Secure Information Exchange in Vehicular and Power Grid Networks

Donghoon Shin, Jinkyu Koo, Madalina Baker, and Prof. Saurabh Bagchi

Contact: Prof. Saurabh Bagchi (sbagchi@purdue.edu; 765-494-3362)

Dependable Computing System Lab (DCSL), School of Electrical & Computer Engineering, Purdue University

Introduction

- Over the past years, ad hoc wireless networks (AWNs) have received great attention as a promising technology for a variety of applications.
- However, it is a difficult task to keep the communications secure when the AWNs are under the attack. The nodes are inherently vulnerable to attacks because they are usually deployed in non-protected environments. → Security issues
- In addition, once nodes are deployed, it is a challenging task to send and receive timely updates: Nodes are typically located in hard-to-reach places and state update or dissemination consumes significant energy.
  → Reprogramming issues

DCSL answers the questions below♪

How can we make AWNs operate reliably and securely?♪	How can we collect events in AWNs quickly and securely?♪	How can we reprogram nodes in AWNs efficiently?♪
↓	↓	↓
Optimal Multi-Channel Monitoring	Secure Event Collection♪	Efficient Code Dissemination♪

Optimal Monitoring in Multi-Channel Wireless Networks

- Deploy a set of monitoring nodes being trusted to monitor the behaviors of other nodes in multi-channel AWNs
- Optimal placement and channel selection of monitoring nodes

Where to place a given number of monitoring nodes among several possible locations in the network and which channels to tune their radios to, in order to maximize the detection coverage?

Equivalently

Given a set of monitoring nodes deployed in the network, how to select a subset of monitoring nodes to be activated and channels for the selected monitoring nodes, in order to maximize the detection coverage?

Greedy Algorithm♪

- At each iteration, pick the pair of monitoring node not yet selected and channel that gives maximum coverage improvement
- Repeat above process until a given number of monitoring nodes is chosen or all sensor nodes are covered

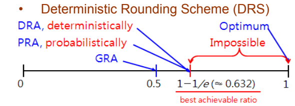
LP Rounding Algorithm— PRA, DRA♪

Basic steps of LP rounding algorithms

1. Formulate a given optimization problem into an integer linear program (ILP)
2. Transform the ILP to an LP by relaxing the integer constraints
3. Solve the LP relaxation (using one of many existing LP solvers)
4. Round the optimal solution of LP relaxation

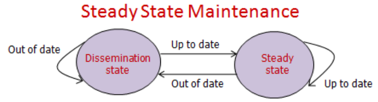
- We develop two different rounding schemes
  - Probabilistic Rounding Scheme (PRS)
  - Deterministic Rounding Scheme (DRS)

Worst Performance Guarantees of Proposed Algorithms♪



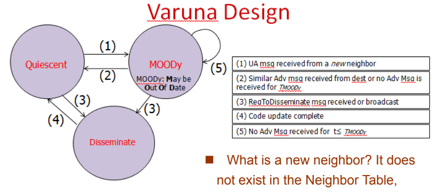
Efficient Code Dissemination

Steady State Maintenance



- Any multi-hop wireless network has to be kept up-to-date as new code or new state is generated at the base node.
  - In this work we use code dissemination as a specific example of state dissemination
- Why is there a cost in the steady state?
  - Dynamic network topology: Caused by transient link failures, node mobility, incremental node deployment, etc.
  - Nodes may remain disconnected from the network for some time and may miss the state update
  - After they come out of disconnection, they must detect the inconsistency
- Communication between inconsistent nodes is a problem
  - Incorrect data may be propagated through the network
  - Network may become partitioned
- Existing solution:
  - Each node periodically broadcasts advertisements containing metadata, e.g. code version number
  - Steady state energy cost increases linearly with the steady state period – the most dominant phase in a node's lifetime
  - Radio transmissions are the most energy expensive operation

Varuna Design

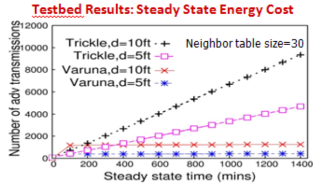


Varuna achieves fixed steady state cost

- After a node downloads a new version of the code, it verifies its metadata with each of its neighbor *only once*
- Steady state energy cost is independent of the steady state period, subject to sufficient memory and reasonable link reliability

- What is a new neighbor? It does not exist in the Neighbor Table, which is cleared when the node boots or its metadata changes
- When neighbor table is full, LRU replacement is used
- *Varuna's invariant*: If a node receives a packet from another node with a lower version of the code metadata than its own, the metadata inconsistency is detected by the receiving node

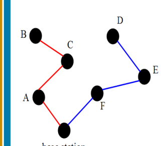
Testbed Results: Steady State Energy Cost



Secure Event Collection

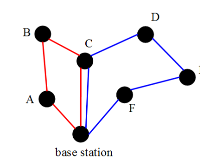
- Problem definition: Collection of a delay-sensitive event, e.g., power line instability, in multi-hop wireless networks, in the presence of attacks.
- Base station needs to check every sensor every P unit time.
- Needs a multi-hop routing to check the node at the distance more than one hop from the base station
- ISSUE: Malicious nodes in the middle of the multi-hop route may drop/delay/modify a critical event report.

Straw-man Solution♪



- Base station checks each line independently, by using ODSBR.
- Any node who has an event to report can put the event log in an ACK packet of the ODSBR.
- If malicious nodes want to stay undetected, they cannot drop/delay/ modify the ACK packet.
- Problem:
  - The ACK mechanism of ODSBR is expensive, due to onion signaturing.
  - This expensive ACK scheme should be used all the times, even if there is nothing to report.

Proposed Solution♪



- Base station keeps circulating a probe token (PT) through each circle.
- On receiving the PT, nodes start its ODSBR timer, expecting to receive an ACK before the timer expires.
- Any node who has an event to report can put the event in the PT with its signature on it.
- High-level ideas:
  - If malicious nodes do not drop the PT, or do not delay the PT for long, the PT returns to the base station within some time. → What is the threshold?
  - In this case, the base station does not need to send ACK packet. Instead, the base station circulates a new PT.
  - This new PT can acknowledge the previous PT before ODSBR timer set up in the previous round expires, if nodes' ODSBR timeout is long enough. → How long?
- Achieved objective:
  - Provides the same security guarantee as ODSBR, but uses the expensive ACK mechanism only when network is under attack.