Robust Communication in Sensor Networks Resistant to Node Compromise and Failures

Saurabh Bagchi
Dependable Computing Systems Lab
School of Electrical and Computer Engineering
Purdue University

Joint work with: Issa Khalil, Gunjan Khanna, Ravish Khosla, Ness Shroff

http://shay.ecn.purdue.edu/~dcsl

Greetings come to you from …

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Motivation

- Sensor networks being deployed in critical military and civilian situations
  - Hostile environment with adversaries in military domain
  - Privacy concerns in civilian domain
  - Tamper proof communication for emergency rescue and relief
  - It is important for sensed data to make its way to command and control center
- Therefore, **dependable sensor networks**

Dependable Sensor Networking

- Dependability is the property of a system to tolerate failures, be it from natural errors or malicious errors, *aka* security attacks

Dependability

Resilience to natural errors, i.e., **Reliability**

Resilience to malicious errors, i.e., **Security**

Why for Sensors?

1. The nodes are failure prone
2. The wireless links are failure prone
3. Placed in hazardous environments
4. Sometimes used for detection of critical events

Why for Sensors?

1. Placed in hostile environments
2. Adversaries have huge gains from compromising sensor network
3. Low cost rules out tamper proof hardware
4. Omni-directional wireless links

Motivation

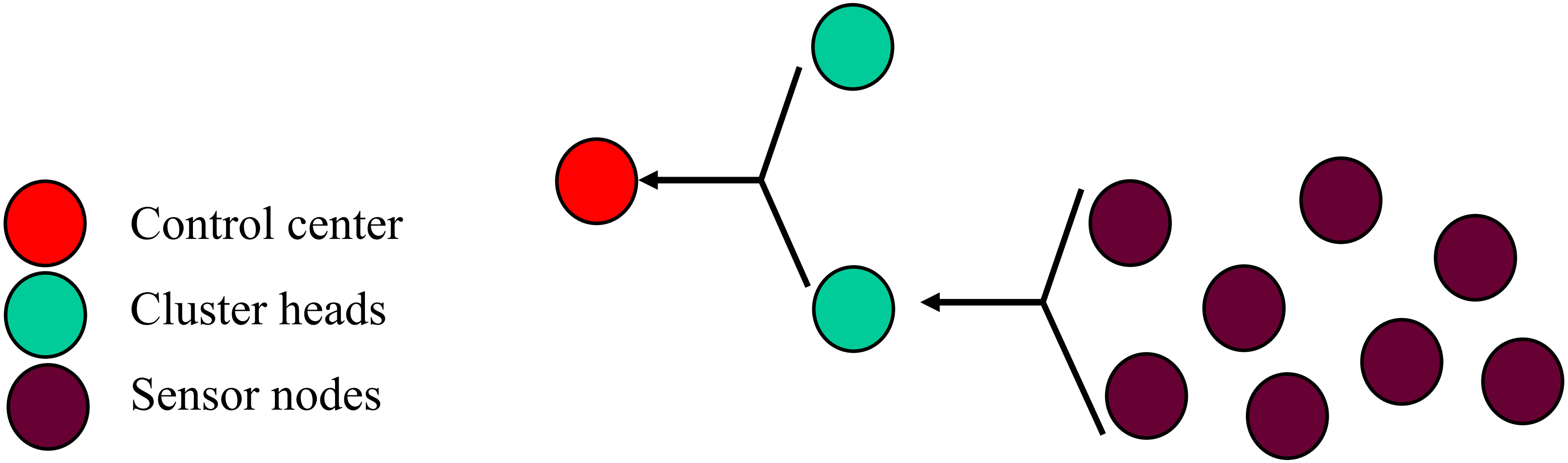
- Reliability in data collection is important but hard to achieve
  - Small energy source
  - Low bandwidth
  - Large scale (ten's of thousands of nodes) with long paths which can have multiple failures
  - Some constraints that technology may *partially* remove for us (compute cycle, memory)
  - Susceptible to collective failures
- Securing communication is important but hard to achieve
  - Traditionally use cryptography techniques for securing communication
  - Cryptography involves keys
  - Key management requires trusted entities
  - Key management requires powerful entities

Outline

- Motivation
- Robust data dissemination
  - **Background**
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

What is data dissemination?

- There are some sources of sensory data
  - Possibly sources with overlapping sensing regions
- There are some nodes interested in sensory data
  - Maybe resource constrained nodes themselves
  - Can be cluster heads in hierarchical communication
  - Alternately, can be a moving data collector



Existing Data Dissemination Protocols

- Data dissemination in sensor networks is a topic receiving enormous interest in the research community
- However, data dissemination in a delay sensitive and energy conserving manner with fault tolerance concerns has received far less attention
- Protocols can be broadly classified into PUSH and PULL based
  - PUSH : Sensors send the data at regular intervals to a sink node
  - PULL : Sensors store the data and data is collected using a polling mechanism

Existing Data Dissemination Protocols

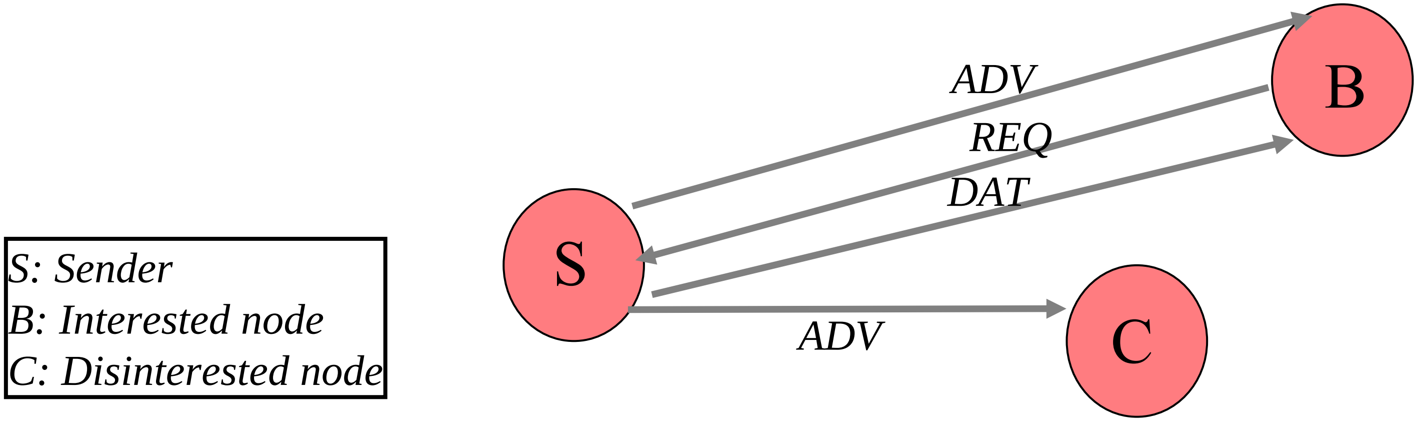
- Broadcast and Gossip have been used to provide reliability but use redundant transmission leading to wastage of energy
- TTDD *[Zhang et al.]*
  - Protocol for data collection by mobile collectors from static sources
  - Sets up a grid structure and proactively determines routing from data source to sink
  - At runtime, when sink needs data it locates a close by "dissemination point" which uses pre-computed route from source to sink
  - Drawbacks: Cost of setting up entire routing grid

Outline

- Motivation
- Robust data dissemination
  - Background
  - **Example protocol: SPIN**
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Example Protocols

