Sleep/Wake Aware Local Monitoring (SLAM)

Issa Khalil , Saurabh Bagchi, Ness Shroff
Dependable Computing Systems Lab (DCSL) &
Center for Wireless Systems and Applications (CWSA)
School of Electrical and Computer Engineering
Purdue University

Work supported by:
NSF-CISE/CNS and
Indiana 21st Century
Research & Technology Fund

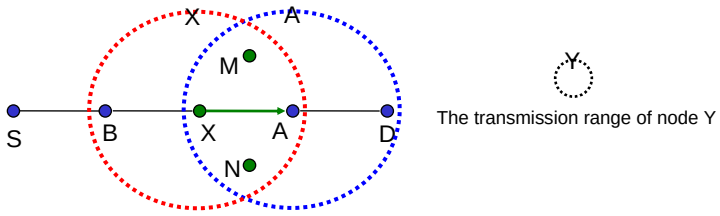
Outline

- Introduction
  - Motivation
  - Background: local monitoring
  - Goal and challenge
  - Solution approach
- Sleep/wake aware local monitoring (SLAM)
  - Assumptions and Sleep-Wake Mechanisms
  - On-demand SLAM: Protocol Description
  - Protocol analysis
  - Simulation results Conclusion

Motivation

- Overhearing in wireless media serves as a primitive building block for collecting information and evidence about network activities
- Target: **W**ireless **A**d **H**oc **a**nd **S**ensor (WAHAS) Networks
- Overhearing is used for many security applications in WAHAS networks
  - Intrusion detection
  - Building trust and reputation among nodes
  - Building secure routing protocols
  - Local monitoring (our work)

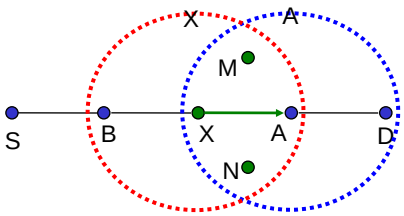
Background: Local Monitoring



- A collaborative detection strategy in which a node monitors the traffic going in and out of its neighbors.
- Assumptions
  - Each node knows its first-hop and second-hop neighbors
  - Requires each node to include the ID of the prev-hop in the forwarded packet
- *A guard* of a node A over the link X → A is any node that lies within the transmission range of both X and A
  - Example: M, X, and N are the guards of node A for the link from X to A

Background: Local Monitoring

- Local monitoring can be used to detect different kinds of attacks by different kinds of checks on the incoming and outgoing traffic
- Local monitoring provides the building block for detecting the following malicious actions
  - Fabrication
  - Modification
  - Delay
  - Drop
  - Misrouting



Goal and Challenge

- Overhearing in local monitoring incurs listening energy cost
  - Guards have to be awake all the time
  - Listening energy is a dominant source of energy consumption and is therefore critical in energy constrained WAHAS networks
- Goal of this work: Minimize energy overhead of monitoring
- Challenge: Perform the sleep/wake of guards securely and efficiently
  - Required guards to monitor a communication should be woken up
  - Quality of detection should not be significantly affected

Solution Approach

- We design a protocol called **Sl**eep/wake **A**ware Local **M**onitoring (SLAM)
  - Wake up guards only when needed
  - Verify that a node is waking up guards as needed
- Contributions
  - Conserve energy through sleep/wake aware local monitoring
  - Provide on-demand sleep/wake algorithm
  - Demonstrate the security coverage is not reduced due to the energy conserving mode of SLAM
  - Evaluate SLAM through extensive simulations and analysis

Outline

- Introduction
- Sleep/wake aware local monitoring (SLAM)
  - Assumptions and Sleep-Wake Mechanisms
  - On-demand SLAM
    - Protocol Description
    - Protocol analysis
    - Simulation results
- Conclusion

