TCP PERFORMANCE OVER MULTIPATH ROUTING IN AD HOC NETWORKS

Developed by
Adila Arif
Ummarah Naseem

Supervised by
Dr. S. Tauseef-ur-Rehman
Prof. Dr. Khalid Rashid

Department of Computer Science
Faculty of Applied Sciences,
International Islamic University, Islamabad
(2004)

In the name of **ALMIGHTY ALLAH,**
The most Beneficent,
The most Merciful

Department of Computer Science
International Islamic University Islamabad

FINAL APPROVAL Dated: 02nd July 2004

It is certificate that we have read the thesis submitted by Ms. Adila Arif Reg. No. 83-CS/MS/02 and Ms. Ummarah Naseem Reg. No. 93-CS/MS/02 and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the MS Degree in Computer Science.

Committee

External Examiner
Dr. Muhammad Qasim Rind
Ex-Director General Establishment Division,
Government of Pakistan,
Islamabad

Internal Examiner
Dr. Sikandar Hayat Khiyal
Head,
Department of Computer Science,
Faculty of Applied Sciences,
International Islamic University
Islamabad

Supervisor
Prof. Dr. Khalid Rashid
Dean,
Faculty of Applied Sciences &
Faculty of Management Sciences,
International Islamic University
Islamabad

2/7/04

Supervisor
Dr. S. Tauseef-ur-Rehman
Head,
Department of Telecommunication Engineering &
Computer Engineering,
Faculty of Applied Sciences,
International Islamic University
Islamabad

A dissertation submitted to the
DEPATMENT OF COMPUTER SCIENCES
INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
as a partial fulfillment of the requirements
for the award of the degree of
MS Computer Science

DEDICATION

Dedicated to our supervisor Dr.S.Tauseef-ur-Rehman

And

To our loving parents

DECLARATION

We hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that we have developed this thesis and software entirely on the basis of our personal efforts made under the sincere guidance of our teachers. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Adila Arif
83-CS/MS/02
Ummarah Naseem
93-CS/MS/02

ACKNOWLEDGEMENTS

All praise to Almighty Allah, the Most merciful and beneficent Who enabled us to complete this thesis. We pray to Almighty Allah that may this research work be used for the betterment of humanity and Muslim Ummah.

We like to express our highest gratitude and sincere thanks to our supervisors Prof.Dr.Khalid Rashid and Dr.S. Tauseef-ur-Rehman who kept our moral high by their suggestions and appreciations. And last but the least we like to acknowledge the support of our families and friends who helped us in every step of this thesis.

Adila Arif
83-CS/MS/02
Ummarah Naseem
93-CS/MS/02

PROJECT IN BRIEF

Project Title:	TCP Performance Over Multipath Routing in Ad Hoc Networks
Objective:	To improve the path availability in Ad hoc Networks
Undertaken By:	Adila Arif, Ummarah Naseem
Supervised By:	Prof. Dr. Khalid Rashid Dean, Faculty of Applied Sciences & Faculty of Management Sciences, International Islamic University Islamabad Dr. S. Tauseef-ur-Rehman Head, Department of Telecommunication Engineering & Computer Engineering, Faculty of Applied Sciences, International Islamic University Islamabad
Simulator Used:	Network Simulator 2 (NS2)
System Used:	Intel PIV 1.80 GHz
Operating System:	Windows 2000
Date Started:	1st October, 2003
Date Completed:	22nd June, 2004

Abstract

This study focuses on improving the path availability using Split Multipath Routing Protocol. Our study shows that this protocol is better suited for mobile ad hoc networks because it generates less control overhead and manages the mobility in a more efficient manner. It builds maximally disjoint routes. Multipath routes help in minimizing route recovery and reduces control overhead. Distributing traffic into multipath prevents nodes from being congested. This routing protocol is able to copy with the new characteristics that a MANET provides like changing topology, limited power supply and moving nodes.

ABBREVIATIONS

ABR:	Associativity Based Routing
AODV:	Ad Hoc On-Demand Distance Vector
AODV-BR:	Ad Hoc On-Demand Distance Vector with Backup Routes
DH:	Data Header
DSR:	Dynamic Source Routing
FSR:	Fisheye State Routing
GPS:	global Positioning System
IETF:	Internet Engineering Task Force
LAN:	Local Area Network
LAR:	Location Aided Routing
LBR:	Location Based Routing
LSA:	Link State Advertisement
LSRR:	Loose Source Record Routes
MANET:	Mobile Ad Hoc Networks
MOR:	Multipath On Demand Routing
NR:	No Route
NS2:	Network Simulator
ODSMR:	On-Demand Split Multipath Routing
QoS:	Quality of Service
RC:	Route Control Message
RE:	Route Entry
RREP:	Route Reply
RREQ:	Route Request
RRER:	Route Error
RTO:	Retransmission Timeout
SMR:	Split Multipath Routing
TCP:	Transport Control Protocol

WRP: Wireless Routing Protocol

GloMoSim: Global Mobile Information Systems Simulation

UCLA PCL: UCLA Parallel Computing Laboratory

SNT: Scalable Network Technologies:

