

Wireless Sensor Network Performance Monitoring

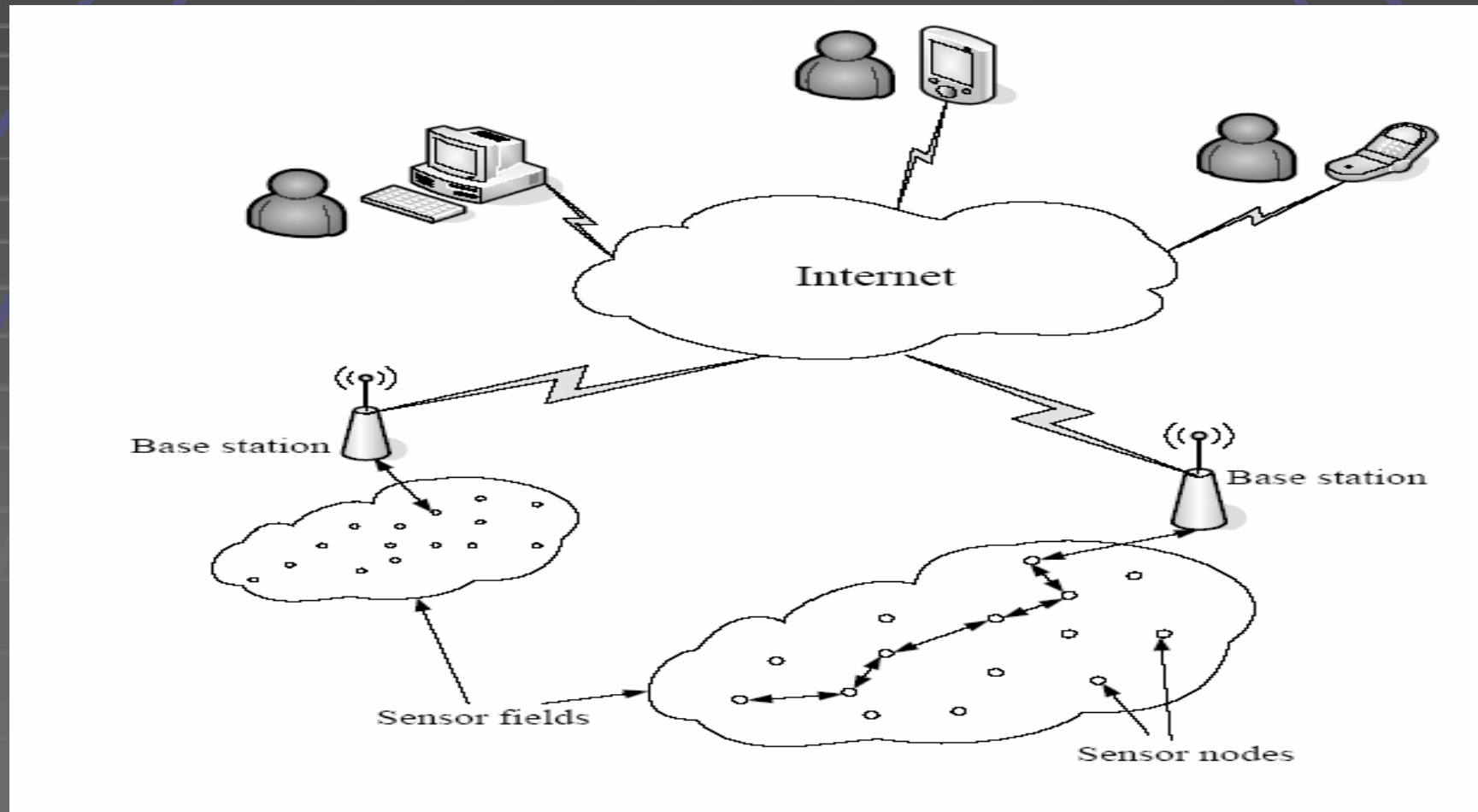
Yaqoob J. Al-raisi & David J. Parish
High Speed Networks Group
Loughborough University

MSN Coseners
12-13th July 2007

Overview

- The problem we are trying to solve and why it is exist.
- How it has been solved by others and solutions' limitations.
- The proposed algorithm characteristics, novelty and function.
- Empirical and simulation experimental results.
- Conclusion and future work.

Wireless Sensor Networks (WSNs)



Characteristics

- Small, disposal, low power consumption network nodes.
- Reduce network and deployment cost.
- Enable network to be deployed in any terrain.
- Extend the virtual functionality of traditional networks.
- Increase the reliability of network collected data.

Challenges

- Limited energy and communication resources.
- No dominant protocols for all network applications.
- Frequent changes in connectivity, link failure and the change node status.
- Unbalanced network traffic.
- Change the degree of network's tolerance to data changes and losses.

Deviation of nodes' operation impact

- Use more network resources and reduce network lifetime.
- Reduce the reliability of collected data.

Related work

- Fault-Tolerant Tools
- Data Cleaning Tools
- Diagnosis tools
- Performance monitoring/measuring Tools
- Used only one level parameter
- Definition of outlier/fault
- Used method (either complex or not considering loss)
- Consume resources in packet exchange
- Clear data without indicating the confidence level
- Map of common neighbourhood function

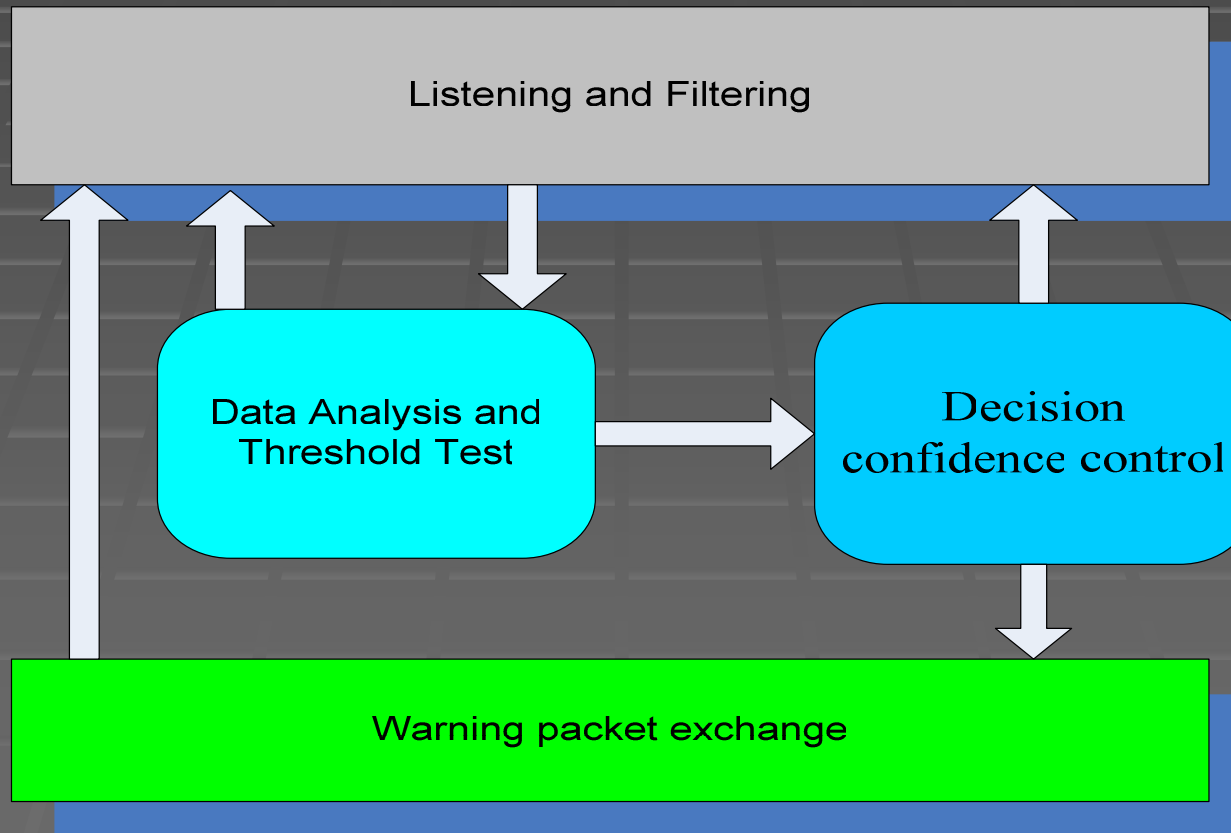
Main method limitations

- The impact on network lifetime
- Tracking of crash faults.
- Tracking of either application- or communication-level parameters.
- The generated warning packets.