Assumptions & Attack Model

- System assumptions
  - Static network
  - Bi-directional links
  - Existing key management protocol that can distributed pairwise symmetric keys to any two nodes
  - Each node is equipped with passive or low-power wakeup antenna
- Attack Model
  - The adversary may not follow the sleep/wake protocol
  - The adversary may not provide the detection according to local monitoring
  - Adversary nodes may collude
  - The adversary can be more powerful than legitimate nodes
  - Physical layer attacks (DoS against wakeup antenna, jamming) are out of scope

Sleep/Wake Mechanisms

- Synchronized sleep/wake
  - All the nodes wake up and go to sleep in a synchronized manner
  - Examples: S-MAC and SPAN
  - Require accurate time synchronization
  - Wastage of energy in scenarios with rare events
  - Adaptation: Guards naturally sleep and wake up when there may be traffic to monitor
- Application-specific sleep/wake
  - Protocol for maintaining specific network properties such as *k*-connectivity or *k*-coverage
  - Adaptation: Change parameters of sleep/wake algorithm, e.g., increase *k*
- On-demand sleep/wake
  - Sleeping nodes are triggered to wake up when needed for communication
  - Requires low-power or passive wakeup antenna
  - Adaptation: On-Demand SLAM – Forms the topic of the rest of this presentation

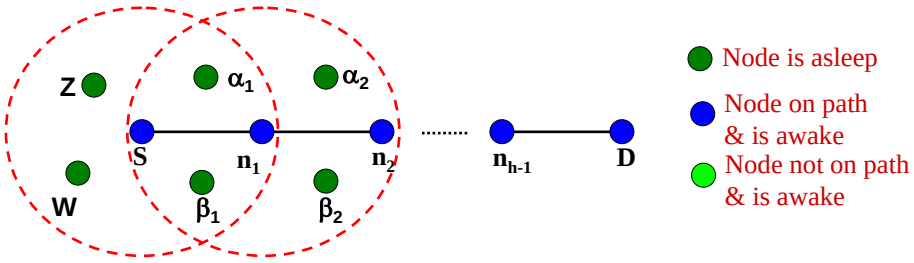
Outline

- Introduction
- Sleep/wake aware local monitoring (SLAM)
  - Assumptions and Sleep-Wake Mechanisms
  - On-demand SLAM
    - Protocol Description
    - Protocol analysis
    - Simulation results
- Conclusion

On-Demand SLAM

- Node wake up does not depend on assumption of a specific communication pattern
  - Nodes can be woken up when there is communication
  - Most energy efficient mode of communication
- Each node has two antennas
  - Wakeup antenna
    - Low-power or passive
    - Low bandwidth
    - Remains awake at all times
    - Available commercially and in research labs
  - Regular antenna
    - Always asleep except when triggered for sending/receiving data
    - Used for data communication and has higher power consumption

On-Demand SLAM: Overview



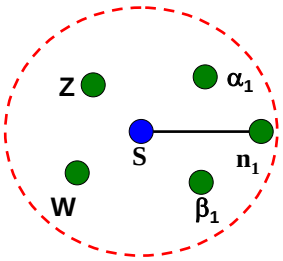
- Initially all the nodes are sleeping
- *S* wants to send a packet, *S* wakes up
- *S* sends a wakeup signal to its one-hop neighbors
- All the one-hop neighbors of *S* wake up
- Non-guards go back to sleep
- *S* sends the packet to n_1
- n_1 wakes up its neighbors
- n_1 sends the packet to n_2
- Guards go back to sleep
- The process continues until the packet reaches *D*

Security of SLAM: Rules of the Guard

- In baseline local monitoring, the responsibility of a guard of n_i over a given link is to verify that:
  - n_i forwards the packet within a time threshold (delay/drop)
  - n_i does not modify the packet it is forwarding (modification)
  - n_i forwards the data to the correct next hop (misrouting)
  - n_i only forwards if a packet is sent on the $n_{i-1} \rightarrow n_i$ link (fabric.)
- SLAM introduces a fifth responsibility
  - n_i should wake up the guards for the communication on the $n_i \rightarrow n_{i+1}$ link *before* forwarding the packet on that link

