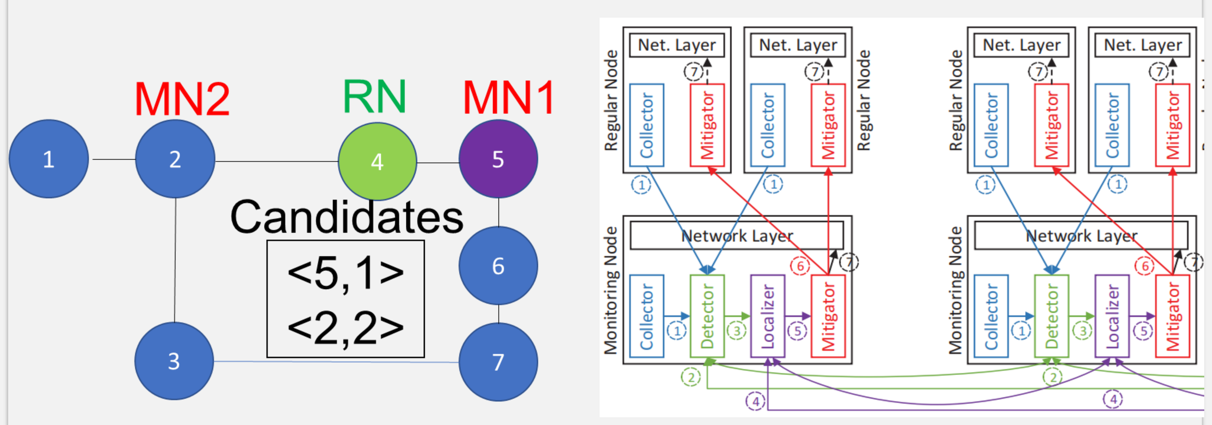
INSIGHTS

1) Considerable change in performance metrics of a compromised node's neighbors during an attack
2) Change in the performance metrics is significant for nodes a few hops from the compromised node



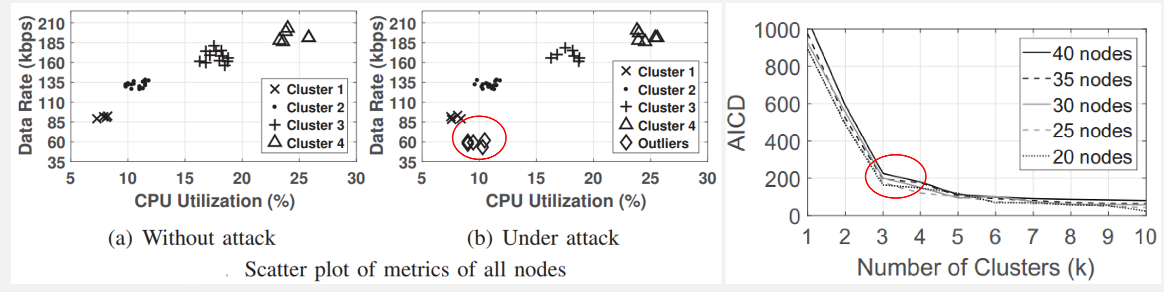
Absolute change in CPU util. of compromised nodes' n-hop neighbors

DISTRIBUTED ATTACK MONITORING



- Idea: An IoTN can be partitioned dynamically into monitoring nodes and regular nodes.by selecting monitoring nodes at regular intervals
- Method:
  (i) Each node selects its candidate from 1-hop neighbors based on no. of the candidates' 1-hop neighbors
  (ii) A different candidate is selected if a given candidate was already selected in a previous round
  (iii) All nodes send performance metrics from a collection module to their selected monitoring node
  (iv) A detector, locator and mitigator block at each monitor node run a performance metric based attack detection and mitigation scheme

ATTACK DETECTION



- Key Observations:
  1. Performance metrics at all nodes before an attack belong to distinct clusters
  2. The no-attack Aggregate Intra Cluster Distance (AICD) for K-means Clustering and Number of clusters(k) shows a knee in [3,6]
  3. Metrics at victim nodes change significantly, leading to outliers
- Method:

  1) Initialization phase after electing monitors
  (i) Aggregate metrics from other monitor nodes
  (ii) Use metrics of smallest k NL addresses as initial centroids of k clusters
  (iii) Determine k in [3,6]
  (iv) Collect metrics of all monitored nodes and cluster them until centroids are stable.
  (v) Save cluster label for all monitored nodes

