COOPERATION ENFORCEMENT METHODS
IN
AD HOC NETWORKS

Dissertation submitted to the
Jawaharlal Nehru University, New Delhi
In Partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY
In
COMPUTER SCIENCE AND TECHNOLOGY

By
GHYANI UMESH KUMAR MAURYA

UNDER THE SUPERVISION OF
Mr. SUSHIL KUMAR

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067, INDIA
JULY-2012

JAWAHARLAL NEHRU UNIVERSITY

SCHOOL OF COMPUTER & SYSTEMS SCIENCES

NEW DELHI-110067, INDIA

CERTIFICATE

This is to certify that the dissertation entitled **"COOPERATION ENFORCEMENT METHODS IN AD HOC NETWORKS"** being submitted by **Mr. Ghyani Umesh Kumar Maurya** to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi,** in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science and Technology,** is a record of bona fide work carried out by him under the supervision of **Mr. Sushil Kumar, Asst. Professor**.

This work has not been submitted in part or full to any university or institution for the award of any degree or diploma.

Mr. Sushil Kumar
(Supervisor)
SC&SS, JNU, New Delhi

Dean
SC&SS, JNU, New Delhi

JAWAHARLAL NEHRU UNIVERSITY

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES

NEW DELHI-110067, INDIA

DECLARATION

This is to declare that the dissertation entitled "**COOPERATION ENFORCEMENT METHODS IN AD HOC NETWORKS**" is being submitted to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi,** in partial fulfillment of the requirements for the award of degree of **Master of Technology** in **Computer Science and Technology,** is a record of bona fide work carried out by me.

The matter embodied in the dissertation has not been submitted in part or full to any university or institution for the award of any degree or diploma.

GHYANI UMESH KUMAR MAURYA

M. Tech (2010-2012), SC&SS, JNU,
New Delhi-110067

Dedicated to

MY FAMILY..

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor, **Mr. Sushil Kumar** for his encouragement, enthusiasm and direction provided to me during the entire duration of this dissertation. Without his valuable thoughts, recommendation, freedom and faith in me, I would have never been able to complete my work.

I am grateful to my parents, my brother and my sister for their unconditional supports and immense love and also for my nieces and nephew for their soothing affection.

I wish to thank all my fellow batch mates and friends for their help and support and, if not so then at least for their fictitious praising. I gratefully acknowledge the unselfish help given to me by them who inspired me greatly through many interesting discussions, support, feedbacks and critical remarks. Few among them are Ramesh, Amit, Ravi, Om Prakash, Kunal, Anil and Aarti.

Finally, I want to thank the almighty whose choicest blessings have sailed me through this research effort.

Umesh Kumar

ABSTRACT

From the very basic nature of the ad hoc network, it is clear that it is a self organizing and for the transmission of the data packets it depends mostly and widely on the nodes cooperation among each other. Whenever there is a path which is multihop, the sending node has to rely on the intermediate nodes for taking the packets further till destination. But there is problem associated with this, as we know that the nodes in an ad hoc network mostly run on the battery power and also, channel bandwidth is constraint, nodes may decline to cooperate and very existence of the network will collapse. This decline of cooperation may manifest in many ways. Selfishness is one kind of behavior so is the maliciousness. So we need to establish certain kinds of cooperation among the nodes voluntarily, forcefully or with some incentives.

Here we have done our work based on trust and activity, game theoretical model and genetic algorithms which has been mentioned in "Preventing selfish behavior in ad hoc networks". In this particular proposed algorithm we have suggested some basic and fundamental changes in the underlying protocol and have designed a novel concept to calculate activity level in the protocol. Also we have suggested a novel formula to calculate pay off for the source nodes in the game. By using these formulae and the pay off tables proposed we are getting better result in the term that normal nodes are able to send more fractions of packets.

The whole algorithm and protocol has been simulated and compared on MATLAB through object oriented programming in C++. It is observed form the simulation results that if the value of exponent varies from 1.1 to 1.4in the distribution for the payoff of a source nodes, the selfish behavior of nodes suppressed more rapidly.

CONTENTS

List of Figures and Tables

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Mobile Ad Hoc Networks

1.1 Introduction

Mobile ad hoc network popularly known as MANET is a self-organizing network with nodes distributed distantly and running on the battery powers. It is characterized by not having any fixed infrastructure and central governing body for the network. Channel bandwidth is also constraint in this type of networks. In ad hoc networks there are some nodes which are participating in the network and, nodes have low transmission range. In ad hoc network nodes are mobile so persistently moving here and there and changing their neighborhoods. Due to mobility of nodes, topology is also very dynamic. Nodes in ad hoc network are linked with wireless links and working like the routers by itself. So for any type of communication in the network if it takes multihop route, nodes have to rely over other intermediate nodes. Due to limitation of power as, nodes are running on battery powers, and channel bandwidth nodes may decline the request. These types of networks are very much vulnerable for various types of attack from the selfish and malicious nodes in the network. [1]

Commonly three types of the traffic are found in the mobile ad hoc network. First is peer to peer in which nodes are communicating to the nodes which are one hop away from it. Second is distant communication in which, nodes are communicating to the nodes which are two or more hop away to it. In this traffic type, nodes are working like forwarding nodes and having the route information about its neighbor. Thirdly when the nodes are moving around, routes are getting stale so quickly that we always need to reconstruct the route.

Now let's have a bit more elaborative discussion about basic features of the mobile ad hoc network.[1]

1. Nodes are autonomous: In the MANET nodes are autonomous body and they can work like a node as well as router also. They can be sender and the receiver of the data.

2. Distributed operation of the network: MANET works in the distributed way. There is no any central control for the nodes, so work of routing and management of the network is distributed over the network among different nodes.
3. Running on the battery power: Nodes in the MANET are working on the battery power so energy is always a basic constraint in this type of network.
4. Multihop routing: Routing is basically multihop in this type of network as nodes are moving here and there over the network.
6. Dynamic topology: Network topology is very dynamic due to mobility of the nodes in the network. It is always changing so that route found previously will be of no use after some time and reconstruction of it is mandatory.
7. Fluctuation in link: Link capacity in the MANET is highly fluctuating due the different kinds of hindrance for which wireless links are susceptible. It has low bandwidth, susceptible to the noise, interference and fading. There may be heterogeneity in the links also.
8. Low capacity of the nodes: Nodes in the ad hoc networks are of very low CPU power, low memory so computation power is low and slow.[2]

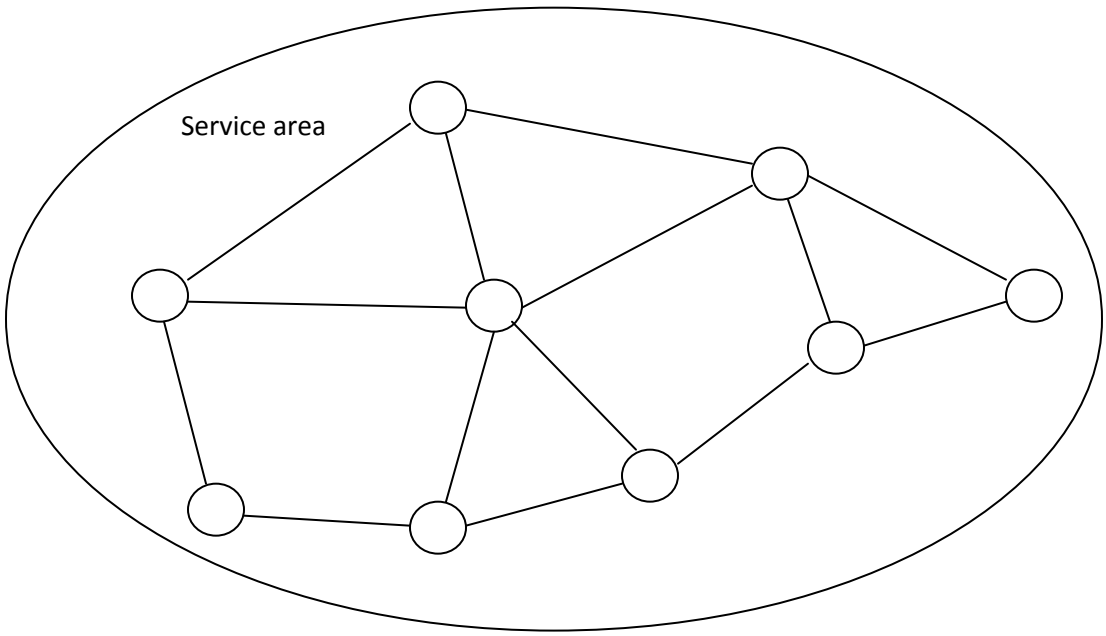


Figure 1.1 Mobile Ad Hoc Networks

1.2 Applications of MANET

A MANET can be used in the varying field of applications where little or no infrastructure in existing. Use of MANET is increasing by leap and bounds with the increasing supply of the different kinds of hand held communication devices. Ad hoc network allows easy set up of communication networks and it is also very easy in the term of adding or removing the nodes at the network, so convenient and the simplicity are the keys. Its application is very diverse from small and hand held system to the big computers to the moving vehicles. Apart from the traditional use of the ad hoc network in the form of replacement of wired network a wide area of applications have also emerged in the recent few years and its use is swelling as no before. Some important areas, where it is used extensively, are as follows. [1]

1. Military applications: The rudimentary form of the ad hoc network initially took its course from this field only. Most of the time the general appliances used by the army personnel's are equipped with some sort of communication device which inevitably must be wireless in the making. This enables the widespread use of ad hoc network in military as it is needed in the patrolling area.
2. Calamity demise: At the time of any form of huge disaster either man made or natural calamity MANET could be very helpful indispensably. In these kinds of situation it is anticipatory that hardly any essential infrastructures could survive. So among the members of the rescue team communication is established in the form of ad hoc network as the part of management.
3. As per the name: Ad hoc network can be used as per there their name in the form of the ad hoc network as and when it is required like in seminars or conferences, in class rooms etc. it is small and local use of this type of networks.
4. Personal use: Ad hoc networks can be used personally either when few of our friends are over there with handy communication devices.[2]

1.3 Challenges of MANET

As from our discussion above seemingly some reader may find illusion that ad hoc network is a mere simple concept and hardly, it has any implacable intricacies. But to be very clear that perception is totally contrary to the reality we are mentioning some very potent thread and challenges in the realization of the ad hoc network.

1. Routing: routing in ad hoc network is really a grave challenge as there are many existing protocol in this regard in near wireless network. Since nodes are continuously moving here and there in the network packet forwarding to a specific destination is difficult task as route may go stale. Multihop routing is even difficult. It is also a difficult task to decide between proactive and reactive kinds of protocols with varying circumstances in network.
2. Security concerns: Inheriting almost all the problem from the wireless network ad hoc network has its own associated problem as well when concerning about the security and reliability of the network. As from the very trait of the wireless medium it is vulnerable to the outer attack and to be betrayed. Due to distributed operation of the nodes authentication is a key issues all time. Wireless characteristic of the medium also aggravate the case of reliability in the form of narrow bandwidth, packet loss, data transmission error etc.
3. QoS: Providing different quality of service in the dynamic environment is a challenge. The inherent stochastic feature of the communication quality avoids any given guarantee to any node in this network for service.
4. Power: Since almost all the light weight hand held systems are running on the battery power so we must use only power efficient protocols for any level of communication for longer service.
5. Internetworking: it is not always necessary that all the communication is restricted to a specific network. We may need sometimes to switch over different network. For this homogeneity of the protocol shared is expected. So for this purpose we insure such kind of the protocols.