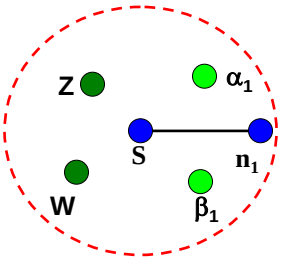
SLAM Variant 1: All-neighbors SLAM (A-SLAM)

* Wakeup signal broadcast to all the first-hop neighbors
+ Simple communication of wakeup signal
+ Knowledge of node location is not required
– Extra energy wasted in waking up irrelevant nodes



SLAM Variant 2: Guards-only SLAM (G-SLAM)

* Wakeup signal sent only to the relevant nodes (next-hop & guards)
+ More energy efficient
– Requires the knowledge of node locations within twice the transmission range
– Sophisticated wakeup signals and wake up hardware



Reducing End-to-End Delay

- Wakeup antenna has warm-up time which may increase end-to-end delay
- Send the wakeup signal at the earliest possible time
  - Pipeline the process of sending wakeup signal and data traffic
  - Increase the listening energy
- Definitions
  - $T_{control}$: time to send the wakeup signal
  - T_{warmup}: time to warm-up (sleep to wake)
  - T_{data} : time to send a packet (the worst case time which includes time to send the packet and channel contention)
  - T_w : time a node continues to be awake after being woken up

Reducing End-to-End Delay

- Two different cases
  - Case 1: $T_{control} + T_{warmup} < T_{data}$ (common case)

$$\Omega_{SLAM-Add}(h) = \Omega_{SLAM}(h) - \Omega_{Base-LM}(h) = T_{control} + T_{warmup}$$

  - Delay is constant independent of number of hops
  - Case 2: $T_{control} + T_{warmup} > T_{data}$

$$\Omega_{SLAM-Add}(h) = \Omega_{SLAM}(h) - \Omega_{Base-LM}(h) = h \cdot \tau + T_{data}$$

  - Delay is the product of the number of hops and the difference in time between ($T_{control} + T_{warmup}$) and T_{data}

Outline

- Introduction
- Sleep/wake aware local monitoring (SLAM)
  - On-demand SLAM
    - Protocol description
    - Protocol analysis
    - Simulation results
- Conclusion

SLAM Coverage: Proof Sketch

- $S \rightarrow n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \ldots \rightarrow n_{h-1} \rightarrow D$
- *Proposition*: SLAM does not cause loss in detection coverage
- *Lemma*: For each n_i in the path from $S \rightarrow D$
  - The guards of n_i (g_i) are awake and monitoring
- *Base case*: S is honest, thus g_1 is awake
- *Inductive hypothesis*: $g_1 \ldots g_i$ are woken up
- *Prove*: g_{i+1} is woken up
  - If n_i is honest it wakes up g_{i+1}
  - Else, g_i will wake up g_{i+1}
- *Result*: The required guards are always woken up either directly or indirectly

Outline

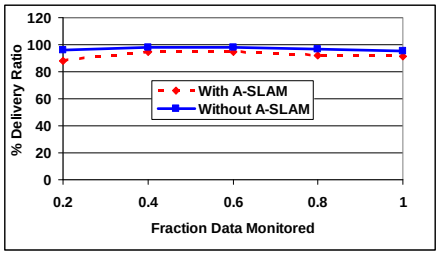
- Introduction
- Sleep/wake aware local monitoring (SLAM)
  - On-demand SLAM
    - Protocol description
    - Protocol analysis
    - Simulation results
      - Simulation setup
      - Effect of fraction of data monitored (f_{dat}) on output metrics
      - Effect of number of malicious nodes (M) on output metrics
      - Effect of traffic load ($1/\mu$) on output metrics
      - Energy savings
- Conclusion