- SPIN (**S**ensor **P**rotocols for **I**nformation via **N**egotiation) [Balakrishnan *et al.*]
  - Use meta data transmissions to reduce redundant transmissions
  - Advertise the data prior to sending the data
  - Efficient in case of collisions
  - Mix of Push and Pull mechanisms



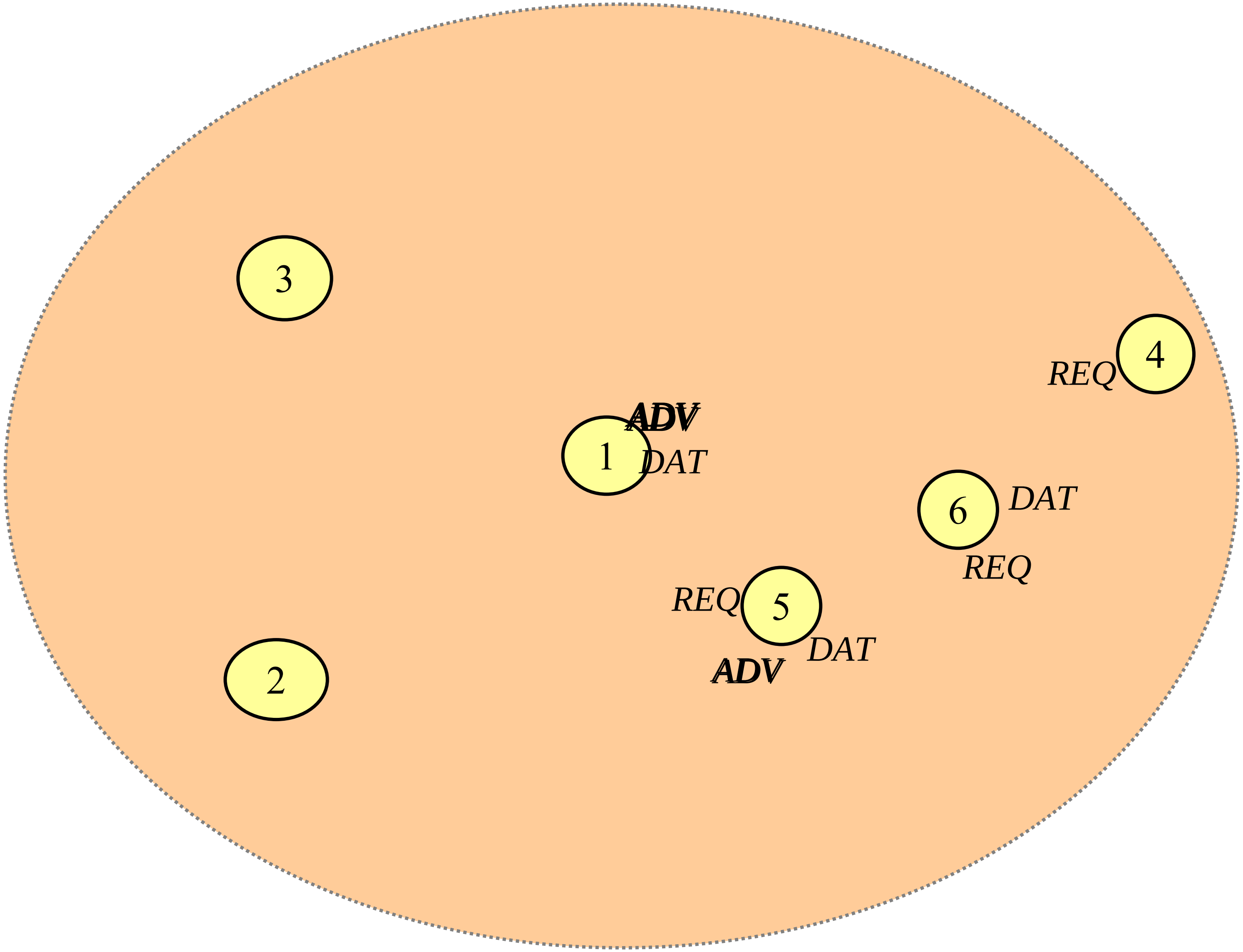
Reliability in Existing Protocols

- Current protocols are not designed to address the issue of failures in the sensors
  - Either the data is lost in case of a failure
  - Broadcast and Gossip do address failures as by-products but are wasteful in terms of resources
- Protocols use direct communication between the nodes and the base stations
  - Not feasible in practical larger sensor networks
- Several times a central controller is employed leading to a violation the distributed nature of the protocol
  - Setting up grid structure in the TTDD

Outline

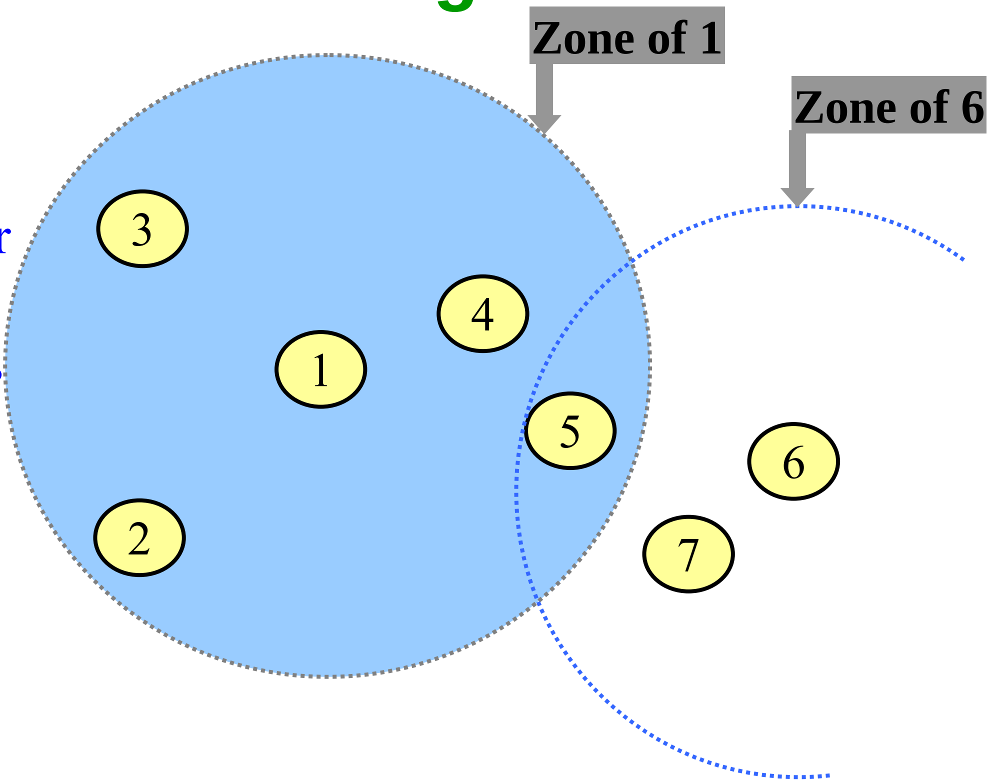
- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - **Our protocol: SPMS**
  - **SPMS: Failure scenario**
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Shortest Path Minded SPIN (SPMS)



Shortest Path Minded SPIN: Design Features

- Zone
  - Maximum distance a node can reach using the maximum power level
  - Node can adjust its power levels to reach all nodes (neighbors) in its zone
  - Routing tables for neighbors in the zone using Bellman Ford
  - Tables contain the power level for each neighbor

