Reputation Based Scheme to enforce Cooperation in Wireless Ad-hoc Networks

Carryoles Maria
Liang Huo
Mani Prashanth Varma Manthena

Dr. Ertan Onur
Kishor Chandra Joshi

IEEE
Advancing Technology
for Humanity

Introduction

- Virtual Currency Schemes
- Reputation Based Schemes

Reputation Based Scheme

- ❑ Global Reputation
- ❑ Local Reputation

Agenda

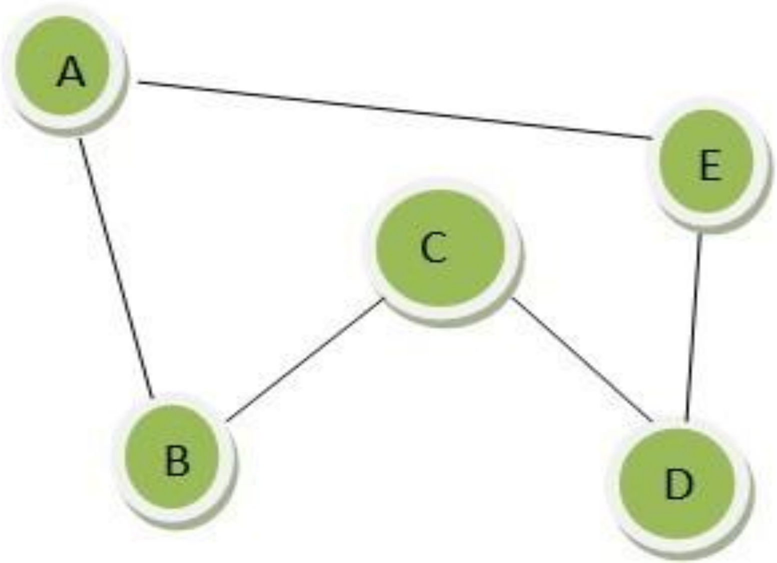
- ❑ **Related Work**
- ❑ **Problem Statement**
  - ▪ Assumptions
  - ▪ Constraints
- ❑ **Our Approach**
  - ▪ Results
- ❑ **Conclusion**

Related Work

- CONFIDANT
  - Based on DSR
  - Uses Global Reputation
- CORE
  - Global Reputation
- OCEAN
  - Based on DSR
  - Uses local reputation

Problem Statement

"Once your problem is clearly defined, your job is half done."



Problem Statement

To state:

- What **problems** do we need to solve
- What **assumptions** do we have to make
- What **constraints** do we have

Reputation Value

- R(A,B) is the rep. value of B in A’s rep. table

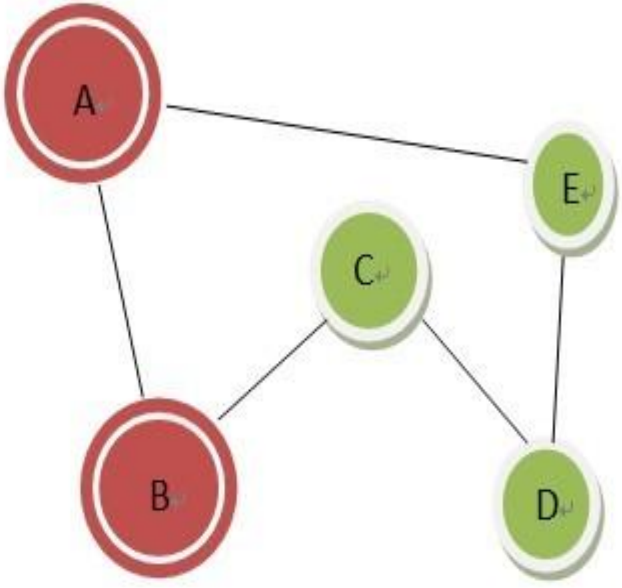
$$R(A,B) = \frac{\text{Number of packets forwarded by node B}}{\text{Total number of packets received by node B from A}}$$

- Rep. Table of A

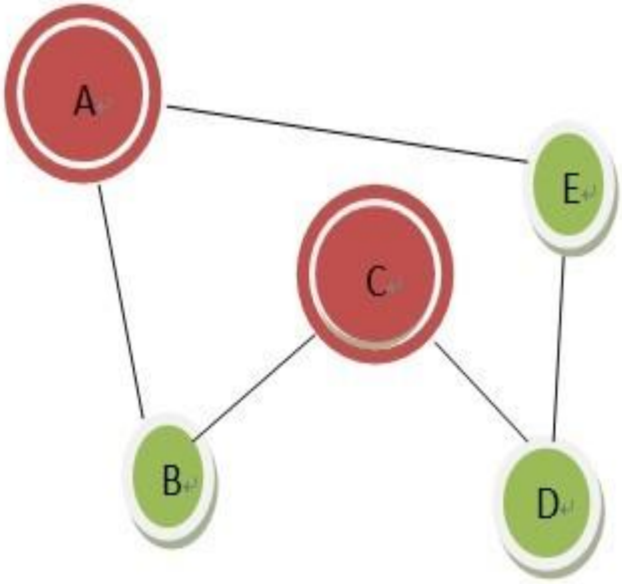
Node ID	Rep. Value
B	0.85
C	0.60
D	0.28
E	0.52