Chapter 2

MANET Routing Protocols

Since we are going to use the routing protocols in the ad hoc network as the basic and subordinate structure on which, our cooperation enforcement method will work. So here we are going to have a comprehensive discussion about some of the routing protocols, apparently those which will be used widely in our proposed methods and related work as well. [1]

2.1 Categorization of ad hoc routing protocols.

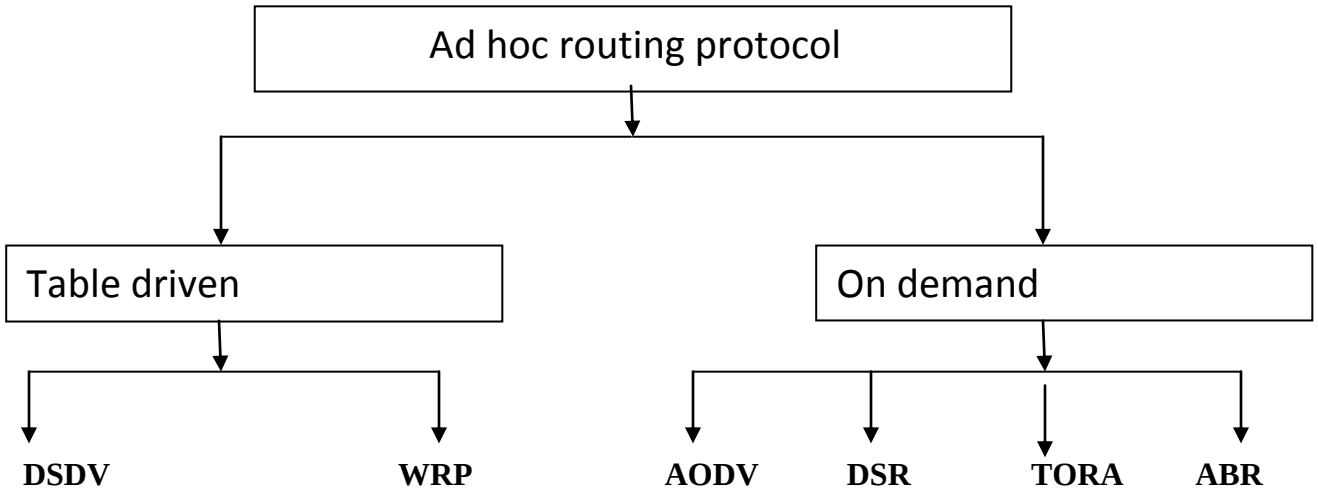


Figure 2.1 Hierarchy of ad hoc routing protocol

2.2 DSDV

DSDV is a proactive table based routing algorithms principally works according to the Bellman Ford algorithm. It guarantees loop free routing protocol. Let's understand how it works. Cost metric used in this protocol is the number of hops. Each node maintains a table for all the nodes in the network for routing purpose. We will discuss each feature one by one.[1]

1. **Sequence number:** DSDV uses sequence no. for each sent packet so that there will be no looping.
2. **Each node maintains table and exchanges with each other.**

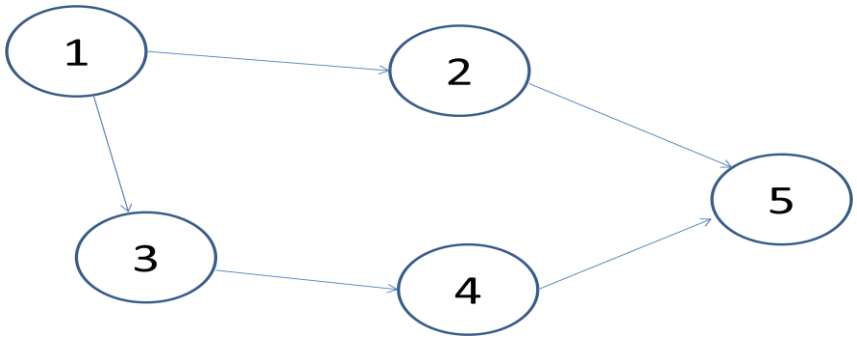
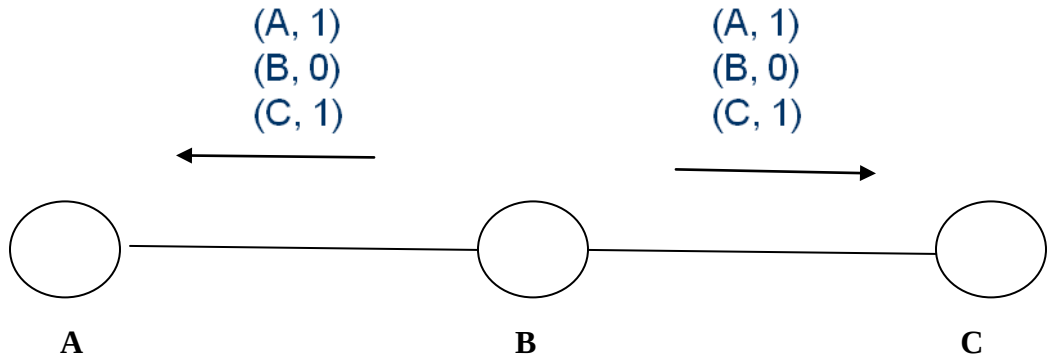


Figure 2.2 DSDV routing

From the above figure 2.2 let node 1 wants to send the packet to node 5 so in this case it will take the shorter path 1-2-5 as per Bellman Ford algorithm.

Now let us take an example of the updating of the table.



Des	Next	Met	Oth
A	A	0	
B	B	1	
C	B	~~3~~ 2	

Des	Next	Met	Oth
A	A	1	
B	B	0	
C	C	1	

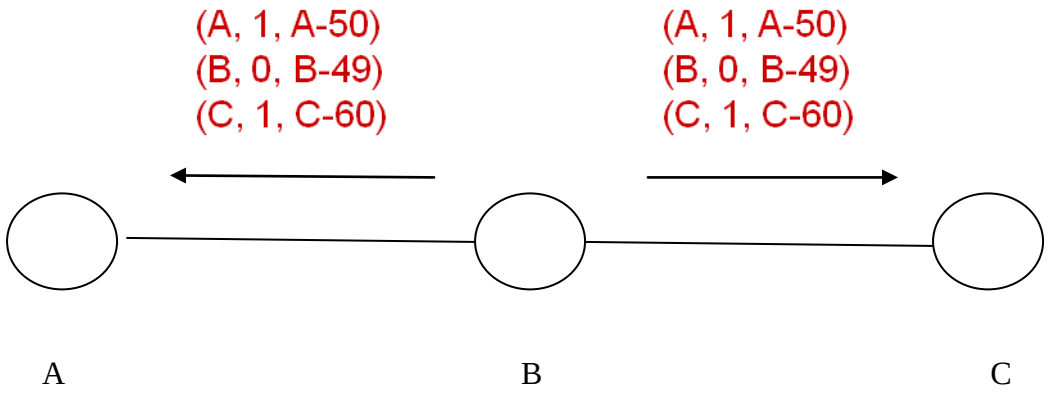
Des	Next	Met	Oth
A	B	~~3~~ 2	
B	B	1	
C	C	0	

Figure 2.3 Exchange of information for updating table

But problem in this type of exchange for the updating of the table may create loop in the network and also it may turn into the count to infinity problem. So from here onward we will discuss the working of DSDV based on table and sequence no.

3. Sequence number in the table

Let's have the above example for this purpose.



Des	Next	Met	Seqno
A	A	0	A-50
B	B	1	B-49
C	B	3	C-60

Des	Next	Met	Seqno
A	A	1	A-50
B	B	0	B-49
C	B	2	C-60

Des	Next	Met	Seqno
A	A	1	A-50
B	B	2	B-49
C	B	0	C-59

Figure 2.4 exchanging information based on sequence number.

From the figure 2.4 we can see that when B is sending updated information for the updating of table it increases its sequence number so that nodes A and C come to know that this is fresh information and to be used. In this way DSDV avoids looping in the network if any node is down or have joined the network. [1]

4. Response to topology change

DSDV responds in two ways for the topology change in the network

(1) Immediate exchange: information about route change, route broken or any related information is immediately propagated on the network.
(2) Full update: if any relevant information has been changed so for then the whole table is exchanged or sent for the updating purpose.
(3) Incremental update: if any relevant information has been changed the only that part of the table is propagated on the network.

5. Response to new node:

When new node joins to the network it advertises its sequence number like it may be D-000 and broadcast it over the network. Let say from the above figure D joins the network near to C. So C can hear this broadcast and come to know that someone has joined the network and asking for the information of the network. So it sends the table for the updating to both B and D. In this way all the nodes on the network come to know about D and D has the information about all other node. Here due to the introduction of sequence number no looping and count to the infinity problem may occur.

6. Disadvantages of DSDV:

Due to the constraint on channel bandwidth continuous information sharing may cause delay. There is no provision of sleeping on the node so running on battery it may be a costly affair for the nodes. Moreover after this whole affair worrying aspect is this, that most of the information will not be used and will go stale.

2.3 AODV

Unlike the DSDV, AODV is a reactive routing protocol. It is not necessary to broadcast every change in the network. If the link breakage does not affect the ongoing transmission then it is not to be broadcasted over the network and if it affects then the information is rendered to affected node only. It is very appreciable in this protocol that local nodes movement have local effect only and for the extent possible network wide proliferation and propagation of information is averted. It also provides unicast and multicast communication and bidirectional communication links are used. It maintains only active route ant sequence number is incorporated for annulling loops. Whenever a route is expired it is discarded to avoid overhead of the maintenance of the

unused information. AODV can find multiple routes from the source to the destination but only one route is implemented at one time, again to avoid burden. Because at one time we can use only one route and if after some time if this route is broken then we can't use the stored route either as no way to know that whether it is available. This was all about AODV in one say, now let's have a proper view from inside one by one.

Destination address
Next hop address
Destination sequence number
Life time

Figure 2.5 Route table in AODV

1. Ad hoc on demand distance vector routing properties:

AODV searches for a route as and when required and also it does not have route information from each node to every other node at one time. Routes are maintained for the time it is being used and just after discarded altogether. Each node maintains it's monotonically increase sequence number for this purpose and to avoid loop. Sequence number is incremented each time when node sees any change in the neighborhoods and corresponding is rendered to it.

AODV uses routing table for the routing information, one for the unicast and other for the multicast information. The route table may store the following information in the table form. Each node maintains the precursor nodes through which the route will pass. Each time when route is used its life time is updated. [3]

2. AODV route discovery:

Whenever a node has to send a packet to a specific destination, first it checks its route table that whether it has route for that destination if yes, then it forwards the packet to the next hop and if no then it start a process called route discovery. It goes as follows.

Route discovery process starts from the generation of a route request packet, RREQ by source node. This packet may contain the following fields, source node IP address, source node's current sequence number, destination IP address, destination sequence number. Packet also contains broadcast ID number and it is incremented each time when broadcast takes place. This ID number with the source IP address gives the unique identification to the subsequent RREQ's. Broadcasting of the RREQ's takes place in the form of flooding. [3]

1. Flooding of the control packet takes place like this. Source node forwards the packets to its neighbors'. All the nodes receiving this packet subsequently flood the packet in the same manner to their respective neighbors'. Addition of the sequence number averts the possibility of exchanging same packet time and again. Packet reaches to the destination provided it is reachable from the source. The whole process the depicted in the figure 2.6.