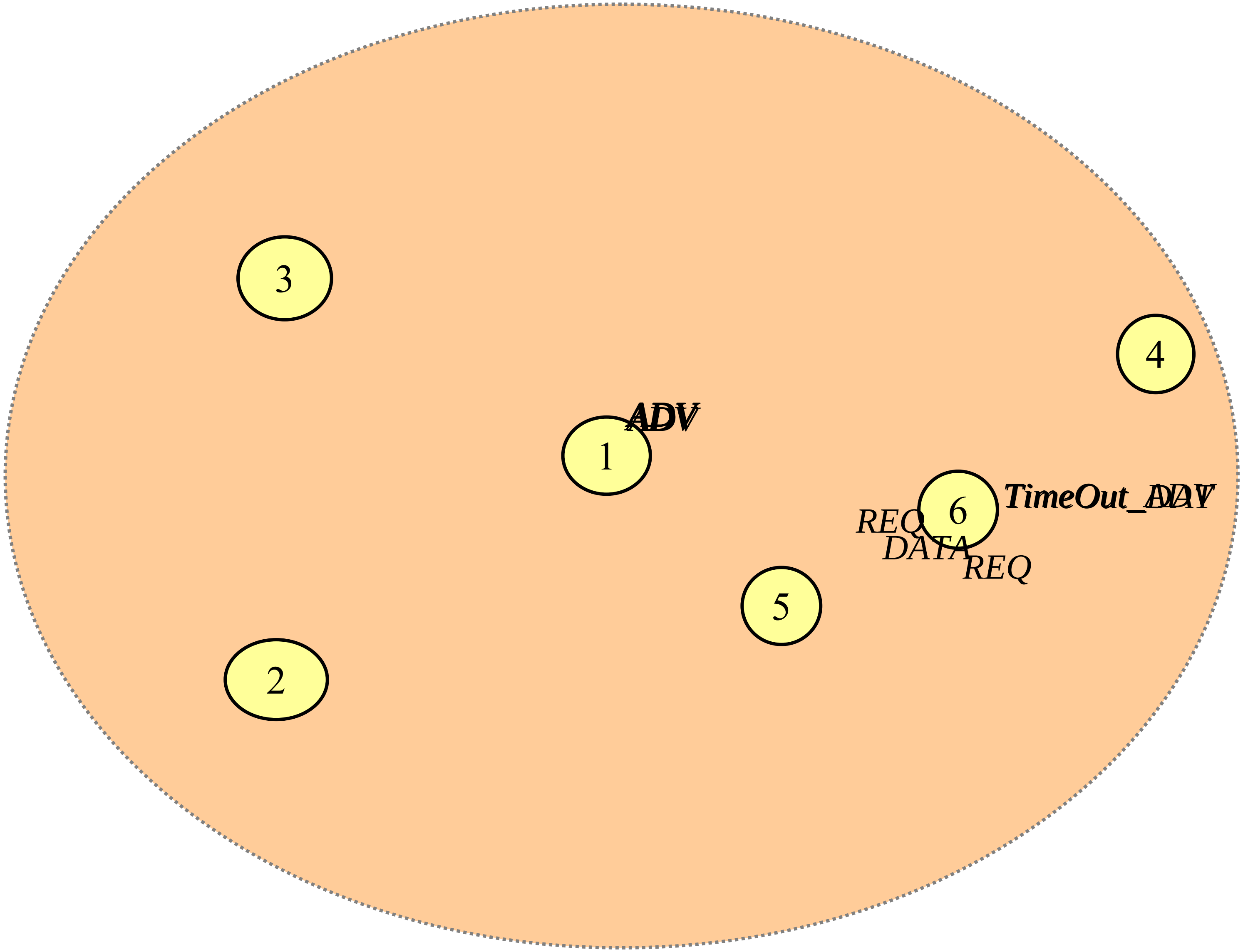


- Timers
  - TimeOut_{ADV} : Nodes wait for the data to come to the ***nearest node*** before sending REQ
  - TimeOut_{DAT}: Nodes wait for the data after sending the REQ packet

SPMS Protocol : Failure Scenario

- Failure of an intermediate node
  - Could take place before or after sending the ADV
  - Not sending an ADV can be misinterpreted as failure
  - Node stores the neighbors which have advertised the data
    - PRONE : Primary Originator Node
    - SCONE : Secondary Originator Node
- Resilience to Failures
  - After a $\text{TimeOut}_{\text{ADV}}$ expires, node sends the request to PRONE through the shortest path
  - DATA is received using the same path if there is no failure
  - Incase of a failure $\text{TimeOut}_{\text{DAT}}$ occurs
  - Node directly sends the REQ packet to PRONE
  - In case PRONE is also not responding then the REQ is sent to SCONE

SPMS : Failure Scenario

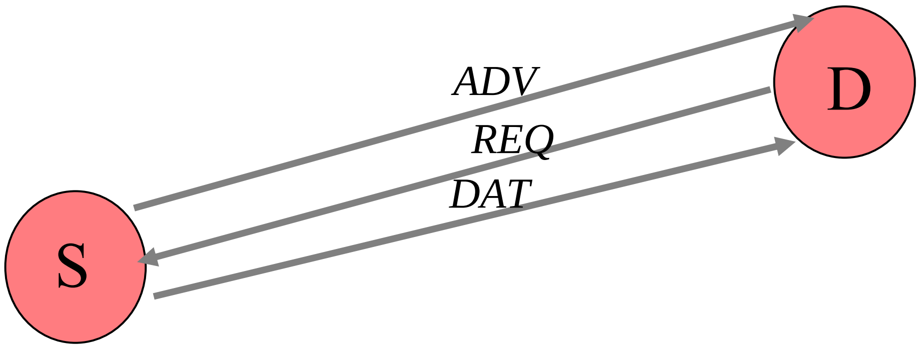


Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - **Energy and delay analysis**
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Energy and Delay Analysis

- Time to get data from source to adjacent destination is defined as T_{round}

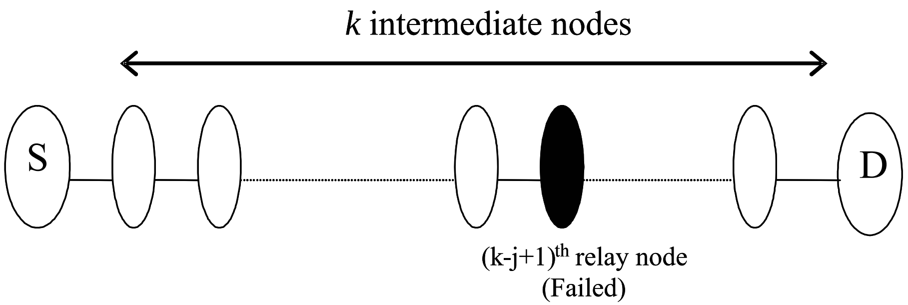


$$T_{round} = G.n_1^2 + A.T_{tx} + T_{proc} + G.n_s^2 + R.T_{tx} + T_{proc} + G.n_s^2 + D.T_{tx}$$

