

# **A Hybrid Approach for Misbehavior Detection in Wireless Ad-Hoc Networks**

**S. Balachandran, D. Dasgupta, L. Wang**

**Intelligent Security Systems Research Lab  
Department of Computer Science  
The University of Memphis**

# Introduction

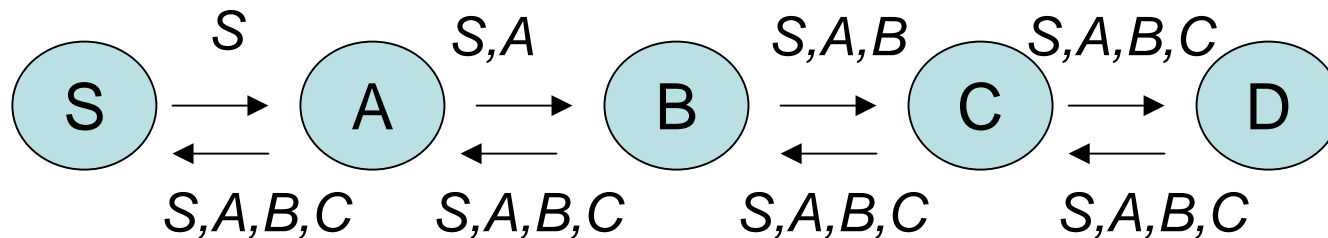
- Ad-hoc Networks are autonomous and dynamic (No Access points/Master-Slave relationship exist) network of mobile devices.
- Ad-hoc Wireless Networks - Nodes act as both data terminals and data transfer equipments. So all nodes are equivalent.
- The mobile nodes form an arbitrary topology where routers are free to move randomly and arrange themselves.

# Dynamic Source Routing Protocol (DSR)

- **Simple and efficient routing protocol designed for multi-hop wireless networks.**
  - requires no existing network infrastructure;
  - able to adapt to rapid topological changes.
- **Protocol Specifics**
  - Source first discovers the entire path to the destination (**Route Discovery**);
  - Packets from the source carry this path in their headers and intermediate nodes forward the packets to the nexthop in the path (**Source Routing**);
  - Source may need to rediscover the path if the current path becomes broken (**Route Maintenance**).

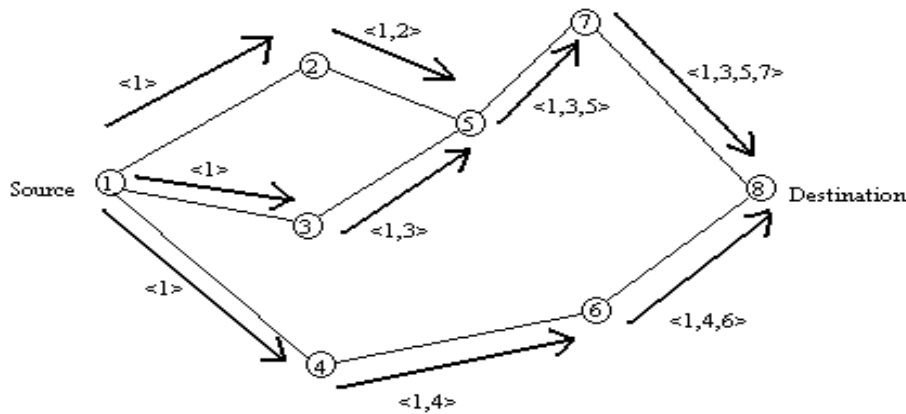
# Route Discovery Procedure

- Source S broadcasts a "Route Request" to all nodes within transmission range;
- Neighbor appends its own address to the route record in the request and broadcasts the packet;
- The process is repeated until the request reaches the destination;
- Destination D returns the path in a "Route Reply" to S;

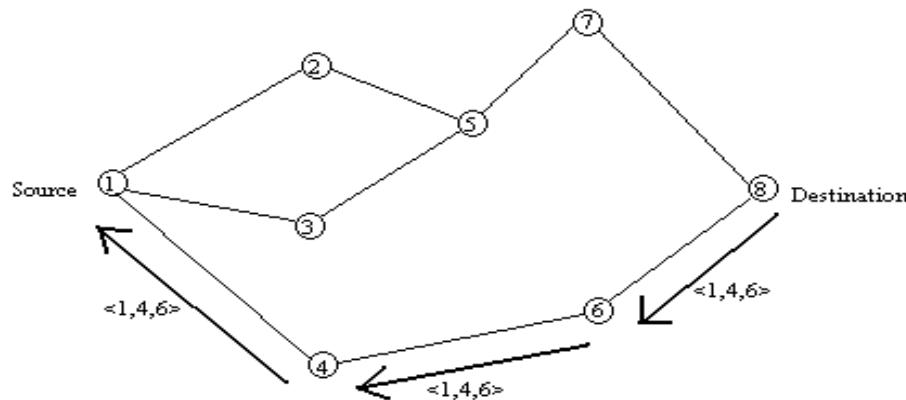


# Further Illustration of DSR

- DSR (Dynamic Source Routing)
  - Source Routed On Demand routing protocol



(a) Building Record Route during Route Discovery



(b) Propagation of Route Reply with the Route Record

- Route Discovery (Phase I)
- Route Maintenance (Phase II)

# Conflict Resolution

- **A intermediate node may receive multiple copies of the Route Requests from different neighbors**
  - Solution: forward the first one, discard the subsequent ones;
  - Why? the first one carries the shortest path (or the path with the shortest delay);
- **The source may receive multiple paths from the destination;**
  - Solution: cache all or some of them;
  - Why? we have a backup when the best path fails.