Main study objective

- To develop a low resource usage tool that tracks network nodes' operation deviation 'on-line' before they degrade collected data reliability and network lifetime.

Algorithm



Algorithm Characteristics and novelty

- The utilization of all neighbour nodes' information in range.
- The extraction of its metrics from network protocols' stored parameters.
- The use of a simple statistical method with tolerance to loss.
- The definition of tracked deviation.
- The relation between high and low network levels by testing the distortion.
- The test of possible communication available.
- The positive tracking of neighbour detection and control of warning packet release.

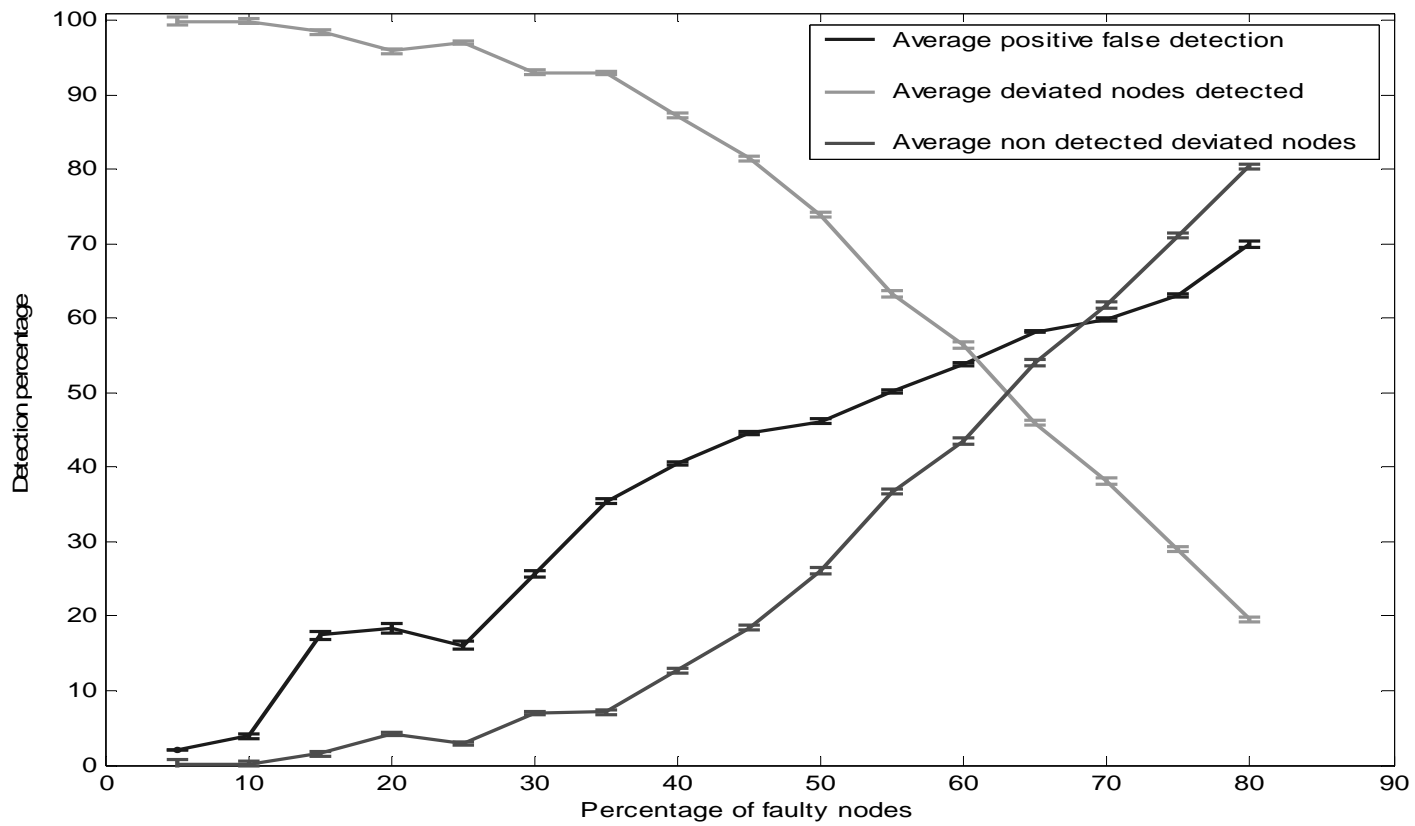
Algorithm event detection

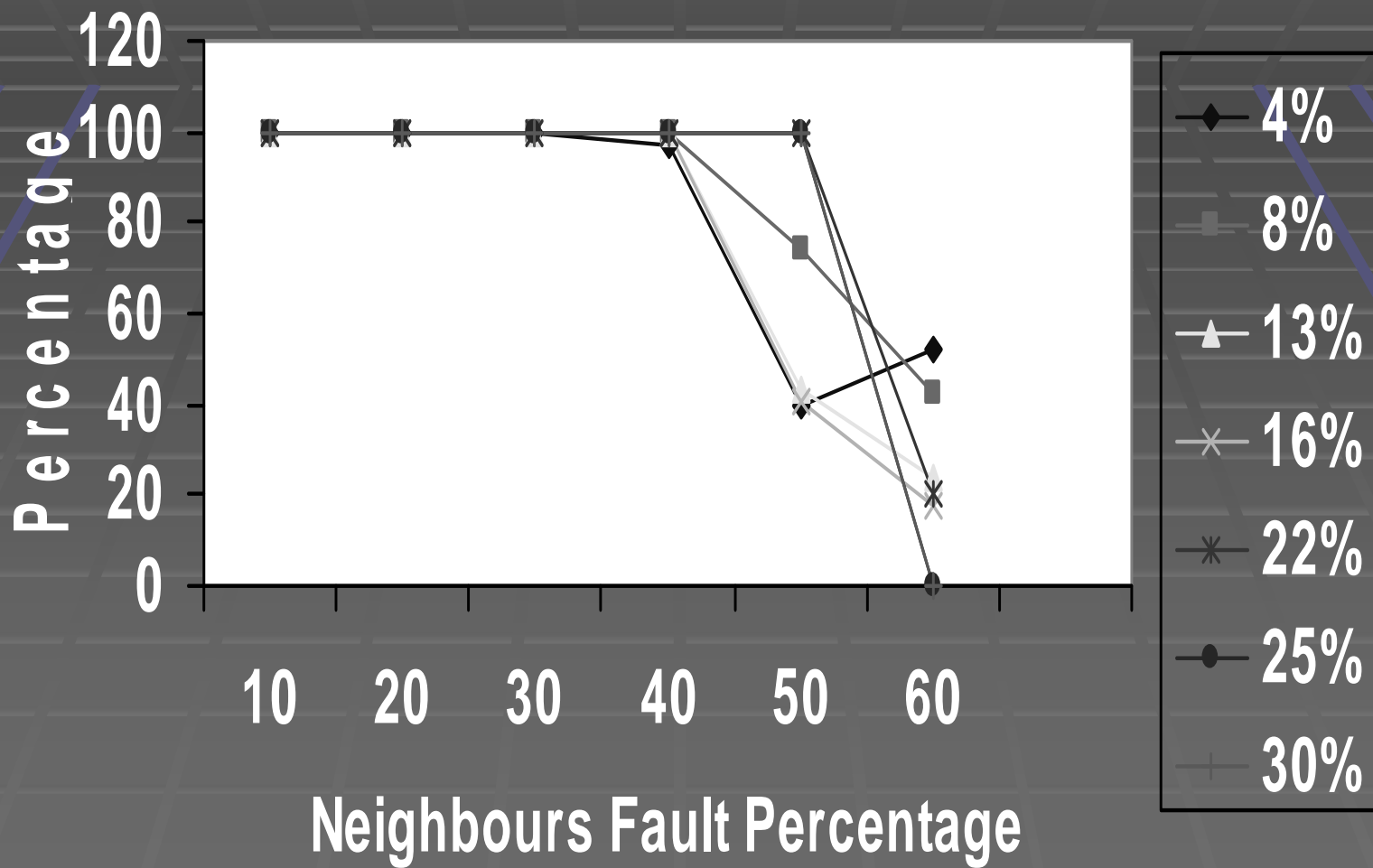
- Node malfunction.
- Neighbourhood malfunction.
- Coverage problem.
- Inefficient power consumption.
- Communication problem.