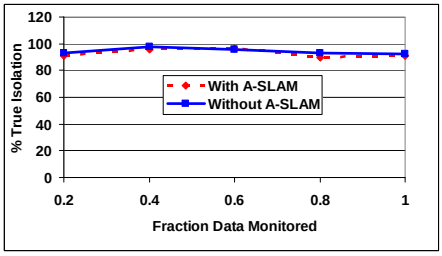
Simulations: Simulation Setup

- Data communication model: any node to any node
- Node distribution: Randomly on a fixed-size field
- Attack model
  - Attacker nodes are internal compromised nodes randomly selected from network
  - Colluding nodes establish a wormhole, then perform selective forwarding
- Two scenarios
  - Local monitoring with A-SLAM
  - Local monitoring without A-SLAM (baseline)
- The network simulator ns-2 is used
- The output parameters are measured at the end of simulation time (1500 s)

Effect of Fraction of Data Monitored (f_{dat})

% Delivery ratio = % (Packets received/Packets sent)
% True Isolation = % (# malicious nodes completely isolated/# malicious nodes)
Inputs: M = 4; N = 100; 1/ μ = 0.1 packets/s; T_{warmup} = 5ms, T_w = 30ms

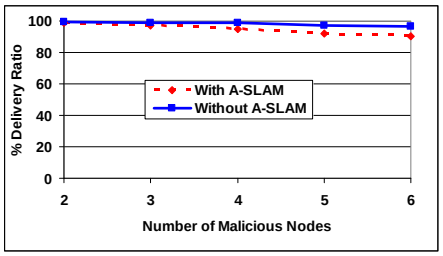


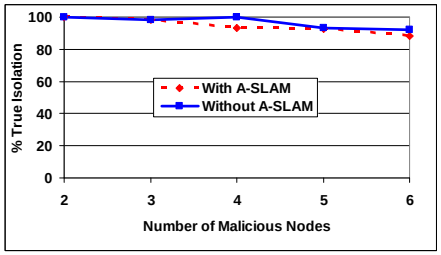


- High delivery ratio & high detection coverage even with the existence of attack
- Changing f_{dat} does not significantly change output metrics due to adaptive detection strategy
- Applying sleeping through SLAM does not significantly degrade the coverage or the delivery ratio

Effect of # Malicious Nodes (M)

% Delivery ratio = % (Packets received/Packets sent)
% True Isolation = % (# malicious nodes completely isolated/# malicious nodes)
Inputs: f_{dat} = 0.6; N = 100; 1/ μ = 0.1 packets/s; T_{warmup} = 5ms, T_w = 30ms

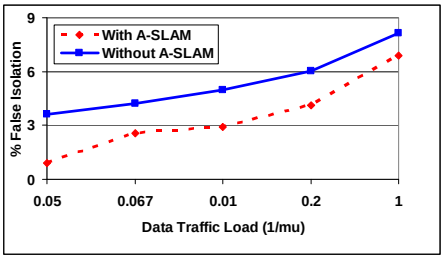


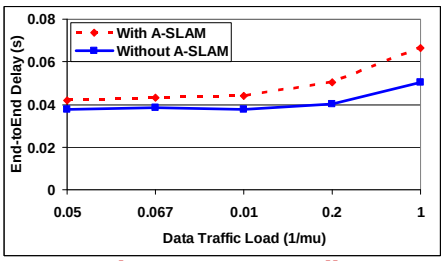


- As M increases, delivery ratio slightly decreases due to the increase in the traffic dropped by malicious nodes before being detected
- As M increases, coverage slightly decreases due to less number of good guards and increase in the probability that a malicious node is at the network's edge
- SLAM performs comparably to baseline local monitoring

Effect of Traffic Load (1/ μ)

% False isolation = % (# good nodes isolated/# nodes)
End-to-end delay = time of receive – time of transmit
Inputs: M = 4; N = 100; f_{dat} = 0.6 packets/s; T_{warmup} = 5ms, T_w = 30ms