Reputation Collection

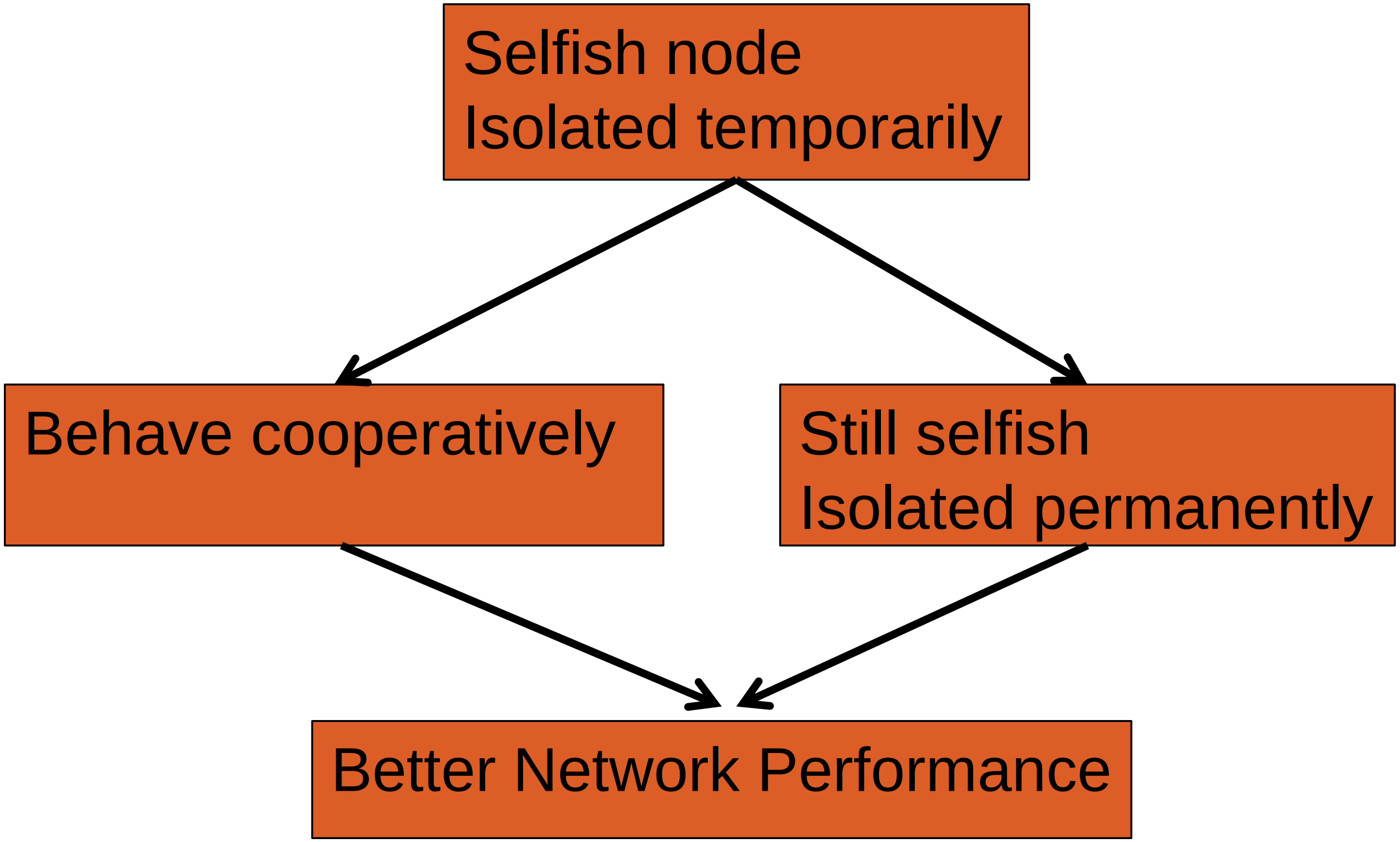
Direct Monitoring **&** Indirect Monitoring