$$T_{round} = G.n_1^2 + (A+R+D).T_{tx} + 2T_{proc} + 2G.n_s^2$$

Energy and Delay Analysis

- In case of *K* relay nodes between two nodes



$$Delay_{failurefree} \leq (K-1)T_{round} + TOut_{ADV} + T_{c2}$$

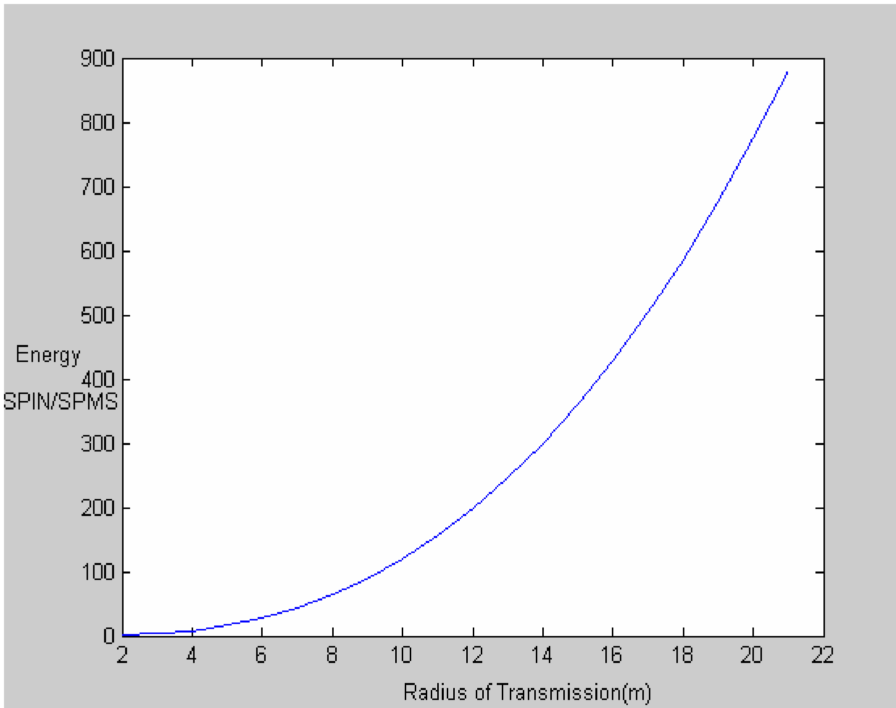
$$Delay_{failure} = (k-j)T_{round} + TOut_{ADV} + G.ns^2 + TOut_{DAT} + 2G.nj^2 + (R+D)T_{tx} + 2T_{proc}$$

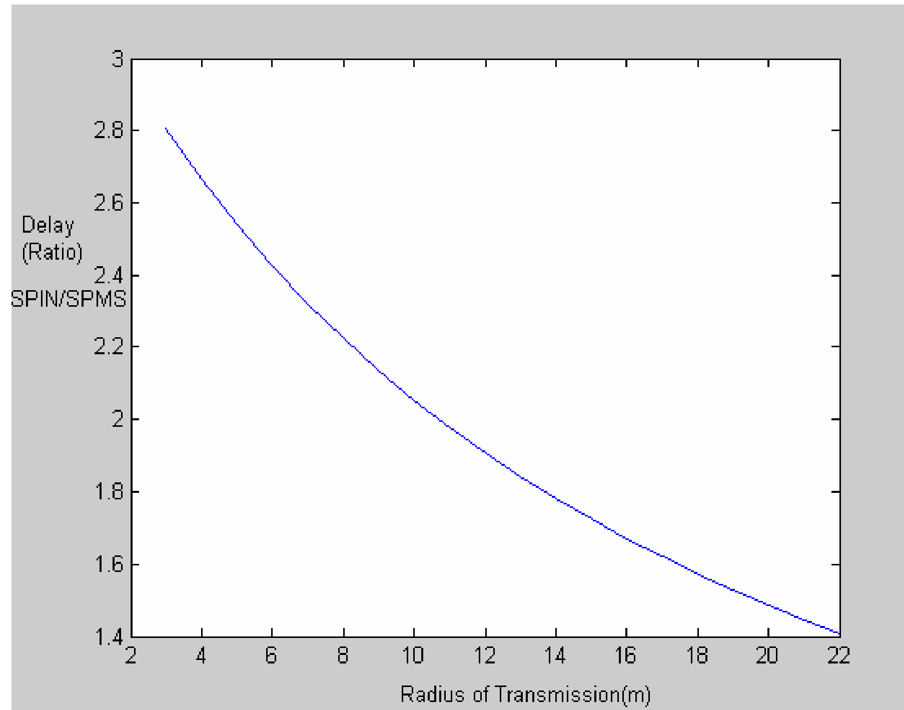
- The ratio of energy between SPIN and SPMS can be given by :

$$E_{SPIN} = (A+D+R).E_1 + (A+D+R).E_r \quad E_{SPMS} = k.A.E_1 + k.(D+R).E_m + k.(A+D+R).E_r$$

$$E_{SPIN} : E_{SPMS} = \frac{E_1 + E_r}{k.f.E_1 + k.E_m + k.E_r}$$

Energy and Delay Comparisons: Equation Plots





SPIN uses more energy than SPMS as relay nodes increase.

Delay advantage of SPMS decreases as relay nodes increase.

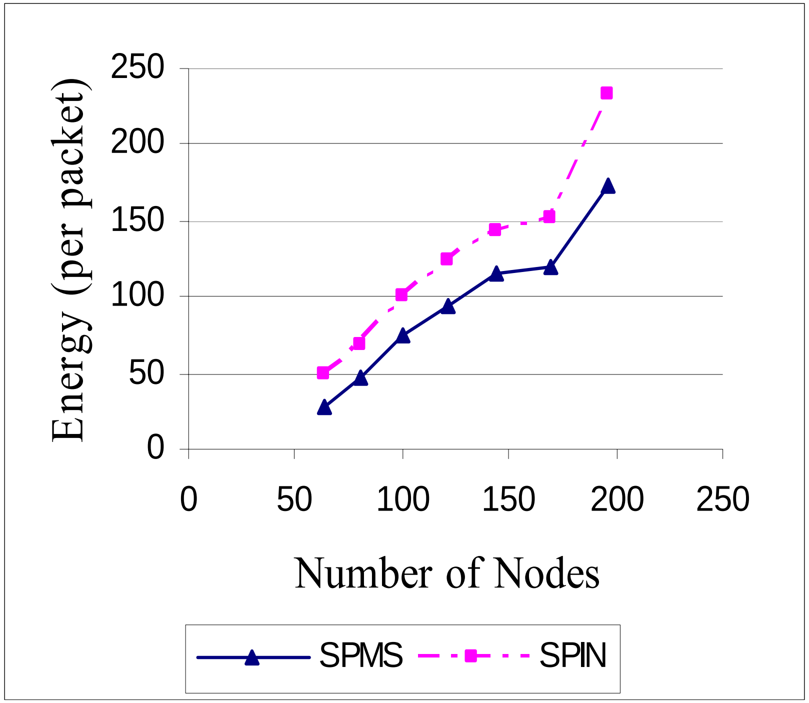
Outline

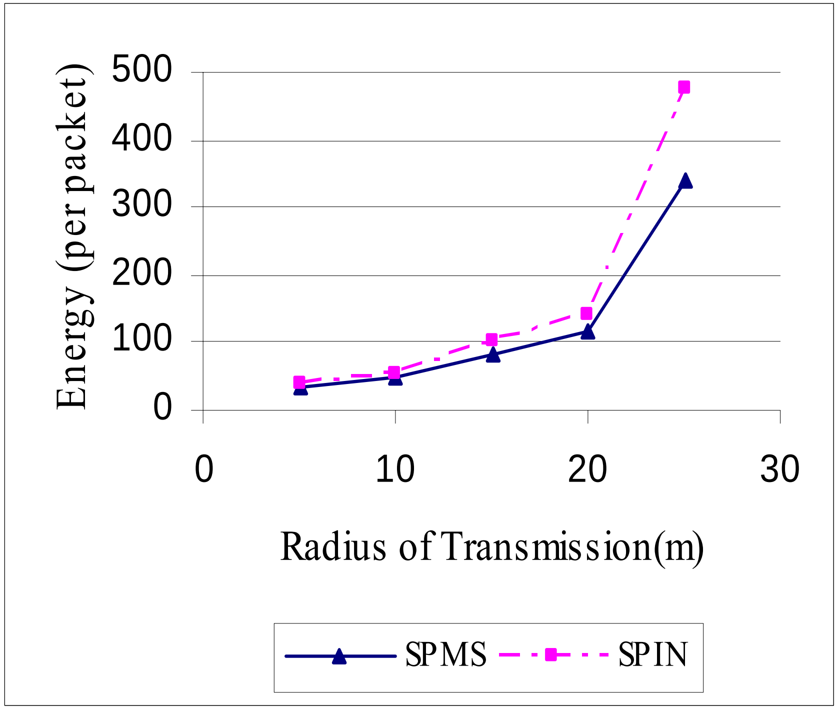
- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - **Simulation results**
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Simulations