# Misbehaving nodes

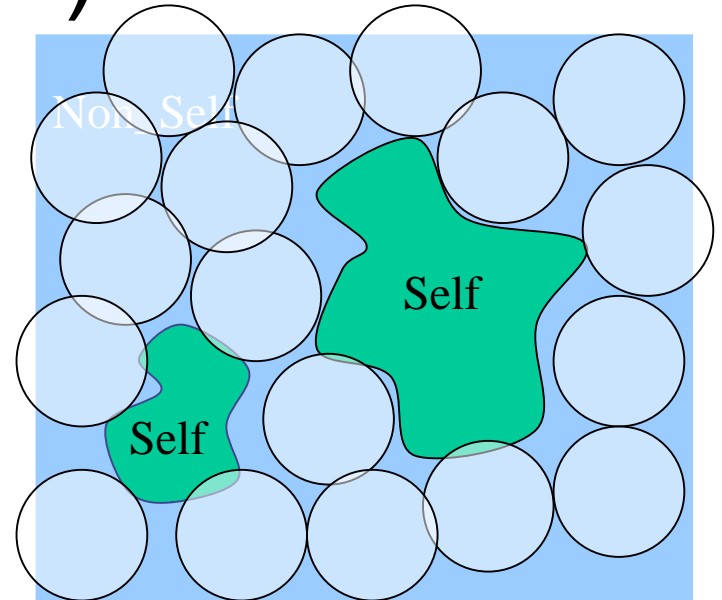
- Route Maintenance
  - Route Error Packet -> fatal transmission problem
  - Acknowledgements -> verify correct route links
- Security Reasons:
  - » Vulnerabilities - eavesdropping and spoofing of MAC address
  - » Lack of infrastructure – so impossible to assign well-defined roles such as trusted third parties.
  - » Lack of dedicated routers – Impact performance of packet forwarding and routing operations because of malicious network behavior.

# Types of Misbehavior

- Nodes neither forward route requests nor send the route replies.
- Nodes do not forward data packets.
- Simulation setting
  - No. misbehaving nodes: 5, 10 or 20
  - Misbehavior probability: 0.8 (a misbehaving node behaves badly 80% of the time in the simulation run).

# Negative Selection Algorithm (Forrest'94)

- There exist efficient algorithms that runs on linear time with the size of self (for binary representation).
- Efficient algorithm to count number of holes.
- Theoretical analysis based on Information Theory.



Given a problem space  $S$ , that represents the normal behavior of a system, the characteristic function of  $S$ , defined as

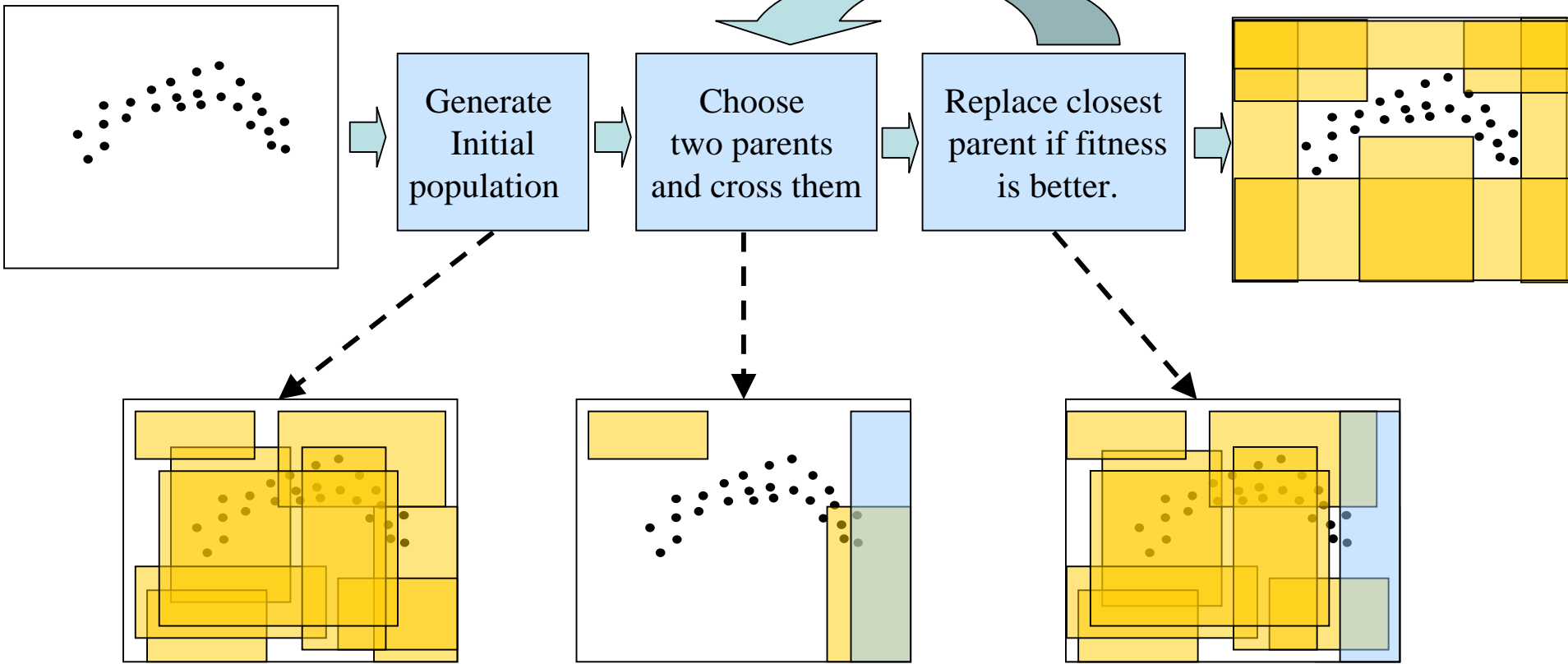
$$S \subseteq [0,1]^n, S \cup N = X, S \cap N = \phi$$

$$\chi_S(p) = \begin{cases} 1, & \text{if } p \in S \\ 0, & \text{if } p \in N \end{cases}$$