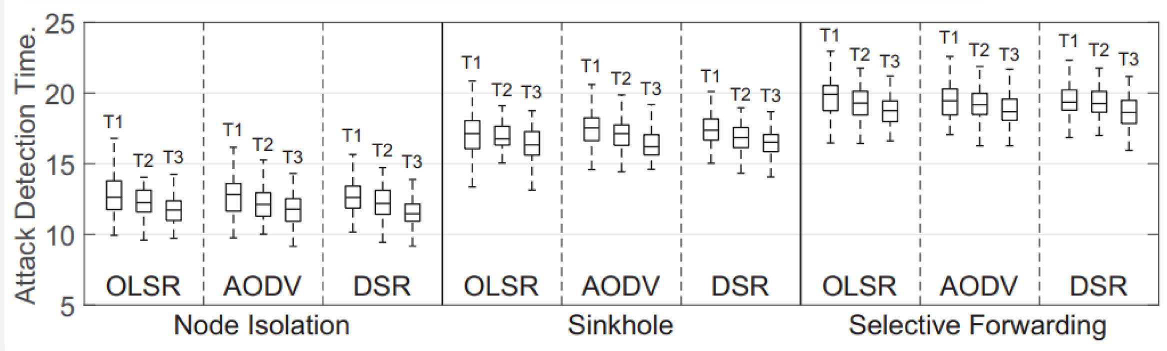
  2) Detection phase :
  (i) Check if the cluster label for the performance metrics of a node has changed.
  (ii) Check if the position of the metrics from the saved cluster center is above a threshold.
  (iii) If either are true, increment a distrust index value for the node. If not, decrement it.
  (iv) If distress index crosses above or below a threshold, notify own localizer block and the detector blocks of all other monitoring nodes.

ATTACK LOCALIZATION & MITIGATION

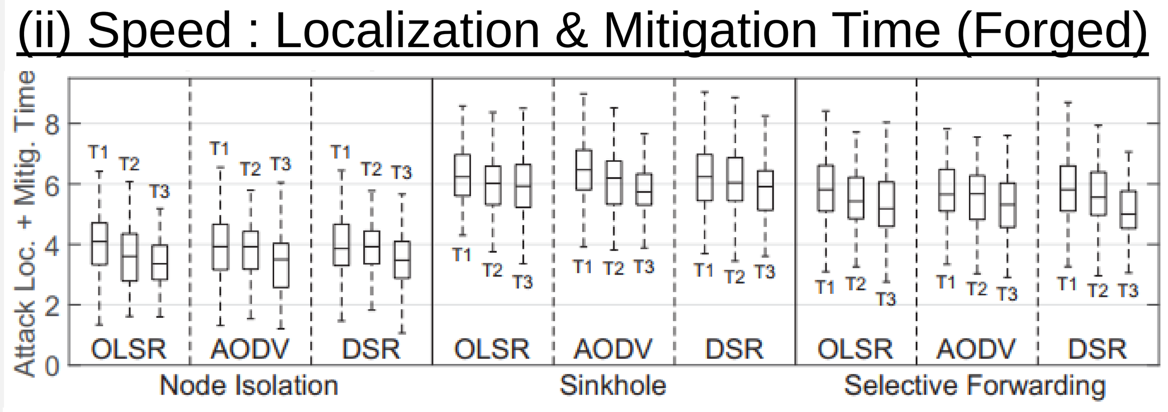
- Localization:
  - Idea: No need for a compromised node to report its metrics truthfully as its effects can be seen from its effect on local neighbors
  - Method:
  (i) Assign a malice score to a node at the localizer block by taking a weighted average of the number of its 1-, 2-, and 3-hop 'suspicious' neighbors.
  (ii) Aggregate malice scores from all other localizer blocks and inform the mitigator block of all malice scores.
- Mitigation:
  - Idea: If a malicious node is isolated,it's neighboring 'suspicious' nodes return to usual behavior
  - Method: Until the mitigation block keeps receiving malice scores:
  (i) Select a node with highest malice score
  (ii) Notify all mitigation blocks of the node's immediate neighbors to start isolation
  (iii) Notify the local NL to isolate the node.