- SPMS protocol is simulated in ns-2 and compared with SPIN
  - We vary the transmission radius and the number of nodes
- Crossbow data sheet is used to calculate the power spent in transmission and receiving packets.
  - Nodes can only transmit at 5 energy levels considered in our experiments
  - ADV and REQ packet are considered to be 2 bytes and DATA packets are 40 bytes long
  - Inter packet arrival time is exponential
- Experiments are carried out for two topologies
  - **All to All communication :** Every node requests data from every other data
  - **Cluster Based Hierarchical Communication:** Cluster heads collect the data and send it to the sink using SPMS
- Experiments for failure free and failure scenarios
  - Failures are transient and follow exponential inter-arrival times

Results for Failure Free Scenario: Energy Metric

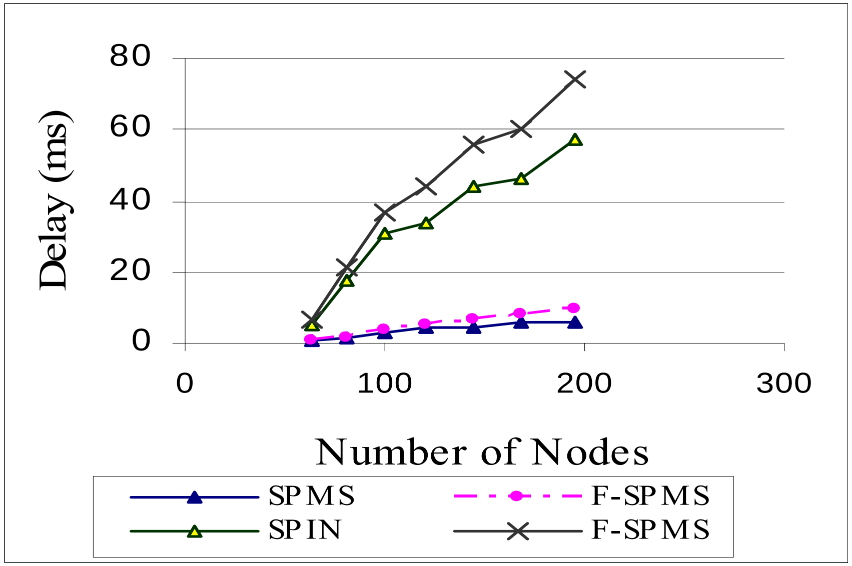


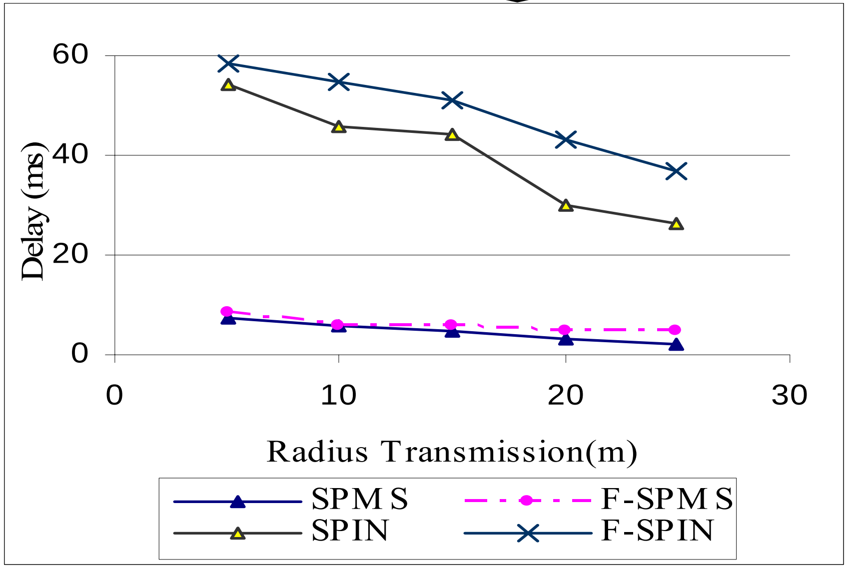


SPMS saves about 23-46% energy compared to SPIN with varying number of nodes

Results for Failure Free and Failure Scenario: Delay Metric

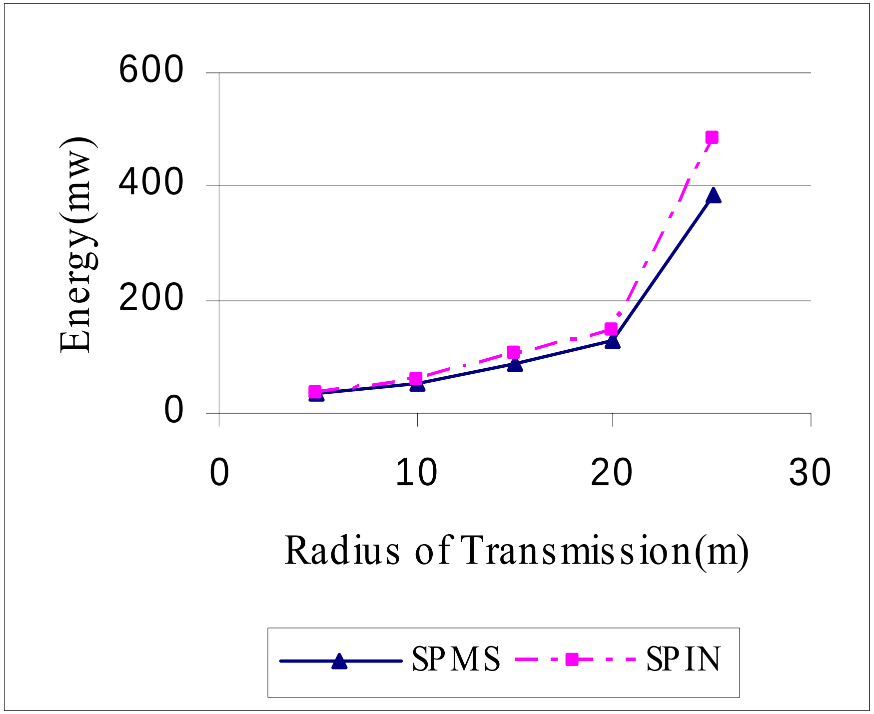
SPIN incurs 10 times more delay



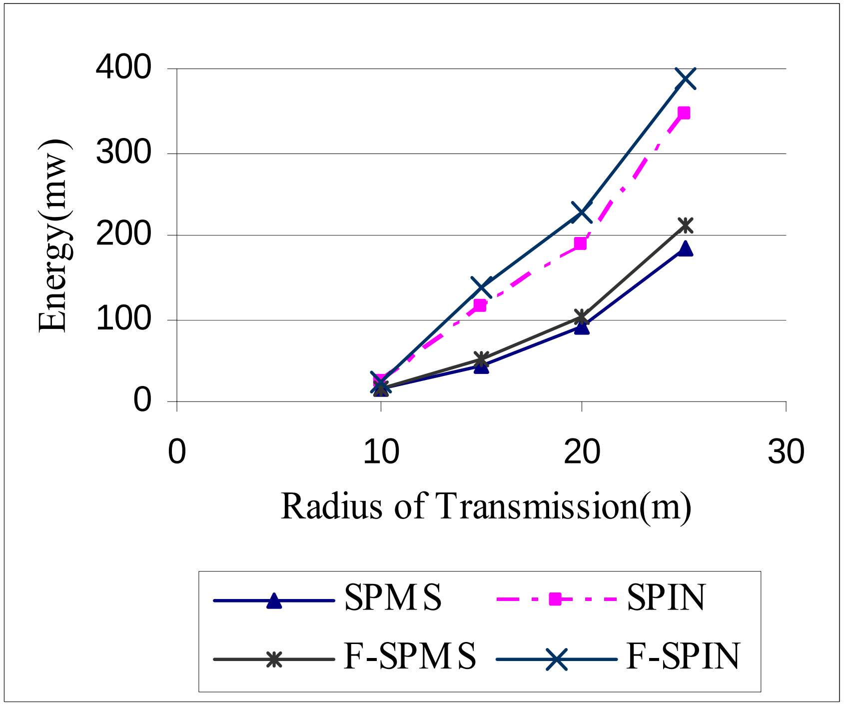


- Delay gradient is steeper for SPIN with increasing number of nodes
- Delay decreases with radius of transmission
- SPMS disseminates data much faster compared to SPIN in both failure and failure free scenarios.

Energy Metric : Mobile Nodes and Cluster Based Communication



Mobile Nodes



Cluster Mechanism

SPMS saves about 21% energy compared to SPIN even with mobility.

SPMS saves 59% energy in Cluster Based Hierarchical communication.

Current Work … Coming Soon

- Failure optimized SPMS
  - Avoid sending REQ through a suspected failed path
  - Inform neighbors of suspected failed path