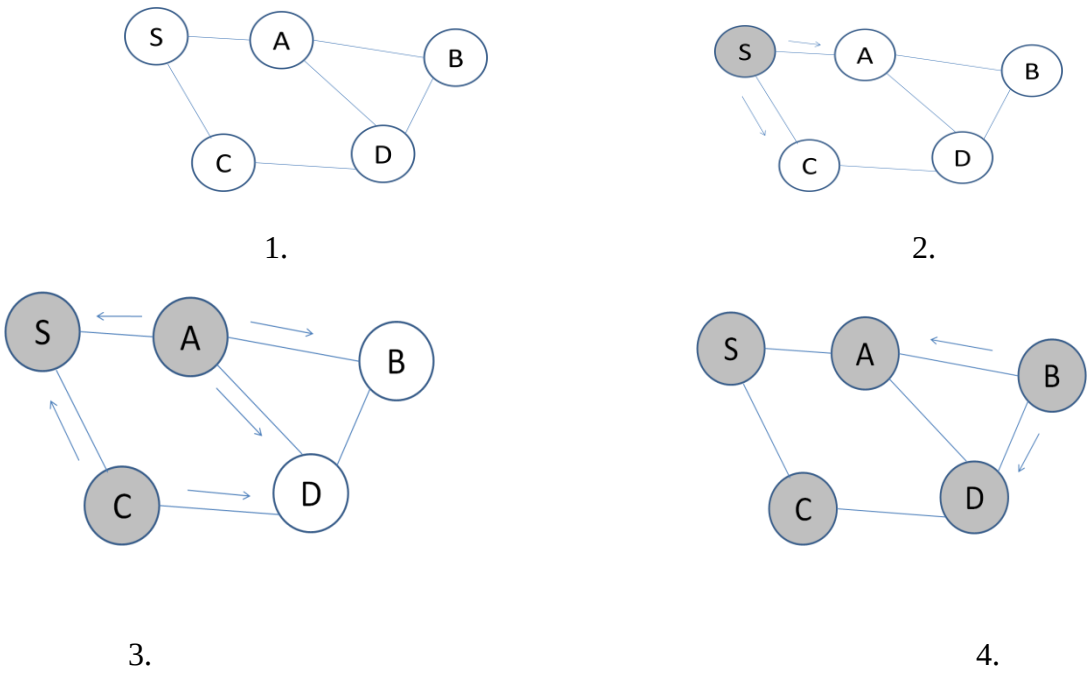


Figure 2.6 Flooding (1) source sends RREQ (2) A and C get the packets

(3) A and C forward the packets (4) D gets the packet

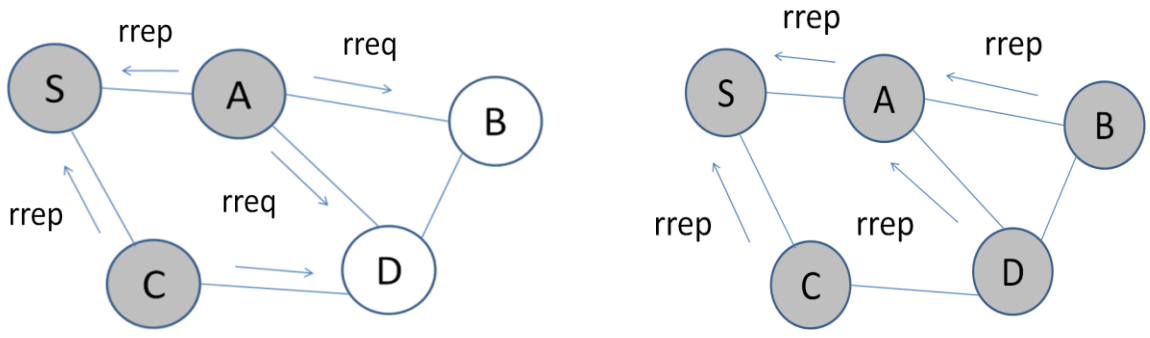
From the figure as it is apparent that source node S has the packet to send D. It first of all sends RREQ packets to A and C. A and C, after getting the packet, forward it to D, B and S. Since S

once has forwarded the packet it does not take any action and it possible due to the sequence number. In this way D got the packet and does not forward it any further as this is meant for it.

3. Flooding of data packets are done by many protocols as DSR, AODV etc. This control packet in the form of RREQ is used for the path discovery. Discovered route is subsequently used for the transmission of the data packets. This type of protocol is more efficient when rate of data transmission is very low. In that case maintaining of route in advance may cost heavily. Reliability of data transmission is very high due to multiple path available and freshness of path both.

4. Disadvantages of flooding may also be very high in some cases. Data packets may be delivered to too many nodes that do not need to receive it. Potentially low reliability of data delivery as it uses 802.11 MAC which provides unreliable service, also nodes may transmit simultaneously to same node and data loss may occur.

5. Once an intermediate node receives RREQ it sets reverse path to its source node simultaneously. Reverse path entry may consist of source IP address, source sequence number, number of hops to the source node, IP address of the node from which it got the packet. Using this reverse route nodes can send RREP packet to the source node so that it can send data packets. When the RREQ reaches to the destination it incorporate RREP packet and sends it back to the route from which it got RREQ. For any conflict in this regard sequence number is used judiciously.



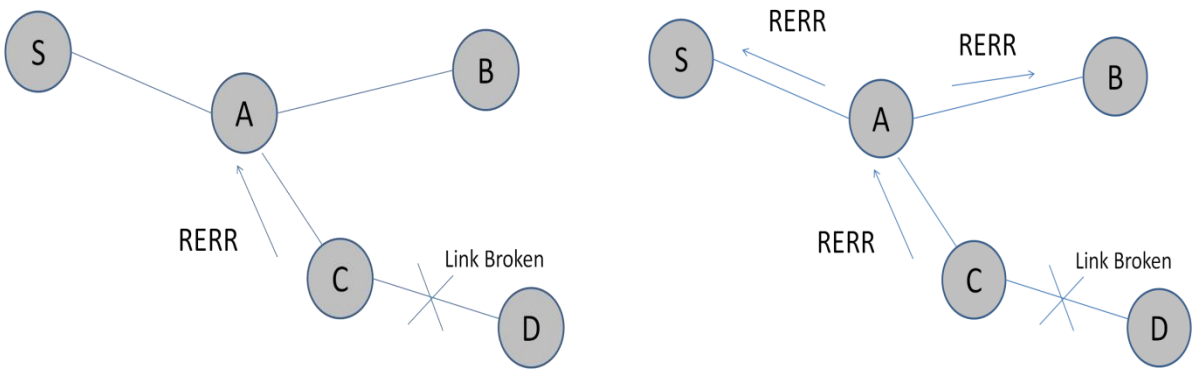
1. 2.

Figure 2.7 Reverse paths setting in AODV

From the figure 2.7 now we will try to understand that how AODV tries to establish the reverse path from the source to destination. As it is depicted in figure 2.7 1 that as A receives the packet RREQ from source node S it immediately sets a reverse link to S with the following settings, source IP address, destination S, next node S, and the HOP COUNT=1. Now it checks whether it has the path for D, if fails then it rebroadcast the packets in neighbors'. In this way D also receives the request and all set to reply the same. D replies this packet with following settings, current sequence number of the destination (D), HOP COUNT=0, and life time of the packet. If any intermediate node knows the route to the destination it also can reply for RREQ with the following settings, destination sequence number, its distance from the destination in term of HOP COUNT, and life time of the packet. In case a node receives multiple RREP, it forwards only it first one which it gets otherwise it also can forward the one which has greater sequence number.

6. AODV does its data delivery in this way. From the figure 2.7 2 S creates forward route entry for D and sets the fields as, Dest. =D, Next= A, HOP COUNT=3. There is also a provision of TIMEOUT in AODV. Table entry for the RREQ's AND RREP's will be deleted after a sufficient time in which communication can take place.

In the case of route failure at any juncture of communication, the node in upstream at which it occurs creates the RERR packet and signifies to all concern nodes which are associated with this path and may use it in near future. In the figure 2.8, let route breaks as shown by cross mark. C is the immediate node in the upstream so it will form the RERR packet and sends it back to the A, similarly A forward it to S. in this way S comes to know that the route for a particular destination has been broken. It starts the process of rediscovery of route afresh.



1. 2.

Figure 2.8 Link breakage and RERR creation

7. Link failure detection in AODV is done by the periodically exchange of HELLO packets in the neighbors'. If no reply will come in specified time it is assumed that the link is broken. Now will see that how sequence number can avert the loop in the network.

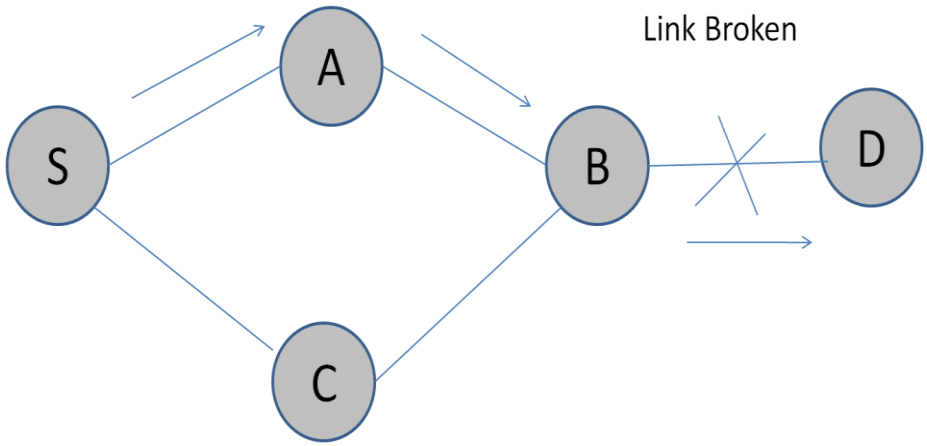


Figure 2.9 Looping case and remedy in AODV

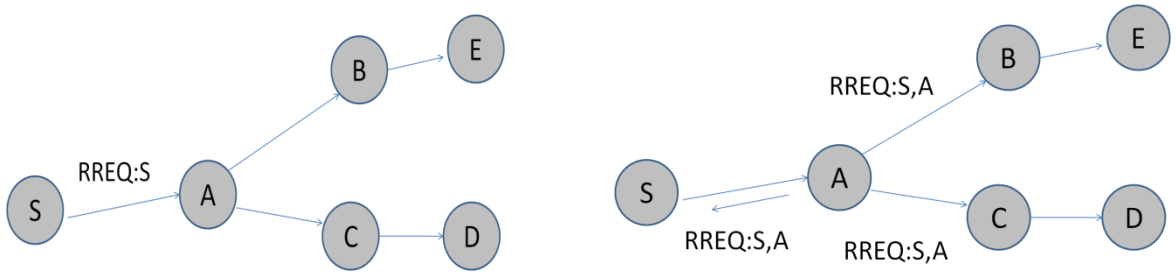
In the figure 2.9, we can see apparently that prior to link breakage there was a route from node S to node D. Now let after breakage of this link from B to D, B signifies it via A but somehow this RERR packet may go lost. After this let say B starts route discovery for the node D and S got

this RREQ from the node C. since S knows the route for D via A it replies to B. This may turn into a loop from S-A-B-C-S. But due to sequence number S will see that the message it got from C has higher sequence number for B than that it has in its own table, so it comes to know that is has withered information and need to be updated. [3]

2.4 Dynamic Source Routing (DSR)

DSR is also a reactive source initiated routing protocol for MANET. It has the capability to use unidirectional link also. When a node need to send a packet to a destination and it does not have the route information then it starts the route discovery procedure that's why it is reactive. One advantage with DSR is that, there is no need of periodic routing packets like HELLO message in AODV. It uses cache to store the route.

1. When source node start route discovery it floods RREQ message in its neighborhood with following settings, source address, destination address, request ID which is to be determined by the sender. Each intermediate nodes add its own identifier also when they forwards the message.



1. 2.

Figure 2.10 DSR route request

In figure 2.10 we can see the procedure by which S sends the REREQ packets to A and in figure 2.10 2, A also does the same by adding its own identifier. This process will go on till D receives the message. Here there may be reversal of the link if it is using IEEE 802.11 MAC;

since in that case route will be bidirectional otherwise there is another procedure of reply in DSR as well.

2. Route reply in DSR can take in following way. If one directional communication is allowed the RREP need to find reverse route from D to S. So D starts route discovery for S and the route from S to D will be piggybacked in the RREP packet itself, so at the end when S will get RREP packet from D, it will get the required path also provided S is reachable from D by at least one unidirectional path.

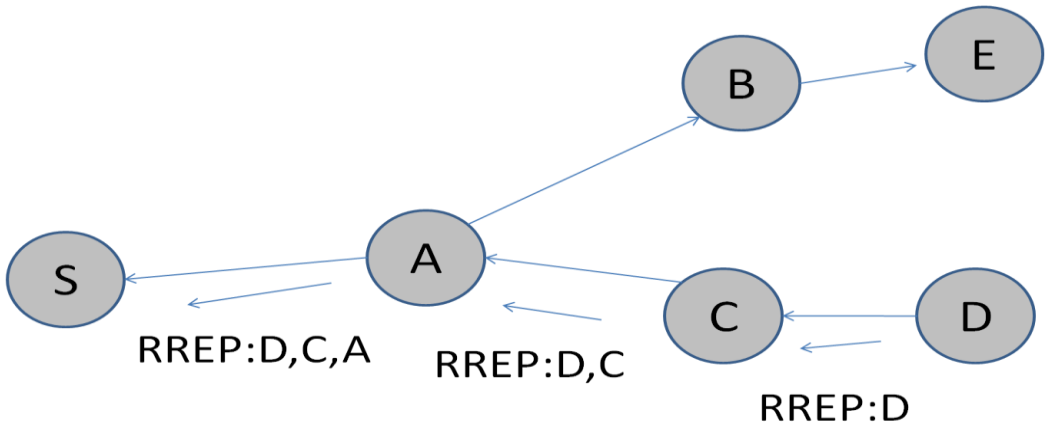


Figure 2.11 Route reply in DSR

In the figure 2.11 it is visible that node D starts sending the RREP packet, first it sends it to C. C gets the packet and adds its own identifier into it and forwards it to A. In this way this message will reach to the node S. S will cache this whole route from S to D and starts data delivery. In each packet it sends, the whole route is in the header of the packets.

