MANET ROUTING PROTOCOL USING TRUST METRIC

by

Md. Amran Hasan Dipu, ID#2005-2-80-018
Md. Tanzil Khan, ID#2005-2-80-041

Submitted to the
Department of Electrical and Electronic Engineering
Faculty of Sciences and Engineering
East West University

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering
(B.Sc. in EEE)

Summer-2009

Approved by

Rahman 29/10/2009

Dr. Mohammad Ghulam Rahman
Thesis Advisor

A Haque 25.11.09

Dr. Anisul Haque
Chairperson

ABSTRACT

Mobile Ad Hoc Networks (MANETs) are communication networks in which all nodes are mobile and communicate with each other via wireless connections. Main aim of MANET is to establish communication route from source node to destination node through a community of mobile devices that interconnect, interact and collaborate with each other. An important and essential issue for MANET is routing protocol design, which is a major technical challenge due to the dynamism of the network. In this thesis we present the state-of-the-art review and comparison for typical representatives of routing protocols designed for MANET. The performance of ad hoc networks depends on the cooperative and trust nature of these devices or nodes. The initiator node has to rely on other intermediate nodes to relay the message to the destination node. In this paper, we propose a trust management scheme for MANET environments, where trust value is utilized for establishing the communication link between the node that initiates the communication (initiator node) and the destination node. We utilize graph theoretic approach in our proposed scheme to find the path and we utilize both the direct trust and recommendation of the neighboring nodes for computing the trusted communication path. By this trust management scheme the highest trusted but minimum routing path is achieved from the initiator node to the destination node. Our trust management scheme can be included in any efficient routing algorithm to achieve highest trusted but minimum routing path from the initiator node to the destination node.

ACKNOWLEDGEMENT

We would like to express our deep gratitude and appreciation to our supervisor Dr. Mohammad Ghulam Rahman, Assistant Professor, Department of the Electrical and Electronic Engineering, East West University, for his excellent guidance and encouragement. The thesis completion is a result of his unparallel support and care.

Special thanks to Dr. Anisul Haque, Chairperson of EEE Department, East West University for his, guidance, support and care.

We also thank all the respected faculty members of the EEE department of East West University for their continuous inspiration and support.

We are thankful to M.A Jobayer Bin Bakkre, the Lab-in-Charge of VLSI Lab and all other Lab-in-charges of Electrical, Electronics and Tele-communications Labs.

Special thanks to our friend Mohammad Farhad Hossain for his moral support and helpful discussion during this work.

We are grateful to our family members and classmates for being there when we needed their help.

Above everything we thank Almighty Allah, for giving us the ability, scope, patience and energy to complete this commitment efficiently.

AUTHORIZATION PAGE

We hereby declare that we are the authors of this thesis. We authorize East West University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Dipu 29/10/2009

Md. Amran Hasan Dipu
(ID#2005-2-80-018)

Md Tanzil Khan 29/10/2009

Md. Tanzil Khan
(ID#2005-2-80-041)

We further authorize East West University to reproduce this thesis by photocopy or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Dipu 29/10/2009

Md. Amran Hasan Dipu
(ID#2005-2-80-018)

Md Tanzil Khan 29/10/2009

Md. Tanzil Khan
(ID#2005-2-80-041)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

Mobile ad hoc networks (MANET) are networks in which routing is based on multi-hop routing from a source to a destination node or nodes. These nodes generate traffic, to be forwarded to some other nodes or a group of nodes. Due to the dynamic nature of ad hoc networks, traditional fixed network routing protocols are not suitable. These networks have quite a many constrains because of uncertainty of radio interface and its limitations. Also some terminals have limitations concerning battery energy in use. There are numerous applicable protocols for ad hoc networks. Each of these protocols is designed to perform its task as well as it is applicable according to its design criteria. The protocol to be chosen must cover all states of a specified network and never is allowed to consume too much network resources by protocol overhead traffic.

MANETs are a new paradigm of networks, offering unrestricted mobility without any underlying infrastructure. An ad-hoc network is a collection of autonomous nodes or terminals that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Each node in a mobile ad-hoc network functions both as a host and a router and the control of the network is distributed among the nodes. Hence, each node has to rely on other nodes in setting up a successful session. Trust plays an important part in such scenarios. Hence an appropriate trust management scheme is needed to cope with the nature of MANET environment. Trust is a before-security issue in ad hoc networks. By clarifying the trust relationship, it will be much easier to take proper security measures and make correct decision on any security issues. Yan *et al.* in their paper [1] suggested that data protection approach, secure route selection or any other decision related to security should be based on trust analysis and evaluation among network nodes.

Trust can be considered as the link between observations (trust evidence). In our trust management scheme we utilize the numerical value of trust between the nodes as the computational means to evaluate trust relationship between entities. It is a common tendency to adopt policy of trusting entities that are trusted by entity that one trusts. Hence, trust would thus propagate through the network and become accorded when one entity can reach another entity via at least one trusted path. In our approach we utilize two modes of trust relationship. First way is through direct observation of other nodes' behavior and the second way is through recommendations from other nodes.

Papadimitratos and Haas [2] suggested that cryptographic mechanisms can be employed for establishing the trust association in mobile ad hoc networks. But due to the unpredictable and highly dynamic nature of MANET, it is difficult to establish some type of secret among the nodes. Hence trust management between nodes can be done through a central trusted authority or a group of nodes acting as certification authorities (CA) to issue membership certificates [3, 4] or require the nodes to issue certificates to each other [5] in order to create a web-of-trust [6]. In the former case, node(s) acting as certification authority could be a malicious node. Also due to the highly dynamic nature of MANET environment, nodes may leave the network and may cause shortage of CAs that can issue certificate shares. While the latter assumes that trust is transitive and could potentially cause serious security breaches. In our approach, the initiator node is responsible for establishing the trust management scheme with other nodes in the network. There has been work done on trust computation based on interactions with one-hop physical neighbors [7]. In the approach proposed here, we extend our trust evaluation to multi hop distant nodes.

In chapter 2 we described MANET and its characteristics. Section 3 depicts different types routing protocols proposed by researchers for MANET environment and comparison of their characteristics. In chapter 4 we elaborate trust management scheme for MANET environment that is based on interactions within piconet members i.e one-hop physical neighbors. We extend our trust evaluation to multi hop distant nodes. In chapter 5 we present the simulation and result. And finally we conclude and recommend future work in chapter 6.

CHAPTER 2

MANET

2.1. Introduction

As the importance of computers in our daily life increases, it also sets new demands for connectivity. Wired solutions have been around for a long time but there is an increasing demand on working through wireless solutions for connecting to the Internet, reading and sending E-mail messages, exchanging information in a meeting and so on. There are solutions to these needs, one being wireless local area network that is based on IEEE 802.11 standard. However, there is increasing need for connectivity in situations where there is no base station (i.e. backbone connection) available (for example two or more PDAs need to be connected). This is where ad hoc networks step in.

Mobile Ad Hoc Networks (MANETs) are communication networks in which all nodes are mobile and communicate with each other via wireless connections. There is no fixed infrastructure. All nodes are equal and there is no central control or overview. There are no designated routers: all nodes can serve as routers for each other, and data packets are forwarded from node to node in a multi-hop fashion. A MANET can be useful in all those situations in which for necessity or other practical or economical reasons no fixed network infrastructure is available, such as military activities in enemy territory, disaster recovery operations or big conference rooms. On the other hand, it is quite clear that in the more technologically advanced societies in the near future networking will be pervasive and highly heterogeneous: mobile ad hoc networks, body area networks, GSM/GPRS networks, satellite networks, the wired Internet will all be somehow interconnected, creating a sort of gigantic hybrid mobile ad hoc network.

2.2. What is MANET

In Latin, ad hoc means "for this," further meaning "for this purpose only.¨ It is a good and emblematic description of the idea why ad hoc networks are needed. Ad hoc network can be set up anywhere without any need for external infrastructure (like wires or base stations).

They are often mobile and that's why a term MANET is often used when talking about Mobile Ad hoc NETworks. MANETs are often defined as follows: MANET is an autonomous system of mobile routers (and associated hosts) connected by wireless links the union of which forms an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. The strength of the connection can change rapidly in time or even disappear completely. Nodes can appear, disappear and re-appear as the time goes on and all the time the network connections should work between the nodes that are part of it. As one can easily imagine, the situation in ad hoc networks with respect to ensuring connectivity and robustness is much more demanding than in the wired case. A simple structure of MANET is shown in Figure 2.1.

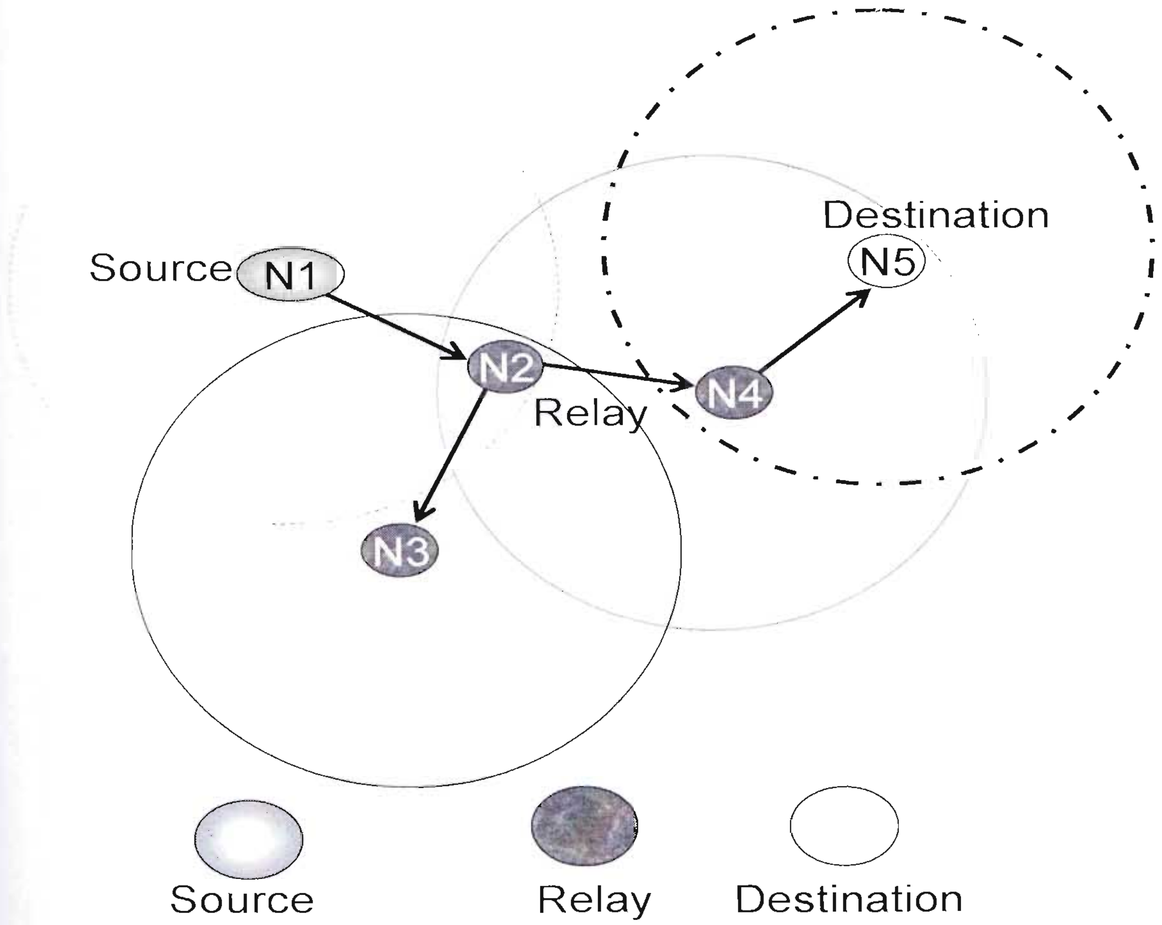


Figure2.1: A Simple Mobile Ad-hoc Network

2.3. Characteristics of MANET

A MANET consists of mobile terminals -- herein simply referred to as "nodes"-- which are free to move around arbitrarily. These nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or on very small devices. MANET may operate in isolation, or may have gateways to and interface with a fixed network. In the latter operational mode, it is typically envisioned to operate as a "stub" network connecting to a fixed internetwork. Stub networks carry traffic originating at and/or destined for internal nodes, but do not permit exogenous traffic to "transit" through the stub network.

MANET nodes are equipped with wireless transmitters and receivers using antennas which may be omni directional (broadcast), highly-directional (point-to-point), possibly steerable, or some combination thereof. At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multihop graph or "ad hoc" network exists between the nodes. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters.

MANETs have several salient characteristics:

i) Dynamic topology: Nodes are free to move arbitrarily; thus, the network topology-- which is typically multihop -- may change randomly and rapidly at unpredictable times, and may consist of both bidirectional and unidirectional links.

ii) Bandwidth-constrained, variable capacity links: Wireless links will continue to have significantly lower capacity than their wired counterparts. In addition, the realized throughput of wireless communications--after accounting for the effects of multiple access, fading, noise, and interference conditions etc.--is often much less than a radio's maximum transmission rate. One effect of the relatively low to moderate link capacities is that congestion is typically the norm rather than the exception, i.e. aggregate application demand will likely approach or exceed network capacity frequently. As the mobile network is often simply an extension of the fixed network infrastructure, mobile ad hoc users will demand similar services. These demands will continue to increase as multimedia computing and collaborative networking applications rise.

iii) Energy-constrained operation: Some or all of the nodes in a MANET may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation.

iv) Limited physical security: Mobile wireless networks are generally more prone to physical security threats than the fixed-cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered. Existing link security techniques are often applied within wireless networks to reduce security threats. As a benefit, the decentralized nature of network control in MANETs provides additional robustness against the single points of failure of more centralized approaches.

These characteristics create a set of underlying assumptions and performance concerns for protocol design which extend beyond those guiding the design of routing protocol within the higher-speed, semi-static topology of the fixed Internet.

2.4. Application Areas of MANET

There is current and future need for dynamic ad hoc networking technology. The emerging field of mobile and nomadic computing, with its current emphasis on mobile IP operation, should gradually broaden and require highly-adaptive mobile networking technology to effectively manage multihop, ad hoc network clusters which can operate autonomously or, more than likely, be attached at some point(s) to the fixed Internet.