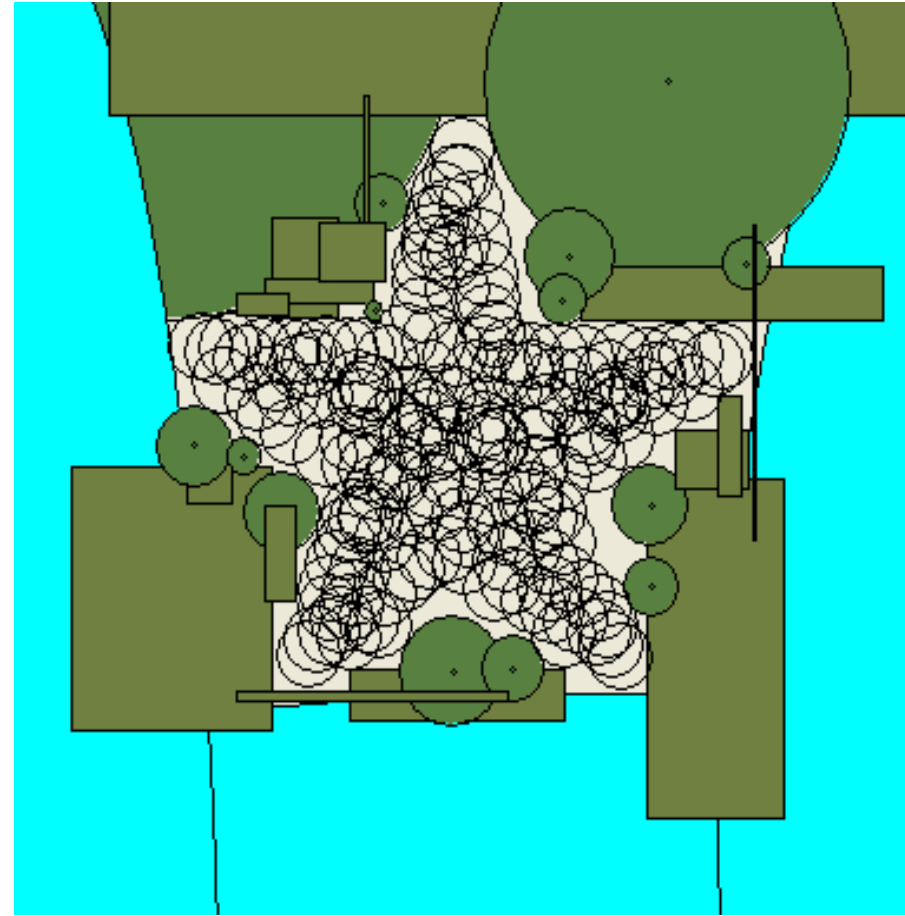
is used to distinguish between self and non-self.

# RNS Rule Evolution: Block Diagram

Self Data



# RNS Multi-shaped Detectors

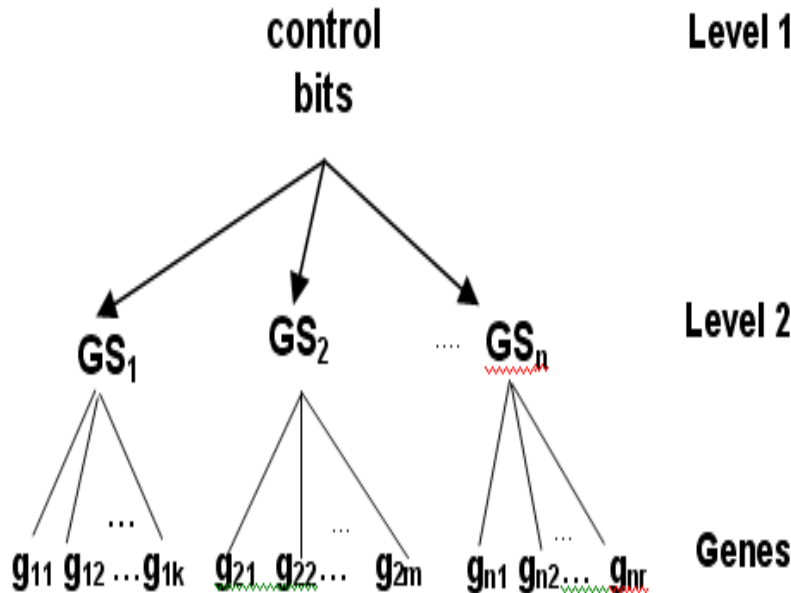


# Evolving Multi-shaped Detectors

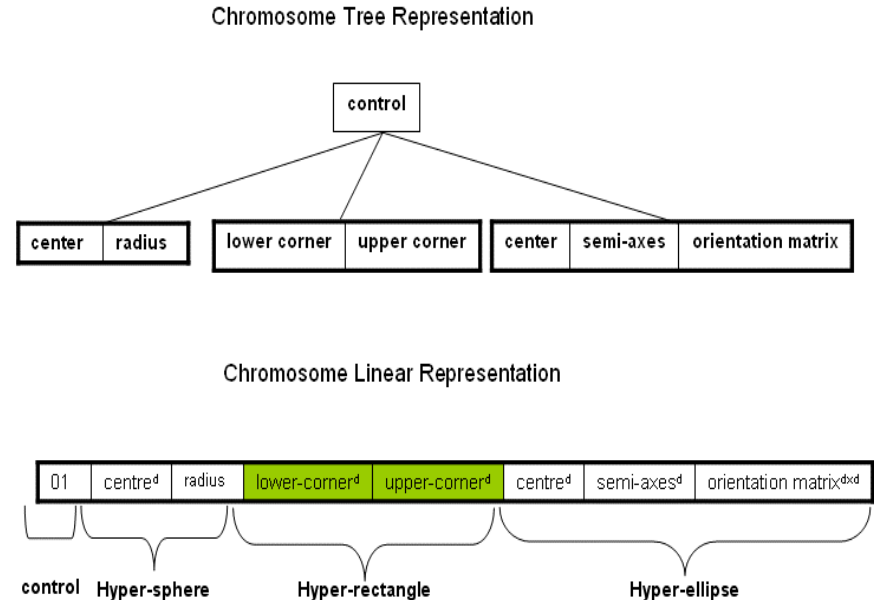
- Structured GA (sGA)