3. New route discovery for the same destination should be postponed or delayed for some time if the node is currently unreachable to avoid overhead. For this purpose DSR uses exponential back off procedure. All the packets which are received during the back off will be cached for later treatment. In DSR some sort of optimization is also used. As whenever a node will get any route by any means it stores that. For example let us consider a hypothetical case that there is a path

from S to D as S-A-F-J-K-D. Then the entire node in this path will save the route as it may be use full for future because in this way F also comes to know route to D, same for others. [4]

Chapter 3

Related Work Based On Game Theory

A mobile ad hoc network is an autonomous set of nodes connected by the means of wireless links. It is characterized by no supporting infrastructures and high mobility of nodes within the system. Routing and packet forwarding capability is built in the nodes itself. Nodes can directly communicate to each other if they are in the transmission range of each other. If they are not in the direct range of each other (any two nodes) then, they have only one option for communication and that is by the means of relay transmission. In this system intermediate nodes forward the packets on the behalf of other node. This is only possible when nodes are ready to cooperate with each other up to a certain level. It means that in this type of systems, nodes have two types of packets one which they generate for their own need and the others which are coming from any other node to forward further. Since in this type of network the topology changes very rapidly and in an unpredictable manner that, to establish any type of cooperation is a tough task. In this type of system there may be four types of nodes. [5]

1. Cooperative nodes: Nodes which follows the standard of network for forwarding and routing purpose and always comply with them.
2. Malicious nodes: These nodes always try to jeopardize the system by misrouting flooding and dropping the packets.
3. Selfish node: these nodes always try to conserve their own resources as battery power for future use and not providing any support to network but always try to take services from the network for its own purpose. This is also termed as the exploitation of the network.
4. Inactive nodes: these nodes are very lazy in the term of awaking time and it is due to some constrained e.g. the energy constraint.

Selfish behavior of the nodes may be very threat full for the network as nodes in the ad hoc network are mostly characterized by low battery power, so nodes may have great temptation to

save the battery power for future use and, are not participating in the routing or forwarding the packets. [6]

There may be mainly two type of the node misbehavior in a network in terms of network layer.

1. Routing misbehavior: it's characterized by failure of the node to comply with the standard of the routing protocol.
2. Forwarding misbehavior: deny to work in accordance with the standard for the data transfer protocol.

3.1 Basics of Game Theory

Game theory basically is the branch of decision theory concerned with interdependent decisions. The problems of interest involve multiple participants, each of whom has individual objectives related to a common system or shared resources. Because game theory arose from the analysis of competitive scenarios, the problems are called games and the participants are called players. But this theory is not constrained to sport only but it can be applied with any problem in which there is competition or moreover in those situations in which, one player strategy depends on the other players move. Some areas of applications for the game theory are the sociology, political science, economics, mathematics, computer science etc. All type of problems in these fields requires a strategic thinking to opt for the best plan which could be deployed. In gaming, player's actions are referred to as moves. The role of analysis is to identify the sequence of moves that we should use. A sequence of moves is called a strategy, so an optimal strategy is a sequence of moves that results in our best outcome.

There may be two fundamental type of the game one is sequential and other is simultaneous. In sequential game player takes move one by one but in simultaneous game they all act simultaneously. We distinguish between these two because they require different analytical approaches. [7]

3.1.1 Sequential Game

To analyse sequential game first of all we construct game tree for all the possibilities and then,

we apply the basic principle which is called as “look ahead and reason back”.
This goes in the following way [7]

1. We look ahead to the very last decision, and assume that if it comes to that point, the deciding player will choose his/her optimal outcome.
2. Back up to the second-to-last decision, and assume the next player would choose his/her best outcome, treating the following decision as fixed (because we have already decided what that player will pick if it should come to that).
3. Continue reasoning back in this way until all decisions have been fixed.

Now we will take an example of the sequential game and will analyse it on our own parameters. Let there is a canteen in the JNU named GANGA which has monopoly over the price till now and making a profit of Rs. 500 per day. Now we also want to enter in this field and want to open a canteen. So there may be two cases for us, either we will open the canteen or will not. Let say if we will open the canteen there is two choices for GANGA either they will accept the competition or they will fight a price war. If they choose the former we both will make a profit of Rs. 200 each (this is due to fact that previously GANGA was having the price monopoly). And if they will opt for the price war it will incur loss of Rs. 200 for them and Rs. 300 for us. Now by analysing this problem we will see which decision we should take.
From the figure 3.1 now we will learn that how, look ahead and analyse back, works. First of all we go till end here two options for GANGA either, make a profit of Rs. 200 or bear a loss of Rs. 200. No doubt that being a no lunatic person they will choose a profit so it ultimately shows that now will enter the market. It is a very simple though exemplary example. We can opt more complex problem which we contain more stages to be analysed.

3.1.2 Simultaneous Game

From the name itself it is suggestive that, it is different from the sequential game as here there may not be any last stage and so look ahead and reason back will not work. Here we will consider a very simple but most famous example of its own kind, named as “prisoner’s dilemma”.

It is formulated as follows that there are two thieves who are caught in a police raid. They are interrogated in two different cells and given the following conditions.

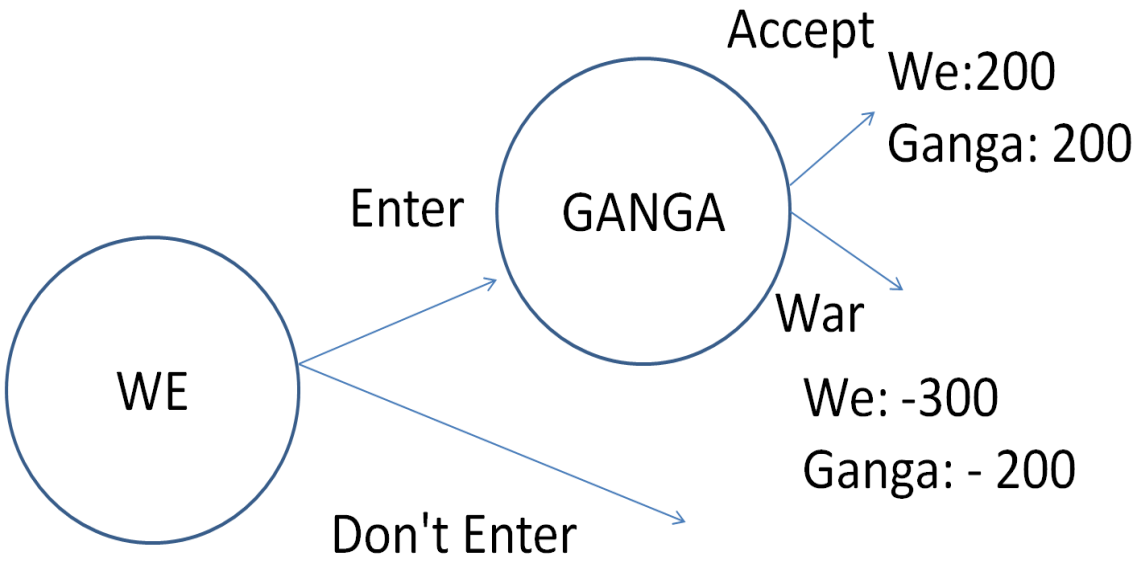


Figure 3.1 Game tree for the sequential game

1. If they both will confess they will be imprisoned for the term of 10 years.
2. If only one confess then he will be termed for 1 year and the other one for the 15 years.
3. And if they both will not confess then they will be equally termed for 3 years each.

Here we can't use look ahead and reason back as none of the decision is made first. This whole combination easily can be shown by game table.

The game table clearly shows that if the second player confesses then first player either will get 10 years term or 25 years depending upon its own decision so definitely he will confess. If the second player holds out the decision then again first player will confess as he will get only 1 year term. Second player also thinks identically. There are so many things in this game which we need to understand clearly. A dominant strategy has payoffs such that, regardless of the choices of other players, no other strategy would result in a higher payoff. [7]

		First prisoner's decision	
		confess	Hold out
Second prisoner's decision	confess	10	25
	Hold out	1	3

Figure 3.2 Prisoner's dilemma game table

This simplifies the decision for a considerable extent that if you have dominant strategy, just you use it. As we have already decided that both the player will confess. Both players also have dominated strategy, with payoffs no better than those of at least one other strategy, regardless of the choices of other players. This also makes the decision simple that dominated strategy should never be used. Third observation here is that if both the player uses their optimal strategy they will not reach to the optimal solution. Here we can see the value of communication, if they could talk to each other they both would have ended up with 3 years each. But it will not happen.

So from these all discussion we can conclude that if you have dominant strategy, use it otherwise look for dominated strategy and eliminate it from the table. Then again check for dominant strategy and so on. Till we find a dominant strategy of game is no further reducible. [7]

3.2 Game Theory in Ad Hoc Network

Here we will talk about ad hoc networks mainly in which the individual rational decision makers (players) are the nodes and such a model is called game. Each player is free to choose a action or move towards the game from a specified set of actions and the resulting outcome of the move is based on interaction among the nodes and effect of this move when all other players have utilizes their moves. This resulting outcome function is called utility function or the payoff function related to that node. [8]

3.2.1 Mathematical formulation

Mathematical form of a game is described by the following expression

G = <N, A, $\{u_i\}$>

Where N= {1, 2, 3, 4… n} is the set of players.

If $A_i$is the set of action available to player I then A= $A_1 \times A_2 \times \ldots \times A_n$

And $\{u_i\} = \{u_1, \ldots, u_n\}$ is the set of utility function which each player I wants to maximize, where u_i:A→R. [8]

3.2.2 Nash Equilibrium and Pareto Optimality

Let action chosen by the player I is a_i and for all the other players it is denoted by a_{-i} then combined of these is called action tuple a. So from above discussion we are in a situation that we can understand Nash Equilibrium. It is defined as that there is an action tuple where no individual player can benefit by unilateral deviation means it has no incentive to do it.

Let $a^* = (a_1^*, a_2^* \ldots a_n^*)$ is a NE then $u_i\ (a_i^*, a_{-i}^*) \geq u_i\ (a_i^*, a_{-i}^*)\ \forall a_i \epsilon A_i$ and for all $i\ \varepsilon$ N.

On the other hand Pareto optimality is used to measure the efficiency of the outcome. It is defined as an action tuple a is called Pareto optimal if there is no other action tuple b such that $u_i(b) \geq u_i(a) \forall\ i\ \varepsilon$ N and for some kϵ N,$u_k(b) > u_k(a)$. [9]

3.3 Why Game Theory?

Since over a decade people interest in the field of game theory has increased so its use in the ad hoc network. It is very well suited for such type of networks where nodes are distributed over an area and work as a player in a non-cooperative game. To analyze such type of arrangement game theory provides convenient way. In the ad hoc network nodes are constantly moving here and there and joining and leaving the network, in such a scenario whether we can certain that such a situation have any steady state and if yes, then whether the system will ever converge with such a set of strategy. This type of question can be answered with game theory. [8]

3.3.1 Modeling of routing techniques in ad hoc network (A general View)

In the recent years the game theory is applied on the network layer routing for different type of

routing techniques as link state routing, distance vector routing, and multicast routing. In analysis we try to compare and contrast different techniques. This may be basically based on the following parameters:

1. Soundness: it decides whether the router have the correct information about the network topology for the routing purpose under the rapid changing environment.
2. Convergence: how much time the routers are taking to find such information or to have a view of network topology.
3. Network overhead: it shows the amount of data transferred among the nodes for the convergence purpose.

Routing in ad hoc network is modeled as the zero sum game in which nodes plays against the network. In a zero sum game utility of payers are negative to each other and try to maximize its utility and other tries to minimize(it also maximizes but in negative terms it is going to decrease). The two cost component in this game may be network overhead and the performance metric.
Like for soundness if all the nodes have correct view of the network when the game ends then cost is 0 and if any of these have incorrect view then cost may be 1. The objective of the routers is to minimize the cost. Another issue related to routing is how to suppress selfish behavior of the routers.

3.3.2 Selfish behavior in packet forwarding and collapse of network

In the network there may be many nodes who try to conserve its resources and not participate in packet forwarding in a multi hop routes. With the help of above given model in table 1 we can understand that how game theoretical model finds a problem in a network and also finds the equilibrium condition. If we consider that the energy is the constrained then in a single stage game equilibrium solution is only one that none of the node is going to cooperate.
Let's have the strategy s= { $s_1, s_2, \dots, s_n$}
And let $\sigma = \{k \varepsilon N \mid s_k = 1\}$ the utility of any node k$\varepsilon\sigma$ is given by $u_k(s)$= (|σ |-1) - s_k = | σ |-2. Now consider that the node k unilaterally deviates to the strategy of not participating then its utility is given by |σ |-1. Since this value is greater than the previous value of | σ |-2 it means that strategy s only can be in a Nash Equilibrium if $\sigma = \emptyset$.