Some applications of MANET technology could include industrial and commercial applications involving cooperative mobile data exchange. In addition, mesh-based mobile networks can be operated as robust, inexpensive alternatives or enhancements to cell-based mobile network infrastructures. There are also existing and future military networking requirements for robust, IP-compliant data services within mobile wireless communication networks [8] -- many of these networks consist of highly-dynamic autonomous topology segments. Also, the developing technologies of "wearable" computing and communications may provide applications for MANET technology. When properly combined with satellite-based information delivery, MANET technology can provide an extremely flexible method for establishing communications for fire/safety/rescue operations or other scenarios requiring rapidly-deployable communications with survivable, efficient dynamic networking. There are likely other applications for MANET technology which are not presently realized or envisioned by the authors. It is simply put, improved IP-based networking technology for dynamic, autonomous wireless networks.

2.5. Advantages/ Disadvantages

2.5.1. Advantages

The following are the advantages of MANETs:

- They provide access to information and services regardless of geographic position.
- These networks can be set up at any place and time.
- These networks work without any pre-existing infrastructure.

2.5.2. Disadvantages

Some of the disadvantages of MANETs are:

- Limited resources. Limited physical security.
- Intrinsic mutual trust vulnerable to attacks. Lack of authorization facilities.
- Volatile network topology makes it hard to detect malicious nodes.
- Security protocols for wired networks cannot work for ad hoc networks.

2.6. MANET Architecture (Piconets/ Scatter-nets)

2.6.1. Piconets

A small area of network is called Piconets. Up to eight mobile devices may be networked together in a master-slave relationship, called a piconet. Figure-2.2 represents a simple piconet.

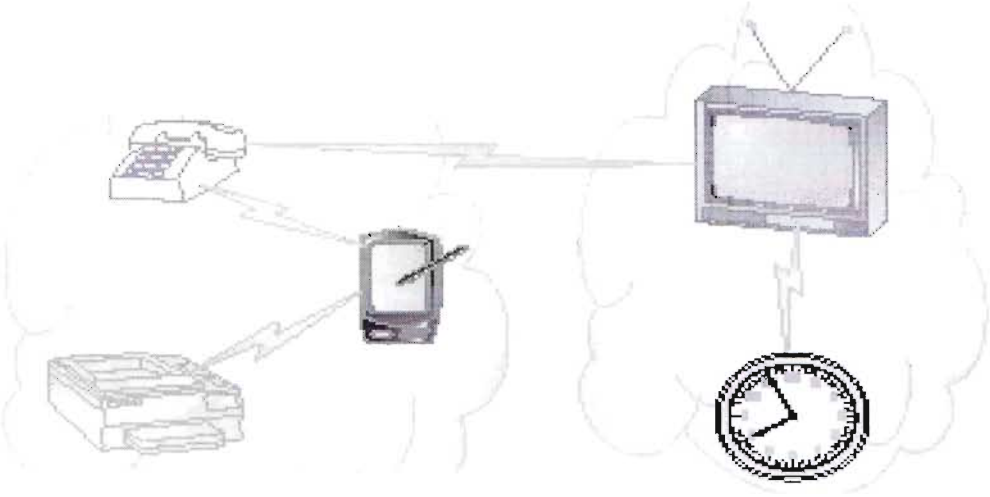


Figure2.2: A Small Area of Network (Piconet)

2.6.2. Scatter-net

The series of piconets often referred to as a scatter-net, allows several devices to be internet worked over an extended distance. Following Figure 2.3 presents a scatter-net.

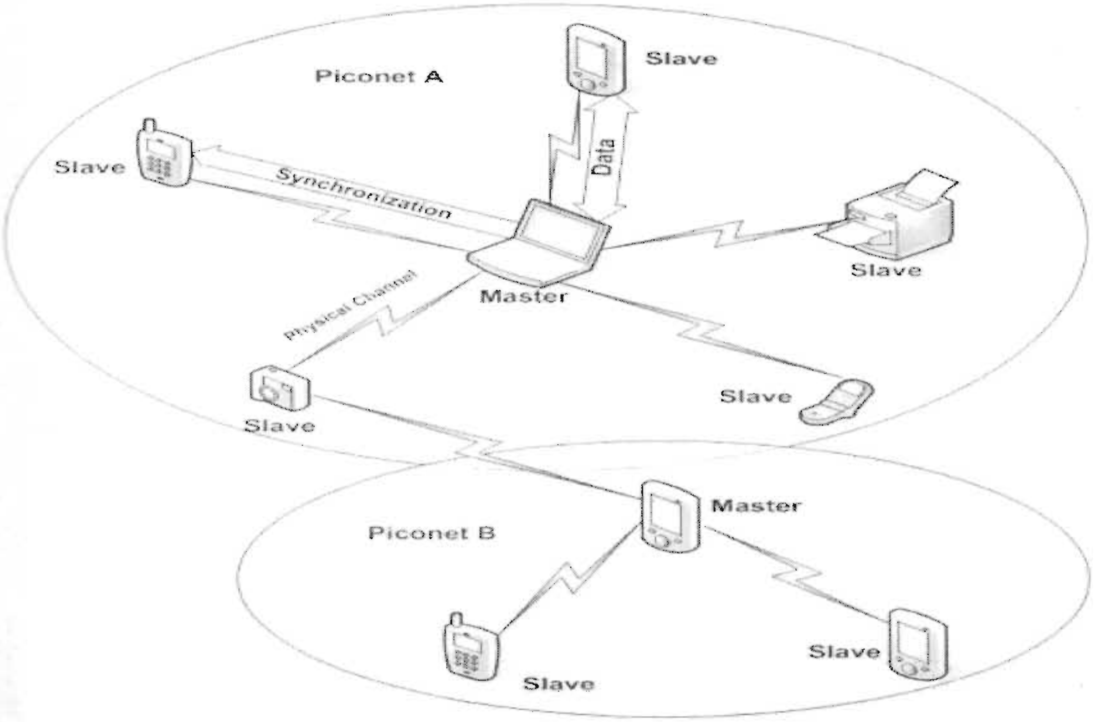


Figure 2.3: A Scatter-net

CHAPTER 3

ROUTING PROTOCOLS

3.1. Introduction

Mobile networks can be classified into infrastructure networks and mobile ad hoc networks [9] according to their dependence on fixed infrastructures. In an infrastructure mobile network, mobile nodes have wired access points (or base stations) within their transmission range. The access points compose the backbone for an infrastructure network. In contrast, mobile ad hoc networks are autonomously self-organized networks without infrastructure support. In a mobile ad hoc network, nodes move arbitrarily, therefore the network may experiences rapid and unpredictable topology changes. Additionally, because nodes in a mobile ad hoc network normally have limited transmission ranges, some nodes cannot communicate directly with each other. Hence, routing paths in mobile ad hoc networks potentially contain multiple hops, and every node in mobile ad hoc networks has the responsibility to act as a router.

Mobile ad hoc networks (MANET) are networks, in which routing is based on multi-hop routing from a source to a destination node or nodes. Ad hoc network is a multi-hop wireless network, which consists of a number of mobile nodes. These nodes generate traffic to be forwarded to some other nodes or a group of nodes. Due to a dynamic nature of ad hoc networks, traditional fixed network routing protocols are not feasible. These networks have quite a many constrains because of uncertainty of radio interface and its limitations e.g. available bandwidth. Also some terminals have limitations concerning battery energy in use.

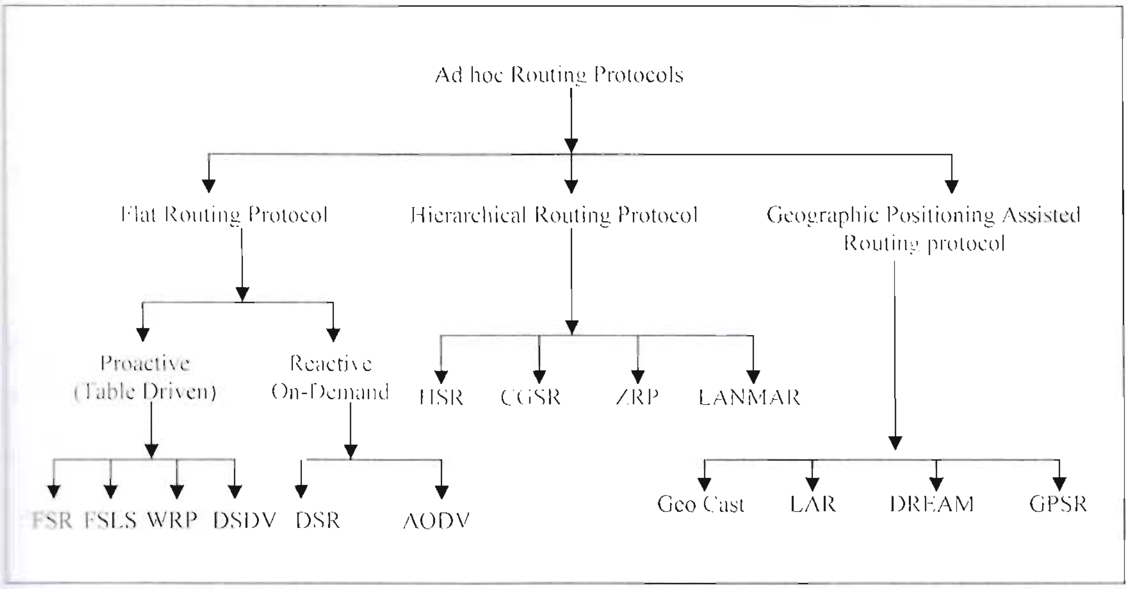
There are numerous applicable protocols for ad hoc networks, but one confusing problem is the vast number of separate protocols. Each of these protocols is designed to perform its task as well as it is acceptable according to its design criteria. The protocol to be chosen must cover all states of a specified network and never is allowed to consume too much network resources by protocol overhead traffic.

3.2. Types of Routing Protocols in MANET

Following categories of routing protocols for MANET are developed by the researchers:

- Flat Routing Protocols
- Hierarchical Routing Protocols
- Hybrid Routing Protocols
- Global Positioning System (GPS) Assisted Routing Protocols

Table 3.1: Classification of Ad-hoc Routing Protocols



3.2.1. Flat Routing Protocols

Table-driven (Proactive) and on-demand (Reactive) routing are called Flat-routing protocols. Flat routing protocols adopt a flat addressing scheme. Each node in routing has an equal role, functionality and capability. As network size increases, flat routing schemes become infeasible due to link and processing overhead.

3.2.1.1. Example of Table-driven Routing (Proactive) Protocol

- DSDV (Destination-Sequenced Distance-Vector)
- WRP (Wireless Routing Protocol)
- FSR (Fisheye State Routing) Protocol

3.2.1.2. Example of On-Demand Routing (Proactive) Protocol

- DSR (Dynamic Source Routing)
- AODV (Ad-Hoc On-Demand Vector Routing)
- DSR and AODV are Included in the Internet-Draft
- TORA (Temporally Ordered Routing Algorithm)
- ABR (Associativity-Based Routing)

3.2.2. Hierarchical Routing Protocol

Most Transmission Control Protocol/Internet Protocol (TCP/IP) routing is based on a two-level hierarchical routing in which an IP address is divided into a network portion and a host portion. Gateways use only the network portion until an IP datagram reaches a gateway that can deliver it directly. Additional levels of hierarchical routing are introduced by the addition of sub-networks.

3.2.3. Hybrid Routing Protocol

This type of protocols combines the advantages of proactive and reactive routing. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. The choice for one or the other method requires predetermination for typical cases. Short distance destination nodes use proactive routing to maintain routing information. Long distance destination nodes do not maintain routing information due to large overhead.

Example of Hybrid Routing Protocol:

- ZRP (Zone Routing Protocol)
- LANMAR (Landmark Ad Hoc Routing Protocol)
- OORP (Order One Routing Protocol)

3.2.4. Global Positioning System (GPS) Assisted Routing Protocol

The advances in the development of GPS nowadays make it possible to provide location information with a precision within a few meters. It also provides universal timing. While location information can be used for directional routing in distributed ad hoc systems, the universal clock can provide global synchronizing among GPS equipped nodes. Additional care must be taken into account in a mobile environment, because locations may not be accurate by the time the information is used. All the protocols surveyed below assume that the nodes know their positions.

Example of (GPS) Routing Protocol:

- LAR (The Location-Aided Routing Protocol)
- DREAM (Distance Routing Effect Algorithm for Mobility)
- GeoCast (Geographic Addressing and Routing)

3.3. Characteristics of Routing Protocols

To compare and analyze mobile ad hoc network routing protocols, appropriate classification methods are important. Classification methods help researchers and designers to understand distinct characteristics of a routing protocol and find its relationship with others. Therefore, we present protocol characteristics which are used to group and compare different approaches in section 3.4. These characteristics mainly are related to the information which is exploited for routing, when this information is acquired, and the roles which nodes may take in the routing process.

3.3.1. Proactive, Reactive and Hybrid Routing Protocol

One of the most popular method to distinguish mobile ad hoc network routing protocols is based on how routing information is acquired and maintained by mobile nodes. Using this method, mobile ad hoc network routing protocols can be divided into proactive routing, reactive routing and hybrid routing.

A proactive routing protocol is also called "table driven" routing protocol. Using a proactive routing protocol, nodes in a mobile ad hoc network continuously evaluate routes to all reachable nodes and attempt to maintain consistent, up-to-date routing information. Therefore, a source node can get a routing path immediately if it needs one.

In proactive routing protocols, all nodes need to maintain a consistent view of the network topology. When a network topology change occurs, respective updates must be propagated throughout the network to notify the change. Most proactive routing protocols proposed for mobile ad hoc networks have inherited properties from algorithms used in wired networks. Using proactive routing algorithms, mobile nodes proactively update network state and maintain a route regardless of whether data traffic exists or not, the overhead to maintain up-to-date network topology information is high. In Section 4, we will give introductions of several typical proactive mobile ad hoc network routing protocols, such as the Wireless Routing Protocol (WRP) [10], the Destination Sequence Distance Vector (DSDV) [11] and the Fisheye State Routing (FSR).

Reactive routing protocols for mobile ad hoc networks are also called "on-demand" routing protocols. In a reactive routing protocol, routing paths are searched only when needed. A route discovery operation invokes a route-determination procedure. The discovery procedure terminates either when a route has been found or no route available after examination for all route permutations.

In a mobile ad hoc network, active routes may be disconnected due to node mobility. Therefore, route maintenance is an important operation of reactive routing protocols. Compared to the proactive routing protocols for mobile ad hoc networks, less control overhead is a distinct advantage of the reactive routing protocols. Thus, reactive routing protocols have better scalability than proactive routing protocols in mobile ad hoc networks. However, when using reactive routing protocols, source nodes may suffer from long delays for route searching before they can forward data packets. The Dynamic Source Routing (DSR) and Ad hoc On- demand Distance Vector routing (AODV) [12] are examples for reactive routing protocols for mobile ad hoc networks.