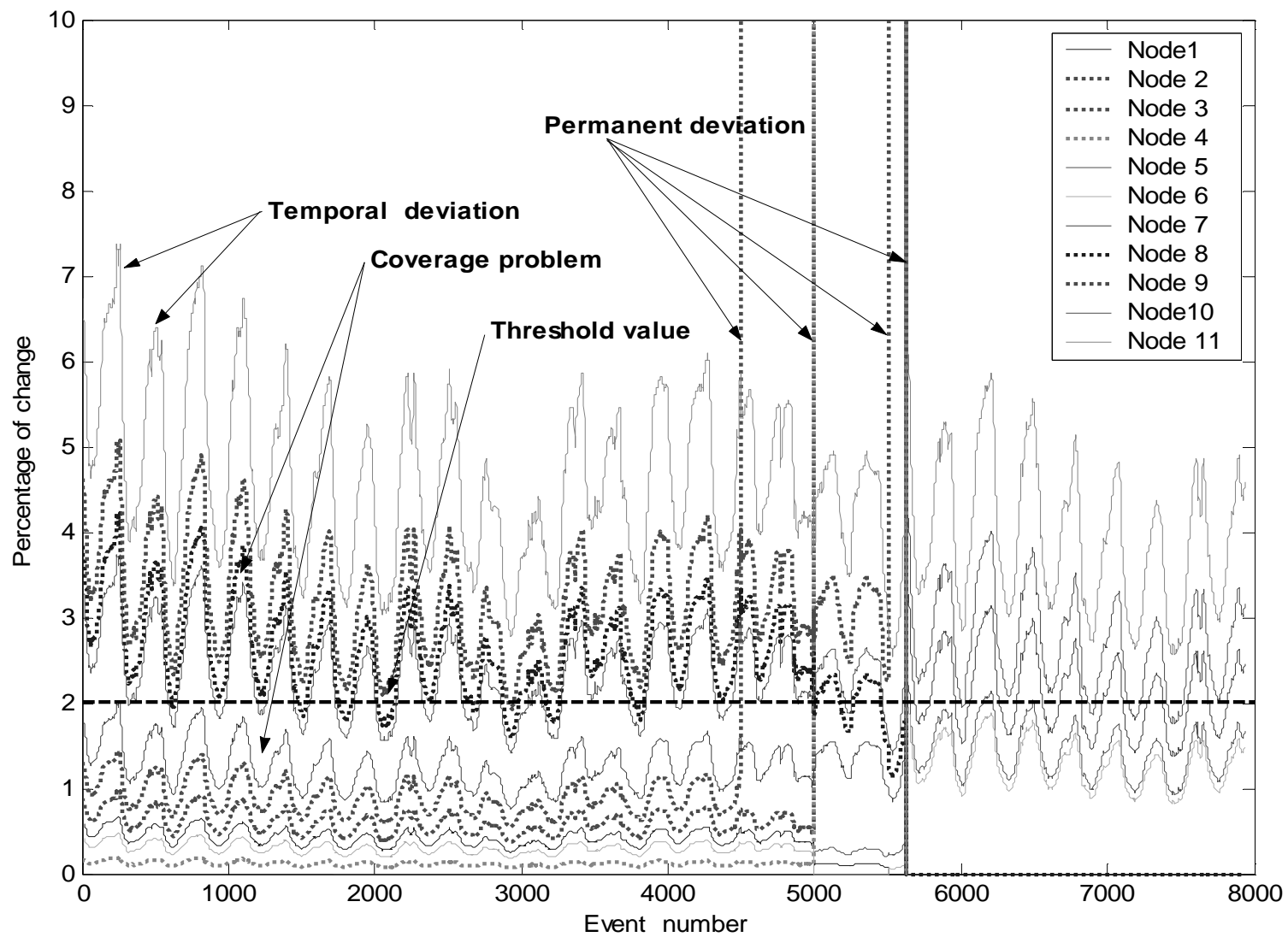
Algorithm performance evaluation

- Lightweight operation (energy consumption, algorithm analysis)
- Robustness (tested under different packet loss, deviated nodes, deviated measurements, dead nodes)
- Adaptability and responsiveness (empirical experiments in single and multi-hop configuration).
- Scalability.