&



Inspire good behavior



Selfish node
Isolated temporarily

Behave cooperatively

Still selfish
Isolated permanently

Better Network Performance

Other Problems

- Preventing liars to transmit fake Rep. values
- Reputation propagation and agreement
- Detecting a selfish node

Problem Statement

To state:

- What **problems** do we need to solve
- What **assumptions** do we have to make
- What **constraints** do we have

Assumptions I

- Well Connected , at least 1 neighbor
- “Neighbor” is 1-hop away
- Rep. value between 0~1
- Rep. value increases when a node cooperate, decreases when it does not

Assumption II

- Cooperation = Relaying packets
- Level of cooperation is defined only on the action of dropping packets, regardless of intentional or unintentional selfish behavior
- Rep. table is updated and propagated proactively

IEEE Advancing Technology for Humanity

Problem Statement

To state:

- What **problems** do we need to solve
- What **assumptions** do we have to make
- What **constraints** do we have

Constraints I

- Dynamic topology, nodes come and go, difficult to keep track of rep. values
- Reputation propagation leads to congestion& large overhead
- Different QoS requirements (packet loss, etc)

Constraints II

- Other reasons to drop packets, e.g. physical collision, faulty nodes with no resources
- Security issues

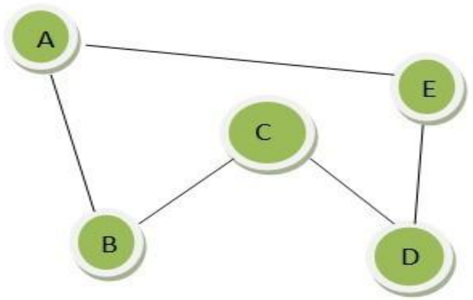
Our Approach

Reputation based algorithm :

- Reputation monitoring and calculations
- Detecting and handling liars
- Reputation propagation
- Reputation agreement
- Handling selfish nodes

Reputation Mechanism

- Network Initialization



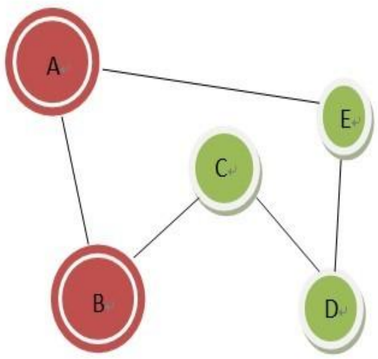
- Creating and initializing reputation tables for each node

e.g. Node A's Reputation table

Node Id	Rep. Value
B	1.00
C	1.00
D	1.00
E	1.00

Reputation Monitoring and Calculations

- Direct monitoring



I. Reputation calculation

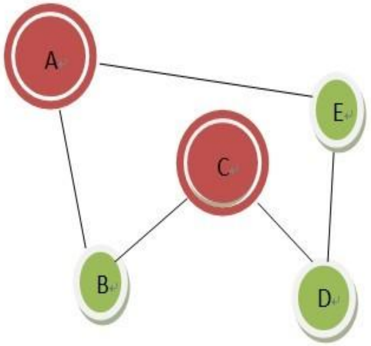
$$R(A,B)=\frac{\text{Number of packets forwarded by node B}}{\text{Total number of packets received by node B from A}}$$

- Updating reputation table and flooding it in the network

e.g. Direct reputation monitoring of Node A on node B gives say 0.50. Then Node A’s reputation table looks as shown below