The sGA interprets the chromosome for evolution as a hierarchical structure, wherein genes at any level can be either active or passive, and high-level genes activate or deactivate sets of low-level genes.

Generalized representation of a chromosome with n different gene



A single chromosome - high level control and low level parameters for 3 different hyper-shapes;



# Objectives in Detector Generation

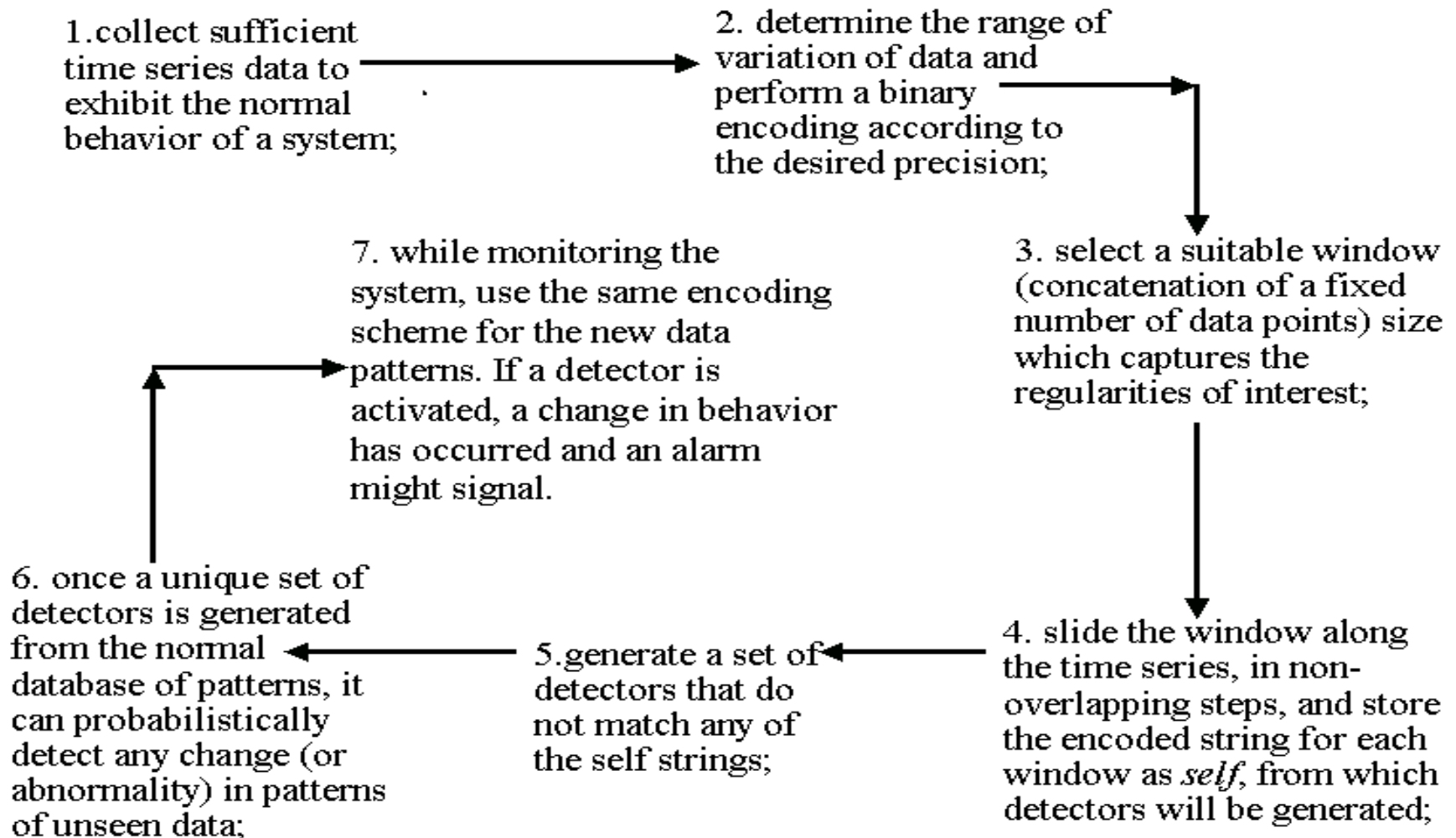
- No self space covered
- Minimize overlap among detectors
- Make the detectors as large as possible and keep them separate from each other, in order to maximize the non-self covering:
  - This is referred to as the coverage parameter in all our experiments;
  - This helps generalize the unknown or non-self space as closely as possible.