Algorithm Detection

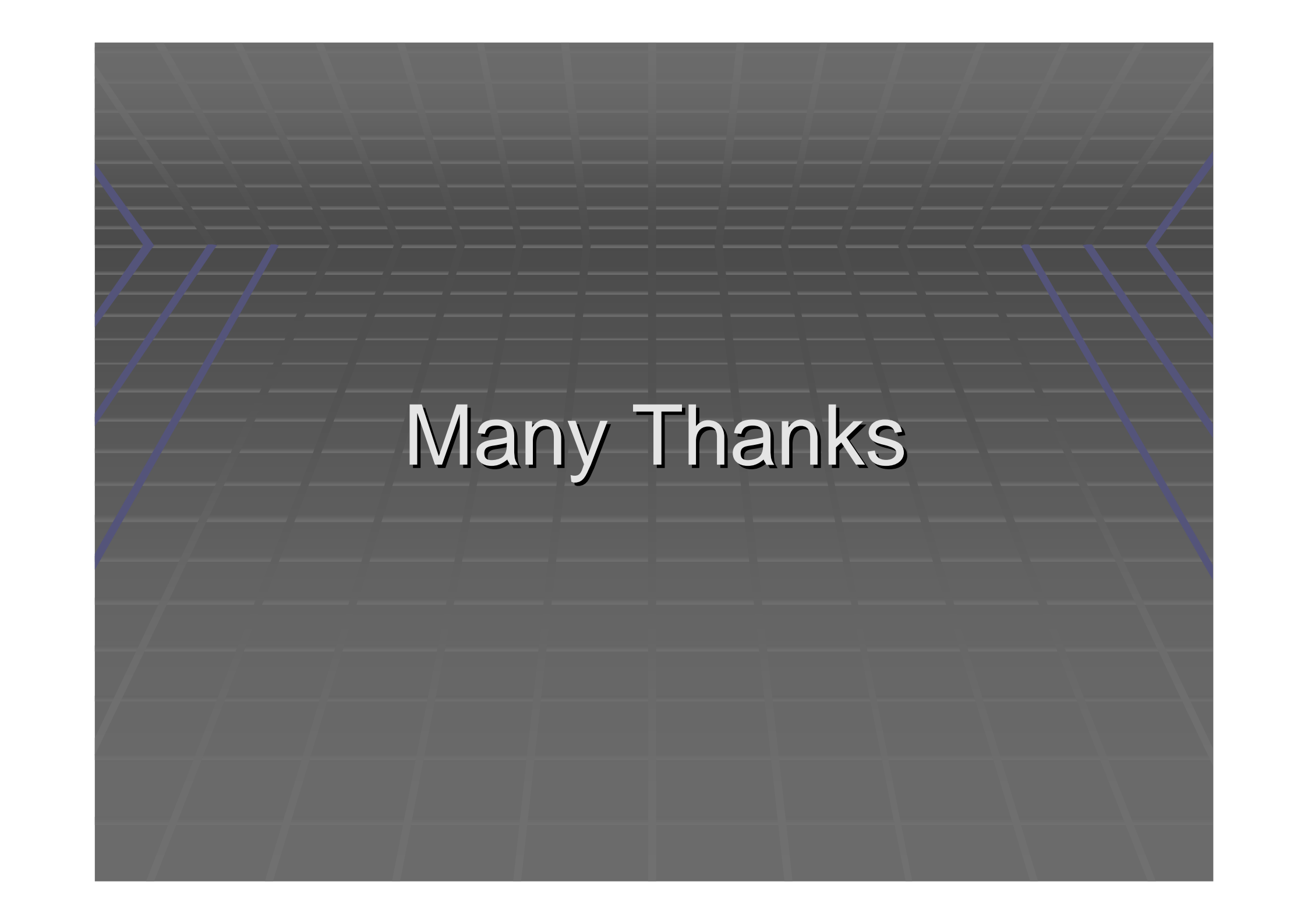






Conclusion and future work

- The experiments showed the efficiency of the passive proposed algorithm in detecting the deviations and low power consumption.
- The experiments detect degradation of the algorithm functionality in very small networks.
- Solve this limitation by cause and effect diagram.
- Test the algorithm in different network applications other than environment monitoring.
- Test the effect of node mobility on the proposed algorithm performance.



Many Thanks

Why it exists

- Low manufacturing material quality and process.
- Limited usage of fault-tolerant and diagnosis techniques.
- Harsh environment the nodes operate in.
- Direct coupling with monitored phenomenon.
- Usage of wireless medium.

Type of deviation

- Systematic: affect the operation cautiously until the problem is solved such as calibration, reduction in operating power.
- Transient: affect the operation until the effect disappears such as random environmental effects and unstable characteristics of hardware.

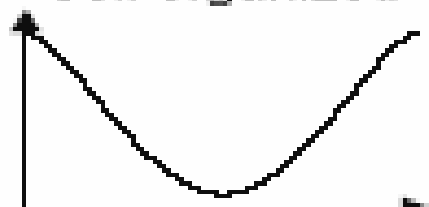
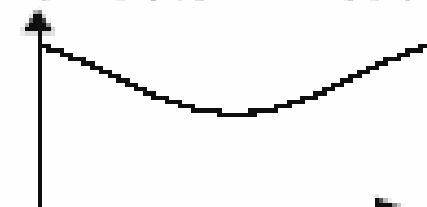
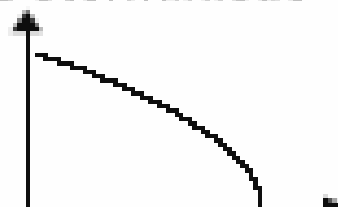
Effect on the quality and quantity

Direct effect	Indirect effect
<ul style="list-style-type: none">▪ Sensor node measurement such as biased, drift and stuck.▪ Dropping exchange packets.▪ Reboot or stop node function.	<ul style="list-style-type: none">▪ Usage of more resources.▪ Reduce the quality or quantity of collected data.

Deterministic

Non deterministic

Self organized



High (in the sink)

Low

Network intelligence

Processor capacity and memory size

High

Low

Accuracy

Raw

Aggregated

Data

Low

High

Redundancy

Low

High

Noise

Low

High

Losses

High

Low

Node price

20

100

100- millions

Number of nodes

The problem we are trying to solve

Maintain the large scale Wireless Sensor Network's collected data quality and quantity at acceptable level.

Algorithm Detection Event	Description	Metrics used to recognize event
Node malfunction	Node measurements deviated from neighbourhood median threshold for monitor window.	Neighbour measurements; neighbour losses.
Coverage detection	Change in correlation between group of neighbour nodes.	Neighbour measurements.
Temporal coverage change	The algorithm detects and releases node malfunction several times within monitor window.	
Neighbourhood malfunction	High variation of neighbourhood median for monitor window.	Neighbour measurements.
Neighbourhood accuracy degradation of collected data	Degree of distortion of collected data accuracy due to loss of neighbour node.	Neighbour losses, neighbour measurements.
Aliveness	No packet received for monitor window.	Neighbour losses.
Connectivity degradation/ Connectivity instability	Neighbourhood median loss is more than 70%. The connectivity is unstable if the algorithm detects frequent disconnection of link between two nodes for three continuous monitored windows.	Neighbour loss.

Listening and filtering module

- Integrate the algorithm functionality into network application flow process.
- Use network application's existing parameters.
- Filter the high deviated parameters.

```
1: Each  $S_i$  sense the phenomenon and wait for time  $T$  to receive  $N(S_i)$  readings
2: IF  $t > T$  THEN
3:   For each unreceived  $x_j^i$  increment  $L_j^i$ ;
4:   IF  $C_{L_j^i} > x_j^i > C_{M_j^i}$ 
5:     Remove  $x_j^i$  from data set and increment  $D_j^i$ 
6:   Calculate  $med_j^i$  of the available  $S_i$  data set
```


Data analysis and threshold test module

- Test median value validity
- Calculate the residual of each neighbour in neighbourhood.
- Test each residual value to predefined threshold.
- Test the validity of data.

```
1: IF  $|med_i - med_{i-1}| > \Delta med$   
   Increment  $M_i$  and let  $med_i = med_{i-1}$   
2:    $d_j = |med_i - x_j^i|$   
3:     IF  $d_j > \Theta_1$  and  $|x_i^i - x_j^i| < \Theta_1$   
4:       Increment  $COV_j^i$   
5:     ELSE increment  $R_i$   
6:       IF  $\frac{R_i}{k} > 40\%$   
7:         Increment  $N_i$   
8:         IF  $\frac{R_i}{k} * d_j > \Theta_1$   
9:           Increment  $D_j^i$ 
```

Decision confidence control module

- Calculate the frequency of deviation existence with the tolerance of the network application protocol
- Request from module 4 to send warning message.

```
1: Calculate  $ML_i$ 
2: IF  $ML_i > 60\%$ 
3:   Send to module 4 a request to send an inefficient power consumption warning message
4:   IF  $M_i > \Theta_M$ 
5:     Send to module 4 a request to send a neighbourhood malfunction due to losses warning message
6:     IF  $COV_j^l > \Theta_C$ 
7:       Send to module 4 a request to send to detecting node j a coverage problem message
8:       IF distortion  $> \Theta_d$  & median of  $L_j^l > 60\%$ 
9:         Send to module 4 a request to send a degrade detection in network functionality message
10:        IF  $D_j^l > \Theta_w$ 
11:          Send to module 4 a request to send a detection of node j malfunction message
```

Warning packet exchange module

- Test if any neighbour sends the same message.
- Test received neighbour warning message.
- Send warning detection message
- Control packet releasing
- Request the protocols to reconfigure the node setting.