Symbol	Meaning
N	The set of nodes in the ad hoc network{1,2,3,…n}
S_k	Action set for the node k ; S_k ={0,1}
s_k	Action of node k: $s_k = 0$(not participating) ands_k=1 (participating)
S	Joint action set; S=$\times_{k\varepsilon N} S_k$.
S	s = $\{s_1, s_2, \dots, s_n\}$; sεS.
$\alpha_k(s)$	Benefit accrued when other nodes participate $\left(e.g., \alpha_k(s) = \sum_{i=1, i\neq k}^{n} s_k\right)$
$\beta_k(s)$	Benefit (or cost) to node k when it participates; for energy constrained nodes it is negative (e.g. $\beta_k(s) = -s_k$).
$u_k(s)$	Utility of node; $u_k(s) = \alpha_k(s) + \beta_k(s)$

Table 3.3 Game Theoretical Model for Node Participation in Ad Hoc Network

From this discussion, it is clear that we need to establish the participation among the node by the means of repeated game model. There are many such models as generous tit for tat (GTFT) and tit for tat (TFT). [8]

All other work in this context, we will discuss in the chapter 5 of this dissertation.

Chapter 4

Related Work Based On Trust and Reputation

Researchers have done a lot of work in this context with full of their Excellency. It is appreciable and mandatory to learn for taking our related work forward if; it could be possible for some extent at all. In this chapter we will discuss some of very basic and fundamental protocols for cooperation enforcement methods based on trust and reputation. Some of these are Watchdog Pathrater, Confidant, Core, Ocean, etc. We will learn know-how about these all protocols one at a time.

4.1 Watchdog Pathrater

In this section we will see how watchdog pathrater work to mitigate the routing misbehaviour and packet forwarding. It is assumed that this protocol is working over DSR until and unless specified separately. There is two entities in this protocol watchdog and pathrater. [10]

4.1.1 Watchdog

Watchdog tries to mitigate the routing misbehaviour by detecting it through the route. We will try to understand the procedure by following figure 4.1.

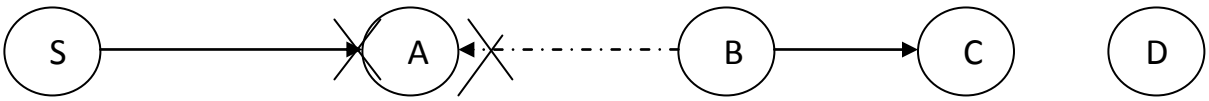


Figure 4.1 Collision of packet when A hears B

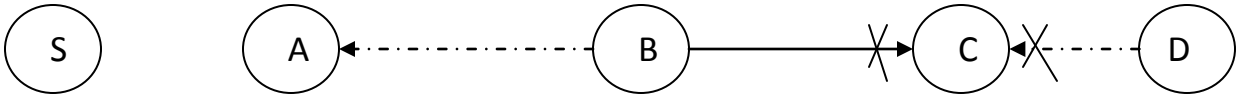


Figure 4.2 Collision at C which A can't hear

From the above figure 4.1 let there exists a path from the node S to the node D. S wants to send a packet to D. Each node in the network is equipped with a mechanism called watchdog. Let see how it will work. Let say A forward the packet to B in order that B can forward it to C and so on. By the time A forward this packet to B it starts listening B to ensure whether B forwards the packet to C or not. In this situation many things can happen. If B will not forward the packet and drop it voluntarily in order to conserve the CPU time or battery power which is the constraint in Ad Hoc network, A will increase counter related to B. After a time if this count will cross a threshold value, B will be declared misbehaving. But there are some problems in this mechanism. Say, B forwards the packet correctly and sends the acknowledgment to A which is received correctly and everything is fine so far. Now it may occur that by the time A will be receiving the acknowledgement from B at the same time S forwards the next packet to A. This certainly will culminate into a head on collision at A. so it may the case that A will increase the count of B for not forwarding the packet and after some time, when this count will accede to a certain brink B will be declared misbehaving node. [10]

If nothing of this kind happened and B forwards the packet to C and also A gets the acknowledgement properly, A will think that transmission is over from his side and he will keep quite. But situation may not be satisfactory at all as, another problem is looming forward. Say, by the time C will be receiving the packet from B, at the same time D starts transmission of certain kind and which may collide with this packet at C (figure 4.2). But the problem is this that A can't take any action in this regard whether packet is properly received by C or not. Now if B is a selfish node then it may decline to its moral responsibility to resend the packet as it has sent the acknowledgement to A and work is done from his side. If not this then B may work maliciously and intensely sends the packet when C is about to get another packet. So from above discussion

it is clear that as per provision each node in the system will keep the record of the nodes to which they listen over time. Now we will use this record to construct the route for a destination in future. [11]

4.1.2 Pathrater

The pathrater run by each node in the network, combines knowledge of the misbehaving nodes with link reliability data to pick the route which is most reliable. Each node in this system maintains a rating about every other node to which it can hear. It calculates a path metric by averaging the rating in the path. We choose this method as it gives comparison of overall reliability of different paths and allows the Pathrater to emulate the shortest path algorithm. If there are multiple paths then we choose the path with highest metric. Here it differs from the DSR which chooses the shortest path in the cache. Since in this mechanism Pathrater constructs the path which a packet has traversed over the time, it this must be built over the DSR. Pathrater assigns values to nodes in order to find the route. It gives value 0.5 to all neutral nodes and 1.0 to self node. It constructs the path with highest metric. It increases the value of nodes in the actively used path over the time. It assigns a highly negative value to misbehaving nodes, and after sufficient time it is gradually increased.

4.2 Confidant

Confidant protocol has the following components.

1. Reputation system
2. Trust manager
3. Path manager
4. Monitor

The finite state machine for the confidant protocol is shown the figure 4.3. Confidant is nothing but the explanation of the following figure and so we will do the same.

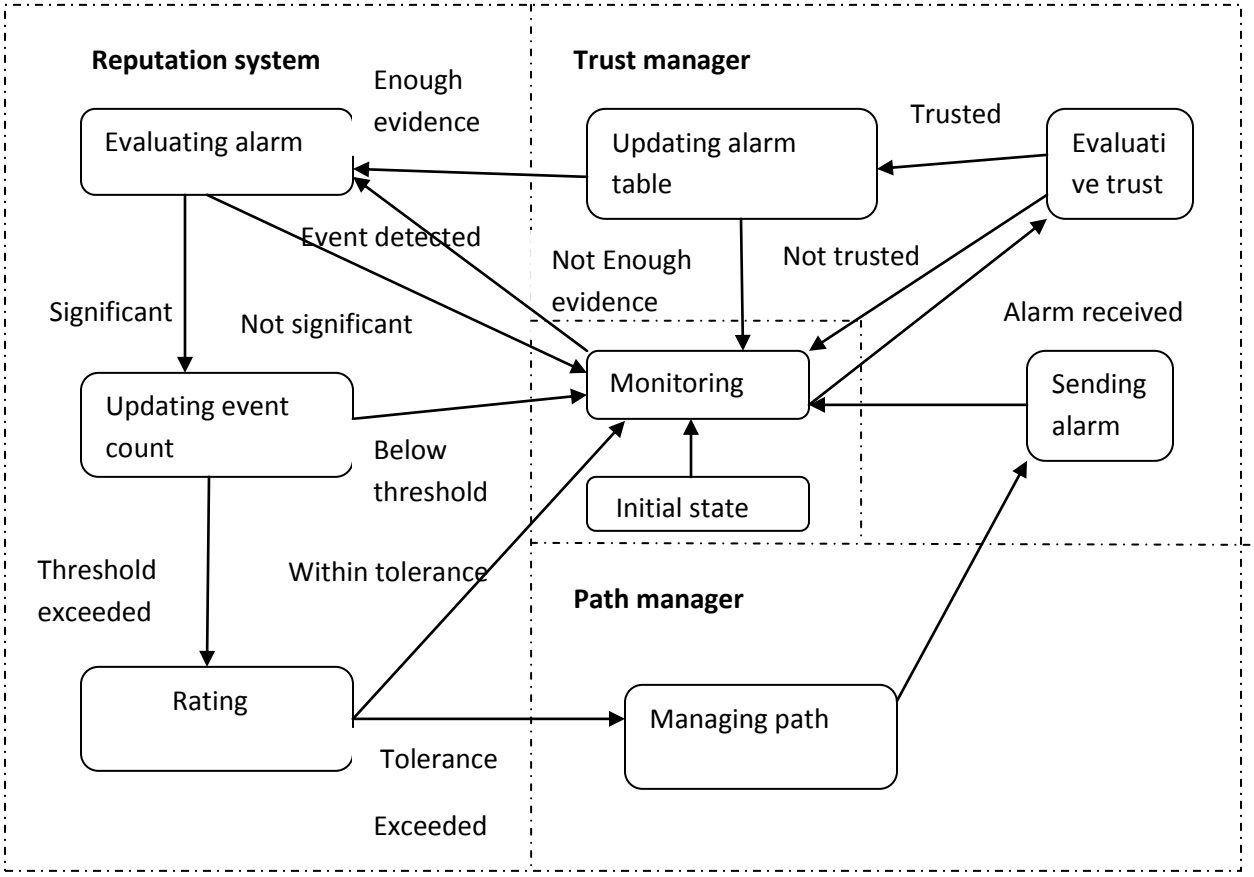


Figure 4.3 Components of confidant protocol

4.2.1 The Monitor

In a wireless networking environment, the nodes most likely to detect non-compliant behavior are the nodes in the vicinity of the offender and in some case the source and the destination, if they detect unusual behavior or do not get proper responses. The latter is not always the case, for instance the case of replay. The latter is not always the case, for instance in the case of replay. One approach to protocol enforcement and detection of damaging behavior suggested here is the equivalent of a neighborhood watch, where nodes locally look for deviating nodes. The nodes of the neighborhoods can be watched for the misbehavior, and also any change in the packet contain can also be detected if we are keeping a copy of the forwarded packet till the transmission is over. For any kind of misbehavior trust manager is called. [12]

4.2.2 The Trust Manager

Trust manager in the ad hoc network has to be distributed. Each node must have its own trust manager. When a potent and assured misbehavior is reported to a node, it immediately sends Alarm message to its entire friends (to those who are there in the list). Alarm message is containing the information about the misbehaving node and extent to which it is misbehaved. After getting the Alarm message the recipient's trust manager will first check the trustworthiness of the sending node. It may be assigned the following labels as unknown, known, fully known etc. According to this information now it is ready to take appropriate action against the reported node. [13]

Trust manager may consist of the following associated components.

1. It contains an alarm table which consist the information about the received alarms.
2. A trust table contains the information about nodes to check against the trustworthiness of the nodes.
3. A friend list of the node to which it sends the Alarm message in case of misbehavior detected.

For routing purpose trust is very necessary when making the following decision.

1. Providing or accepting routing information.
2. Accepting a node as a part of a route and.
3. Taking part in a route originated by other node.

4.2.3 The Reputation System

Reputation systems are used in some online auctioning systems. They provide a means of obtaining a quality rating of participants of transactions by having both the buyer and the seller give each other feedback on how their activities were perceived and evaluated. To avoid a centralized rating, local rating lists and black lists are maintained at each node and potentially ex changed with friends. In the route request nodes can include that black sheep (malicious nodes) be avoided for routing, which also alarms nodes along the way. Nodes can look up senders in the black list (also in the rating list) containing the nodes with bad rating before forwarding anything for them. The problem of how to distinguish alleged from proven malicious nodes, i.e., how to

avoid false accusations, can be lessened by timeout and subsequent recovery or revocation lists of nodes that have behaved well for a specified period of time. Another problem is scalability and how to avoid blown up list, which can also be address by timeout.

The reputation system in this way consist a list against the reputations or the ratings about the nodes and the ratings will be only changed, when it gets the sufficient evidence about the maliciousness of a nodes, which have exceeded a particular threshold value. Rating is then changed according to a function which assigns different weights to different kind of misbehavior. In this way if the rating of a particular node has fallen bellow to a definite threshold value then the path manager is called for action. [13]

4.2.4 The Path Manager

The path manager in this system is intended to perform the following task.