PERFORMANCE EVALUATION

(i) Speed : Detection Time (Forged Metrics)

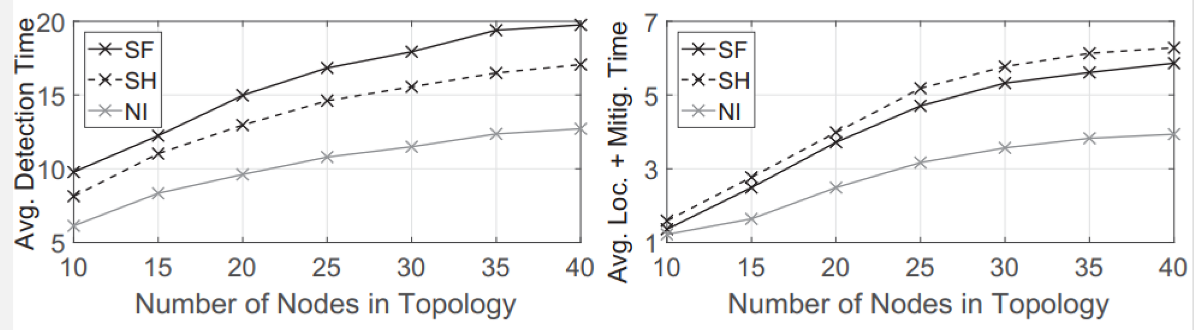


(ii) Speed : Localization & Mitigation Time (Forged)

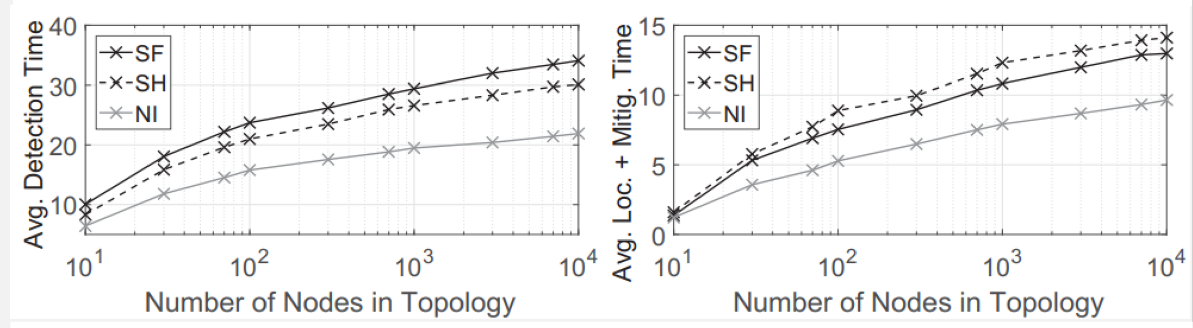


- Minimum detection time for NI and maximum for SF
- Detection times are highest for T1 and lowest for T3.
- Choice of NL protocol has no impact

(iii) Scalability: Impact of Topology Size (Real Testbed)



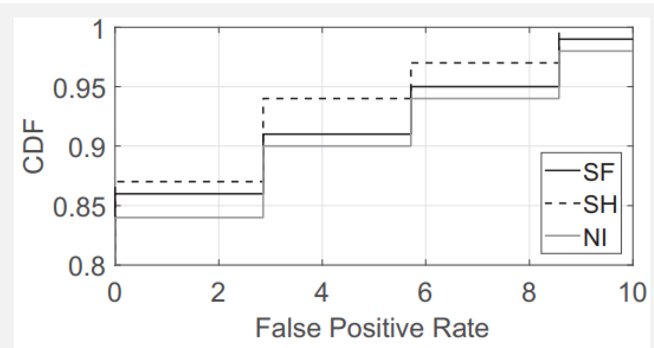
(iv) Scalability: Impact of Topology Size (Simulated Testbed in NS-3)



- Detection time increases only sub-linearly(Real)/ sub-logarithmically(NS3) as no. of nodes increases
- Detection time can be reduced by selecting a larger k at the expense of faster energy depletion of nodes.

(v) Error rates: False Positive and False Negative

- FNR of 0 in 100% of 100 runs and FPR of 0 in at least 84% of 100 runs and 6%FPR at 95[th] percentile



CONCLUSION

Proposed a fully distributed and lightweight framework that detects arbitrary NL attacks, localizes the compromised nodes, and automatically mitigates the attacks by isolating the compromised nodes with a 95[th] percentile FPR of under 6%