1: Receiving neighbour warning

- a) Check received warning with the same module 3 counter of reported node.
- b) IF module 3 counter < 30%
- c) Release 'NO_EVIDENCE_OF_FAULT' message
- d) ELSE flag the stop sending of the same message from the node at this monitoring time.

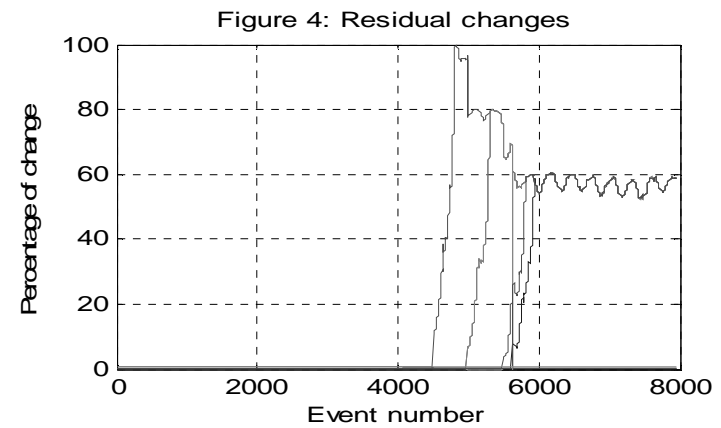
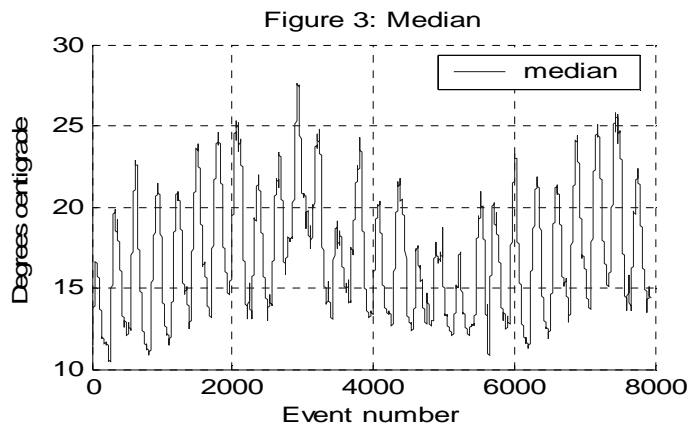
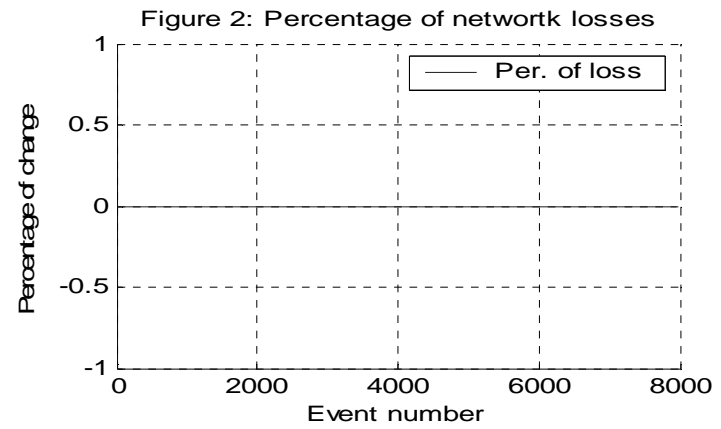
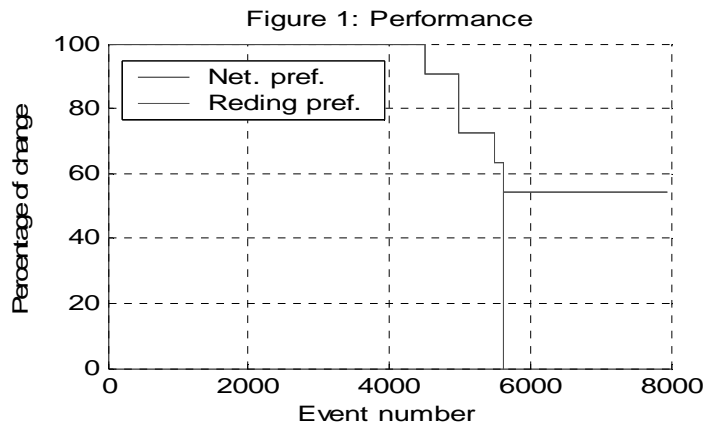
2: Receiving module 3 request

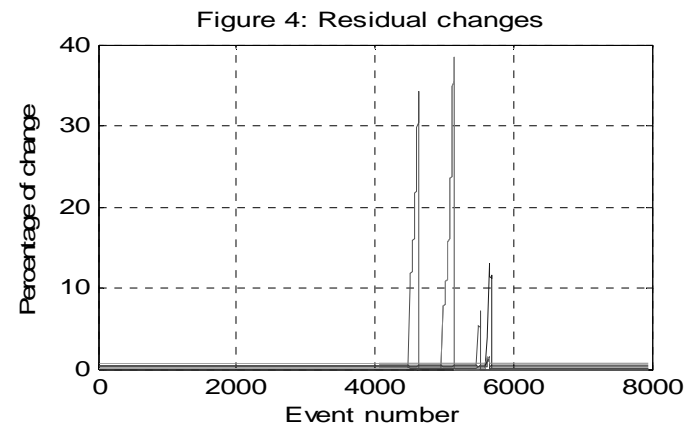
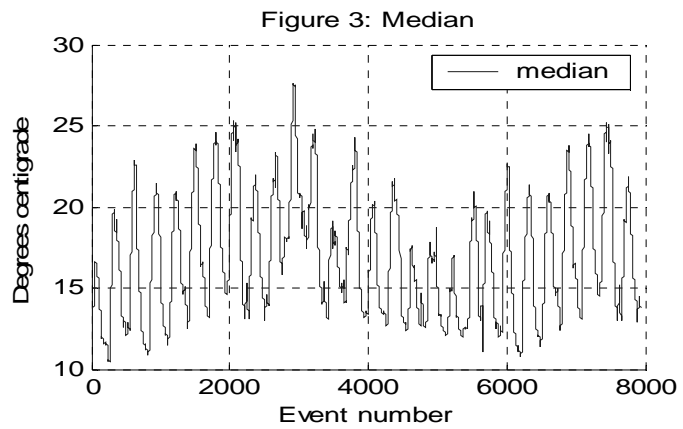
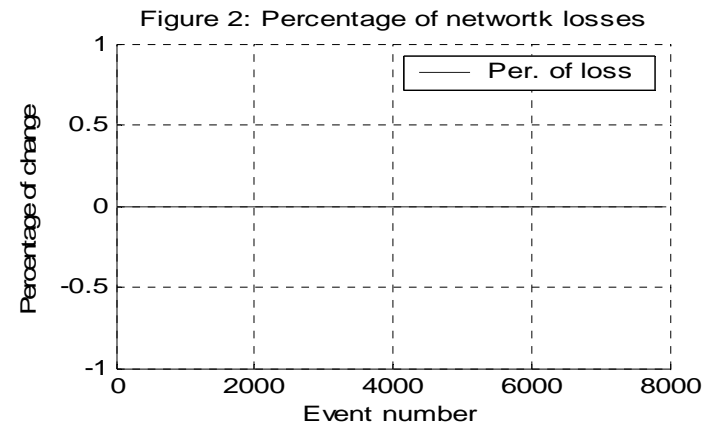
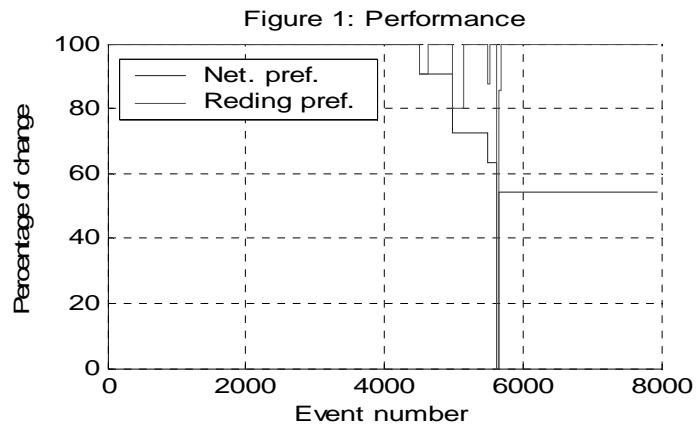
- a) Test stop flag of received request warning
- b) IF flag = 1 discard message
- c) IF send message repeated 3 times send 'FAULT_MESSAGE_STOP' message and flag stop fault counter.
- d) ELSE send the requested message by module 3.