# Advantages of Negative Selection

- From an information theory point of view, to characterize the normal space is equivalent to characterize the abnormal space.
- Distributed detection: Different set of detectors can be distributed at different location
- Other possibilities
  - Generalized and specialized detectors
  - Dynamic detector sets
  - Detectors with specific features
  - Artificial attack signatures

# Attack/Misbehavior Detection

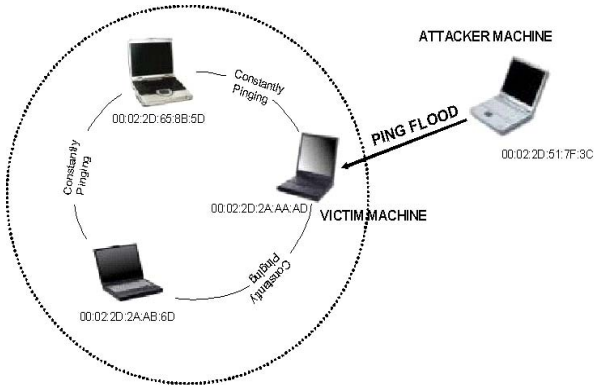
- Dasgupta & Forrest (1996) on time series data, based on the previously discussed *negative-selection algorithm*.



# WLAN attacks

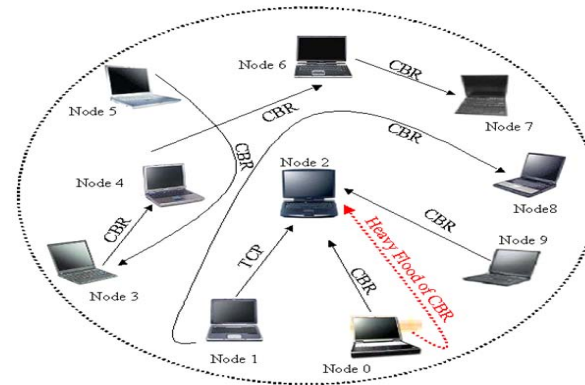
- Study Spoofing and Denial of service (Dos) attacks by examining the MAC layer sequence numbers.
  - The Network traffic subfields, Sequence Number of frame and the Fragment Number which is count of the number of fragments in the frame were analyzed.

# Prior experiments on WLAN



**Table 1.** The Detection result for mixed shaped detectors for various self threshold parameters. (**D.R.** is detection rate, **F.A.** is False Alarm rate, **F.N.** is False Negative is expressed as a percentage over the entire data points, **F.P.** is False Positive)

Ratio	Coverage	D.R.	F.A.	F.N.	F.P.
0.01	97.11	95.1	0	4.75%	0
0.02	96.4	96.45	0	4.05%	0
0.03	97.28	92.05	0	7.75%	0
0.05	98.2	87.1	0	11.57%	0

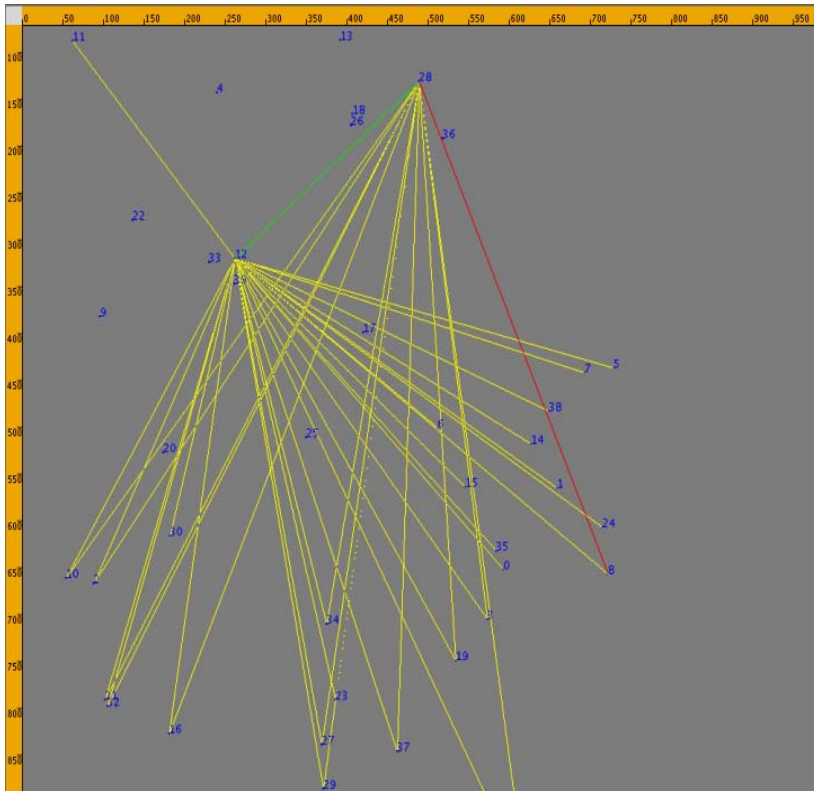


**Table 2.** The Detection result for mixed shaped detectors for various self threshold parameters. (**D.R.** is detection rate, **F.A.** is False Alarm rate, **F.N.** is False Negative, **F.P.** is False Positive)

Ratio	Coverage	D.R.	FA	F.N.	F.P.
0.01	99.9	92.75	0	6.50%	0
0.02	99.86	91.1	0	8.50%	0
0.03	99.775	87.1	0	14.10%	0
0.05	99.77	86.9	0	13.70%	0

# Simulation Environment

- Tool employed – GloMoSim



The visualization tool indicates packet transmissions among nodes within the power range.

Yellow link indicates links within range,

Green indicates successful reception

Red line is indicative of unsuccessful reception.