Hybrid routing protocols are proposed to combine the merits of both proactive and reactive routing protocols and overcome their shortcomings. Normally, hybrid routing protocols for mobile ad hoc networks exploit hierarchical network architectures. Proper proactive routing approach and reactive routing approach are exploited in different hierarchical levels, respectively. In this report, as examples of hybrid routing protocols for mobile ad hoc networks, the Zone Routing Protocol (ZRP), Zone-based Hierarchical Link State routing (ZHLS) and Hybrid Ad hoc Routing Protocol (HARP) [13] will be introduced and analyzed.

3.3.2. Exploiting Network Metrics for Routing Protocol

Metrics used for routing path construction can be used as criteria for mobile ad hoc network routing protocol classification. Most routing protocols for mobile ad hoc networks use "hop number" as a metric. If there are multiple routing paths available, the path with the minimum hop number will be selected. If all wireless links in the network have the same failure probability, short routing paths are more stable than the long ones and can obviously decrease traffic overhead and reduce packet collisions. However, the assumption of the same failure properties may not be true in mobile ad hoc networks. Therefore, the stability of a link has to be considered in the route construction phase. For example, routing approaches such as Associativity Based Routing (ABR) [14] and Signal Stability based Routing (SSR) are proposed that use link stability or signal strength as metric for routing.

With the popularity of mobile computing, some mobile applications may have different QoS requirements. To meet specific QoS requirements, appropriate QoS metrics should be used for packet routing and forwarding in mobile ad hoc networks. As in wired networks, QoS routing protocols for mobile ad hoc networks can use metrics, such as bandwidth, delay, delay jitter, packet loss rate and cost. As an example, bandwidth and link stability are used in CEDAR as metrics for routing path construction.

3.3.3. Multicast Routing Protocols

Most classification methods used for unicast routing protocols for mobile ad hoc networks are also applicable for existing multicast routing protocols. For example, multicast routing algorithms for mobile ad hoc networks can be classified into reactive routing and proactive routing. The Ad-hoc Multicast Routing (AMRoute) and Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) [15] belong to category of proactive multicast routing and the On-Demand Multicast Routing Protocol (DMRP) and Multicast Ad hoc On-demand Distance Vector (MAODV) [16] are reactive multicast routing protocols.

There is a classification method particularly used for multicast routing protocols for mobile ad hoc networks. This method is based on how distribution paths among group members are constructed. According to this method, existing multicast routing approaches for mobile ad hoc networks can be divided into tree based multicast routing, mesh based multicast routing, core based multicast routing and group forwarding based multicast.

Tree based multicast routing protocols can be further divided into source-rooted and core-rooted schemes according to the roots of the multicast trees. In source-rooted tree based multicast routing protocols, source nodes are roots of multicast trees and execute algorithm for distribution tree contraction and maintenance. This requires that a source must know the topology information and addresses of all its receivers in the multicast group. Therefore, source-rooted tree based multicast routing protocols suffer from control traffic overhead when used for dynamic networks. The AMRoute [17] is an example for source-rooted tree multicast routing.

In a core-based multicast routing protocol, cores are nodes with special functions such as multicast data distribution and membership management. Some core-based multicast routing protocols utilize tree structures also, but unlike source-rooted tree based multicast routing, multicast trees are rooted at core nodes. For different core-based multicast routing protocols, core nodes may perform various routing and management functions. For example, in CTB [18] and AMRIS, cores are cross points for all traffic flows of multicast groups and may becomes bottlenecks of the network. On the other hand, in protocols like CAMP, core nodes are not necessarily part of all routing paths.

In a mesh-based multicast routing protocol, packets are distributed along mesh structures that are a set of interconnected nodes. The mesh structure is more robust than the tree structure when used for multicast routing in dynamic networks because a mesh provides alternate paths when link failure occurs. However, the cost for maintaining mesh structures are normally higher than trees. The ODMRP [19] and Core-Assisted Mesh Protocol (CAMP) are mesh-based multicast routing protocols proposed for mobile ad hoc networks.

In the group forwarding based multicast routing, a set of mobile nodes is dynamically selected as forwarding nodes for a multicast group. Forwarding nodes take the responsibility for multicast packet distribution. Using this scheme, it is possible to get multiple routing paths, and duplicate messages will reach a receiver through different paths. ODMRP is a group forwarding based multicast routing protocol using adaptive forwarding groups.

3.4. Analysis and Comparison of Typical Routing Protocols

Because so many routing protocols have been proposed for mobile ad hoc networks, it is impossible to cover all of them in this review. Therefore, this report presents typical protocols selected from the class of similar approaches that can reflect the state-of-the-art of research work on mobile ad hoc network routing. Table3.2 and Table3.3 list the protocols reviewed in this report.

Table 3.2: Unicast Routing Protocols

Uniform Routing Protocol	Proactive Routing Protocol	Wireless Routing Protocol (WRP)	
		Destination Sequence Distance Vector (DSDV) Routing Protocol	
		Fisheye State Routing (FSR)	
		Distance Routing Effect Algorithm for Mobility (DREAM)	Location based Routing
	Reactive Routing Protocol	Dynamic Source Routing (DSR) Protocol	
		Temporally Ordered Routing Algorithm (TORA)	
		Ad-hoc On-demand Distance Vector Routing (AODV) Protocol	
		Location Aided Routing (LAR)	Location based Routing
		Associativity Based Routing (ABR) Protocol	Link-stability based Routing Protocol
		Signal Stability-base adaptive Routing Protocol (SSR)	Link-stability based Routing Protocol
Non-Uniform Routing Protocol	Zone-base Routing Protocol	Zone Routing Protocol (ZRP)	Hybrid Routing Protocol
		Hybrid Ad-hoc Routing Protocol (HARP)	Hybrid Routing Protocol also
		Zone-based Hierarchical Link State Routing (ZHLS)	Hybrid Routing Protocol also
		Grid Location Service (GLS)	Location Service
	Cluster-based Routing Protocol	Clusterhead Gateway Switch Routing (CGSR)	
		Hierarchical State Routing (HSR)	
		Cluster Based Routing Protocol (CBRP)	
	Core-node Base Routing Protocol	Landmark Ad-hoc Routing (LANMAR) Protocol	Proactive Routing Protocol
		Core-Extraction Distributed Ad hoc Routing (CEDAR)	Reactive Routing Protocol
		Optimized Link State Routing Protocol	Proactive Routing Protocol

Table 3.3: Typical Multicast Routing Protocols

	Tree Based	Mesh Based	Core Based	Group Forwarding Based
Ad-hoc Multicast Routing (AMRoute) Protocol	Y		Y	
Ad-hoc Multicast Routing (AMRoute) Protocol utilizing Increasing id-numbers (AMRIS)	Y			
On-Demand Multicast Routing Protocol (ODMRP)		Y		Y
Core-Assisted Mesh Protocol (camp)		Y	Y	
Multicast Ad-hoc On-Demand Distance Vector (MAODV)	Y			

3.4.1. Typical Proactive Routing Protocols

3.4.1.1 The Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) [20] is a proactive unicast routing protocol for mobile ad hoc networks. WRP uses improved Bellman-Ford Distance Vector routing algorithm. To adapt to the dynamic features of mobile ad hoc networks, some mechanisms are introduced to ensure the reliable exchange of update messages and reduces route loops.

Using WRP, each mobile node maintains a distance table, a routing table, a link-cost table and a Message Retransmission List (MRL). An entry in the routing table contains the distance to a destination node, the predecessor and the successor along the paths to the destination, and a tag to identify its state, i.e., is it a simple path, a loop or invalid. Storing predecessor and successor in the routing table helps to detect routing loops and avoid counting-to-infinity problem, which is the main shortcoming of the original distance vector routing algorithm. A mobile node creates an entry for each neighbor in its link-cost table. The entry contains cost of the link connecting to the neighbor, and the number of timeouts since an error-free message was received from that neighbor.

In WRP, mobile nodes exchange routing tables with their neighbors using update messages. The update messages can be sent either periodically or whenever link state changes happen.

The MRL contains information about which neighbor has not acknowledged an update message. If needed, the update message will be retransmitted to the neighbor. Additionally, if there is no change in its routing table since last update, a node is required to send a Hello message to ensure connectivity. On receiving an update message, the node modifies its distance table and looks for better routing paths according to the updated information.

In WRP, a node checks the consistency of its neighbors after detecting any link change. A consistency check helps to eliminate loops and speed up convergence. One shortcoming of WRP is that it needs large memory storage and computing resource to maintain several tables. Moreover, as a proactive routing protocol, it has a limited scalability and is not suitable for large mobile ad hoc networks.

3.4.1.2. The Destination Sequence Distance Vector (DSDV) Routing Protocol

The Destination Sequence Distance Vector (DSDV) [21] is a proactive unicast mobile ad hoc network routing protocol. Like WRP, DSDV is also based on the traditional Bellman-Ford algorithm. However, their mechanisms to improve routing performance in mobile ad hoc networks are quite different.

In routing tables of DSDV, an entry stores the next hop towards a destination, the cost metric for the routing path to the destination and a destination sequence number that is created by the destination. Sequence numbers are used in DSDV to distinguish stale routes from fresh ones and avoid formation of route loops.

The route updates of DSDV can be either time-driven or event-driven. Every node periodically transmits updates including its routing information to its immediate neighbors. While a significant change occurs from the last update, a node can transmit its changed routing table in an event-triggered style. Moreover, the DSDV has two ways when sending routing table updates. One is "full dump" update type and the full routing table is included inside the update. A "full dump" update could span many packets. An incremental update contains only those entries that with metric have been changed since the last update is sent. Additionally, the incremental update fits in one packet.

3.4.1.3. The Fisheye State Routing (FSR) Protocol

The Fisheye State Routing (FSR) [22] is a proactive unicast routing protocol based on Link State routing algorithm with effectively reduced overhead to maintain network topology information. As indicated in its name, FSR utilizes a function similar to a fish eye. The eyes of fishes catch the pixels near the focal with high detail, and the detail decreases as the distance from the focal point increases. Similar to fish eyes, FSR maintains the accurate distance and path quality information about the immediate neighboring nodes, and progressively reduces detail as the distance increases.

In Link State routing algorithm used for wired networks, link state updates are generated and flooded through the network whenever a node detects a topology change. In FSR, however, nodes exchange link state information only with the neighboring nodes to maintain up-to-date topology information. Link state updates are exchanged periodically in FSR, and each node keeps a full topology map of the network. To reduce the size of link state update messages, the key improvement in FSR is to use different update periods for different entries in the routing table. Link state updates corresponding to the nodes within a smaller scope are propagated with higher frequency.

FSR exhibits a better scalability concerning the network size compared to other link state protocols because it doesn't strive for keeping all nodes in the network on the same knowledge level about link states. Instead, the accuracy of topology information is reverse proportional to the distance. This reduces the traffic overhead caused by exchanging link state information because this information is exchanged more frequently with node nearby than with nodes far away.

Adapting the frequency of exchanging link state information according to the FSR

3.4.1.4 Comparison of WRP, DSDV and FSR Routing Protocols

Control traffic overhead and loop-free property are two important issues when applying proactive routing to mobile ad hoc networks. The proactive routing protocols used for wired networks normally have predictable control traffic overhead because topology of wired networks change rarely and most routing updates are periodically propagated. However. periodic routing information updates are not enough for mobile ad hoc routing protocols. The proactive routing in mobile ad hoc networks needs mechanisms that dynamically collect network topology changes and send routing updates in an event-triggered style.

Although belonging to the same routing category for mobile ad hoc networks. WRP, DSDV and FSR have distinct features. Both WRP and DSDV exploited event-triggered updates to maintain up-to-date and consistent routing information for mobile nodes. In contrast to using event-triggered updates, the updates in FSR are exchanged between neighboring nodes and the update frequency is dependent on the distance between nodes. In this way, update overhead is reduced and the far-reaching effect of Link State routing is restricted.

Different mechanisms are used in WRP. DSDV and FSR for loop-free guarantee. WPR records the predecessor and the successor along a path in its routing table and introduces consistence-checking mechanism. In this way, WRP avoids forming temporary route loops but incurs additional overhead. Every node needs to maintain more information and execute more operations. In DSDV, a destination sequence number is introduced to avoid route loops. FSR is a modification of traditional Link State routing and its loop-free property is inherited from Link State routing algorithm.

WRP, DSDV and FSR have the same time and communication complexity. Whereas WRP has a large storage complexity compared to DSDV because more information is required in WRP to guarantee reliable transmission and loop-free paths. Both periodic and triggered updates are utilized in WRP and DSDV; therefore, their performance is tightly related with the network size and node mobility pattern. As a Link State routing protocol. FSR has high storage complexity, but it has potentiality to support multiple-path routing and QoS routing.

3.4.2. Reactive Routing Protocols

3.4.2.1. The Ad Hoc On-demand Distance Vector Routing (AODV) Protocol

The Ad Hoc On-demand Distance Vector Routing (AODV) protocol [12] is a reactive unicast routing protocol for mobile ad hoc networks. As a reactive routing protocol. AODV only needs to maintain the routing information about the active paths. In AODV. routing information is maintained in routing tables at nodes. Every mobile node keeps a next-hop routing table, which contains the destinations to which it currently has a route. A routing table entry expires if it has not been used or reactivated for a pre-specified expiration time. Moreover, AODV adopts the destination sequence number technique used by DSDV in an on-demand way.

In AODV, when a source node wants to send packets to the destination but no route is available, it initiates a route discovery operation. In the route discovery operation, the source broadcasts route request (RREQ) packets. A RREQ includes addresses of the source and the destination. the broadcast ID. which is used as its identifier. the last seen sequence number of the destination as well as the source node's sequence number. Sequence numbers are important to ensure loop-free and up-to-date routes. To reduce the flooding overhead, a node discards RREQs that it has seen before and the expanding ring search algorithm is used in route discovery operation. The RREQ starts with a small TTL (Time-To-Live) value. If the destination is not found, the TTL is increased in following RREQs. In figure 3.1 shown The Route Request Packets flooding in AODV Routing Protocol.