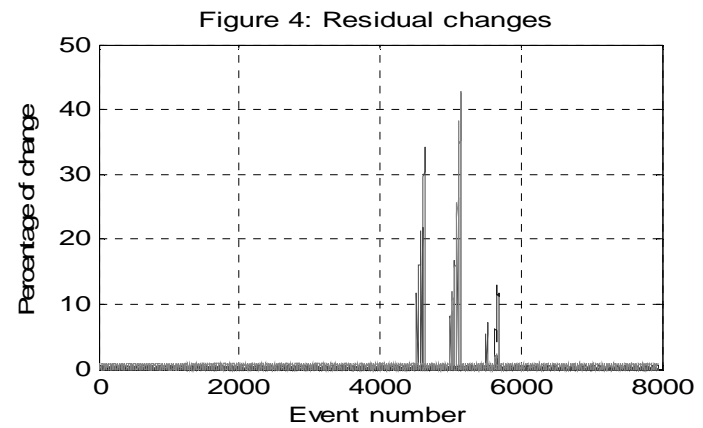
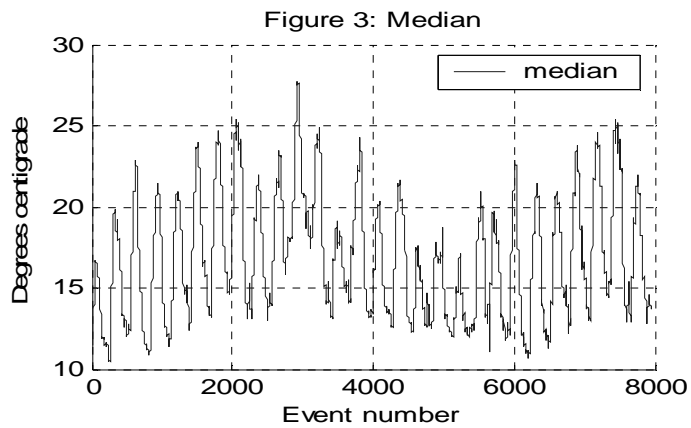
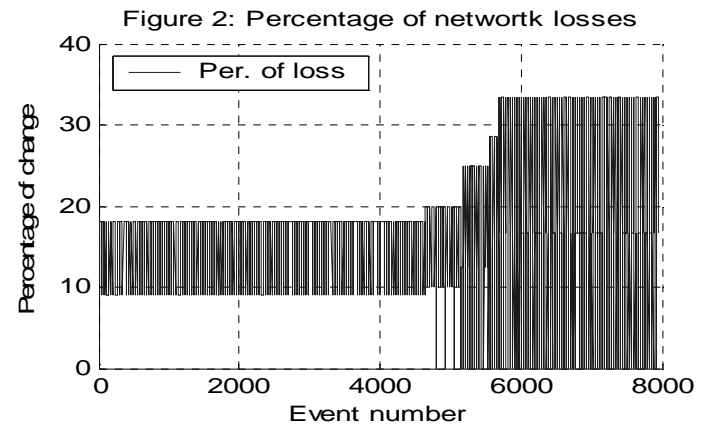
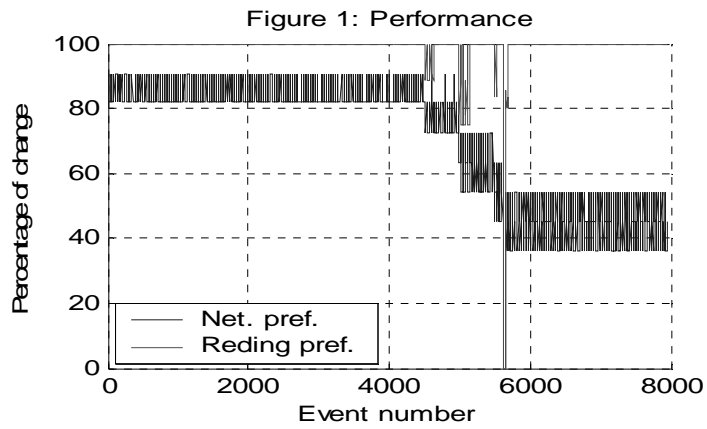
3: Testing warning packet release