# Simulation: Parameter settings

Simulation System parameters		Detection System Parameters	
Parameter	Default value(s)	Parameter	Default value(s)
Routing protocol	DSR	Upper limit for Events sequence sets of a Monitored Node for learning	500
Simulation time	60 mins	Number of subsequences in a sequence set	4
Simulation area in metres	800x1000	Upper limit for the number of events in a sequence set.	40
Number of nodes	40	Upper limit for a sequence set collection	10s
Radio range	380 m	Misbehavior probability	0.8
Propagation Pathloss model	Two-ray	Learning data threshold	0.001 - 0.1
Mobility model	Random way point	Threshold for detection (% of Detection rate)	0.25
Mobility speed (no pauses)	1m/s	Mutation probability	0.05-0.1
Misbehaving nodes	5, 10, 20	Crossover probability	0.6
Traffic type	telnet, CBR	Normalized space range	[0.0, 1.0]
Payload size	512 bytes	Number of dimensions	4, 2
Frequency/rate	0.2-1s		
Radio-Bandwidth/link speed	2Mbps		

# Experimentation - Data Collection and Preprocessing

Packets Transmitted		Packets Received	
Label	Event Type	Label	Event Type
A	RREQ	E	RREQ
B	RREP	F	RREP
C	RERR	G	RERR
D	DATA sent and IP address not of monitored node	H	DATA received and IP destination address is not of monitored node

Collected Sequence example **T**: (EAFBHHEDBHDHDHHDHD)

Subsequences through the following expressions

Expression1:- number of E in sequence (**T**); **#E**

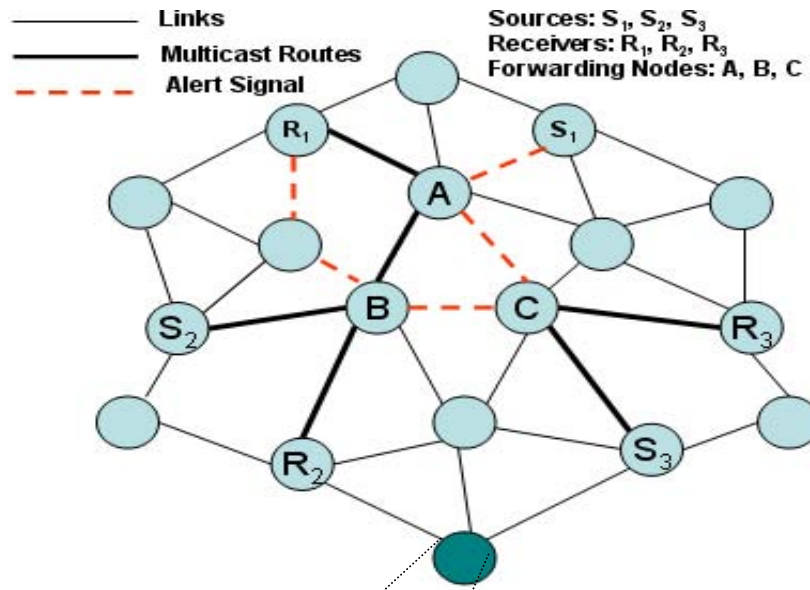
Expression2:-number of E with 1or no event followed by either an A or a B in sequence (**T**); **#(EX(A|B))**.

Expression3:- number of H in sequence (**T**); **#H**

Expression4:- number of H with one or none event followed by a D in sequence (**T**); **#(H?D)**

Subsequences evaluation through the expressions to yield a 4 dimensional vector as shown: **S = (3 2 7 6)**.

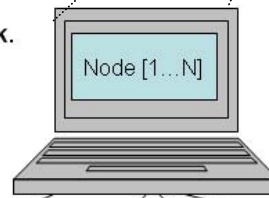
This series of vectors are used both for training as well as detection cases.



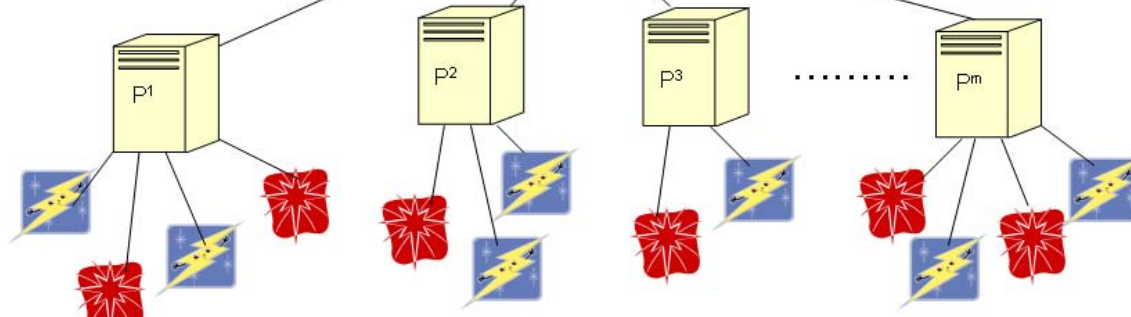
A simplified scenario of a multi-hop ad-hoc wireless network nodes (a single node enlarged showing the basic elements for consideration in a wireless network for attack detection).

There are 1 to N nodes in the network.

Each protocol has a set of events associated with itself which may be different for different protocols.



Each is associated with several different protocol layers.



$$P^1 : e^1_1, e^1_2, \dots, e^1_k ; P^2 : e^2_1, e^2_2, \dots, e^2_k \dots P^m : e^m_1, e^m_2, \dots, e^m_k$$

# Experimentation — Network Event Profile

Shows a snapshot of a profile (logical) collected during the simulation run. **S** denotes an event sequence creation while “-” denotes an absence of any sequence.  $\Delta t$  is used for sampling.