1. It re-ranks the path according to the received information about the security.
2. Deletion of path containing malicious nodes.
3. What action it must take after receiving a request for route from a malicious node.
4. Action on receiving a request for a route which contains malicious nodes in its source route.

All above traits of different components what we discussed above is depicted in self evaluating manner in figure 4.3.

4.2.5 Protocol Description

As shown in the figure 4.3 each node monitors the behavior of its next hop node for any malicious activity. If any suspicious activity occurs the information is passed on to the reputation system for the rating change and proper action against the node. If the event is significant for the node then it is checked whether it has occurred a significant number of times. If it has occurred a significant number of times then node is declared a malicious node. When trust manager receives any Alarm message it checks whether issuer of the message is trustable or not and accordingly it takes the action. [14]

4.3 Core

A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc networks. A simulation study presented in showed that the performance of MANET severely degrades in face of simple node misbehavior. Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks, those functions are carried out by all available nodes. This very difference is at the core of the increased sensitivity to node misbehavior in ad hoc networks. Now first of all we will define the different components of the Core.

4.3.1 Network Entity

The network entity corresponds to a mobile node. Each entity is equipped with a set of Reputation Tables and a watchdog mechanism. The reputation table and the watchdog together constitute the basis of the collaborative reputation mechanism presented in this paper. These two components allow each entity to observe and classify each other entity that gets involved in a request/reply process, reflecting the cooperative behavior of the involved parts. The classification of the entities based on their behavior is then used to enforce the strong binding between the cooperative behavior of a subject and the utilization of the common resources made available by all the other entities of the network. [15]

4.3.2 Reputation table

Reputation is data structure stored in the entity. It has the following components (data in each row).

1. Unique identifier for the entity.
2. A recent observation about the behavior of that entity.
3. A list of the indirect reputation values about entity.
4. Value of reputation defined for predefined function.

4.3.3 The Watchdog Mechanism

The watchdog mechanism implements the validation phase and it is used to detect misbehaving nodes.

4.3.4 Protocol

The Core protocol implements two types of entity, a requester and one or more providers that are within the transmission range of the requester. The nature of the protocol and the mechanisms on which it relies assure that if a provider refuses to cooperate (i.e. the request is not satisfied), then the CORE scheme will react and it will decreasing the reputation of the provider, leading to its exclusion if the non-cooperative behavior persists. Following may be the scenario which may arise between requester and the provider.

1. Protocol execution when there is no misbehavior detected: First, the requestor asks for the execution of a function to the provider. It then will activate the watchdog related to the provider for the required function and waits for the outcome of the watchdog within a predefined time out. Since the two parties correctly behave, the outcome of the watchdog assures that the requested function was correctly executed and the requestor disarms the watchdog. We suppose that the reply message corresponding to the result of the execution of function includes a list of all the entities that correctly participated to the protocol: the requestor uses this indirect information to update its reputation table and enters in an idle mode.
2. Protocol execution when there is a proven misbehavior: Since we suppose that the provider does not cooperate, the outcome of the watchdog will be negative. The requestor will then update the entry in the reputation table corresponding to the misbehaving entity with a negative factor and will enter in an idle mode.
3. Request made by a misbehaving node: Upon receiving the request for the execution of a function the entity checks the reputation value evaluated for the requestor in its global reputation table. If the reputation value is negative then the entity will not execute the requested function. It has then the choice whether to notify or not the denial of service.

4.4 Ocean

The observation-based cooperation enforcement in ad hoc networks, it introduces an intermediate layer that resides between the network and the MAC layers. This layer helps the nodes to make intelligent routing and forwarding decisions. It is designed on the DSR level, but its principles can be applied to other routing protocols, as well. OCEAN relies only on first hand

observations. In this scheme every node maintains ratings for each neighboring node and monitors their behaviors through promiscuous observations. It will be positive or negative, depends on the behavior of the nodes which are expected to forward the packet. The absolute value of decrement is chosen to be lower than that of increment. When value will drop below to a specified threshold also called faulty threshold node will be added to faulty list. This list is to be constructed by nodes itself, means by their own knowledge. This list will be flooded with the route request RREQ message in DSR. A route is rated good or bad, based on whether the next hop in the route belongs to the avoid-list. The receiver of an RREQ decides to drop it or to further process it, if the intersection of the avoid-list and the DSR route in the RREQ packet is void. [10]

In this way, each node along a route, makes its own decision about the trustworthiness of other nodes, and has control only over routes that it belongs to. Every node rejects the data packets if it arrived from the nodes belonging to its faulty list. Thus, misbehaving nodes are eventually isolated over time. However, a second chance mechanism is used to allow nodes that misbehaved in the past to let them become operational again. After a certain period, a misbehaved node is excluded from the faulty list and assigned with a neutral rating. OCEAN uses a different policy to deal with nodes that do not participate in the route discovery process. This policy, affected by the credit-based models, requires no tamper-proof hardware or a central server. Each node measures the behavior of its neighbors by directly interacting with them.

In this chapter and in previous chapter we discussed few very popular and in use cooperation enforcement methods. Some of them were broadly based on the game theoretical model and few were from trust and reputation. There is umpteen other protocol and fields which we left altogether like currency based model, which we hope will be discussed in our next thesis. So above four chapters described basic features of MANET, some of its routing protocols and cooperation enforcement methods. From here onward we embark our journey for only our area concerned and proposed enhancement in existing algorithms.

Chapter 5

Underlying Protocol and Proposed Enhancement

5.1 Introduction

A mobile ad hoc network is an autonomous set of nodes connected by the means of wireless links. It is characterized by no supporting infrastructures and high mobility of nodes within the system. Routing and packet forwarding capability is built in the nodes itself. Nodes can directly communicate to each other if they are in the transmission range of each other. If they are not in the direct range of each other (any two nodes) then they have only one option of communication and that is by the means of relay transmission. In this system intermediate nodes forward the packets on the behalf of other node. This is only possible when nodes are ready to cooperate with each other up to a certain level. It means that in this type of system nodes have two types of packets one which it generates for its own need and the other which is coming from any other node to forward further. Since in this type of network the topology changes very rapidly and in an unpredictable manner that to establishment of any type of cooperation is a tough task. In this type of system there may be four types of nodes: [8]

1. Cooperative nodes: Nodes which follows the standard of network for forwarding and routing purpose and always comply with them.
2. Malicious nodes: These nodes always try to jeopardize the system by misrouting flooding and dropping the packets.
3. Selfish node: these nodes always try to conserve their own resources as battery power for future use and not providing any support to network but always try to take services from the network for its own purpose. This is also termed as the exploitation of the network.
4. Inactive nodes: these nodes are very lazy in the term of awaking time and it is due to some constrained e.g. the energy constraint.

There may be mainly two type of the node misbehavior in a network in terms of network layer.

1. Routing misbehavior: it's characterized by failure of the node to comply with the standard of the routing protocol.
2. Forwarding misbehavior: deny to work in accordance with the standard for the data transfer protocol.

In this discussion we will consider and discuss only about the forwarding misbehaviors. [8]

5.2 Proposed protocol

Here we will discuss a strategy of cooperation enforcement which basically relies on the trust and activity. Let's start our journey of understanding the protocol. First of all we need to understand how; nodes in the network will update their respective trust about other nodes. This mechanism is shown in the figure 5.1.

5.2.1 Trust evolution

Here we will start our discussion from an example. Let say A wants to send a packet to node E. Let assume that route discovery algorithm DSR has been applied and node A is availed from the route from A to E.

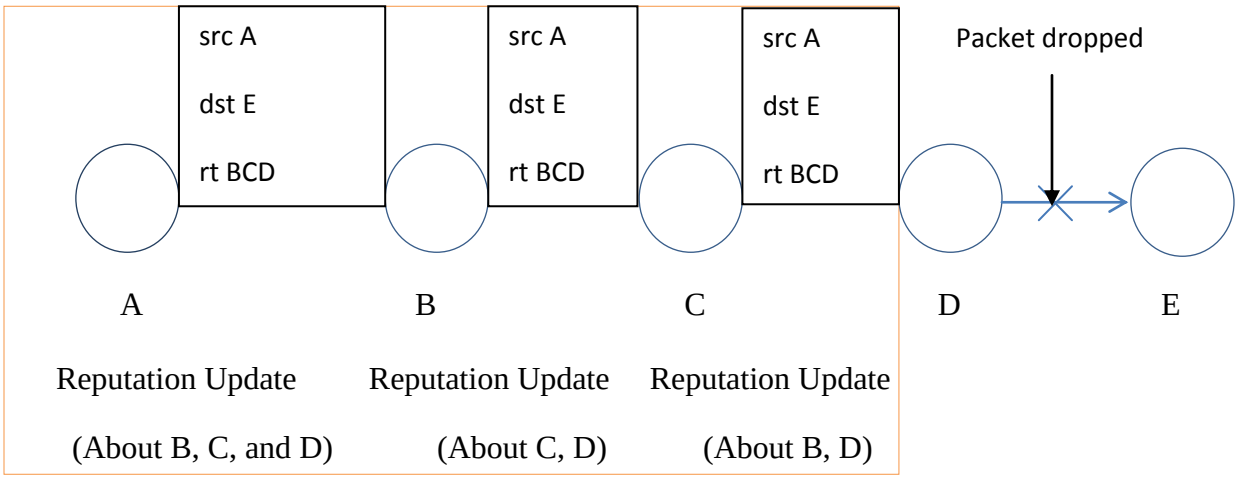


Figure 5.1 Trust update mechanism

A will incorporate that route with the packet and forward it to the node B, B to C and communication will go like this. If everything will be fine packet will be delivered to E. As shown in the figure all nodes in between the route will update reputation about other nodes. Now a situation may be there that node D did not forward the packet correctly and discarded it. In this case C will forwards message about the misbehavior of node D and all other node update the reputation against D. [16]

Now suppose before forwarding the packet of A node B wants to know the trust level of node A.

For this purpose B looks that how many packets has been sent to node A and how many it has Forwarded. Ratio of these two is called *forwarding rate* of node A. B will have a lookup table for deciding the trust and it will check this value from the table. This table is shown in the table 5.1. We can calculate forwarding rate (Fr) = (no of packet forwarded by node A)/ (packet sent to A).

Table 5.1 Showing the Trust Level

Fr	Trust Level
1 to 0.85	3
0.85 to 0.7	2
0.7 to 0.4	1
0.4 to 0	0

5.2.2 Activity Evaluation

Here we need to calculate the activity level of a node. It's a necessity because we can't trust only on the trust level as we know that in ad hoc network due to the battery constraint nodes always have temptation to sleep. Due to this behavior we need to check the activity of the node to ensure that it is not sleeping or at least it will be available to forward the packet. How to calculate activity level this we can understand from the figure below.

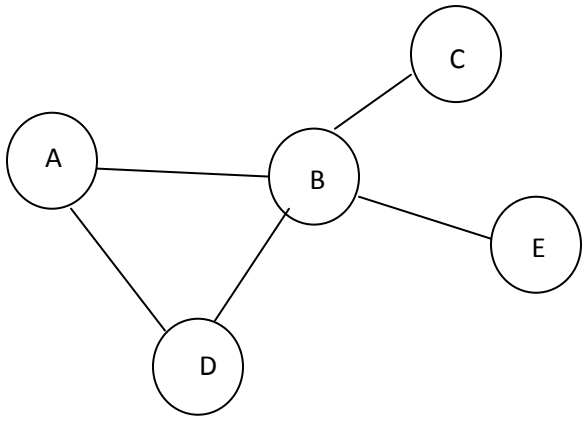


Figure 5.2 A section of network for calculation of activity level

From the figure 5.2 let node B wants to know about the activity level of the node A. it can calculate it from the following formula.

Activity = (Fr of node A)/ ((Fr of node D + Fr of node C + Fr of node E)/3)

Later according to above value of activity we can calculate *Activity level* according to table given below.

Table 5.2 Calculation of activity level

Activity	Activity Level
> 0.9	3
< =0.9 && > 0.7	2
<= 0.7 && > 0.5	1
<= 0.5 && > 0.0	0

5.2.3 Strategy for *strategy*

Decision of a node whether it will forward the packet received or discard depends on the strategy which is coded in the binary form. 0 will stand for discarding of the packet and 1 will stand for

forwarding the packet. Decision predominantly will depend on the trust and activity level of the node wants its packet get forwarded. A sample coding strategy is shown in the figure 5.3 below.