Node Id	Rep. Value
B	0.50
C	1.00
D	1.00
E	1.00

- Indirect Monitoring



- Handling liars

  - Liar test

$$R(A,B) - R(C,B) = L \leq 0.4$$

- If a node passes the liar test i.e. $L \leq 0.4$

I. Its reputation table is used in indirect monitoring

II. Reputation calculations for indirect monitoring
R’(A,B)=R’(C,B)=(R(A,B)+R(C,B))/2

III. Reputation tables are updated and flooded in the network

- Reputation propagation and agreement

  All the nodes have same reputation values for all other nodes in the network

- Detecting selfish nodes

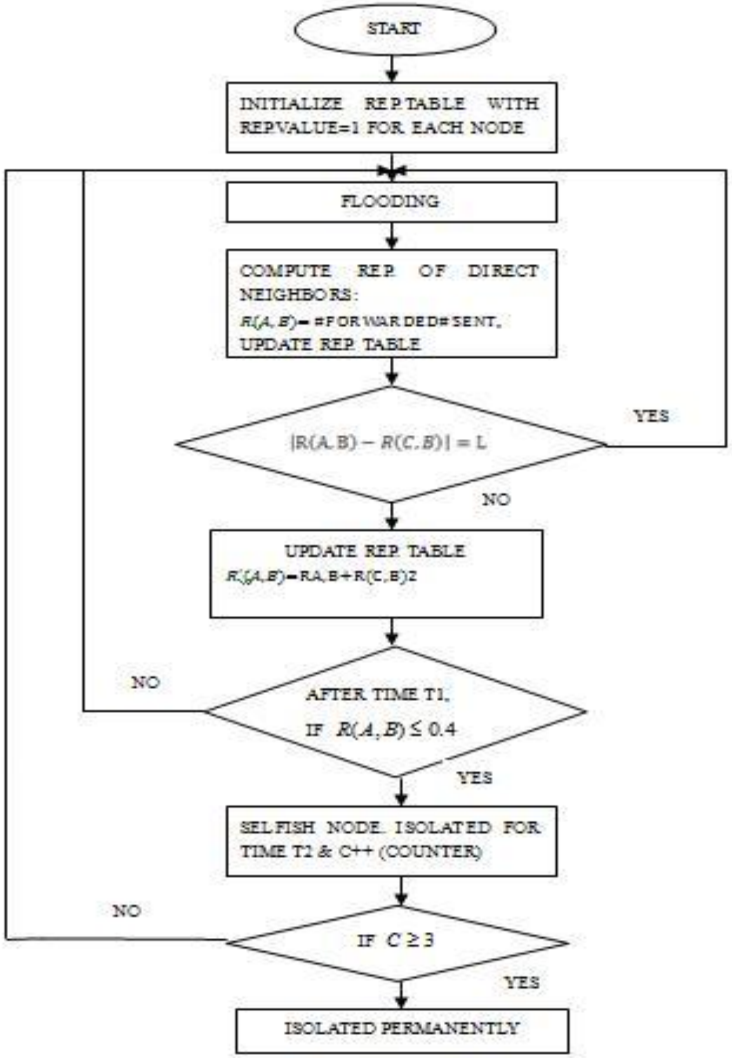
  $R(A,B)<0.4$(Threshold)

- Handling selfish nodes

I. Temporary isolation for certain time

II. Permanent isolation due to repeated selfish behavior

Flowchart



START

INITIALIZE REP.TABLE WITH REP.VALUE=1 FOR EACH NODE

FLOODING

COMPUTE REP. OF DIRECT NEIGHBORS:
$R(A, B)$= #FORWARDED#SENT,
UPDATE REP. TABLE

$|R(A.B) - R(C.B)| = L$

YES

NO

UPDATE REP. TABLE
$R;(A,B)$=RA,B+R(C,B)2

NO

AFTER TIME T1,
IF $R(A,B) \leq 0.4$

YES

SELFISH NODE ISOLATED FOR TIME T2 & C++ (COUNTER)