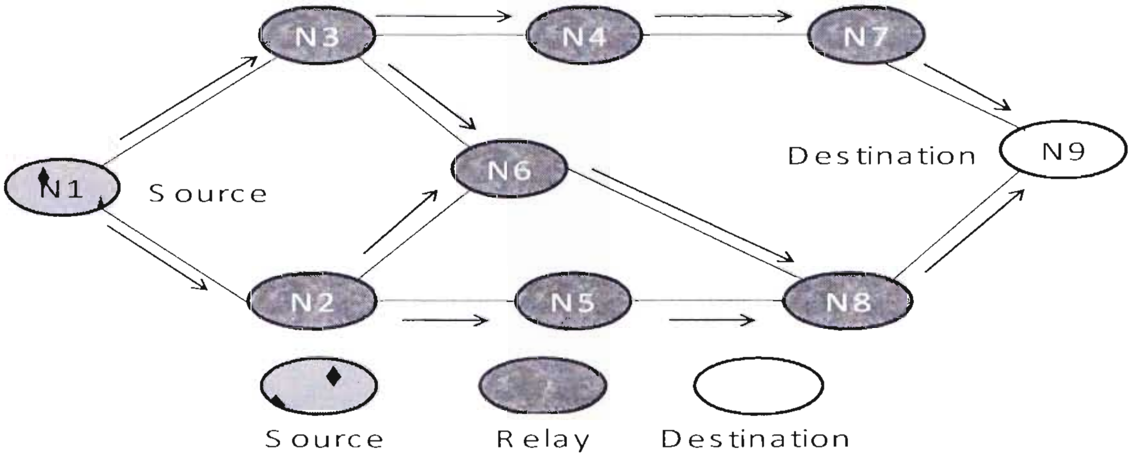


Figure 3.1: The Route Request Packets Flooding in AODV

In AODV. each node maintains a cache to keep track of RREQs it has received. The cache also stores the path back to each RREQ originator. When the destination or a node that has a route to the destination receives the RREQ. it checks the destination sequence numbers it currently knows and the one specified in the RREQ. To guarantee the freshness of the routing information. a route reply (RREP) packet is created and forwarded back to the source only if the destination sequence number is equal to or greater than the one specified in RREQ. AODV uses only symmetric links and a RREP follows the reverse path of the respective RREP. Upon receiving the RREP packet. each intermediate node along the route updates its next-hop table entries with respect to the destination node. The redundant RREP packets or RREP packets with lower destination sequence number will be dropped.

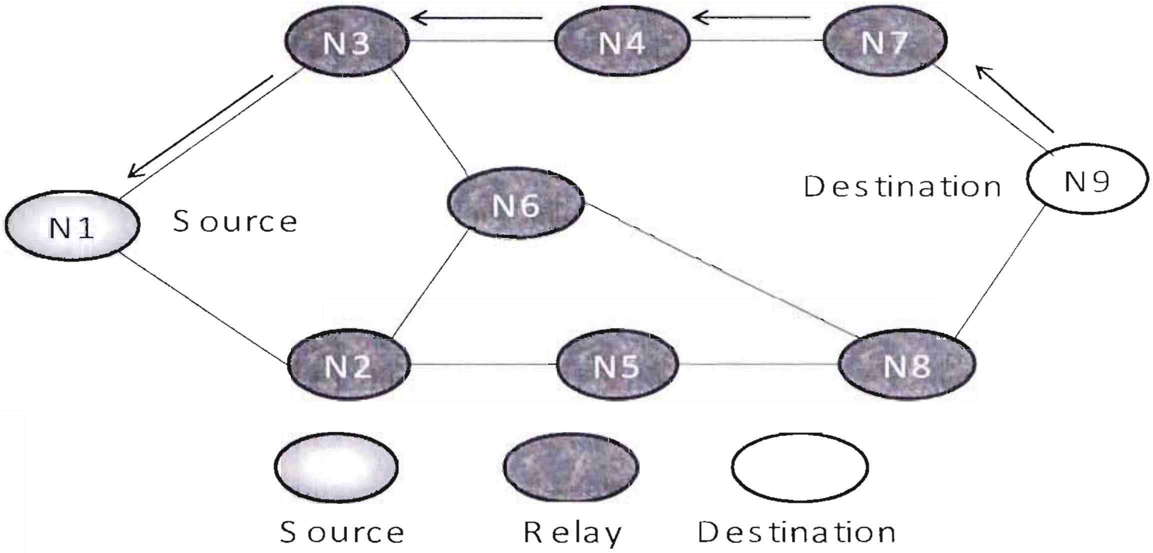


Figure 3.2: The Forwarding of Route Reply Packet in AODV

In AODV. a node uses hello messages to notify its existence to its neighbors. Therefore. the link status to the next hop in an active route can be monitored. When a node discovers a link disconnection. it broadcasts a route error (RERR) packet to its neighbors. which in turn propagates the RERR packet towards nodes whose routes may be affected by the disconnected link. Then. the affected source can re-initiate a route discovery operation if the route is still needed.

3.4.2.2. The Dynamic Source Routing (DSR) Protocol

The Dynamic Source Routing (DSR) [23] is a reactive unicast routing protocol that utilizes source routing algorithm. In source routing algorithm. each data packet contains complete routing information to reach its dissemination. Additionally, in DSR each node uses caching technology to maintain route information that it has learnt.

There are two major phases in DSR, the route discovery phase and the route maintenance phase. When a source node wants to send a packet, it firstly consults its route cache. If the required route is available, the source node includes the routing information inside the data packet before sending it. Otherwise, the source node initiates a route discovery operation by broadcasting route request packets. A route request packet contains addresses of both the source and the destination and a unique number to identify the request. Receiving a route request packet, a node checks its route cache. If the node doesn't have routing information for the requested destination, it appends its own address to the route record field of the route request packet. Then, the request packet is forwarded to its neighbors. To limit the communication overhead of route request packets, a node processes route request packets that both it has not seen before and its address is not presented in the route record field. If the route request packet reaches the destination or an intermediate node has routing information to the destination, a route reply packet is generated. When the route reply packet is generated by the destination, it comprises addresses of nodes that have been traversed by the route request packet. Otherwise, the route reply packet comprises the addresses of nodes the route request packet has traversed concatenated with the route in the intermediate node's route cache.

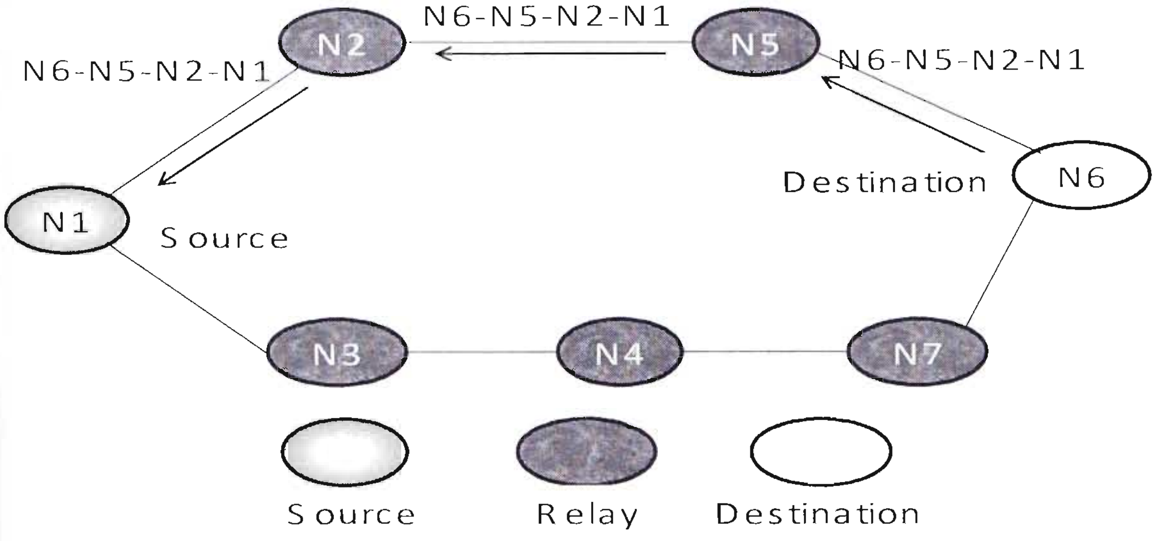


Figure 3.3: Route Reply with Route Record in DSR

After being created, either by the destination or an intermediate node, a route reply packet needs a route back to the source. There are three possibilities to get a backward route. The first one is that the node already has a route to the source. The second possibility is that the network has symmetric (bi-directional) links. The route reply packet is sent using the collected routing information in the route record field, but in a reverse order as shown in Figure3.3. In the last case, there exists asymmetric (unidirectional) links and a new route discovery procedure is initiated to the source. The discovered route is piggybacked in the route request packet.

In DSR, when the data link layer detects a link disconnection, a ROUTE_ERROR packet is sent backward to the source. After receiving the ROUTE ERROR packet, the source node initiates another route discovery operation. Additionally, all routes containing the broken link should be removed from the route caches of the immediate nodes when the ROUTE ERROR packet is transmitted to the source.

DSR has increased traffic overhead by containing complete routing information into each data packet, which degrades its routing performance.

3.4.2.3. The Temporally Ordered Routing Algorithm (TORA)

The Temporally Ordered Routing Algorithm (TORA) [24] is a reactive routing algorithm based on the concept of link reversal. TORA improves the partial link reversal method by detecting partitions and stopping non-productive link reversals. TORA can be used for highly dynamic mobile ad hoc networks.

In TORA, the network topology is regarded as a directed graph. A Directional Acyclical Graph (DAG) is accomplished for the network by assigning each node *i* a height metric *hi*. A link directional from *i* to *j* means *hi* > *hj*. In TORA, the height of a node is defined as a quintuple, which includes the logical time of a link failure, the unique ID of the node that defines the new reference level, a reflection indicator bit, a propagation ordering parameter and an unique ID of the node. The first three elements collectively represent the reference level. The last two values define an offset with respect to the reference level. Like water flowing, a packet goes from upstream to downstream according the height difference between nodes. DAG provides TORA the capability that many nodes can send packets to a given destination and guarantees that all routes are loop-free.

TORA has three basic operations: route creation, route maintenance and route erasure. A route creation operation starts with setting the height (propagation ordering parameter in the quintuple) of the destination to 0 and heights of all other nodes to NULL (i.e., undefined). The source broadcasts a QRY packet containing the destination's ID. A node with a non-NULL height responds by broadcasting a UPD packet containing the height of its own. On receiving a UPD packet, a node sets its height to one more than that of the UPD generator. A node with higher height is considered as upstream and the node with lower height is considered as downstream. In this way, a directed acyclic graph is constructed from the source to the destination and multiple paths route may exist.

The DAG in TORA may be disconnected because of node mobility. So, route maintenance operation is an important part of TORA. TORA has the unique feature that control messages are localized into a small set of nodes near the occurrence of topology changes. After a node loses its last downstream link, it generates a new reference level and broadcasts the reference to its neighbors. Therefore, links are reversed to reflect the topology change and adapt to the new reference level. The erase operation in TORA floods CLR packets through the network and erase invalid routes.

3.4.2.4. Comparison of DSR, AODV and TORA

As reactive routing protocols for mobile ad hoc networks, DSR, AODV and TORA are proposed to reduce the control traffic overhead and improve scalability. In the appendix, their main differences are listed.

DSR exploits source routing and routing information caching. A data packet in DSR carries the routing information needed in its route record field. DSR uses flooding in the route discovery phase. AODV adopts the similar route discovery mechanism used in DSR, but stores the next hop routing information in the routing tables at nodes along active routes.

Therefore. AODV has less traffic overhead and is more scalable because of the size limitation of route record field in DSR data packets.

Both DSR and TORA support unidirectional links and multiple routing paths. but AODV doesn't. In contrast to DSR and TORA. nodes using AODV periodically exchange hello messages with their neighbors to monitor link disconnections. This incurs extra control traffic overhead. In TORA. utilizing the "link reversal" algorithm. DAG constructs routing paths from multiple sources to one destination and supports multiple routes and multicast [25].

In AODV and DSR. a node notifies the source to re-initiate a new route discovery operation when a routing path disconnection is detected. In TORA, a node re-constructs DAG when it lost all downstream links. Both AODV and DSR use flooding to inform nodes that are affected by a link failure. However. TORA localizes the effect in a set of node near the occurrence of the link failure.

AODV uses sequence numbers to avoid formation of route loops. Because DSR is based on source routing, a loop can be avoided by checking addresses in route record field of data packets. In TORA. each node in an active route has a unique height and packets are forwarded from a node with higher height to a lower one. So. a loop-free property can be guaranteed in TORA. However, TORA has an extra requirement that all nodes must have synchronized clocks. In TORA. oscillations may occur when coordinating nodes currently execute the same operation.

Performances of DSDV, TORA. DSR and AODV are compared in [26] based on simulation. The simulation results showed that DSDV performs well when node mobility rates and speed of movements are low. When the number of source nodes is large. the performance of TORA decreases. As shown in [20]. both AODV and DSR perform well for different simulation scenarios. DSR outperforms AODV because it has less routing overhead when nodes have high mobility. A simulation-based comparison of two reactive mobile ad hoc network routing protocols. the AODV and DSR, is reported in [19]. The general result of [27] was that DSR performs better than AODV when number of nodes is small, lower load and /or mobility, and AODV outperforms DSR in more demanding situations.

3.4.3. Zone Based Hierarchical Routing Protocols

3.4.3.1. The Zone Routing Protocol (ZRP)

The Zone Routing Protocol (ZRP) [28] is a hybrid routing protocol for mobile ad hoc networks. The hybrid protocols are proposed to reduce the control overhead of proactive routing approaches and decrease the latency caused by route search operations in reactive routing approaches.

In ZRP, the network is divided into routing zones according to distances between mobile nodes. Given a hop distance *d* and a node N, all nodes within hop distance at most *d* from N belong to the routing zone of N. Peripheral nodes of N are N's neighboring nodes in its routing zone which are exactly *d* hops away from N.

In ZRP, different routing approaches are exploited for inter-zone and intra-zone packets. The proactive routing approach, i.e., the Intra-zone Routing protocol (IARP), is used inside routing zones and the reactive Inter-zone Routing Protocol (IERP) is used between routing zones, respectively. The IARP maintains link state information for nodes within specified distance *d*. Therefore, if the source and destination nodes are in the same routing zone, a route can be available immediately. Most of the existing proactive routing schemes can be used as the IARP for ZRP. The IERP reactively initiates a route discovery when the source node and the destination are residing in different zones. The route discovery in IERP is similar to DSR with the exception that route requests are propagated via peripheral nodes.

3.4.3.2. The Hybrid Ad hoc Routing Protocol (HARP)

The Hybrid Ad hoc Routing Protocol (HARP) [13] is a hybrid routing scheme, which exploits a two-level zone based hierarchical network structure. Different routing approaches are utilized in two levels, for intra-zone routing and inter-zone routing, respectively.

The Distributed Dynamic Routing (DDR) [13] algorithm is exploited by HARP to provide underlying supports. In DDR, nodes periodically exchange topology messages with their neighbors. A forest is constructed from the network topology by DDR in a distributed way. Each tree of the forest forms a zone. Therefore, the network is divided into a set of non-overlapping dynamic zones.