Node 0		$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$	$\Delta t$
	Node 1	S	S	-	-	S	-	S	-	-	-
	Node 2	-	S	S	-	-	S	-	-	S	S
	.	.									.
	.	.									.
	Node N	S	-	-	S	-	-	-	-	-	S
Node 1	Node 0	-	-	-	-	S	S	-	-	-	-
	Node 2	S	-	-	S	S	-	-	-	-	-
	.	.									
	.	.									
	Node N	-	-	-	S		-	-	-	-	S
.											
.											
Node N	Node 0	S	-	-	-	-	-	S	-	S	S
	Node 1	-	-	-	-	S	-	S	-	S	-
	.										
	.										
	Node N-2	-	-	-	S	S	-	-	S	S	-
	Node N-1	-	-	-	-	-	-	S	S	S	S

# Experimentation - Objectives

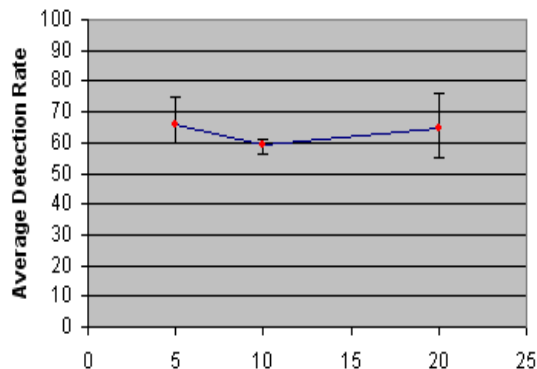
- ***Using All 4 Event types (all 4 parameters)***: - This analysis involves the misbehavior detection for the nodes that are neither replying from its route cache nor forwarding route requests as well as non forwarding data packets.
- ***Using more prominent event types (last 2 parameters)***:- This analysis is more significant and pronounced if the initial route set up for the monitored node is not taken into account for misbehavior. It concerns more those misbehaviors that happen during the **data packets' transfer among the nodes**. This is more crucial for detecting those misbehaving nodes that do not forward data packets. Unless this assumption is made in the simulation, this analysis will fail.

# Experiment - Performance Metrics

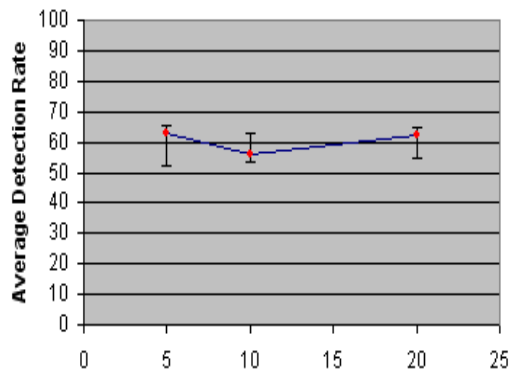
- Average detection rates for the misbehaving nodes; defined as  
**detection rate (D.R.) = (true positives) / (true positives + false negatives).**
- Average False Alarm rates for misclassifying the well behaving nodes;  
defined as  
**false alarm rate (F.A.R) = (false positives) / (false positives + true negatives).**

# Experimental Results

Learning Threshold = 0.002



Learning Threshold : 0.01

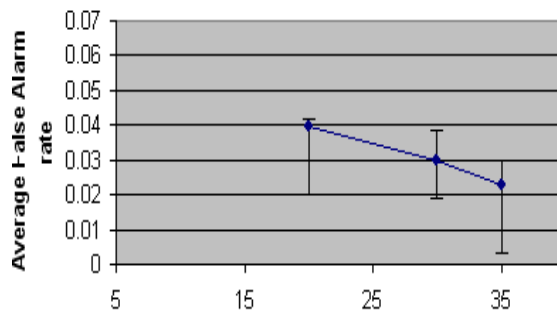


Misbehaving Nodes

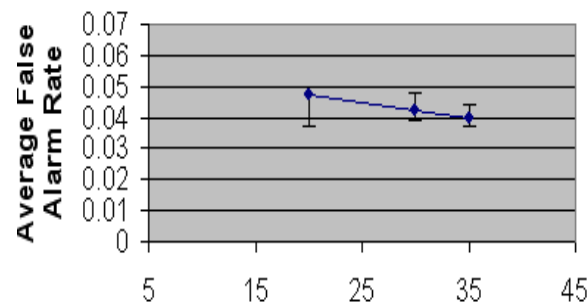
Misbehaving Nodes

Average Detection Rates for all misbehaving nodes - 5, 10 and 25 with learning thresholds of 0.002, 0.01. The Learning Threshold parameter has a significant effect on the detection rate in all three cases.