NO

IF $C \geq 3$

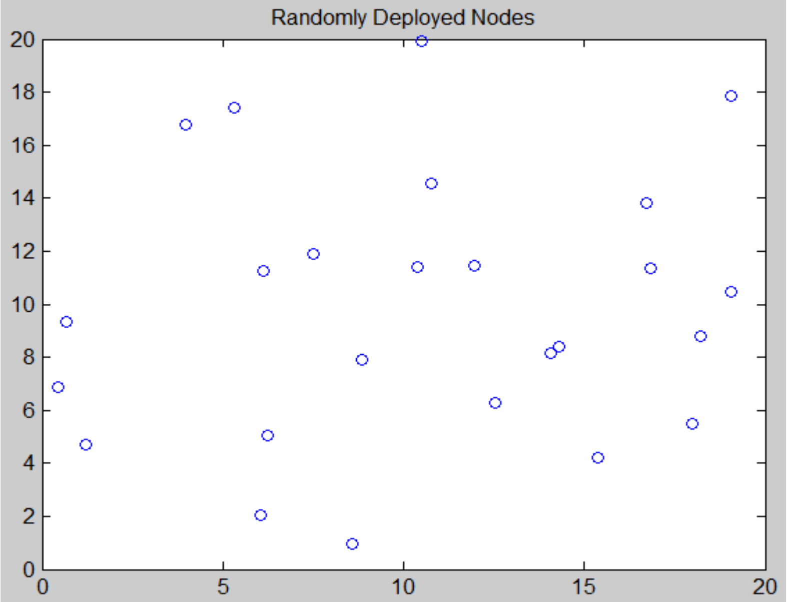
YES

ISOLATED PERMANENTLY

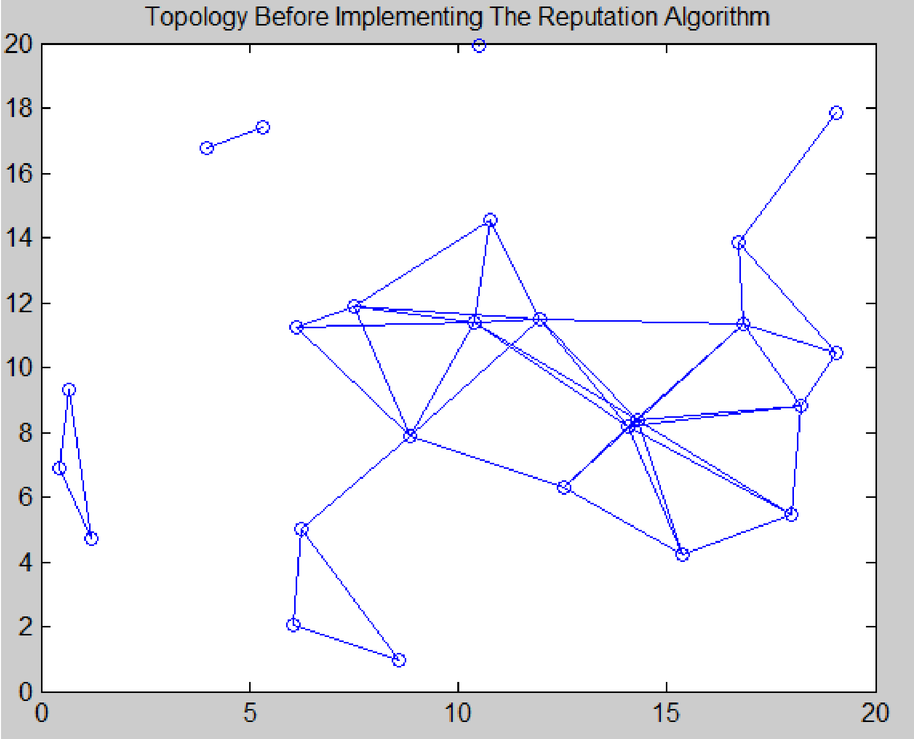
Implementation and Results

Simulation Tool – MATLAB R2012b

- Random deployment of nodes in a field

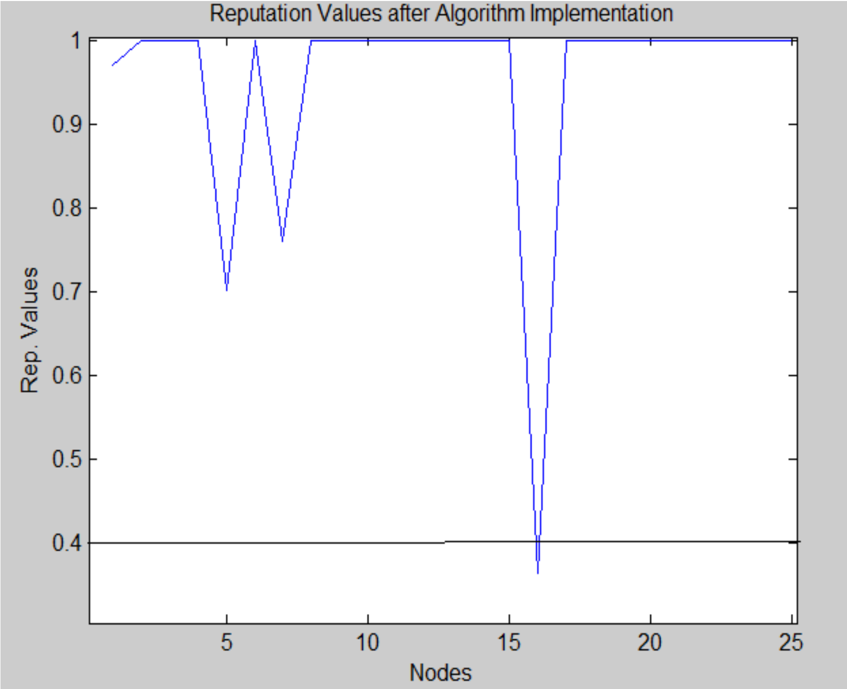


- Topology based on transmission ranges of the nodes

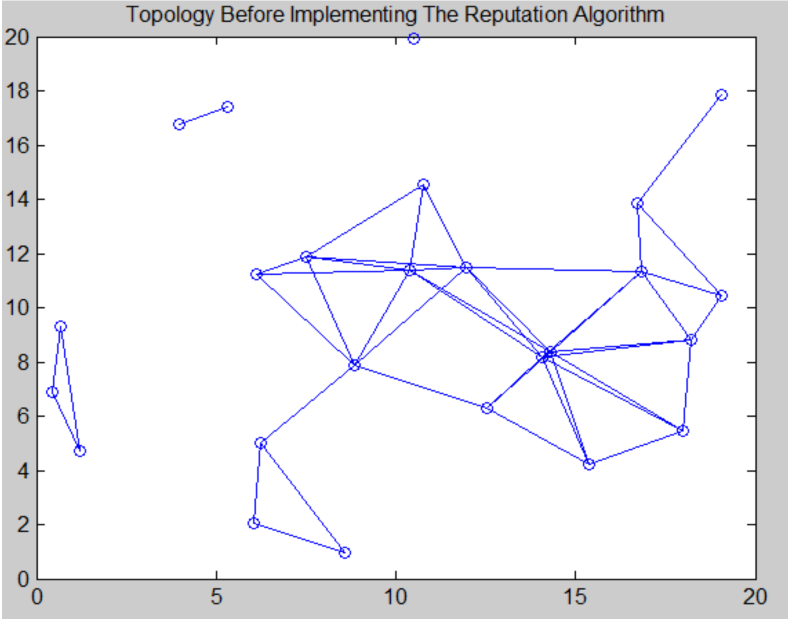


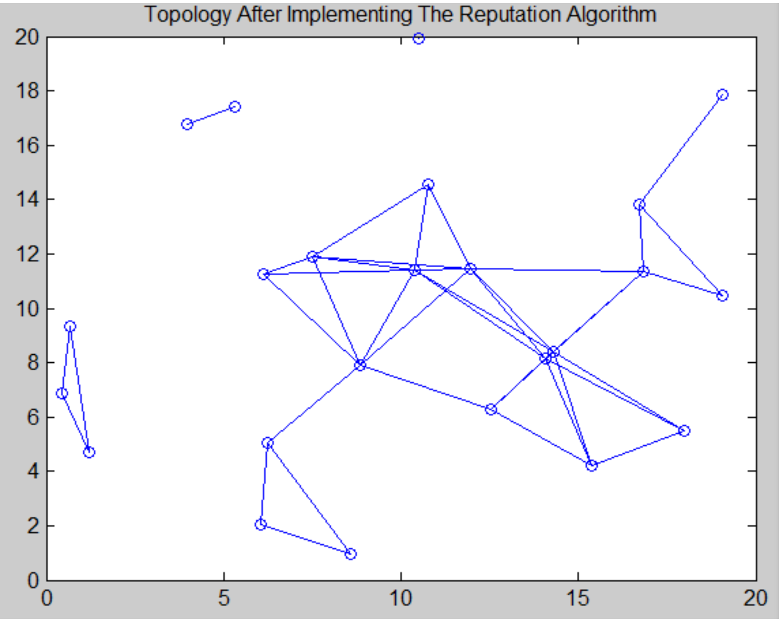
- Creating reputation tables for each node