A mobile node keeps routing information for all other nodes in the same zone. The nodes belonging to different zones but are within the direct transmission range are defined as gateway nodes. Gateway nodes have the responsibility forwarding packets to neighboring zones. In addition to routing information for nodes in the local zone, each node also maintains those of neighboring zones.

As in ZRP, the intra-zone routing of HARP relies on an existing proactive scheme and a reactive scheme is used for inter-zone communication. Depending on whether the forwarding and the destination node are inside the same zone, the respective routing scheme will be applied.

3.4.3.3. The Zone-based Hierarchical Link State routing (ZHLS)

The Zone-based Hierarchical Link State routing (ZHLS) is a hybrid routing protocol. In ZHLS, mobile nodes are assumed to know their physical locations with assistance from a locating system like GPS. The network is divided into non-overlapping zones based on geographical information.

ZHLS uses a hierarchical addressing scheme that contains zone ID and node ID. A node determines its zone ID according to its location and the pre-defined zone map is well known to all nodes in the network. It is assumed that a virtual link connects two zones if there exists at least one physical link between the zones. A two-level network topology structure is defined in ZHLS, the node level topology and the zone level topology. Respectively, there are two kinds of link state updates, the node level LSP (Link State Packet) and the zone level LSP. A node level LSP contains the node IDs of its neighbors in the same zone and the zone IDs of all other zones. A node periodically broadcast its node level LSP to all other nodes in the same zone. Therefore, through periodic node level LSP exchanges, all nodes in a zone keep identical node level link state information. In ZHLS, gateway nodes broadcast the zone LSP throughout the network whenever a virtual link is broken or created. Consequently, every node knows the current zone level topology of the network.

Before sending packets, a source firstly checks its intra-zone routing table. If the destination is in the same zone as the source, the routing information is already there. Otherwise, the source sends a location request to all other zones through gateway nodes. After a gateway node of the zone, in which the destination node resides, receives the location request, it replies with a location response containing the zone ID of the destination. The zone ID and the node ID of the destination node will be specified in the header of the data packets originated from the source. During the packet forwarding procedure, intermediate nodes except nodes in the destination zone will use inter-zone routing table, and when the packet arrives the destination zone, an intra-zone routing table will be used.

3.4.3.4. Comparison of ZRP, HARP and ZHLS

As zone based mobile ad hoc network routing protocols, ZRP, HARP and ZHLS use different zone construction methods, which have critical effect on their performance.

In ZRP, the network is divided into overlapping zones according to the topology knowledge for neighboring nodes of each node. In HARP, the network is divided into non-overlapping zones dynamically by DDR through mapping the network topology to a forest. For each node in HARP, the topology knowledge for neighboring nodes is also needed and the zone level stability is used as a QoS parameter to select more stable route. ZHLS assumes that each node has a location system such as GPS and the geographical information is well known, and the network is geographically divided into non-overlapping zones. The performance of a zone based routing protocol is tightly related to the dynamics and size of the network and parameters for zone construction. However, because zones heavily overlap, ZRP in general will incur more overhead than ZHLS and HARP.

All three zone-based routing protocols presented in this subsection use proactive routing for intra-zone communication and reactive routing for inter-zone packet forwarding. Performance of a zone based routing protocol is decided by the performance of respective proactive and reactive routing protocols chosen and how they cooperate each other.

3.4.4. Cluster-Based Routing Protocols

3.4.4.1. The Clusterhead Gateway Switch Routing (CGSR)

The Clusterhead Gateway Switch Routing (CGSR) [29] is a hierarchical routing protocol. The cluster structure improves performance of the routing protocol because it provides effective membership and traffic management. Besides routing information collection, update and distribution, cluster construction and clusterhead selection algorithms are important components of cluster based routing protocols.

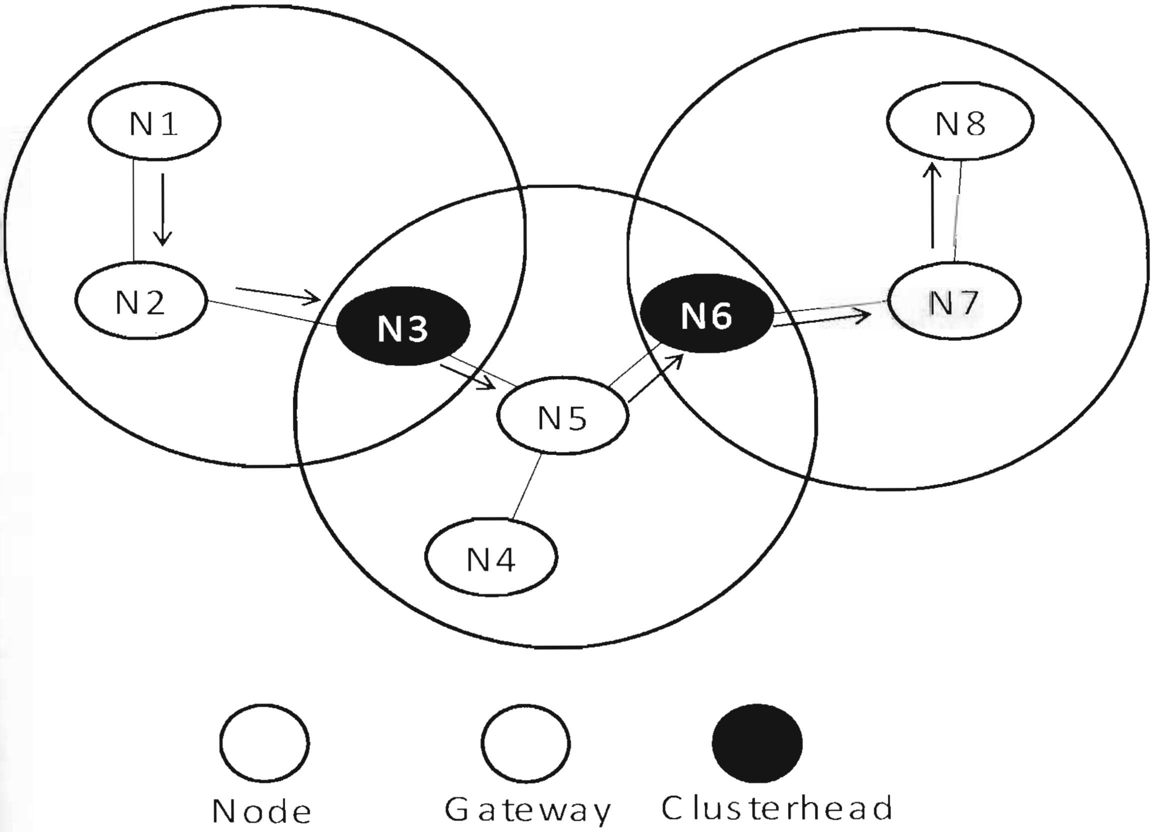


Figure 3.4: Cluster Structure in CGSR

CGSR uses similar proactive routing mechanism as DSDV. Using CGSR, mobile nodes are aggregated into clusters and a cluster-head is elected for each cluster. Gateway nodes are responsible for communication between two or more clusterheads. Nodes maintain a cluster member table that maps each node to its respective cluster-head. A node broadcasts its cluster member table periodically. After receiving broadcasts from other nodes, a node uses the DSDV algorithm to update its cluster member table. In addition, each node maintains a routing table that determines the next hop to reach other cluster.

In a dynamic network, cluster based schemes suffer from performance degradation due to the frequent elections of a clusterhead. To improve the performance of CGSR, a Least Cluster Change (LCC) algorithm is proposed. Only when changes of network topology cause two clusterheads merging into one or a node being out of the coverage of all current clusters, LCC is initiated to change current state of clusters.

CGSR, when forwarding a packet, a node firstly checks both its cluster member table and routing table and tries to find the nearest clusterhead along the routing path. As shown in Figure3.4, when sending a packet, the source (node 1) transmits the packet to its clusterhead (node 2). From the clusterhead node 2, the packet is sent to the gateway node (node 3) that connecting to this clusterhead and the next clusterhead (node 5) along the route to the destination (node 8). The gateway node (node 6) sends the packet to the next clusterhead (node 7), i.e. the destination cluster-head. The destination clusterhead (node 7) then transmits the packet to the destination (node 8).

3.4.4.2. The Hierarchical State Routing (HSR)

The Hierarchical State Routing (HSR) [14] is a multi-level cluster-based hierarchical routing protocol. In HSR, mobile nodes are grouped into clusters and a clusterhead is elected for each cluster. The clusterheads of low level clusters again organize themselves into upper level clusters, and so on. Inside a cluster, nodes broadcast their link state information to all others. The clusterhead summarizes link state information of its cluster and sends the information to its neighboring clusterheads via gateway nodes. Nodes in upper level hierarchical clusters flood the network topology information they have obtained to the nodes in the lower level clusters.

In HSR, a hierarchical address is assigned to every node. The hierarchical address reflects the network topology and provides enough information for packet deliveries in the network. Mobile nodes are also partitioned into logical subnetworks corresponding to different user groups. Each node also has a logical address in the form of <subnet, host>. For each subnetwork, there is a location management server (LMS) which records the logical addresses of all nodes in the subnetwork. LMSs advertise their hierarchical addresses to the top level of hierarchical clusters. The routing information, which contains LMSs' hierarchical addresses, is sent down to all LMSs too. If a source node only knows the logical address of the destination node, before sending a packet, the source node firstly checks its LMS and tries to find the hierarchical address of the destination's LMS. Then the source sends the packet to the destination's LMS, and the destination's LMS forwards the packet to the destination. Once the source knows the hierarchical address of the destination, it sends packets directly to the destination without consulting LMSs.

In HSR, logical addresses reflect the group property of mobile nodes and hierarchical addresses reflect their physical locations. Combining these addressing schemes can improve adaptability of the routing algorithm.

3.4.4.3. Cluster Based Routing Protocol (CBRP)

In the Cluster Based Routing Protocol (CBRP) [30], nodes are divided into clusters and the clustering algorithm is performed when a node joins the network. Before joining, a node is in the "undecided" state. The "undecided" node initiates the joining operation by setting a timer and broadcasts a Hello message. If a clusterhead receives the Hello message, it replies with a triggered Hello message. Receiving the triggered Hello message, the "undecided" node changes its state to "member" state. If the "undecided" node has bi-directional links to some neighbors but does not receive a message from a clusterhead before the local timer generates a timeout, it makes itself a clusterhead. Otherwise, the node remains in "undecided" mode and repeats the joining operation later.

In CBRP, every node maintains a neighbor table in which it stores the information about link states (uni-directional or bi-directional) and the state of its neighbors. In addition to the information of all members in its cluster, a clusterhead keeps information of its neighboring clusters, which includes the clusterheads of neighboring clusters and gateway nodes connecting it to neighboring clusters.

If a source node wants to send a packet but has no active route which can be used, it floods route request to clusterhead of its own and all neighboring clusters. If a clusterhead receives a request it has seen before, it discards the request. Otherwise, the clusterhead checks if the destination of the request is in its cluster. If the destination is in the same cluster, the clusterhead sends the request to the destination, or it floods the request to its neighboring clusterheads. Source routing is used during the route search procedure and only the addresses of clusterheads on the route are recorded. The destination sends a reply including the route information recorded in the request if it successfully receives a route request. If the source doesn't receive a reply in the specified time period, it starts an exponentially backoff algorithm and sends the request later.

The shortening route is proposed in CBRP for performance optimization. Because CBRP uses a source routing scheme, a node gets all information about the route when receiving a packet. To reduce the hop number and adapt to network topology changes, nodes exploit route shortening to choose the most distant neighboring node in a route as next hop.

Another optimization method exploited by CBRP is local repair. Whenever a node has a packet to forward and the next hop is not reachable, it checks the routing information contained in the packet. If the next hop or the hop after next hop in the route is reachable through one of its neighbors, the packet is forwarded through the new route.

3.4.4.4. Comparison of CGSR, HSR and CBRP

Different clustering algorithms have been introduced to group mobile nodes and elect clusterheads in cluster based routing protocols. In HSR, hierarchical addressing is used and the network may have a recursive multi-level cluster structure. Moreover, a location management mechanism is used in HSR to map the logical address to the physical address. CGSR is based on DSDV, a proactive routing protocol for mobile ad hoc networks, and every node keeps routing information for other nodes in both the cluster member table and the routing table. In CBRP, every node keeps information about its neighbors and a clusterhead maintains information about its members and its neighboring clusterheads. CBRP exploits the source routing scheme and the addresses of clusterheads along a route are recorded in the data packets.

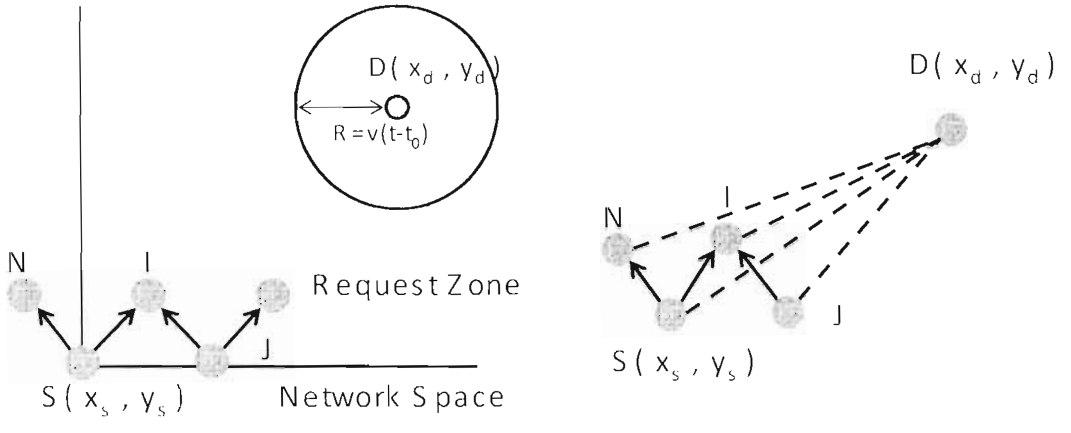
3.4.5. Routing Protocols Using Location Information

3.4.5.1. Location Aided Routing (LAR)

The Location Aided Routing (LAR) [23] is a reactive unicast routing scheme. LAR exploits position information and is proposed to improve the efficiency of the route discovery procedure by limiting the scope of route request flooding.

In LAR, a source node estimates the current location range of the destination based on information of the last reported location and mobility pattern of the destination. In LAR, an expected zone is defined as a region that is expected to hold the current location of the destination node. During route discovery procedure, the route request flooding is limited to a request zone, which contains the expected zone and location of the sender node.