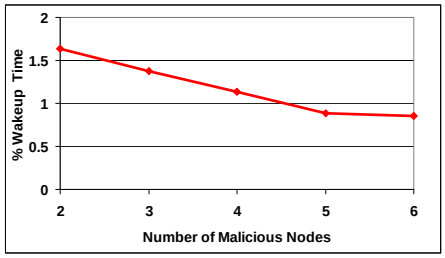


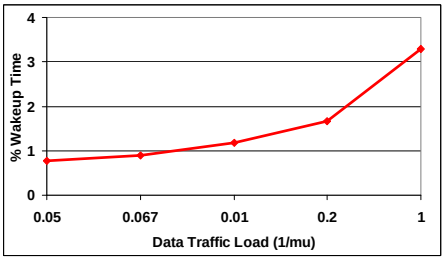


- As traffic load increases, false isolation increases due to increasing collisions
- As traffic load increases, end-to-end delay increases due to increasing channel contention
- SLAM has lower false isolation since some of the packets that may falsely identify a node as malicious may be dropped due to *erroneous sleep*
- End-to-end delay increases in SLAM with higher contention due to increase in the warm-up time

Listening Energy due to A-SLAM

% Wakeup time = % (time a node spent wake up solely for monitoring/total simulation time)
Inputs: f_{dat} = 0.6; M = 4; N = 100; 1/ μ = 0.1 packets/s; T_{warmup} = 5ms, T_w = 30ms





- The wakeup fraction is very small with SLAM
- As M increases, the wakeup time decrease due to the decrease in the number of packets that require monitoring
- As the traffic load increases, the wakeup time increases due to the increase in the number of packets that require monitoring

Conclusion

- Present a modified version of local monitoring
  - Sleep/wake aware local monitoring
  - On-demand sleep/wake mechanism called SLAM proposed
- Provide security analysis of SLAM
  - Does not weaken network security
- Provide end-to-end delay and energy analysis of SLAM
  - Fixed increase in the end-to-end delay may be achieved
  - Considerable energy savings compared to the vanilla version
- Provide extensive simulations to evaluate SLAM performance
  - Comparable performance with the vanilla version
- Future work: Enabling local monitoring based detection and isolation to multi-channel wireless networks

Thanks

Questions?

Backup Slides

WAHAS Networks

- **W**ireless **A**d-**H**oc and **S**ensor networks
- Autonomous system of nodes with no static infrastructure
- All or subset of nodes may move
- Nodes communicate wirelessly in multi-hop fashion
- Often subject to rapid deployment in environments where natural or malicious errors are likely
- Especially attractive in scenarios where it is infeasible or expensive to deploy significant networking infrastructure
- Application examples include: battle field surveillance, medical monitoring, biological detection, home security, disaster recovery

Challenges to Providing Dependability

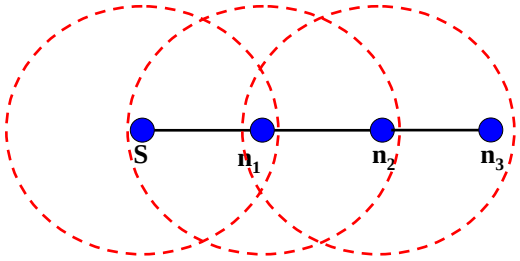
- Open communication media
  - Snooping is easy
- Hostile environment
  - Attackers everywhere
- Limited resources
  - Bandwidth and energy resources are constrained
  - Can not tolerate high overhead measures
  - Easy to launch attacks (cheap devices, no tamper-proof)
- Lack of infrastructure and difficult-to-reach deployments
  - Difficult to manage and control
  - Nodes are easy to access and capture by attackers
- Mobility
  - Dynamic topology
  - Frequent link failures