  - This is more timely than route updates
- Mobility optimized SPMS
  - Avoid Bellman Ford on entire zone if node moves in
  - Incremental computation in a lazy manner

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - **Background: Key management in sensors**
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Key Management in Sensor Networks

- Most nodes have resource constraints
- Dynamic environment where network partitions and failures of nodes and links are not unlikely
- Individual nodes may be compromised
- Two traditional approaches
- Key predistribution: Two extreme examples are
  - Unique key for each node pair
  - Single key for the entire network
- Kerberos-like client-server approach: Privileged nodes distributed in the network for key management functionality

Our Design Goals

- Provide scalable secure key management obeying the constraints of the sensor node
- Remove the requirement of specialized nodes
- Make the protocol resilient to eavesdropping, denial of service, and node compromise attack and natural failures
- Reduce the end-to-end latency of secure data communication
- These goals realized in protocol called SECOS

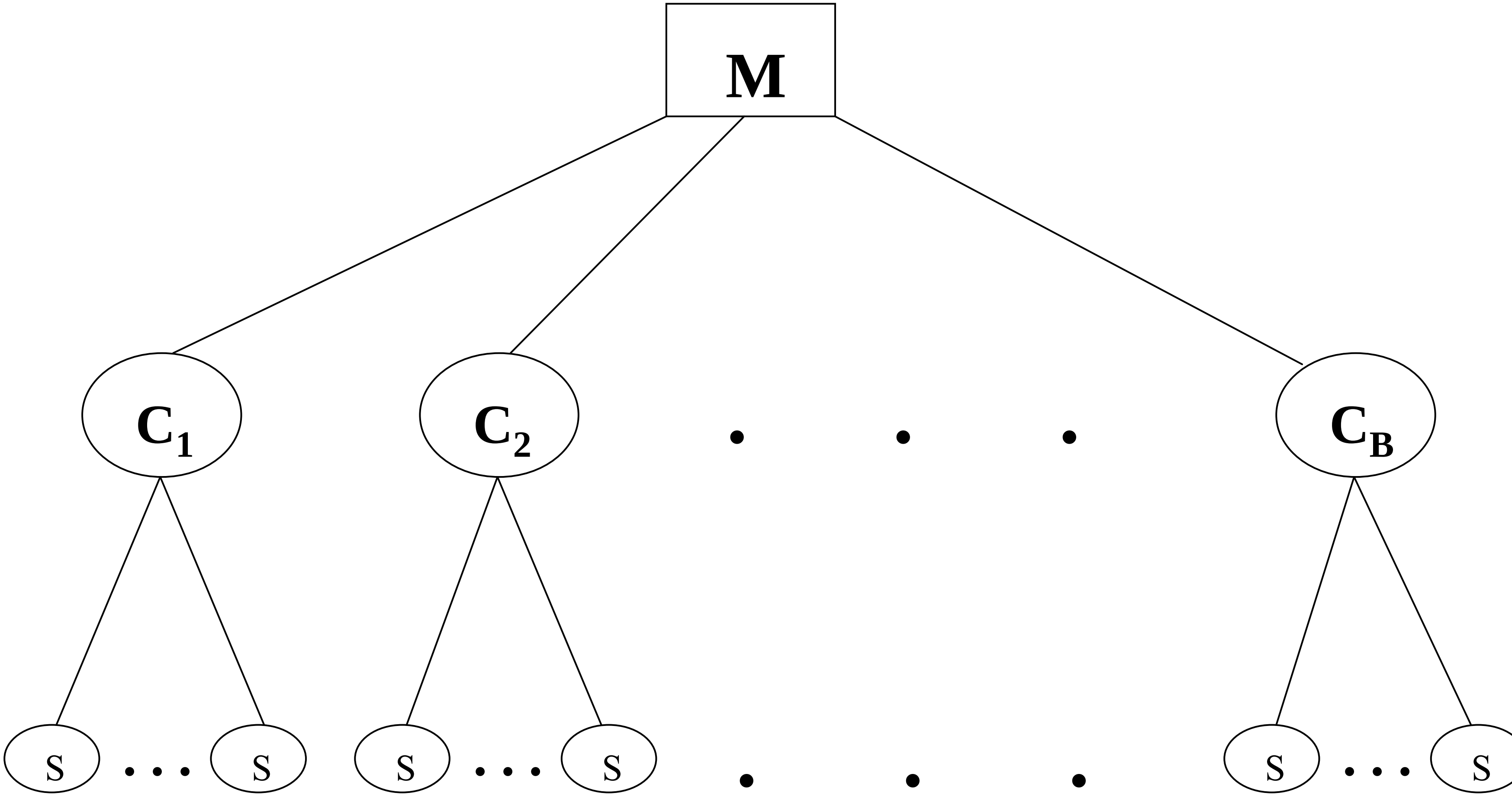
Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - **Our protocol: SECOS**
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

SECOS: High Level Approach

- Divide the sensor field into multiple control groups, each with a control node
- Symmetric cryptographic primitive used, such as DES
- Communication within a group happens using key exchanged through the control node
- Communication across groups happens using key exchanged through multiple control nodes
  - Communication between control nodes happens using key exchanged through base station