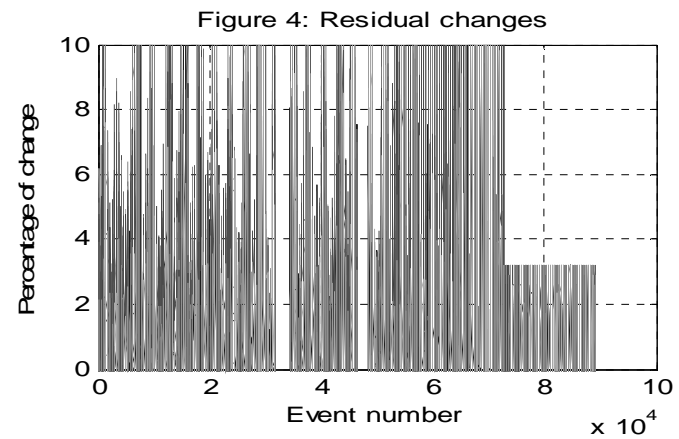
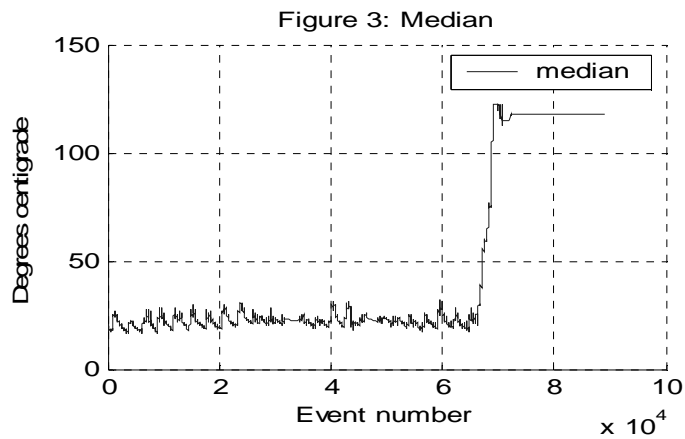
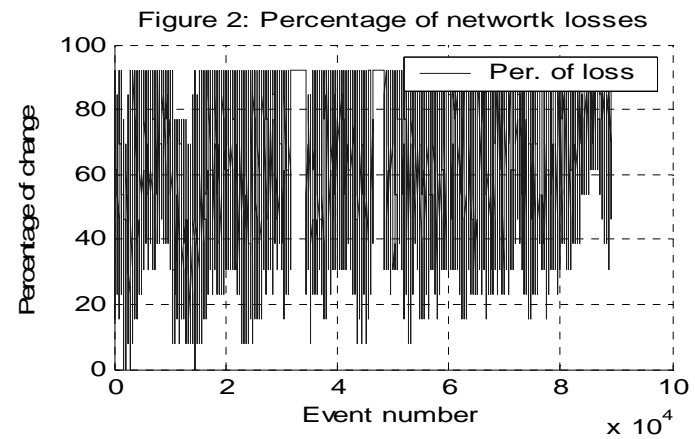
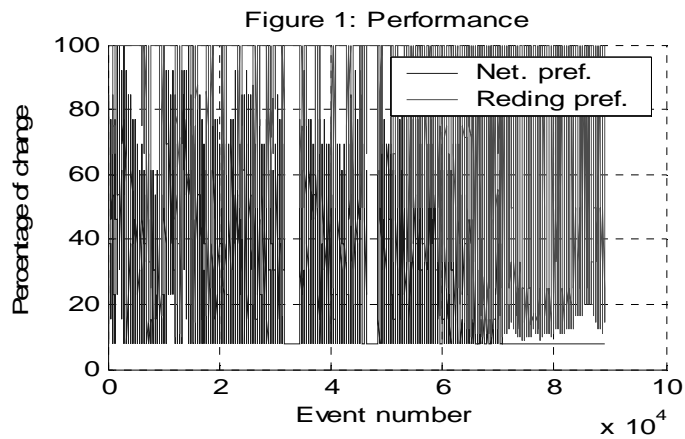
- a) IF detected fault returns to normal reset the same fault counters, send 'FAULT_CLEAR' message and recalculate protocol tables.
- b) IF step 2 and 3-a alternate for the same fault three times in a predefined monitoring window, the module sends an 'TOPOLOGY_UNSTABLE' message to report the detection and flags a permanent fault counter to stop reporting the same fault.
- c) If the number of 'NO_EVIDENCE_OF_FAULT' messages in the neighbourhood exceeds 1, then the warning message intended to be sent to the sink is dropped.

4: By the end of the predefined period reset all counters.