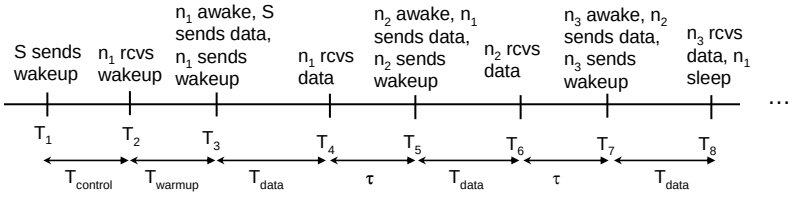
Security Attacks

- Two classes of attacks
  1. Attacks that can be defeated by crypto mechanisms
     - Eavesdropping: Solved by encryption
     - Message tampering: Solved by authentication
  2. Attacks that subvert the functionality of the network
     - Control attacks: Manipulating control traffic (e.g., routing traffic) to disrupt data traffic
       - Examples: ID spoofing and Sybil, sinkhole, rushing, wormhole
     - Data Attacks: Directly manipulate data traffic
       - Examples: Blackhole, grayhole
     - This class cannot be prevented by cryptographic mechanisms alone
- Throughout this work we have addressed both classes with more focus given to the second class

Case 2: ($T_{control}$ + T_{warmup}) > T_{data} – Forwarding node

- $S \rightarrow n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \ldots \rightarrow n_{h-1} \rightarrow D$
- Let $\tau = (T_{control} + T_{warmup}) - T_{data}$
- Three different node role
  - For a node in the route to the destination

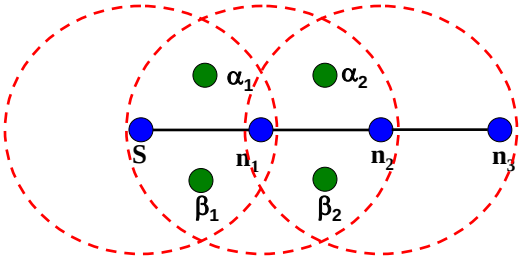


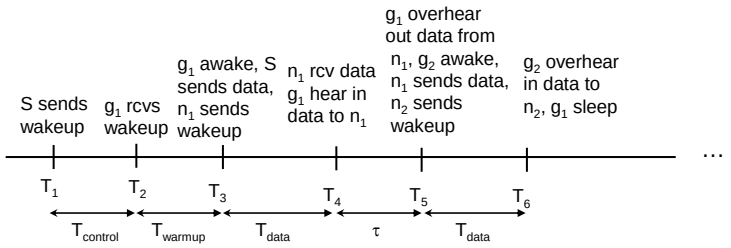


PURDUE UNIVERSITY

Case 2: ($T_{control}$ + T_{warmup}) > T_{data} – Guard node

- $S \rightarrow n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \ldots \rightarrow n_{h-1} \rightarrow D$
- Three different node types
  - For a guard node

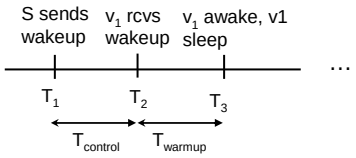


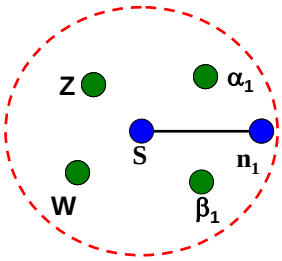


PURDUE UNIVERSITY

Case 2: ($T_{control}$ + T_{warmup}) > T_{data} – Non-guard node

- S→n_1→n_2→n_3→ … →n_{h-1}→D
- Three different node types
  - For a neighbor to the route but is not a guard





Case II: ($T_{control}$ + T_{warmup}) ≤ T_{data}

- The timing schedule can be generated in a similar way
- Please refer to the report

Energy Consumption

- SLAM is compared to an on-demand sleep/wake algorithm with no local monitoring support
- Three different nodes considered
  - Forwarding node
  - Guard node
  - Non-guard node
- The worst case additional energy of SLAM for each node is respectively (single flow, per packet)
  - Forwarding node: T_w . A_{active}
  - Guard node: T_{warmup} . A_{warmup} + T_{data} . A_{active} + T_w . A_{active}
  - Non-guard node: T_{warmup} . A_{warmup} or 0 for G-SLAM