Learning Threshold = 0.002



Learning Threshold = 0.01



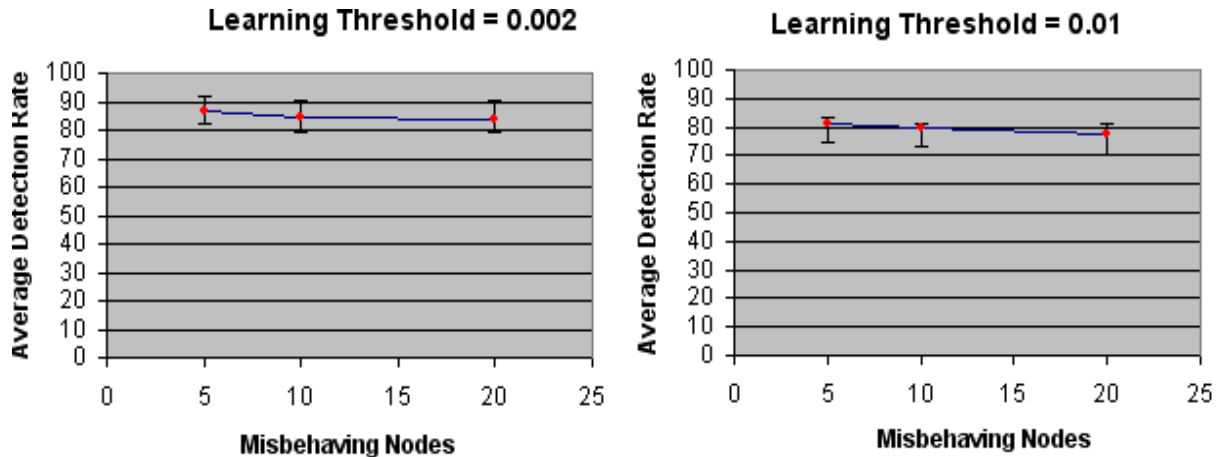
Well Behaving Nodes

Well Behaving Nodes

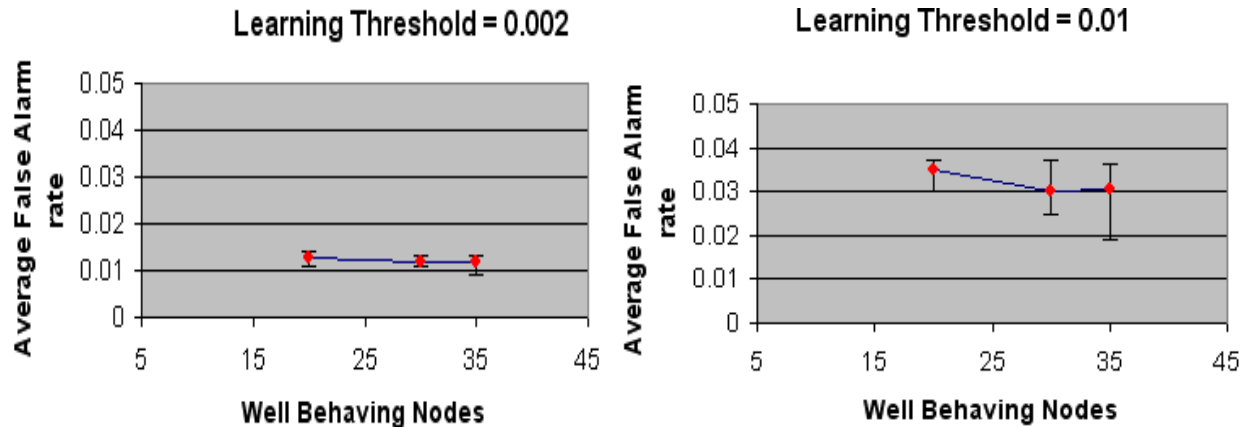
Average misclassification (false alarm rates) for well behaving nodes - 20, 30 and 35 with learning thresholds of, 0.002, 0.01. The false positives are within the upper limit of 0.03 in most cases.

Using all 4 Parameters

# Experiment Results



Average Detection Rates using partial parameters for learning (ignoring the routing setup data) misbehaving nodes - 5, 10 and 25 with learning thresholds of 0.002, 0.01. The Learning Threshold parameter has a significant effect in each of the cases.

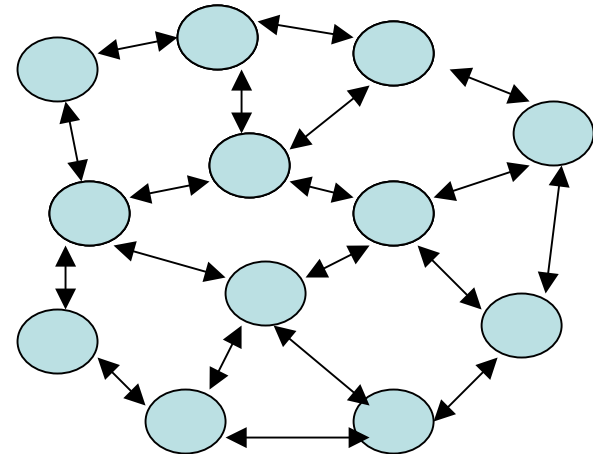


Using Final 2 Parameters

The average misclassification or false alarm rates for the partial events' sequence are within an upper limit of 0.02 in most cases

# Ad-hoc WLAN M&R

- Apply Cellular automata Concepts:
  - neighborhood topology.
  - Local interaction
  - React to changes in their neighbors.
- Challenging Issues:
  - Secure communication.
  - Synchronization.
  - Remote execution.



# Conclusion

- Experiments are performed in simulated environment to detecting the misbehaving nodes by examining the entire network profile.
- successfully identify faulty nodes that, from time to time, suffer from node misbehavior (due to faulty software, hardware/firmware or does not forward packets or are disastrously infected with Internet worms and viruses that maliciously attempt to bring the network down.)
- Behavior-based approach was used that could detect new/unforeseen vulnerabilities and can even contribute to the (partially) automatic discovery of new behaviors.
- Results demonstrated that the our approach were able to detect routing misbehavior

# Intelligent Security Systems Research Lab (ISSRL)

(<http://issrl.cs.memphis.edu>)

At

Center for Information Assurance  
The University of Memphis

- Offering security-related courses
- Developing distributed security agent software (using various Intelligent Techniques) for automated intrusions/anomaly detection and response.



# ISSRL SECURITY PAPERS

- *An Immunogenetic Approach to Intrusion Detection.* In IEEE Evolutionary Computation, June 2002.
- *Evolving Fuzzy Rules for Intrusion Detection.* In Proceedings of Information Assurance Workshop, June 2002.
- *An Intelligent Decision Support System for Intrusion Detection and Response.* Lecture Notes in Computer Science (publisher: Springer-Verlag) May 2001.
- *An Administrative Tool for Distributed Security Task Scheduling.* In Third Annual International Systems Security Engineering Association Conference, March, 2002.

# ISSRL SECURITY PAPERS (Continued)

- *Mobile Security Agents for Network Traffic Analysis.*

In the proceedings of the second DARPA Information Survivability Conference and Exposition II (DISCEX-II), 13-14 June 2001 in Anaheim, California.

- *Immunity-Based Intrusion Detection Systems: A General Framework.* In the proceedings of the 22nd National Information Systems Security Conference (NISSC), October, 1999.

- *Cougaar based Intrusion Detection System (CIDS).*

CS Technical Report No. CS- 02- 001 February 4 2002.