As shown in Figure3.5, there are two different schemes in LAR. In the scheme 1, the source node calculates the expected zone and defines a request zone in request packets, and then initiates a route discovery. Receiving the route request, a node forwards the request if it falls inside the request zone; otherwise it discards the request. When the destination receives the request, it replies with a route reply that contains its current location, time and average speed. The size of a request zone can be adjusted according to the mobility pattern of the destination. When speed of the destination is low, the request zone is small; and when it moves fast, the request zone is large.



Scheme1

Scheme2

Figure3.5: Two different Schemes in LAR

In scheme 2, a source node *S* with coordinate (*xs*, *ys*) calculates the distance *Dist_s* to the destination *D*, whose coordinate is (*xd*, *yd*) before it initiates a route discovery operation. Receiving a route request, a node *I* with coordinate *(xi,yi)* calculates its distance *Dist_i* to the destination *D* and forwards the request only if $Dist\_i \leq Dist\_s + \delta$, otherwise it discards the request. Before forwarding the request, node *I* replaces *Dist_s* with *Dist_i*. The non-zero δ increases the success probability of the route discovery procedure.

3.4.5.2. Distance Routing Effect Algorithm for Mobility (DREAM)

The Distance Routing Effect Algorithm for Mobility (DREAM) [31] exploits location and speed information of mobile nodes for data packet routing. In DREAM, geographical information is used to limit the flooding of data packets to a small region, rather than to merely provide assistance during the route discovery phase in LAR.

DREAM is a proactive routing scheme. In DREAM, the routing table of a node contains location information of all other nodes in the network. When a source wants to send a packet, firstly it checks its routing table and gets the respective location information of the destination. Then, the source forwards the packet to a neighbor in the direction towards the destination. Therefore, the most substantial issue in DREAM is disseminating the location information through the network. To do that, every mobile node sends location updates comprising its location. The frequency of the location update is determined by the distance and node mobility. Considering the distance effect, nodes departing far away normally have a more stable relative location relationship. Consequently, when a node maintains the location information of another one that is far away, less frequent updates are used. Additionally, each location update is tagged with the "life time" which limits the transmission range of the update. Mobile nodes are allowed to adjust transmission frequencies of their location updates according to their mobility patterns.

3.4.5.3. The Grid Location Service (GLS)

The Grid Location Service (GLS) [32] provides a distributed location service for routing in mobile ad hoc networks with large number of nodes.

It is assumed in GLS that every node can get its geographic position information by means of GPS or a similar mechanism. Nodes periodically exchange Hello messages with their neighbors. The Hello messages comprise the sender's position and speed information. Hence, every node maintains a table that contains the identifiers and geographic position information of its neighbors. When using GLS with a forwarding scheme, each node also keeps a routing table; the identifier and geographic position of the destination are contained in packet headers.

There are three main operations in GLS: location server selection, location query request and location server update. In GLS, the geographical area is divided into a hierarchy of grids with squares of increasing size. The originate grid cells are called order-1 squares and the next upper level cells are called order-2 squares, and so on. To avoid overlapping, an order-n square is part of only one order-(n+1) square. So, each node is located in exactly one square of each size. A node maintains its geographic position information into a small set of location servers, which are distributed throughout the network. Unique IDs are assigned to nodes in the network and location server set selection is based on node IDs. A node selects the node with the numerically closest ID as its location severs in a square. Location server nodes are not special nodes and each node in the network can act as a location server for some others.

In a routing protocol using GLS to provide location service, before forwarding packets, a mobile node consults its neighbor table and chooses the node closest to the destination. When the next hop node receives the packet, the same operation will continue until the destination is reached. However, a node may have no routing information about the destination when it receives a packet. Such a situation is called a dead-end in location base routing approaches. When a node encounters a dead-end problem, it sends an error message to the source and GLS comes into play.

When a node needs location information of a destination, it initiates a location query request. At first, it searches location servers of the destination in its order-2 square. The request will be sent to the node whose ID is either equal to the ID of the destination or the smallest of the IDs greater than the destination's. This operation will continue if a location server cannot be found in the current level of square order.

When a node moves, it sends location updates to its location servers in the respective squares. A node doesn't need to know the exact locations of its location servers. In location update forwarding, a similar scheme is used as in the location query procedure. The location update frequency is determined according to the distance that a node has moved since last update.

When a node moves into another grid, its location information stored by its location servers becomes obsolete. Additionally, when a location server moves out of a square, the location information becomes useless. To avoid these failures, a node places a forwarding pointer in the grid it is leaving. This forwarding pointer is broadcasted to nodes in the grid and indicates into which grid the departing node is moving.

3.4.5.4. Comparison of LAR, DREAM and GLS

Location based routing protocols exploit location and node mobility information for the routing process. LAR, DREAM and GLS use the information in different ways and provide different services.

LAR can be integrated into a reactive routing protocol and its main objective is to perform more efficient route discovery and limit the flooding of route request packets. Using LAR, a sender includes its location in the packets. In contrast to LAR, DREAM itself is a proactive routing protocol and every node keeps location information of all participants in the network. In DREAM, the location update frequency is determined by the relative distance between nodes and their mobility characteristics. GLS is not a routing protocol, but only provides a location service. In GLS, every node has several location servers scattered throughout the network which provide location information.

Although the flooding is constrained in both LAR and DREAM by using location information, they are still not suitable for large-scale ad hoc networks. Their poor scalability roots in the directional flooding reactively initiated in LAR and proactive location information flooding in DREAM. In contrast, GLS can be used in large-scale mobile ad hoc networks with high node density. In GLS, a node chooses a small set of location severs throughout the network. Compared to LAR and DREAM, GLS doesn't exploit flooding for location update and query. Hence, its traffic overhead is greatly reduced. Simulation results in [32] showed that GLS has a high query success ratio in large networks with high node density. However, simulation work in [33] also showed that the performance of GLS greatly declines in small size networks with lower node density.

Because LAR is used for route discovery and GLS provides only location service, they should be used with appropriate location based forwarding schemes. However, DREAM itself is a routing protocol and comprises location service and packet forwarding

3.5. Limitations of Above Protocols

All kinds of Routing Protocol are affected by two problems namely:

- Ping-pong effect
- Self-looping effect

3.5.1. Ping-Pong Effect

If any message transition occurs between only two nodes and carry on continuously then this is called ping-pong effect. But in our simulation we can eliminate ping-pong effect.

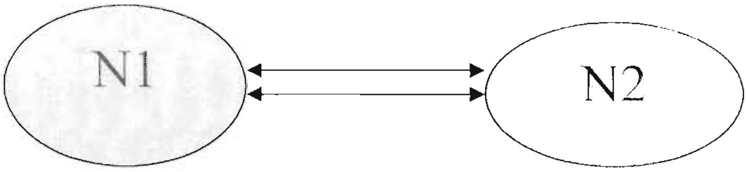


Figure3.6: Ping-pong Effect between two Nodes

5.6.2. Self Looping

If any message from source node and moving around its neighboring nodes but does not reach to destination node, then it's called self-looping. But in our simulation we can eliminate self-looping effect for 20 nodes.

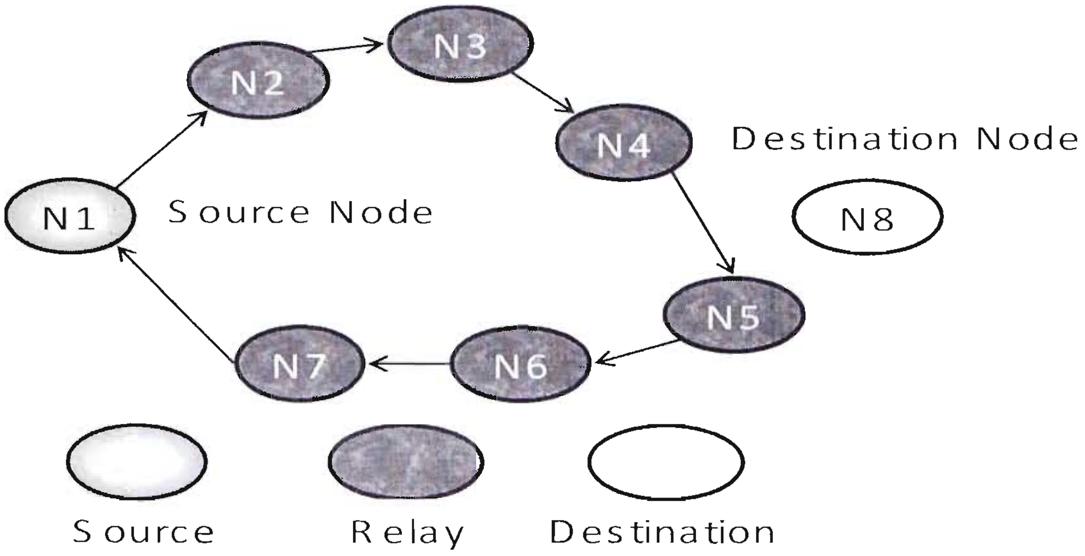


Figure3.7 : Self Looping

3.6. Conclusion

Routing is an essential component of communication protocols in mobile ad hoc networks. The design of the protocols are driven by specific goals and requirements based on respective assumptions about the network properties or application area. The survey tries to review typical routing protocols and reveal the characteristics and trade-offs. In our simulation work we tried to eliminate ping pong effect and self looping of the routing protocols using trust metric.

CHAPTER 4

Trust Management Scheme for MANET Environment

4.1. Introduction

MANETs are self-organizing and adaptive in nature. Securing such networks can be a major challenge. The term 'adaptive' simply implies that an ad-hoc network can take different forms and have highly variable mobile characteristics such as power and transmission conditions, traffic distribution variations, and load balancing. Not only are the ad-hoc networks challenging to design but the need for security services in such networks further complicates the situation.

An ad-hoc network is a collection of autonomous nodes or terminals that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Each node in a mobile ad-hoc network functions both as a host and a router and the control of the network is distributed among the nodes. Hence, each node has to rely on other nodes in setting up a successful session. Trust plays an important part in such scenarios. Hence an appropriate trust management scheme is needed to cope with the nature of MANET environment. Trust is a before-security issue [1] in ad hoc networks. By clarifying the trust relationship, it will be much easier to take proper security measures and make correct decision on any security issues.

Trust can be considered as the link between observations (trust evidence). In our trust management scheme we utilize the numerical value of trust between the nodes as the computational means to evaluate trust relationship between entities. It is a common tendency to adopt policy of trusting entities that are trusted by entity that one trusts. Hence, trust would thus propagate through the network and become accorded when one entity can reach another entity via at least one trust path. In our approach we utilize two modes of trust relationship. First way is through direct observation of other nodes' behavior and the second way is through recommendations from other nodes.

4.2. Trust Issues in MANET

4.2.1. Assumptions and Notations

The following assumptions are considered for the trust management scheme proposed in this thesis:

a. All nodes in the MANET assist in completing the communication session i.e., we ignore the selfish behavior of reluctant nodes.

b. Each node has its own repository of trusted and untrusted nodes based on the previous interactions.

c. The ad hoc network under consideration is a pseudo-open network. Within one hop distance, at least one of the neighboring nodes is in the trusted list of the node which will start the communication. Recommendation trust for a node from the third party node can only be counted if the third party is already in the trust repository of the initiator node.

Throughout the paper the following notations are used:

ω_x → Trust threshold value of node x, above which x trusts other nodes in its repository and can have direct interaction.

δ_x → Cooperative threshold value of node x for other nodes. (To get the service from another node, x's trust value of that node has to be at least equal to cooperative threshold)

$T_x(y)^t$ → Node x's trust on node y at time t. (time notation is omitted when time specific consideration of trust is not considered)

$T_x^d(y)$ → Direct trust of node x on node y.

Other notations are explained when they appear in the paper.

4.2.2. Notions of Trust Establishment

Trust establishment can be viewed as the application of evaluation of a body of trust evidence. The outcome of the trust establishment process is a trust relation. An established trust relation constitutes evidence that can be used in future trust establishment process. Here we are presenting some notions of trust establishment that also hold true in MANET environment.

Self-reinforcibility

Trust, once established in some degree, is often self-reinforcing because individuals have stronger tendencies to confirm their prior beliefs than to disprove them [34]. The converse is also true, as below a certain trust, individuals tend to confirm their suspicions of others [35]. The first part of the rule is based on the idea that, if trust between two nodes is initially above some threshold value, which we call ω_x for node x, then the trust between the two nodes will not decrease below the threshold. This is because, for each member in the interaction, they will tend to look for the best in the behavior of the other and will tend to get it. Since above the threshold, each will tend to receive service from other. Thus trust will increase or at least remain constant. We call ω_x as the service factor of node x.

The converse is true, since, if two nodes in a relationship, trust each other below a certain threshold value, δ ($\delta \neq \omega$), then they will tend not to cooperate with each other.

The first part of the rule is that trust is self-heightening. Consider two nodes, x and y.
If

$$\left(T_x(y)^t > \omega_x\right) \wedge \left(T_y(x)^t > \omega_y\right) \qquad (1)$$

Then,

$$\left(T_x(y)^{t+a} \geq T_x(y)^t\right) \wedge \left(T_y(x)^{t+a} \geq T_y(x)^t\right) \qquad (2)$$

That is, if at time t, x trusts y more than a threshold value and y trusts x more than a threshold value, then at a later time $t+a$, ($a>0$), the amount of trust they will have in each other will be greater than or equal to the amount of trust they have in each other at time t. The second part of the rule is that trust is self-reinforcing downwards also
If

$$\left(T_x(y)^t < \delta_x\right) \wedge \left(T_y(x)^t < \delta_y\right) \qquad (3)$$

Then.

$$\left(T_x(y)^{t+a} \le T_x(y)^t\right) \wedge \left(T_y(x)^{t+a} \le T_y(x)^t\right) \tag{4}$$

Note that ω and δ are not necessarily the same value for each node. neither is $\omega_x = \omega_y$ nor is $\delta_x = \delta_y$ and for any node x. $\delta_x \le \omega_x$. The threshold value δ_x is the cooperative factor. If trust value for any node falls below this threshold. then that node is not considered trustworthy.

Tansitivity

Trust is not transitive in MANET environment. Hence the following can be said for the nodes

$$\left(T_x(y) > T_y(z)\right) \not\Rightarrow \left(T_x(y) > T_x(z)\right) \tag{5}$$

In other words. that node x trusts y by some value and y trusts z by some value. says little or nothing about how much x trusts z.