- Assigning random reputation values to the nodes initially
- Reputation monitoring
  - Direct Monitoring
  - Indirect Monitoring with Liar Test(Handling liars)

- Reputation propagation and agreement
- Detecting selfish nodes



- Handling selfish nodes
  - Temporary isolation
  - Permanent isolation





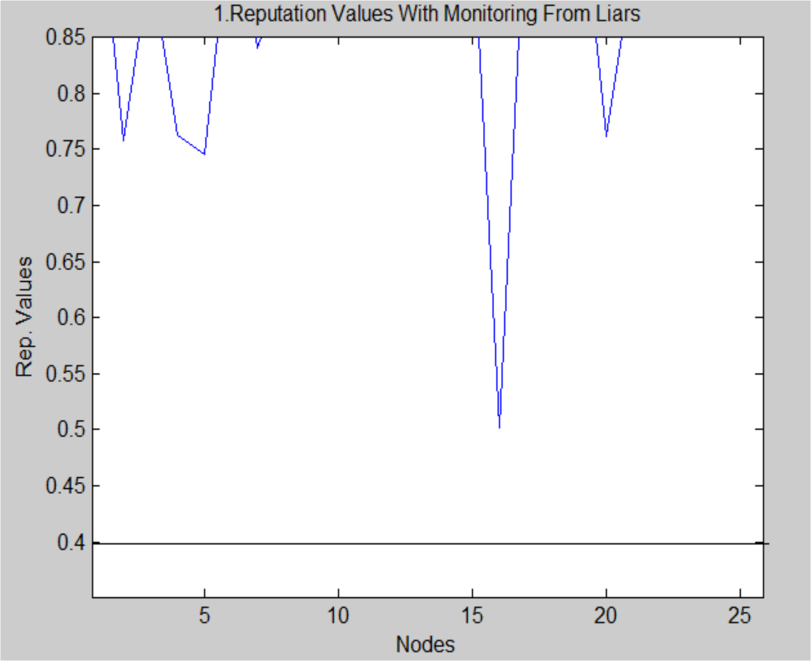
- Algorithm implementation without deviation test i.e. Not handling liars