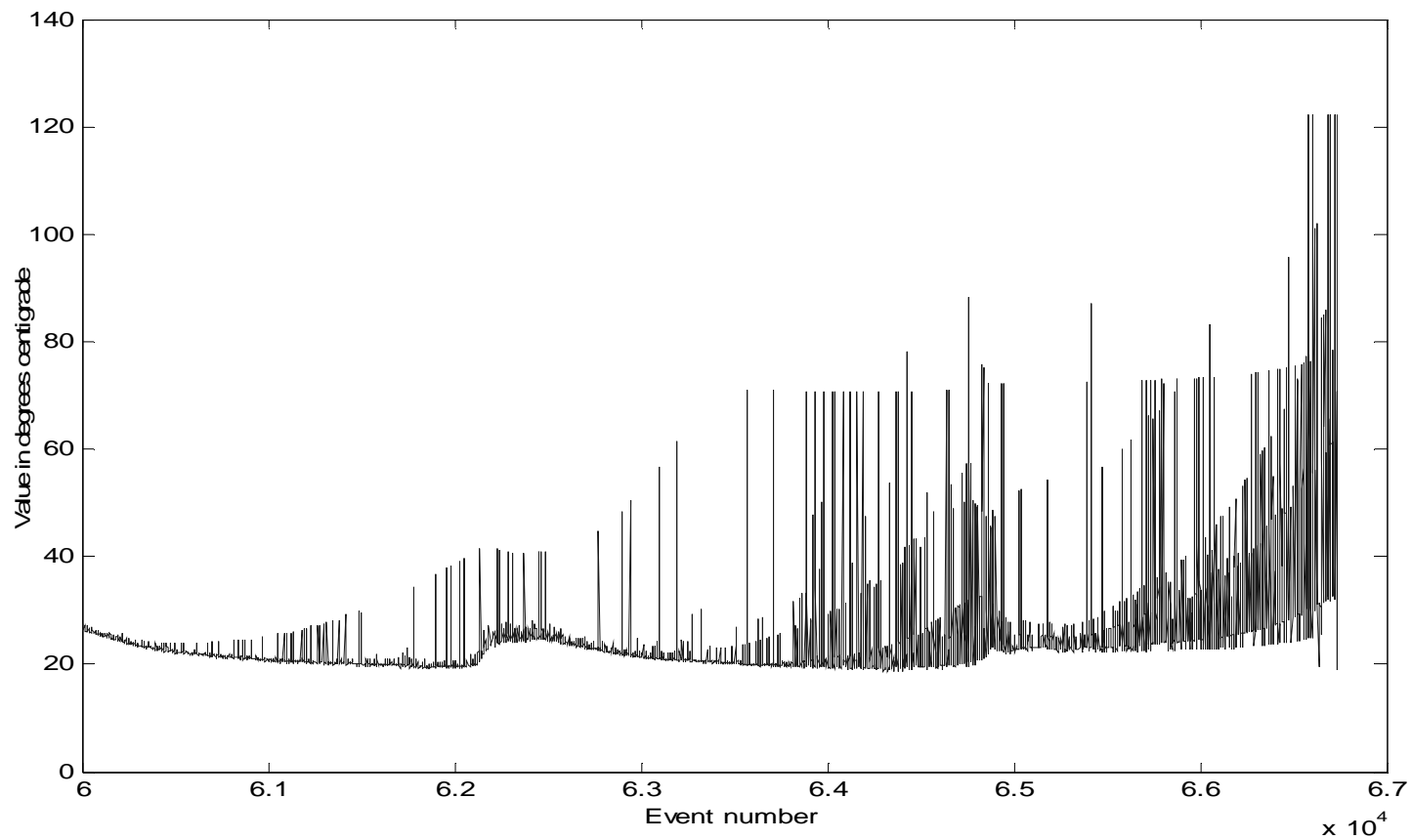




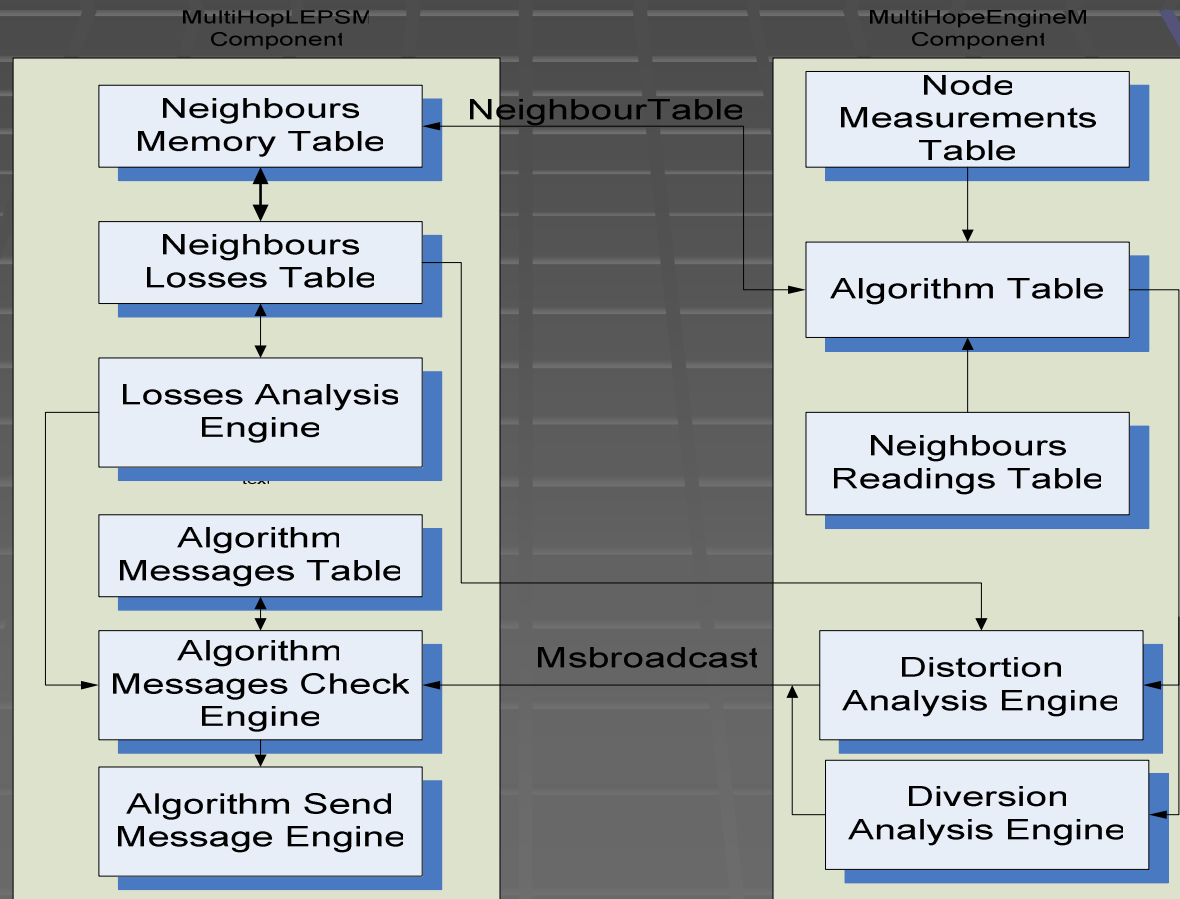




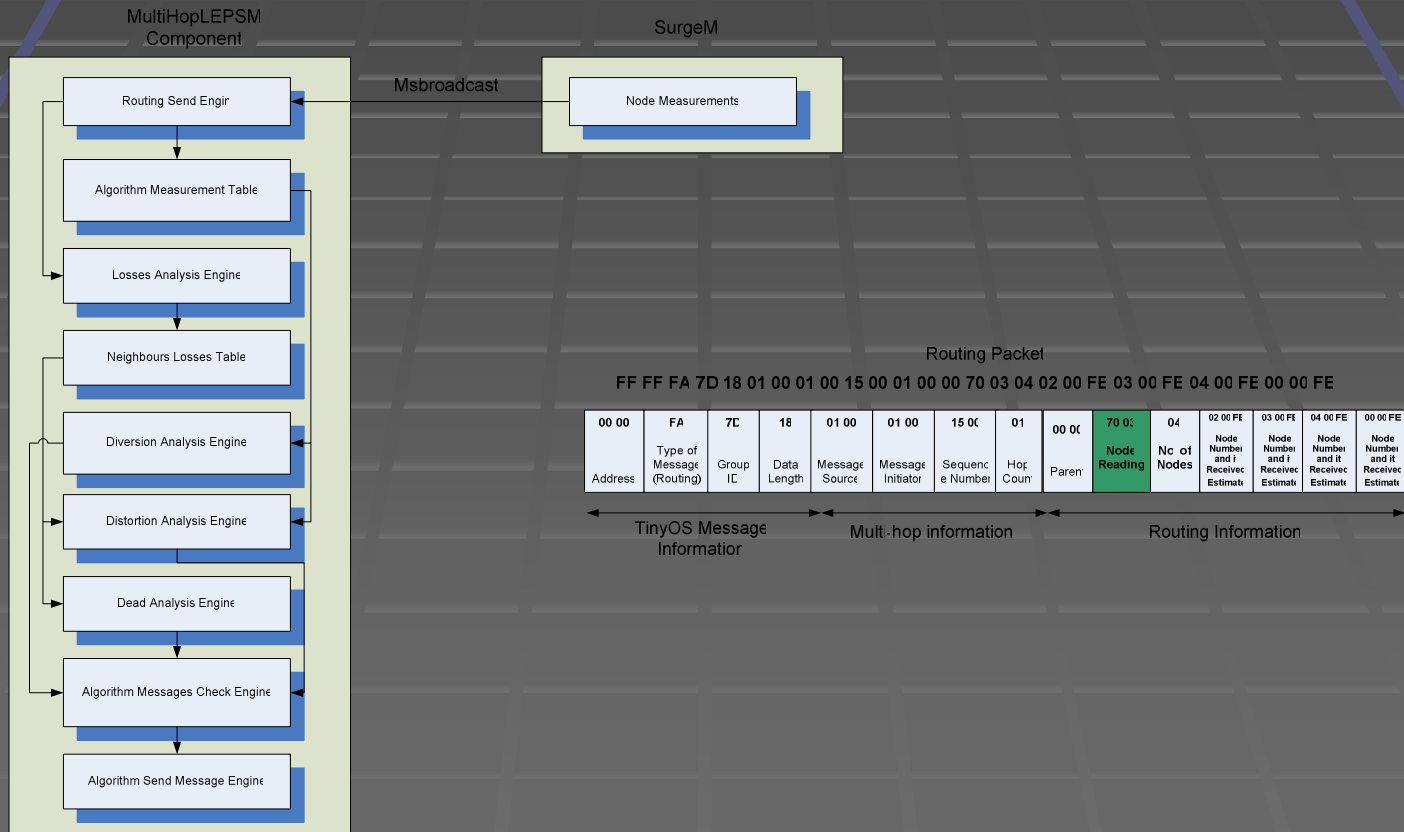
Relating the high and low parameters



Method 1



Method 2



Algorithm warning packet

Warning Message Packet 00 00 0B 7D 0F 01 00 01 00 7C 09 02 0A 00 04 03 00 01 03 00

00 00	0B	7D	0F	01 00	01 00	7C 09	02	0A	00	04	03 00	01	03 00
Address	Type of Message (Warning)	Group ID	Data Length	Message Source	Message Initiator	Sequence Number	Hop Count	Number of Readings	Nc of Deac Nodes Detector	Nc of Diversion Nodes Detection	Suspected Node	Type of Fault	Number of Detected Windows