But in MANET we assume all nodes act rationally in trust establishment (assumption a) and hence the following can be held

$$\left(T_x(y) > T_x(z)\right) \wedge \left(T_y(z) > T_x(w)\right) \Rightarrow \left(T_x(y) > T_x(w)\right) \tag{6}$$

Reversibility

Trust in general is non-reversible. The same is true in ad hoc network. That is $T_x(y) \neq T_y(x)$. Both x and y nodes evaluate $T_x(y)$ and $T_y(x)$. respectively.

4.2.3. Types of Trust

4.2.3.1. Direct Trust

If any node is already enlisted within the repository of trusted list of the initiator node s. then the trust value of node s on the node in question (say node y) is the direct trust value. If the node in question was already engaged in assisting any communication of the initiator within a certain period of time. then it would be in its trust repository. The value of direct trust is represented by $T_x^d(y)$.

4.2.3.2. Recommendation Trust

A recommendation trust relationship exists if node s is willing to accept the experiences with y of third party node o. The third party must be in the trusted repository of the initiator node s with the trust value, $T_s^d(o) \geq \delta_s$.

Recommendation trust is represented by $T_s^O(y)$, where O is the set of other nodes, who are already in the trust repository of the initiator node s and whose trust on node y are utilized by node s in evaluating the recommendation trust. The set of nodes O is defined as: O $\{\forall$ node $o \in O \Rightarrow o$ is in the range of y and $\exists\ T_s^d(o)$ s.t. $T_s^d(o) > \delta_s\}$

The recommendation trust of the neighboring nodes (i.e. within one hop distance of node y) is calculated as the simple average of the product, $T_s^d(o) \times T_o^d(y)$, $\forall$ node $o \in O$ i.e.,

$$T_s^O(y) = \frac{\sum_{o \in O} \left(T_s^d(o) \times T_o^d(y) \right)}{|O|} \qquad (7)$$

where $|O|$ is the cardinality of the set O.

By utilizing the direct trust and the recommendation trust the initiator node s can get the trust estimate of any other node y, $T_s(y)$.

In Figure 4.1, node x is the initiator of a communication and y is already in its trust repository. So node x` trust on y in its repository is $T_x^d(y)$. Node o is a node within one hop distance of node y.

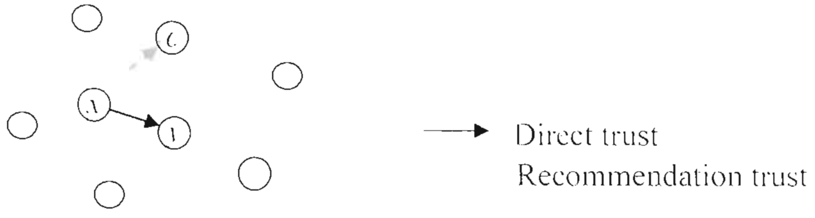


Figure 4.1: Mobile Ad-hoc Network Illustrating Direct and Recommendation Trust

The trust value computed by node s on the basis of node o's trust on node y is considered as the recommendation trust $T_s^o(y)$.

4.3. Trust Estimate Calculation

We utilize both direct and recommendation trust for evaluating the trustworthiness of the surrounding nodes by the initiator node. We call it trust estimate and define the initiator node s' trust estimate on another node y as:

$$T_s(y) = \alpha_1 T_s^d(y) + \alpha_2 T_s^o(y) \tag{8}$$

where α_1 and α_2 are weighting factors, such that $\alpha_1 + \alpha_2$ 1. It always need to keep $\alpha_1 + \alpha_2 = 1$ because self-trust is always unity. Thus by varying α_1 and α_2, node s can vary the weight of direct trust and recommendation trust in calculating the trust estimate, $T_s(y)$. Here $0 \le \{T_s(y), T_s^d(y), T_s^o(y)\} \le 1$. The recommending nodes must be in the trust repository of the initiator node s. If node s' direct trust value on node y, $T_s^d(y) \ge \omega_s$ (ω_s= service trust threshold value of node s) then s can rely on node y in getting the service i.e. transmitting packets to the destination node chain. In this case, α_1=1 and hence α_2=0. The value of $T_s^o(y)$ in equation (8) is the recommendation trust of the neighboring nodes and is calculated by equation (7).

CHAPTER 5

SIMULATION RESULT & ANALYSIS

5.1. Defining the Neighboring Node

5.1.1. Algorithm

Step 1: Define the number of node and the number of piconet

Step 2: Calculate the number of neighbor node around a node as
num_neig_node= (num_node/num_pico_net)+C. where C is an integer

Step 3: Select a node for which neighboring node has to be defined.

Step 4: Define the neighbor nodes around the selected node. The number of neighbor of neighbor node is calculated before in Step 2.

Step 5: decision making: selected node == last node? If yes go to Step 6. Otherwise go to Step 3.

Step 6: End

5.1.2. Simulation result

Neighboring node matrix is as following neig_node:

```
     2    3    4    5    6    7    8    9
     1    3    4    5    6    7    8    9
     1    2    4    5    6    7    8    9
     1    2    3    5    6    7    8    9
     1    2    3    4    6    7    8    9
     2    3    4    5    7    8    9   10
     3    4    5    6    8    9   10   11
     4    5    6    7    9   10   11   12
     5    6    7    8   10   11   12   13
     6    7    8    9   11   12   13   14
     7    8    9   10   12   13   14   15
```

8	9	10	11	13	14	15	16
9	10	11	12	14	15	16	17
10	11	12	13	15	16	17	18
11	12	13	14	16	17	18	19
12	13	14	15	17	18	19	20
12	13	14	15	16	18	19	20
12	13	14	15	16	17	19	20
12	13	14	15	16	17	18	20
12	13	14	15	16	17	18	19

Neighbor of node of some selected node is given below:

For node 1 neighbor nodes are:	2	3	4	5	6	7	8	9
For node 2 neighbor nodes are:	1	3	4	5	6	7	8	9
For node 20 neighbor nodes are:	12	13	14	15	16	17	18	19
For node 15 neighbor nodes are:	11	12	13	14	16	17	18	19

5.2. Calculating the Trust Estimate Matrix

5.2.1. Algorithm

Step 1: Define the direct trust matrix

Step 2: Select a node in respect of which the trust estimate matrix of other node will be calculated

Step 3: Select a node for which the trust estimate will be calculated

Step 4: Decision making: selected node in Step 2 == selected node in Step 3? If yes then set the trust estimate as 1 and go to Step 3. Otherwise go to Step 5.

Step 5: Find the direct trusts value of the selected node in Step 3 with respect to other nodes in system except the selected node in Step 2 along with the direct trust value of each other nodes with respect to the selected node in Step 2.

Step 6: Multiply the each conjugate direct trust value and add up all the multiplied value

Step 7: Divide the value founded in Step 6 with the number of trust conjugate and it is the estimated trust value of selected node in Step 3 respect to the node selected in Step2.

Step 8: Decision making: selected node in Step 3 == last node? If yes, go to Step 9. Otherwise go Step 3.

Step 9: Is the selected node in Step 2 is the last node? If yes, go to Step 10. Otherwise go to Step 2.

Step 10: End.

5.2.1.1. Example of Directed Trust matrix for a Five (5) Node System

```
T_d =

    1.0000    0.1122    0.7241    0.9862    0.8289
    0.3941    1.0000    0.2816    0.4733    0.1663
    0.5030    0.4668    1.0000    0.9028    0.3939
    0.7220    0.0147    0.7085    1.0000    0.5208
    0.3062    0.6641    0.7839    0.8045    1.0000
```

5.2.1.2. Example of Estimate Trust matrix for a Five (5) Node System

```
Ts_y =

    1.0000    0.1005    0.6381    0.8674    0.6852
    0.2892    1.0000    0.2366    0.3334    0.1495
    0.5135    0.3450    1.0000    0.8347    0.3739
    0.6384    0.0322    0.6272    1.0000    0.4613
    0.3174    0.4897    0.6567    0.6697    1.0000
```

5.2.1.3. Trust Estimate Graph for 100 Nodes

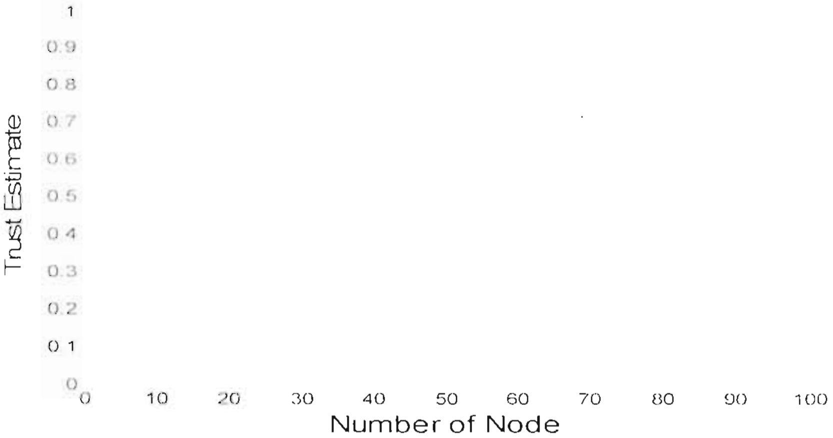


Figure 5.1: Trust Estimate for α_1 =0.7 & α_2 =0.3

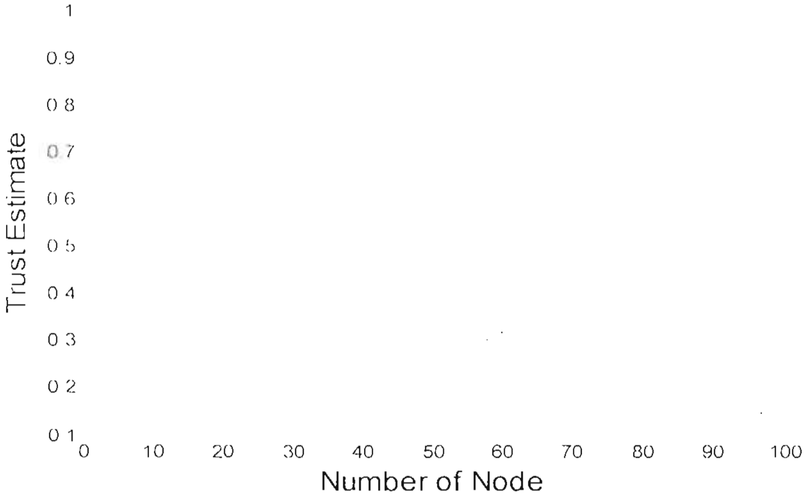


Figure 5.2: Trust Estimate for α_1 =0.5 & α_2 =0.5

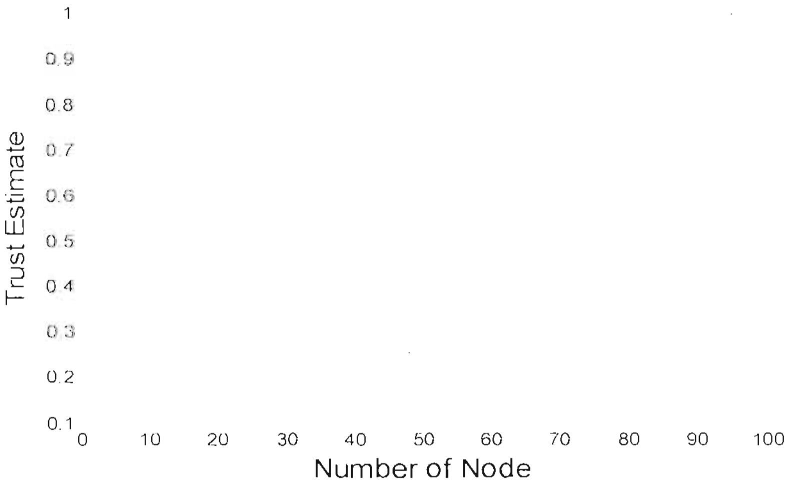


Figure 5.3: Trust Estimate for α_1 =0.3 & α_2 =0.7

For Trust estimate calculation. Here we used 100 nodes (For example). if we decrease the value of α_1 then trust level also decreased and if we increase the value of α_1 then trust level also increased. From Figure5.1~5.3 we see how trust level is varied with the value of α_1.

5.4. Trusted Path Generation Algorithm

5.4.1. Algorithm

Step 1: Define the initial and destination node.

Step 2: Define the maximum number of transit and a variable that will give the number of transit. Initially set the variable to 1.

Step 3: Set the initial node as the first node in the trusted path matrix.

Step 4: Decision making: initial node == destination node? If yes go to Step 25. Otherwise go to Step 5.

Step 5: Is the initial node is in the piconet that consist destination node? If yes. set the destination node as the next node in trusted path matrix and go to Step25 Otherwise go to Step6.

Step 6: Decision making: number of transit> maximum number of transit? If yes. go to Step 24. Otherwise go to Step 7.

Step 7: Get the neighbor of the initial node.

Step 8: Get the direct trust and the estimated trust of the neighbor of the initial node with respect to the initial node.

Step 9: Get the neighbor nodes that have the direct trust which is greater than the service trust threshold value. Is the number of nodes greater than zero zero? If yes. go to Step 10. Otherwise go to Step 12.

Step 10: Is the number of nodes got in Step 9 is equal to one (1). If yes, select the node as the next node in the trusted path matrix and go to Step 13. Otherwise go to Step 11.

Step 11: Find the node that have maximum direct trust and select it as the next node in the trusted path matrix and go to Step 13.

Step 12: Find the node that has the maximum estimated trust form the estimated trust matrix and set it as the next node in the trusted path matrix and go to Step 13.

Step 13: Increase the number of transit by 1 and set the next node and current node in previous state as current node and previous node in this stage respectively.

Step 14: Does the number of transit exceed the maximum number of transit? If yes, go to Step 24. Otherwise go to Step 15.

Step 15: Is the current node in trusted path matrix is the destination node. If yes, go to Step 25. Otherwise go to Step 16.

Step 16: Is the current node is in the piconet that consist destination node? If yes, set the destination node as the next node in trusted path matrix and go to Step 25. Otherwise go to Step 17.

Step 17: Get the neighbor of the current node.

Step 18: Deduct the neighbors of the previous node in the trusted path along with the previous node.

Step 19: Find the direct trust and the estimated trust of the nodes founded is Step 17 with respect to the current node.

Step 20: Get the neighbor nodes that have the direct trust which is greater than the service trust threshold value. Is the number of nodes greater than zero? If yes, go to Step 21. Otherwise go to Step 23.

Step 21: Is the number of nodes got in Step 17 is equal to one (1). If yes, select the node as the next node in the trusted path matrix and go to Step 13. Otherwise go to Step 22.

Step 22: Find the node that have maximum direct trust and select it as the next node in the trusted path matrix and go to Step 13.