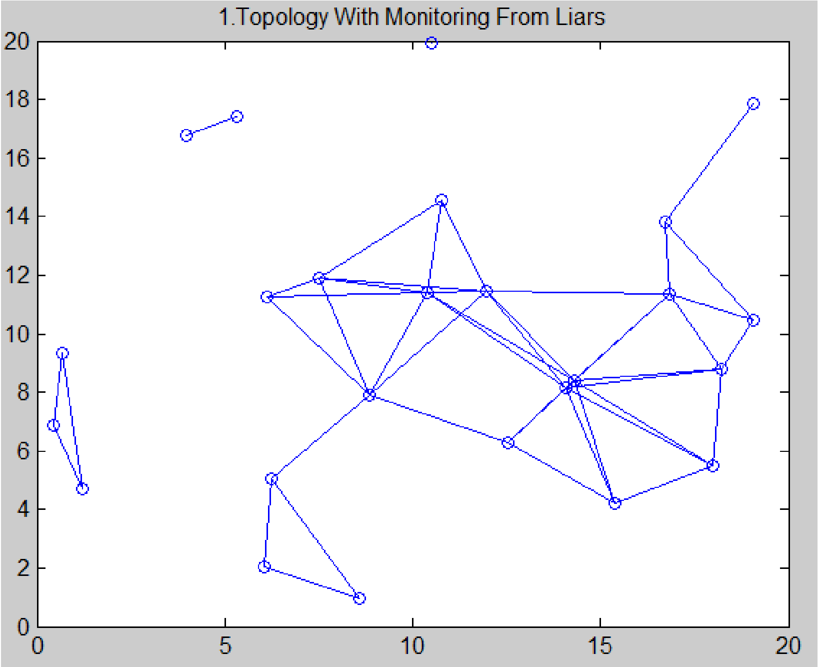
- Two cases can be studied here

I. Case 1 : Selfish nodes still find a spot in the network

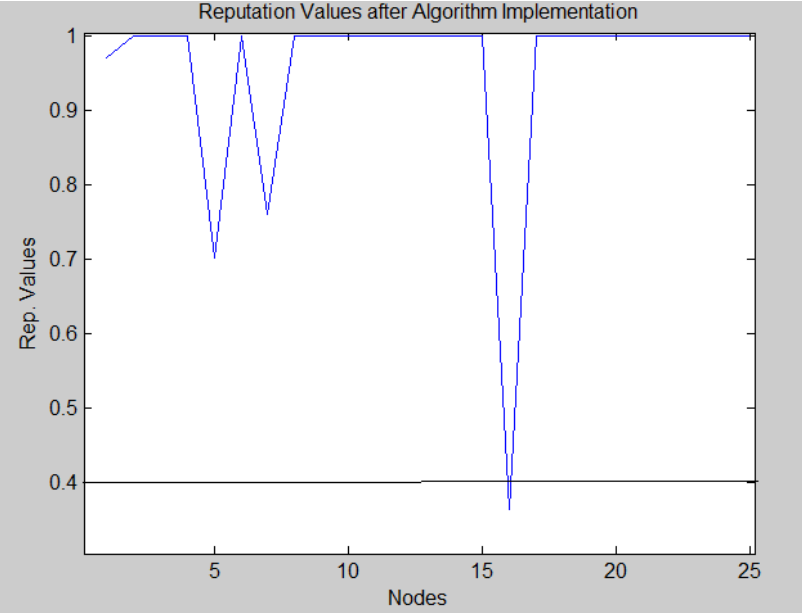
II. Case 2 : Cooperating nodes may be isolated

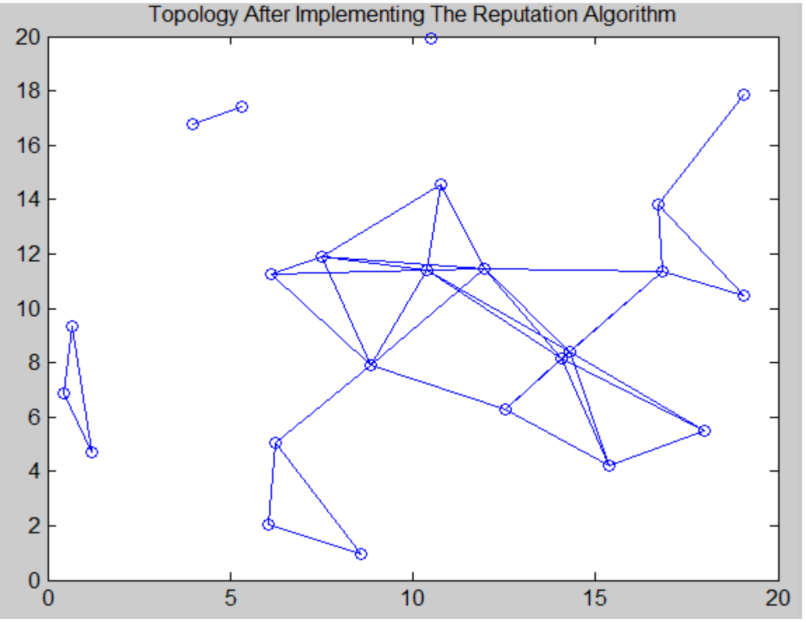
- Case 1


1.Reputation Values With Monitoring From Liars
Rep. Values
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
5
10
15
20
25
Nodes