Trust →	0				1				2				3				
Activity →	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Decision →	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0

Figure 5.3 A sample of strategy for the nodes

The last entry in the second row shows the strategy for an unknown node.

5.2.4 Lets play

We define a game in ad hoc network in which a particular node wants to send a packet to some other node and intermediate nodes participate in that. Number of participant depends on the length of the path. Source node and intermediate nodes are called the participants. Intermediate nodes are chosen randomly. Let's take an example. Let say node A wants to send a packet to D in the figure 5.4. Then in this case this will be called A's game and A, B, C all are participants.

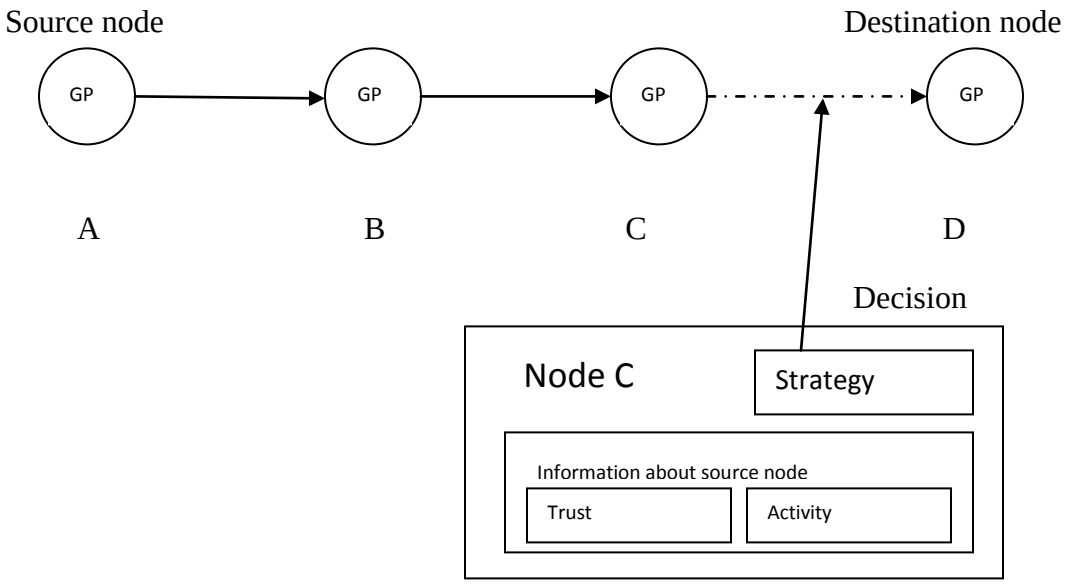


Figure 5.4 Ad hoc network sample game

Node takes decision based on their strategy whether to forward a packet or not. It is clearly depicted in the above figure. [16]

5.2.5 Payoff to the participants and Fitness of the nodes

In this part we will try to do some enhancement over the original protocol. Here we will discuss and design three types of the payoff tables. One for the intermediate nodes which will heavily depends on their decision to forward or discard the packets and trust and activity level of the source node. Second table will be for rewarding the nodes to go in sleep mode and save the battery and their interaction with the network. Such a table is depicted below in table 5.3.

Now we will discuss the pay off for the source node. Here our proposal is that payoff for all the source node should not be the same. As we know that this network is full of the nodes which are showing selfish behavior so in such a situation if a node is able to get its packet forwarded further must be rewarded with greater pay off. For this purpose here rather than using the constant pay off we will use a function for this purpose which will depend on number of the hop, packet will travel.

Let x is the number of hop so pat off for this will be $pay\ off = x * a^{-x}$. ……..*5.1*

For moderate value of a, we can see the performance of this function in figure 5.5 below.

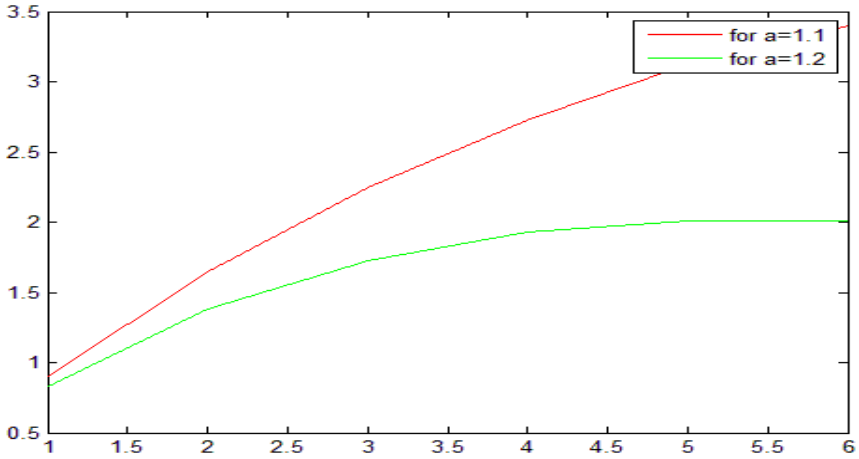


Figure 5.5 performance of function 5.1

Here we can see the moderating behavior of this function as number of hop is increasing. So according to the size of network and number of selfish node we can set the value of a most preferably between 1.1 and 1.4. It will help and suppress the selfish behavior rapidly. Table for pay off to the nodes going in sleep mode is shown bellow.

Table 5.3 Payoff for going in sleep mode

Mode of the interface in a round	Payoff(per round)
Sleep	5
idle	0

Payoff for intermediate nodes is depicted in the Table 5.4 below.

In equation 5.2 sop stand for the pay off which corresponds to sending own packet, fpb stands for forwarding packet on behalf of others, pdp stands for pay off discarding packets, psm stands for pay off in sleep mode and nae is net total events occurred.

Table 5.4 Payoff for intermediate nodes

Reliability of source node		Payoff	
Trust	Activity	Forward	Discard
3	3	15	0
	2	12	2
	1	8	3
	0	2	6
2	3	12	2
	2	8	6
	1	5	8
	0	1	10
1	3	10	5
	2	6	7
	1	2	9
	0	0	12
0	3	6	5
	2	3	7
	1	1	9
	0	0	14

Fitness can be calculated from the formula below. [16]

$$\text{Fitness} = (sop + fpb + pdp + psm)/ nae \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.2$$

5.2.6 Evaluation and evolution of strategy

Strategy of each player will be evaluated in ad hoc network tournament game. In every round each node is source of packet only once and it will act like intermediate player many times. How this whole procedure will work it is written in box below.

Algorithm for playing game

Step 1: Let K is the number of node in tournament and i= 1:K, r=1 and R is total round to be played.

Step 2: we will randomly select intermediate nodes and destination node.

Step 3: For each available path from source of packet to destination we will calculate reputation value based on trust and reputation.

Step 4: Play the game as shown in figure 5.4.

Step 5: Update the pay offs of the corresponding nodes including source node.

Step 6: Update the reputation about participant.

Step 7: If i<K i++.

Step 8: If r<R r=r+1, go to step 1 else stop the game.

In this discussion we will check our proposal against only two types of the nodes. First is the normal node (NP) which will cooperator and follow the protocol and second is the selfish node (SP) which always has the temptation to violate the protocol. In first hand all the nodes of normal player will be given random strategy and later they will evolve by the means of genetic algorithm operators (selection and reproduction). Such a diagram is depicted below in figure 5.6. [7]

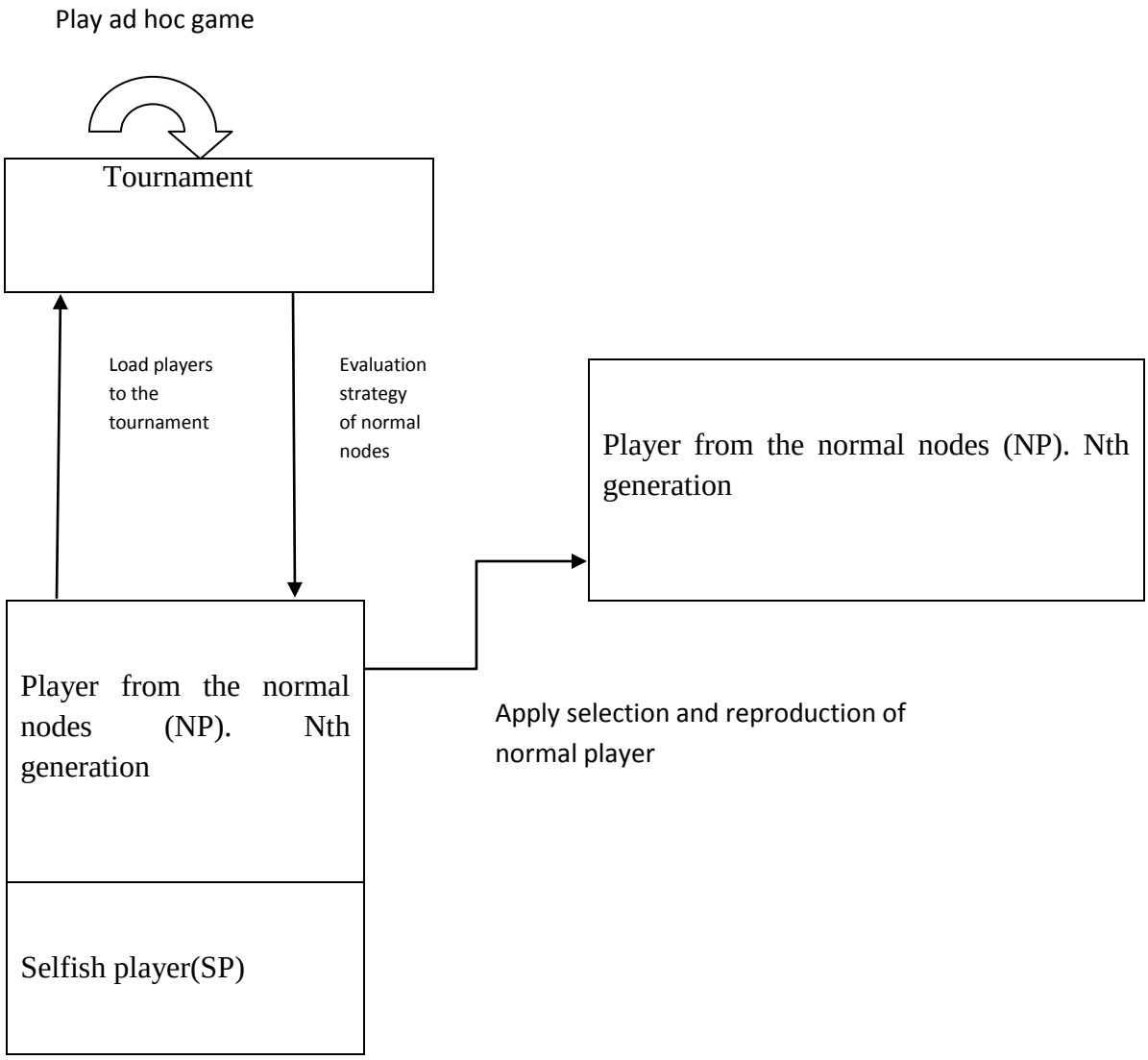


Figure 5.6 evolution of the population

The simulation results and performance of this whole protocol, which we discussed above, will be discussed in next chapter. [17]

Chapter 6

Simulation Result Description, Conclusion and Future Work

6.1 Tool wielded

We are using Matlab in this case for simulating and programming our protocol. MATLAB is a numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, [18] plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN. Here we have written our program in C++ and interacted with MATLAB. C++ is an object oriented language and capable of handling distributed programming. [19]

6.2 Know how about object-oriented programming

Object-oriented programming (OOP) is a programming paradigm using "objects" – data structures consisting of data fields and methods together with their interactions – to design applications and computer programs. Programming techniques may include features such as data abstraction, encapsulation, messaging, modularity, polymorphism, and inheritance. Many modern programming languages now support OOP, at least as an option. Let us take an example how we created our node in C++ with the help of classes. [20, 21]

```
Class node
{
Name of the node;
Initial forwarding values;
Strategy of the nodes;
Connection information about other nodes;
Number of packet received;

Public:
Function to operate on these data above.
}
```

With above class we can create different node object and can simulate our network. [21]

6.3 Simulation environment and result analysis

Here we will simulate a hypothetical wireless which will be simulated as ad hoc only through different settings (due to programming constraint). The actual set of nodes and sample and related information about network is mentioned in the box below.