Step 23: Find the node that has the maximum estimated trust form the estimated trust matrix and set it as the next node in the trusted path matrix and go to Step 13.

Step 24: End the process and print 'Destination node cannot be reached for maximum transit restriction'.

Step 25: End the process and print the trusted path matrix.

CHAPTER 6

CONCLUSION & RECOMMENDATION

In this thesis we propose a trust management scheme applied in MANET routing protocol. We utilize the random value of the trust between the nodes as the computational means to find out the minimum but highly trusted path between the initiator and the destination node and the initiator node is exclusively in control of generating the trusted path based on his direct trust and recommendation from the other nodes. Like cryptographic approach the source node does not have to depend on other nodes as CAs to provide the secret shares which is the bottleneck of the cryptographic approach in MANET environment. Our trust management scheme can be incorporated in any efficient routing algorithm to achieve highest trusted but minimum routing path from the initiator node to the destination node. In our approach we assumed that all nodes assist in completing the communication session. But due to energy constraint of the mobile nodes, some nodes may not participate in the session and act selfishly. On top of AODV our protocol can be implemented so that reluctant nodes can be eliminated.

Our future work is to simulate with 40 nodes as at present simulation with more than 40 nodes destination node is not reachable due to self-looping effect. Even though we eliminated ping-pong effect but self looping effect is still persist for large number of nodes.

Reference:

[1] Z. Yan, P. Zhang, and T. Virtanen, "Trust Evaluation Based Security Solution in Ad Hoc Networks", in Proceedings of the Seventh Nordic Workshop on Secure IT Systems (NordSec 2003), 15-17 October 2003, Gjovik, Norway.

[2] P. Papadimitratos and Z. J. Haas, "Securing Mobile Ad Hoc Networks", The Handbook of Ad Hoc Wireless Networks pp. 551 - 567, 2003.

[3] L. Zhou and Z.J. Hass, "Securing Ad-hoc Networks", IEEE Network magazine, 13(6), November/December 1999.

[4] C. Davis, "A Localized Trust Management Scheme for Ad Hoc Networks", in Proceedings of Third International Conference on Networking (ICN04), March, 2004.

[5] J. Hubaux, L. Buttyan and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks", in Proceeding of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Long Beach, CA, October 2001.

[6] P. Zimmermann, *The Official PGP User's Guide*, MIT press, 1995.

[7] S. Buchegger and J.-Y. L. Boudec, "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-Hoc Networks. In Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), Sophia-Antipolis, France, 2003.

[8] B. Adamson, "Tactical Radio Frequency Communication Requirements for IPng", RFC 1677, August 1994.

[9] IETF Manet Chapter,http://www.ietf.org/html.charters/manet-chapter.html.

[10] S. Murthy, and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", ACM Mobile Networks and Application Journal: Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[11] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", ACM Computer Communication Review, Vol. 24, No.4, (ACM SIGCOMM'94) Oct. 1994, pp.234-244.

[12] C.E. Perkins and E.M. Royer, "Ad Hoc on Demand Distance Vector Routing", in Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, (WMCSA '99) 1999, pp.90 - 100.

[13] N. Nikaein. C. Bonnet and N. Nikaein, "HARP - Hybrid Ad Hoc Routing Protocol", in Proceeding of International Symposium on Telecommunications (IST 2001). Iran/Tehran 2001.

[14] C.-K. Toh. "Associativity Based Routing for Ad Hoc Mobile Networks". Wireless Personal Communications Journal: Special Issue on Mobile Networking and Computing Systems. March 1997. pp.103-139..

[15] C.W. Wu and Y.C. Tay. "AMRIS: A Multicast Protocol for Ad Hoc Wireless Networks". in Proceedings of IEEE MILCOM'99, Atlantic City, Nov. 1999

[16] E. M. Royer and C. E. Perkins. "Multicast Operation of the Ad hoc On Demand Distance Vector Routing Protocol", ACM MOBICOM. Aug.1999.

[17] M. Liu. R. Talpade, A. McAuley, and E. Bommaiah. "AMRoute: Ad-hoc Multicast Routing Protocol", Technical Report, CSHCN T. R. 99-1, University of Maryland.

[18] A. Ballardie, "Core Based Trees (CBT version 2) Multicast Routing", Internet Request for Comment 2189, September 1997.

[19] S.J. Lee. M. Gerla and C.C. Chiang. "On Demand Multicast Routing Protocol". in Proceedings of IEEE WCNC'99. New Orleans. pp. 1298-1302. Sept 1999

[20] S. Murthy. and J.J. Garcia-Luna-Aceves. "An Efficient Routing Protocol for Wireless Networks". ACM Mobile Networks and Application Journal: Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[21] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", ACM Computer Communication Review, Vol. 24. No.4, (ACM SIGCOMM'94) Oct. 1994. pp.234-244.

[22] G. Pei, M. Gerla and T.-W. Chen, "Fisheye State Routing in Mobile Ad Hoc Networks". In Proceedings of the 2000 ICDCS Workshops, Taipei, Taiwan. Apr. 2000. pp. D71-D78

[23] D. Johnson and D. A. Maltz. "Dynamic Source Routing in Ad Hoc Wireless Networks', Mobile Computing (T. Imielinski and H. Korth. eds.). Kluwer Acad. Publ., 1996.

[24] V. Park, and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification", IETF Internet draft. 1997.

[25] L. Ji and M. S. Corson. "A Lightweight Adaptive Multicast Algorithm". in Proceedings of GLOBECOM'98, November 1998.

[26] J. Broch. D.A. Maltz. D.B. Johnson, Y.C. Hu, and J. Jetcheva. "A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", in Proceedings of MOBICOM. 1998. pp.85-97.

[27] S. R. Das. C. E. Perkins and E. M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks", in Proceedings of the IEEE InfoCom. March 2000.

[28] Z. J. Haas and M.R Pearlman. "The Zone Routing Protocol (ZRP) for Ad Hoc Networks". IETF Internet draft . August 1998

[29] C. C. Chiang. T. C. Tsai. W. Liu and M. Gerla. "Routing in Clustered Multihop. Mobile Wireless Networks with Fading Channel, in Proceeding of the Next Millennium IEEE SICON. 1997.

[30] Mingliang Jiang, Jinyang Li and Y. C. Tay, "Cluster Based Routing Protocol (CBRP)". Internet draft,draft-ietf-manet-cbrp-spec-01.txt.

[31] S. Basagni. I. Chlamtac. V. Syrotiuk and B. WoodWard. "A Distance Routing Effect Algorithm for Mobility (DREAM)". In Proceedings of 4th MOBICOM, 1998

[32] J. Li. J. Jannotti. D. S. J. De Couto, D. Karger and R. Morris. "A scalable Location Service for Geographic Ad Hoc Routing", in Proceedings of MOBICOM'2000. Boston. MA. USA. 2000.

[33] N. K. Guba and T. Camp. "Recent work on GLS: a Location Service for an Ad Hoc Network". in Proceedings of the Grace Hopper Celebration (GHC). 2002

[34] R. A. Hinde. and J. Groebel. "Cooperation and Prosocial Behavior". Cambridge University. Press. 1991.

[35] R. T. Golembiewski. and M. McConkie. "The Centrality of Interpersonal Trust in Group Process". pp. 131-185. Theories of Group Processes; Edited by Carry L. Cooper. Wiley & Sons Inc.

APPENDIX: Mathlab code

```
%Dipu & Tanzil  Final Project_EEE498

clear all
clc
num_node=input('give the number of nodes \n');
ini_node=input('give the initial node \n');
desti_node=input('give the destination node \n');
num_piconet=input('give the number of piconet\n(make sure that number of node can be
totally devided by the piconet number)\n');
max_transit=input('give the number of maximum transit \n');
num_neig_node=(num_node/num_piconet)+3; %%The number of neighbor node.
alpha_1=0.7;
alpha_2=0.3;
ws=0.8;
dels=0.5;
n=1:num_node;
l=length(n);

%------------------ Defining the neigbouring node ------------------------%
for i=1:num_node
    p=i-1; q=num_node-i;
    if (rem(num_neig_node,2)==0)
        dr=num_neig_node/2;
        if (p<=dr)
            a=1:i-1; %a is a number
            b=(i+1):i+(num_neig_node-length(a));
            neig_node(i,:)=[a,b]; % Neighboring node matrix
        elseif (q<=dr)
            b=i+1:num_node;
            a=i-(num_neig_node-length(b)):i-1;
            neig_node(i,:)=[a,b];
        else
            a=i-dr:i-1;
            b=i+1:i+dr;
            neig_node(i,:)=[a,b]; % Neighboring node matrix
        end
    else
        dr1=(num_neig_node-1)/2;
        dr2=(num_neig_node+1)/2;
        if(p<=dr1)
            a=1:i-1;
            b=i+1:i+(num_neig_node-length(a));
            neig_node(i,:)=[a,b];
        elseif(q<=dr2)
            b=i+1:num_node;
            a=i-(num_neig_node-length(b)):i-1;
            neig_node(i,:)=[a,b];
        else
            a=i-dr1:i-1;
            b=i+1:i+dr2;
            neig_node(i,:)=[a,b];
        end
    end
end

%------------------ Defining the nodes in hop ----------------------------%
node_in_piconet=num_node/num_piconet; % the number of node in a piconet
m=1;
for i=1:node_in_piconet:l
    PICONET(m,:)=[n>=i & n<i+node_in_piconet].*m; % matrices of piconet
```

```
   m=m+1;
end
PICONET=sum(PICONET);

%------------ Calculating the trust estimate matrix --------------------%
T_d=eye(l,l)+(triu(ones(l,l),1)+tril(ones(l,l),-1)).*rand(l,l); % T_d= direct trust
for i=1:l;
   for j=1:l;
      for k=1:l;
         if (i==j)
            Ts_y(i,j)=1;                  %Ts_y= estimate trust
         else
            if (j==k)
             Ts_o(k)=0;
            else
               if (T_d(i,k)>=dels)
               Ts_o(k)=T_d(j,k)*T_d(k,j);
               else
               Ts_o(k)=0;
               end
               Ts=sum(Ts_o)/nnz(Ts_o);
              %using eqa-7 from Ch-4,nnz=number of non zero Matrix element.
               Ts_y(i,j)=alpha_1*T_d(i,j)+alpha_2*Ts; %using eqa-8 from Ch-4
            end
         end
      end
   end
end
plot(n,Ts_y(1,:),'*')
line([0,l],[ws,ws])
line([0,l],[dels,dels])
```

```
axis([0 1 0 1])
xlabel('Nodes number'.'Fontsize'.14)
ylabel('Trust estimate'.'Fontsize'.14)

%----------Transition from one node to another to reach the destination nodes-----------%
Relay(1)=ini_node; % ini_node=initial node and relay(1) are equal
i=2;
while (1)
   if (i>=2 & i<=max_transit)
      if(PICONET(ini_node)~=PICONET(desti_node)) % desti_node=the destination nodes
         n_nodes=neig_node(ini_node.:);
         dtrust_neig=T_d(ini_node.n_nodes); % dtrust_neig= neigbouring node of direct trust
         estrust_neig=Ts_y(ini_node,n_nodes); );
         %estrust_neig=neibouring node of estimate Trust
         dtrust_ws=[dtrust_neig>=ws].*dtrust_neig;

         if(sum(dtrust_ws)~=0)
            [val1 pos1]=max(dtrust_ws); %val=value & pos=position
            prsnt_node=n_nodes(pos1): % prsnt_node=present node
            Relay(2)=prsnt_node: %now  present node  is relay(2)
            pre_node=ini_node: % present node and initial node are equal

            if(prsnt_node==desti_node)
               break
            end
         else
            [val2 pos2]=max(estrust_neig):
            prsnt_node=n_nodes(pos2): %now position 2 of n_node is  present node
            Relay(2)=prsnt_node; %now present node  is relay(2)
            pre_node=ini_node: % pre_node=pervious node & initial node are equal
      if(prsnt_node==desti_node)
        break          %break terminats the execution of while loop
      end
    end
  else
    Relay(2)=desti_node; %now desti_node is relay(2)
    break
  end
elseif(i>2 & i< max_transit)
    if(PICONET(prsnt_node)~=PICONET(desti_node))
      if(PICONET(prsnt_node)~=PICONET(pre_node))
        pre_neig=(nonzeros((PICONET==PICONET(pre_node)).*n))';
        prsnt_neig=neig_node(prsnt_node,:);
        n_nodes=(setdiff(prsnt_neig(:),pre_neig(:)))';
        %setdiff=when prsnt_neig &pre_neig vectors returns the values in prsnt_neig
         % that are not in pre_neig

        dtrust_neig=T_d(prsnt_node,n_nodes);
        estrust_neig=Ts_y(prsnt_node,n_nodes);
        dtrust_ws=[dtrust_neig>=ws].*dtrust_neig;

        if(sum(dtrust_ws)~=0) %sum of dtrust_ws is not equal zero
          [val1 pos1]=max(dtrust_ws); %value1 & position1 are maximum dtrust_ws
          pre_node=prsnt_node;
          prsnt_node=n_nodes(pos1);
          Relay(i)=prsnt_node;

          if(prsnt_node==desti_node)
            break
          end
        else
        [val2 pos2]=max(estrust_neig); %value2 & position2 are maximum dtrust_ws
        pre_node=prsnt_node;
        prsnt_node=n_nodes(pos2);
        Relay(i)=prsnt_node;

        if(prsnt_node==desti_node)
            break
        end
    end
else
    pre_neig1=[pre_node]';
    prsnt_neig1=neig_node(prsnt_node,:);
    n_nodes1=(setdiff(prsnt_neig1(:),pre_neig1(:)))';
   dtrust_neig1=T_d(prsnt_node,n_nodes1);
    estrust_neig1=Ts_y(prsnt_node,n_nodes1);
    dtrust_ws1=[dtrust_neig1>=ws].*dtrust_neig1;

    if(sum(dtrust_ws1)~=0)
        [val1 pos1]=max(dtrust_ws1);
        pre_node=prsnt_node;
        prsnt_node=n_nodes1(pos1);
        Relay(i)=prsnt_node;

        if(prsnt_node==desti_node)
            break
        end
    else
        [val2 pos2]=max(estrust_neig1);
        pre_node=prsnt_node;
        prsnt_node=n_nodes1(pos2);
        Relay(i)=prsnt_node;
```

```
                       if (prsnt_node==desti_node)
                           break
                       end
                   end
               end
            else
                Relay(i)=desti_node:
                    break
            end
    else
        fprintf('Destination node can not be reached for maximum transit restriction.\n')
    break
    end
    i=i+1;
end
Relay

%---------------------------- END ---------------------------------------%
```