SECOS: High Level Approach



S: Sensing Node C_i : Control Node M: Base Station

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - **SECOS: Key elements**
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Failure and Resource Model

- Base station is fixed, secure, and has no resource constraints
- All other nodes are generic sensor nodes and have all the typical resource constraints
- Links may be subjected to eavesdropping and message tampering
- Nodes may be subjected to denial of service attacks and may be compromised
  - "Don't trust thy neighbor"

Building Blocks for SECOS

- Purging key caches: Caches provide benefits in latency and energy consumption but lead to vulnerability
- Key refreshment: Either periodically or when triggered by anomalous event
- Rotate privileged node role: Since we do not assume specialized protected nodes for key management functionality

Keys used in SECOS

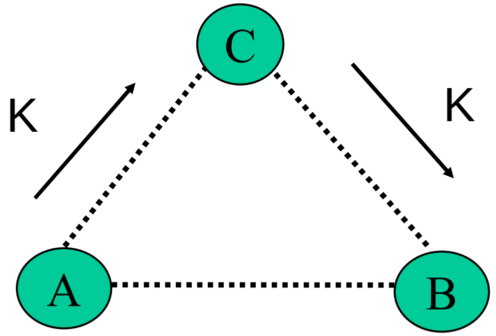
- Master key: Unique key shared between each node and the base station
  - Burnt in at time of deployment
- Volatile secret key: Used for key generation of other keys such as session key
  - Provided to a node at deployment time
  - Changed after each key generation
- Session key: Used for secure communication between two end points
  - $K_{XY(2)} = MAC_{K_{XY(1)}}(counter_{XY} \oplus K_{XY(v)} \| 1)$
- MAC key and random number generator key: Not discussed here
- Counters for semantic security

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - **Communication within control group**
  - Communication across control group
  - Analysis
  - Simulation results
- Take away lessons

Communication within Control Group: Soln I

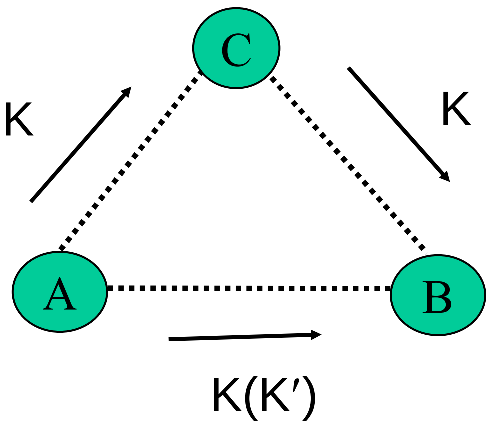
- Control node establishment and establishment of secure channel between control node and other nodes done



- Con: Compromised control node can expose communication between A and B

K

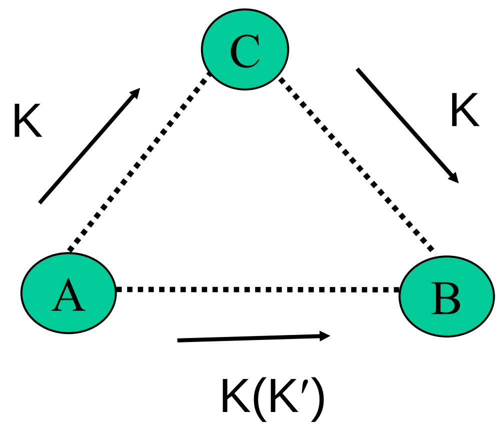
Communication within Control Group: Soln II



- Control node has access to K but not K(K′)
- Hence, it cannot get K′, the session key between A and B

- Con: If C colludes with a node that is on the path from A to B and gets K(K′)

Communication within Control Group: Soln III



Extract K′ from K(K′)

Use K″ from earlier control node. Session key is $K'' \oplus K'$

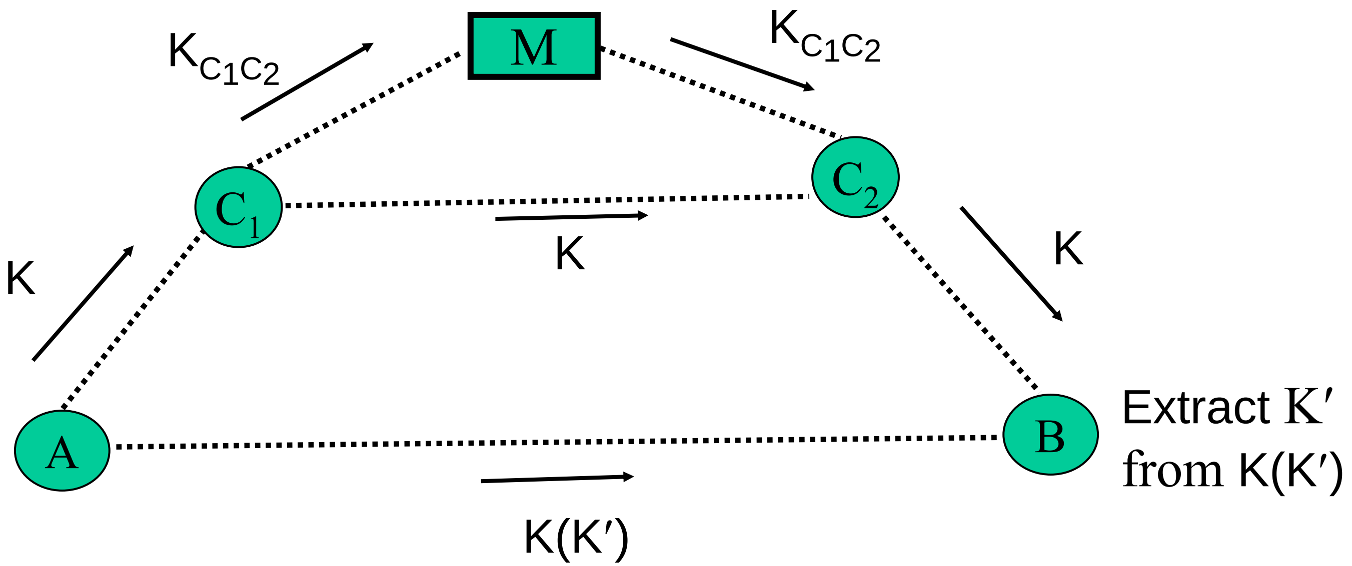
	C_1	C_2	…	C_n
K_0	K_1'	K_2'		K_n'
	$K_1 = K_0 \oplus K_1'$	$K_2 = K_1 \oplus K_2'$		$K_n = K_{n-1} \oplus K_n'$

- Con: If adversary crypt-analyzes K_0 *and* compromises $C_1, \ldots, C_n$

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - **Communication across control group**
  - Analysis
  - Simulation results
- Take away lessons