1. Set of nodes 20.
2. Transmission range is set hypothetically.
3. 5 selfish nodes.
4. Underlying routing protocol is DSR
5. Length of strategy 17
6. One point crossover and 2 bit mutation is used.
7. A function for deciding the trust based on packet received and packet forwarded is used.
8. A function for deciding the activity based on packet received and packet forwarded is used.
9. A function to decide whether the packet will reach to destination or not is used
10. If packet reaches the destination, updating mechanism for reputation is used through a different function.
11. An all together dummy environment is created through programming which may have its own faults but enough for our simulation and desired result.

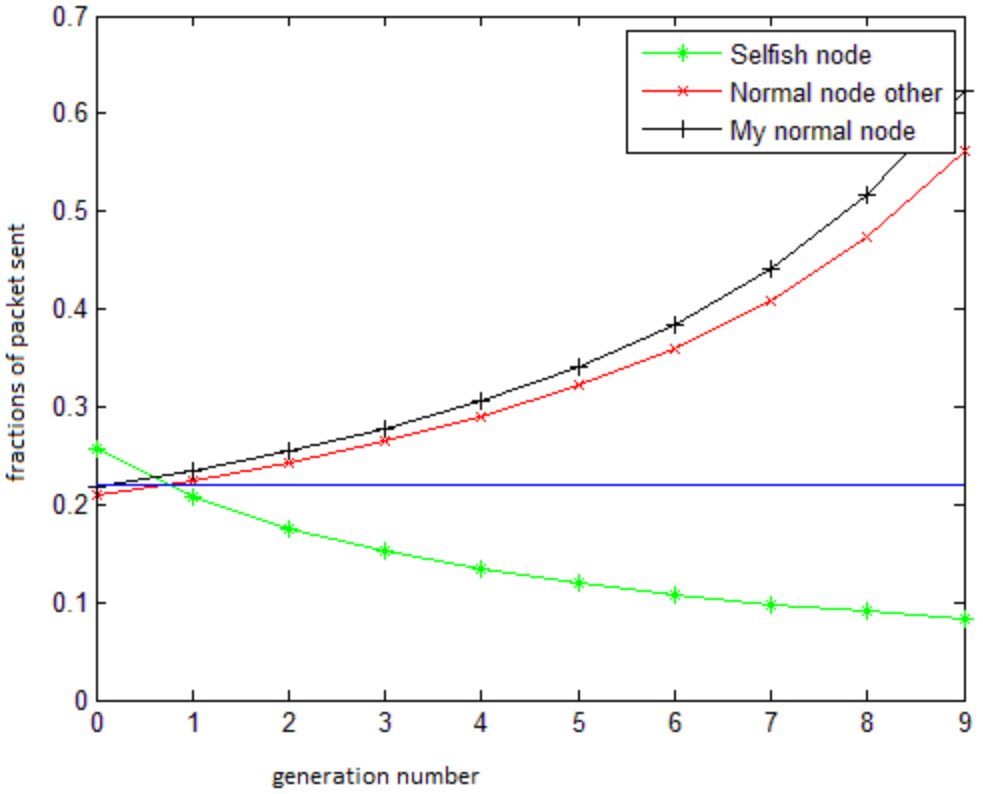


Figure 6.1 Diminishing selfish behaviour and evolving peformance of normal nodes

The simulation results are shown in the figure 6.1 and 6.2. In figure 6.1 it is shown that how selfish nodes start and, they are able to send even more fractions of packets initially with randomly distributed strategies. But after 2 to 3 evolution of generation, normal nodes come to know that network has some selfish nodes and they started declining to send packets of those nodes. And from here onward they, by their continued evolution, are able to increase their fraction of packets successfully sent. Due to change in fitness evaluation and payoff for source node we can see the clear difference in two set of normal performing nodes. For example for the generation number 0 means initially when game starts due to randomness in the strategy selfish nodes perform well but after the first generation their performance started declining and it continues onward. For generation number 2 selfish nodes send 18 percents while normal nodes send 25 percents, for generation number 6 differences is going to even wider from 7 percents to 22 percents. It is also apparent from the figure that the difference between our proposed protocol

and underlying protocol is also increasing in terms of percentage of packets sent. For generation number 4 it was 2 percents but for generation number 9 it is almost 5 percents. [16, 20, 21]

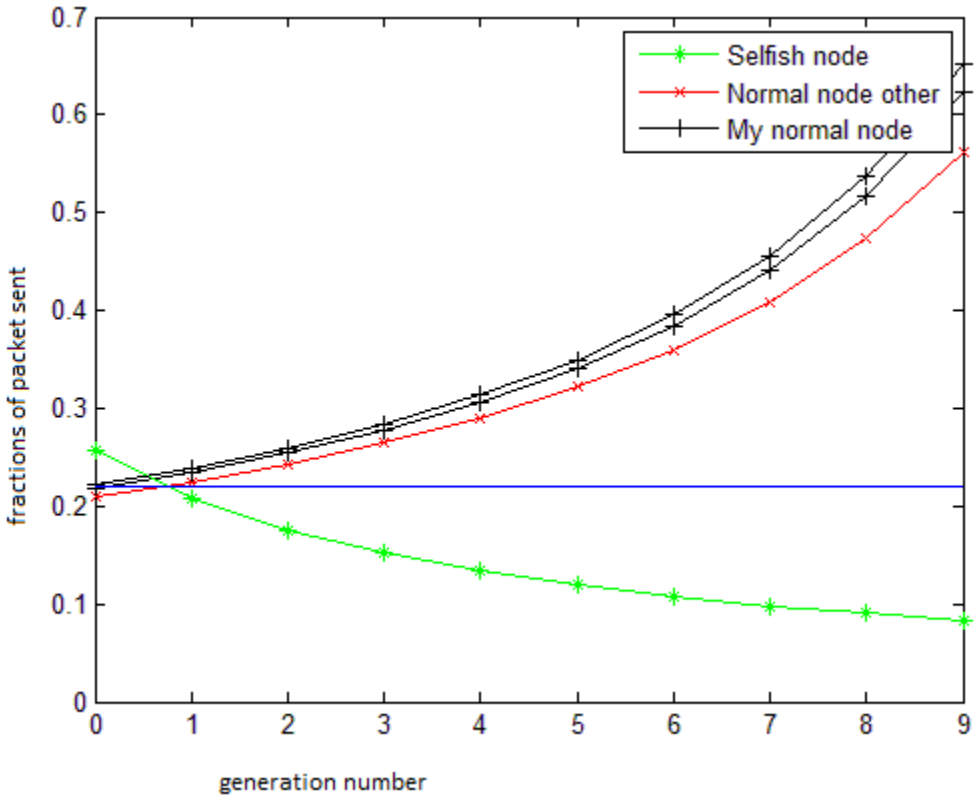


Figure 6.2 Relative prformance of normal nodes and with changing value of a.

In figure 6.2 a comparison between performances of two sets of normal nodes are shown for different value of a in equation 5.1. But the range of a between 1.2 and 1.4 is valid for small network only. When we increase value of a percentage of packets sent is definitely increasing but for underlying protocol it is constant. By the time generation number reaches 9 difference is almost 1 percent.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Performance of Ad Hoc network heavily depends on the nodes cooperation participating in the packet forwarding. It is quite vulnerable due to its very nature of distributed function over an area without any central authority. Also due to limited channel bandwidth and less capable battery power nodes always have the temptation to decline from packet forwarding. We derived a frame work using game theory and genetic algorithm which equipped with reputational behavior of node, is giving desire result for some extent. We developed a enhanced algorithms built over "preventing selfish behavior" in ad hoc network which will further help to suppress selfish behavior of node and they will not be able to take the services from the network.

In our proposed protocol we took a new set of payoff tables and also incorporated formulae to calculate different payoff. Also we used all together new set of formulae to calculate the activity ant trust level of the nodes. Here we took help of genetic algorithms operators to evolve the strategy for the nodes by the help of single point crossover and 2 bit mutation. The result which we got from the simulation is good in the term of sending fractions of packets successfully for the normal nodes and the selfish nodes. It is also evident from the result that, as the generation of evolution increases, the selfish behavior of selfish nodes are detected and packets sent by them is gradually decreasing.

At last we will conclude that the idea used in this dissertation is quite logical and supplementary to the original algorithm, it, if incorporated with it is giving better result apparently and must be appreciated.

7.2 Future Work

There are numerous vistas in the field of Ad Hoc network to work on. We chose a tiny field out of this. There are many methods for cooperation enforcement. We acquired one or two out of that and did some work. We will take this whole journey on even diverse kind of environments in upcoming time. Chances and need for improvements are immense. We are also trying to develop protocol which neither depends on incentives nor reputation. Such an algorithm is there OMH. In future we transform over platform from MATLAB to NETWORK SIMULATOR. Three we can compare our protocols in even close to reality situations. In future we will trudge a way of wisdom and try to hang on our best endurance and intelligence.

Bibliography

1. C. Shiva Ram Murthy, and B. S. Manoj, "Ad-hoc Wireless Networks: Architectures and protocols," Copyright © 2004 by Pearson education, Inc. pp 12-19, 255-312.

2. J. Zhao Sun," Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing", MediaTeam, Machine Vision and Media Processing Unit, Infotech Oulu, 2002.

3. B. Awerbuch & A .Mishra, "Ad hoc On Demand Distance Vector (AODV) Routing Protocol", Department of Computer Science Johns Hopkins, 2004.

4. B. Awerbuch & A. Mishra, "Dynamic Source Routing (DSR) Protocol", Department of Computer Science Johns Hopkins, 2004.

5. B. Wang, Y. Wu, Z. Ji, K.J. Ray Liu, and T. Charles Clancy, "Game theoretical mechanism design method",ieee signal processing magazine [74] November 2008.

6. L. Buttyan and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks, ACM/Kluwer Mobile Networks and Applications (MANET) Special Issue on Mobile Ad Hoc Networks", vol. 8, no. 5, Oct. 2003.

7. A. Kumar Dixit, and J. Nalebuff Barry, "Thinking Strategically", New York: W. W. Norton & Co., 1991.

8. V. Srivastav, J. Neel, A.B. Mackenzie, R. Menon, L. A. Dasilva, J. E. Hicks, J. H. Reed, and R. P. Gilles,"Using game theory to analyses the wireless ad hoc network",IEEE Communications Surveys & Tutorials • Fourth Quarter 2005.

9. M. Felegyhazi. L. Buttyan and J.-P. Hubaux,, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks", Mobile Computing, IEEE Transactions on, vol.5, no.5, pp. 463-476, May 2006 doi: 10.1109/TMC.2006.68.

10. G. F. Marias,y, P. Georgiadis, D. Flitzanis and K. Mandalas," Cooperation enforcement schemes for MANETs: A survey", Department of Informatics and Telecommunications, University of Athens, 15784, Greece, Department of Informatics, Athens University of Economics and Business, 10434, Greece, 2007.

11. S. Marti, T.J Giuli, K. Lai, M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks”, Department of Computer Science Stanford University USA, 2000.

12. J. Mundinger, J. Le Boudec. "Analysis of a Reputation System for Mobile Ad-Hoc Networks with Liars,"wiopt, pp. 41-46, Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, andWireless Networks (WiOpt'05), 2005.

13. S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol ", in Proc. 3rd ACM Intl. Symp., on Mobile Ad Hoc Networking and Computing, Jun ’02.

14. A Marianne, M. El-Kassas AzerSherif Abdel, F. Hassan Wahab, S. El-Soudani Magdy “A Survey on Trust and Reputation Schemes in Ad Hoc Networks ”IEEEComputer society magzine Magazine, vol. 43, no. 7, July 2005.

15. P. Michiardi and R. Molva, “CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks”, In Proc. 6th IFIP Commun. and Multimedia Security Conf., Sept. ’02.

16. M. Seredynski, P. Bouvry and M. A. Klopotek,”Preventing selfish behavior in ad hoc networks” 2007 IEEE congress on evolutionary evaluation.

17. D. Whitley, “A Genetic Algorithm Tutorial”, Colorado State University, 2005.

18. C. Xenophontos, “A Beginner’s Guide to MATLAB”, Loyola College, 2008.

19. A. H. Register, “A Guide to MATLAB Object-Oriented Programming” , Georgia Tech Research Institute Atlanta, Georgia, U.S.A, 2008.

20. H. Scheldt,” C++: The Complete Reference Third Edition”, Osborne McGraw-Hill, 2006 edition.

21. http://en.wikipedia.org/wiki/Object-oriented_programming.

22. Bryan’s Dynamic Source Routing FAQ , http://www.skynet.ie/~bryan/dsr_faq/