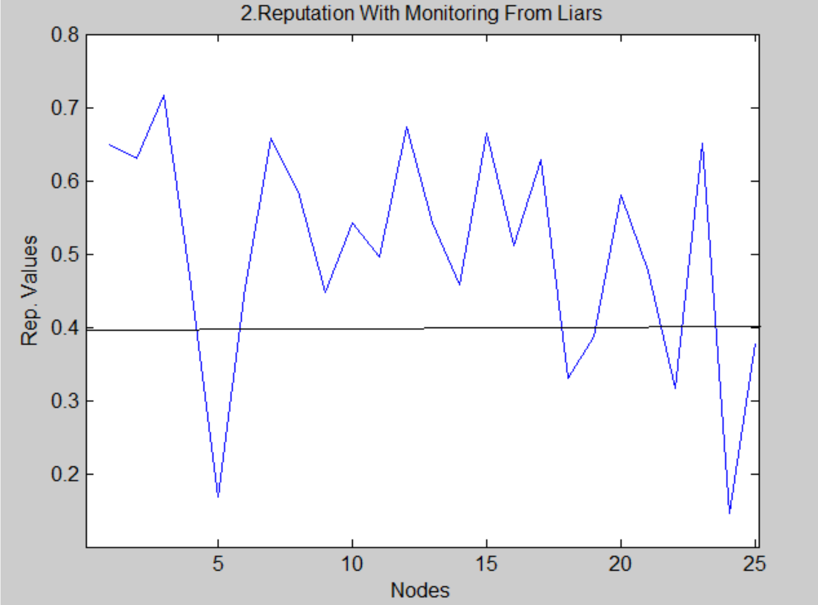

1.Topology With Monitoring From Liars
20
18
16
14
12
10
8
6
4
2
0
0
5
10
15
20

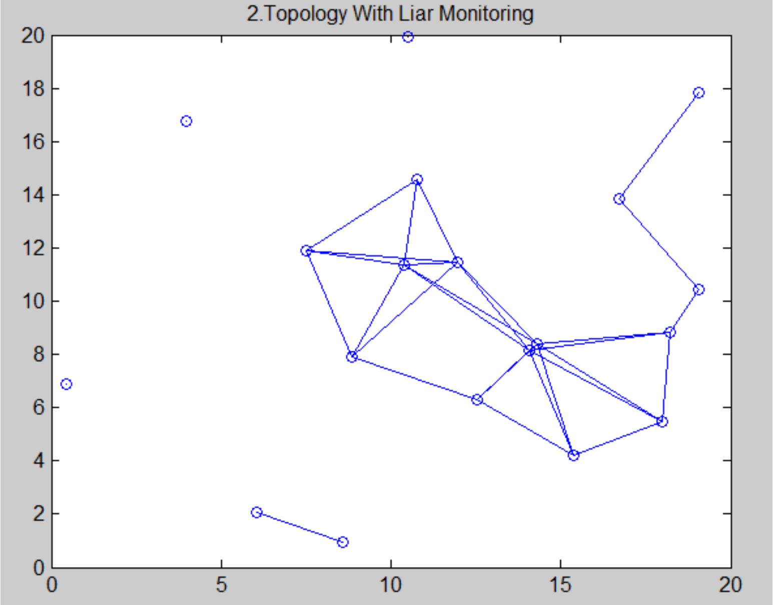
- Reputation algorithm with liar test i.e. Handling liars





Case 2


2.Reputation With Monitoring From Liars
Rep. Values
0.8
0.7
0.6
0.5
0.4
0.3
0.2
5
10
15
20
25
Nodes


2.Topology With Liar Monitoring
20
18
16
14
12
10
8
6
4
2
0
0
5
10
15
20

Conclusion

- Cooperation results in good system performance
- Selfish nodes should be inspired for good behavior
- Liars must be handled in reputation based algorithms

Future Work

- We would like to measure network performance using our algorithm over a wide period of time in real world situations using a different simulation tools like OPNET or NS2
- We would also like to implement security which is an important concern in ad-hoc networks

Thank you !!