Communication across Control Group



- Expensive communication protocol
- Note asymmetry in the two phases

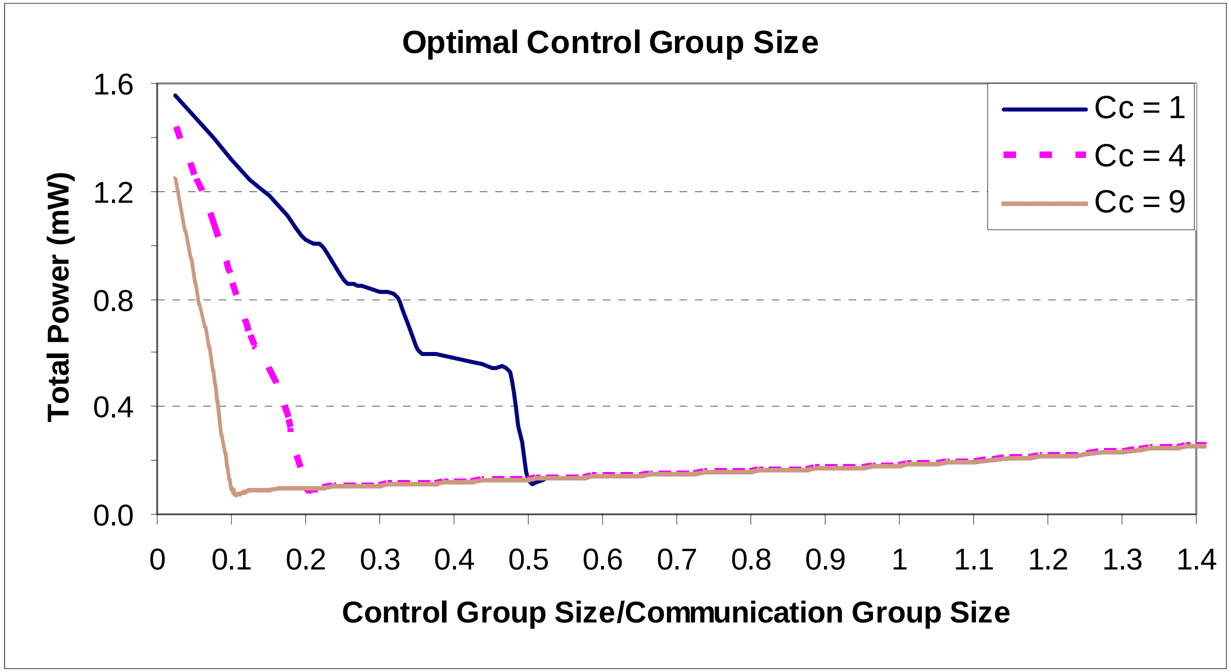
Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - **Analysis**
  - Simulation results
- Take away lessons

Control Group Size

- Upper bound imposed by the resource constraint on control node
- Energy wise optimal control group size determination has two opposing pulls
  - Larger size avoids expensive inter-group communication
  - Smaller size minimizes the number of hops to the control node
  - Control cache comes to the rescue
- Energy curve is discontinuous due to different cases
  - Hit in regular cache
  - Miss in regular cache, communication within control group
  - Miss in regular cache, outside control group, hit in control cache
  - Miss in regular cache, outside control group, miss in control cache

Analytical Result



N = 2000 nodes, H_m = 100, H = 10, G_c = 200, β_c = 0.2, E = 100 pJ, R = 128 bit

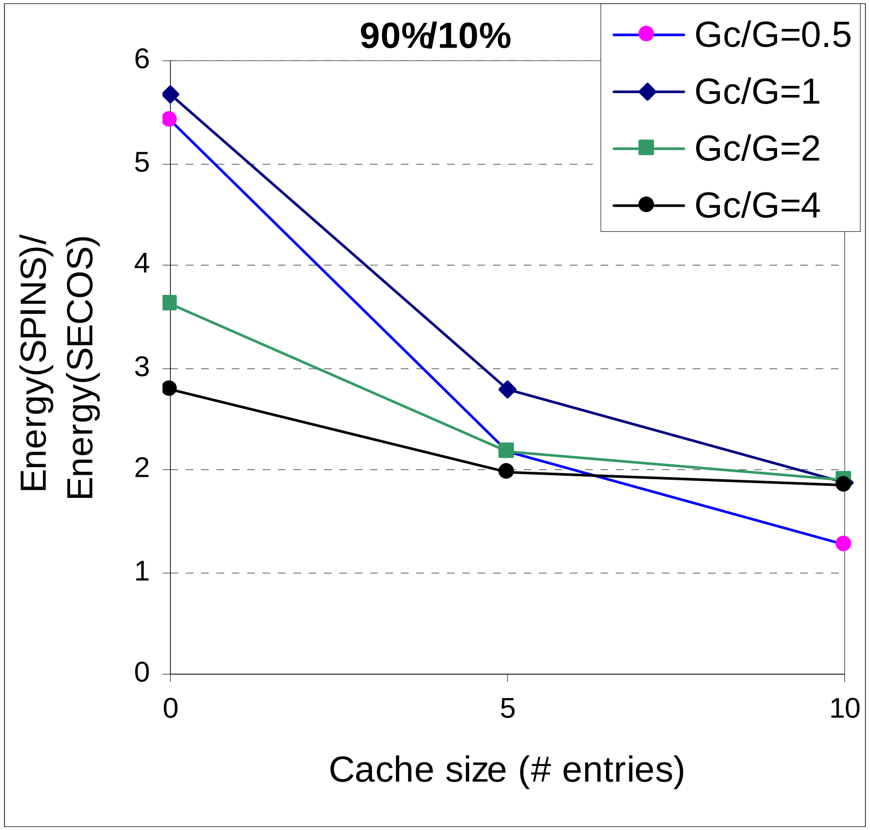
- Estimate for optimal point is size of control cache = number of control groups in a communication group – 1
- Operating point determined by energy wise optimal size and the max size given by resource constraints

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - **Simulation results**
- Take away lessons

Simulation Results

- Comparison with SPINS which uses base station as intermediary for node to node communication



N=200, μ=20 s, λ=5 s, G=10, C_C=5, τ_C=200 s, τ_S=200 s

- As cache size increases, SECOS and SPINS perform similarly
- Inter-group communication is more expensive in SECOS than SPINS
- It is important to choose the control cache size carefully

Conclusion

- Demonstrated a protocol called SECOS for energy efficient key management in sensor networks
- SECOS is resilient to different kinds of attacks – eavesdropping (discussed here), denial of service, and node compromise (discussed here)
- Claim: Compromising any number of nodes in the network does not compromise the session between two legitimate nodes
- Future Work:
  - Impact of neighbor watch on the energy efficiency of the protocol
  - Secure topology building and maintenance with SECOS

Outline

- Motivation
- Robust data dissemination
  - Background
  - Example protocol: SPIN
  - Our protocol: SPMS
  - SPMS: Failure scenario
  - Energy and delay analysis
  - Simulation results
- Secure communication
  - Background: Key management in sensors
  - Our protocol: SECOS
  - SECOS: Key elements
  - Communication within control group
  - Communication across control group
  - Analysis
  - Simulation results
- **Take away lessons**

Take Away Lessons

- Communication protocols in sensor networks have to be designed with
  - Failures in mind
  - Node compromise in mind
- Trade-offs exist between latency and energy consumption and customizable protocols that fit different regions of trade-off curve are desirable
- Desirable characteristics of large class of sensor network communication protocols
  - No privileged nodes
  - No node trusted completely

Questions Anyone?

Issa Khalil

Gunjan Khanna

Ness Shroff

- "*Fault Tolerant Energy Aware Data Dissemination Protocol in Sensor Network*," Gunjan Khanna, Saurabh Bagchi, Yu-Sung Wu. At IEEE Dependable Systems and Networks Conference (DSN 2004), June 28-July 1, 2004, Florence, Italy.
- "*Analysis and Evaluation of SECOS, A Protocol for Energy Efficient and Secure Communication in Sensor Networks*," Issa Khalil, Saurabh Bagchi, Ness Shroff. Submitted to Ad-hoc Networks Journal, September 2004. Available as CERIAS Tech Report from home page.