Table of Contents

Ch. No. *Contents* *Page No.*

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. Introduction

The objective of our research is to improve the path availability in Mobile Ad Hoc Environment using multipath routing. Previous research on multipath routing mostly used UDP traffic for performance evaluation. When TCP is used, we find that most times using multipath simultaneously improves the TCP performance. We have test another strategy called Split multipath routing protocol which is an on-demand routing protocol. On-demand routing in particular, is widely developed in bandwidth constrained mobile wireless ad hoc networks because of its effectiveness and efficiency. SMR establishes and utilizes multiple routes of maximally disjoint paths which helps minimizing route recovery process and control message overhead in Ad Hoc Networks.

radios are being increasingly deployed in common activities. Applications such as conferences, meetings, lectures, crowd control, search and rescue, disaster recovery, and automated battle fields typically do not have central administration or infra structure available. In these situations, ad hoc networks, consist of hosts which are equipped with portable radios. They must be deployed without any wired based station. In ad hoc networks, each host must act as a router since routes are mostly multi hop. Nodes in such a network move arbitrarily, thus network topology changes frequently, unpredictably, and may consist of unidirectional links as well as bi-directional links. More over, wireless channel bandwidth is limited. The scarce bandwidth decreases even further due to the effects of multiple access, signal interference and channel fading. Network host of ad hoc networks operate on constraint battery power, which will eventually be exhausted. Ad hoc networks are also more prone to security threats. All these limitations and constraints make multi hop network research more challenging.

1.1 Challenges in routing and multipath

Routes in ad hoc networks are multi hop because of the limited propagation range of wireless radios. Since nodes in the network move freely and randomly, routes often get disconnected. Routing protocols are thus responsible for maintaining and reconstructing the routes in a timely manner as well as establishing the durable routes. In addition, routing protocols are required to perform all the above tasks without generating excessive control message overhead. Control packets must be utilized efficiently to deliver data packets, and be generated only when necessary. Reducing the control overhead can make the routing protocol efficient in bandwidth and energy consumption. Multipoint communications have emerged as one of the most researched areas in the field of networking.

1.2 Accomplishment and Contributions

Our accomplishments, which are elaborated throughout this dissertation, can be broadly listed as follows:

- Studied and compared the simulation performance of various routing protocols in ad hoc networks [1].
- Performed simulation of up to 10 nodes and evaluated ad hoc routing protocol scalability [2]. We also introduce several schemes to improve the protocol performance in large networks [2].
- Designed on demand unicast protocols that build multiple routes. Ad hoc on demand Distance Vector with Backup Routes (AODV-BR) [3] is a scheme applied to the existing AODV protocol to construct multiple back up routes without generating additional control overhead. Backup routes are utilized when the primary route is disconnected. On the other hand, Split Multipath Routing (SMR) builds maximally disjoint multiple routes and distributes the traffic into multipaths.
- Proposed the On-Demand Split Multipath Protocol (ODSMR) [4,5,6]. SMR builds the mesh structure on demand to provide multiple paths. The mesh makes the protocol robust to mobility. We implemented the protocol in simulation platform using Network Simulator (NS2). The protocol is recently approved standard at the IETF (Internet Engineering Task Force) MANET (Mobile Ad hoc Networks) Work Group.

1.3 ROUTING

This section introduces the subject of routing and different techniques involving it. Different characteristics of the routing protocols are presented and discussion of the characteristics which are suitable for our scenario is presented.

The routing process determines the paths between nodes in the network. It is the routing protocols function to control these events. This is very complicated matter due to large involvement of all nodes in the network. There are several different factors to take into consideration when determining the paths between the nodes. One of the most challenging is the dynamics nature of the network. It is hard to keep track of the all nodes whereabouts at all times. As a result of this, a reactive method has been developed to solve this issue. Unlike the proactive method the reactive does not need to know the nodes location at all times. Instead it only needs to make a request for a path when it

with a very low rate of mobility.

1.3.1 Traditional routing algorithms

To understand the routing principals in a MANET there is a good idea to take a look at the conventional routing algorithms such as distance vector, link state, flooding and source routing. This is because many of the routing protocols for a MANET have a traditional routing concept as underlying algorithms.

Distance vector

The distance vector technique is based on that every node maintained a forwarding table with the best route to every node in a network. In a certain time interval the information is sent to every neighboring node in the network. These nodes then conduct a comparison between their own routing table and the received one. If the distance between any nodes in the received table is smaller compared to the one At hand, the node updates the routing table with the new values. If the values that is in the forwarding table is from the node that now is sending a new value, the node updates the forwarding table

Flooding

With this technique every packet is sent to every node in the network and is broadcasted by the receiving nodes exactly once. Each node on receiving the packet, broadcast it to every neighboring node, except the source (the node from which the packet is received). These, neighboring nodes, in turn do the same and so on. To avoid retransmission of the same packet twice every packet is triggered with a source address and a sequence number which serves as a unique identifier. With these identifiers, each node keeps track of the packets which they have transmitted.

This approach has a very high consumption of network resources since every packet is sent to every possible node to ensure that the packet arrives to its destination. On the other hand it results in an extremely high delivery ratio.

Link State Routing

Link state routing works almost like distance vector when it comes to the usage of a forwarding table. What differentiates them is how the table is updated. Link state generates its table so that every node keeps a map over the nodes in the network. From this map every node can use a shortest path algorithm to decide which way is the shortest to each destination and hence know what the next hop should be in the forwarding table.

When there is a change in the network, for example a node connects or disconnects, a message is sent throughout the network to announce the change. The message is called a link state advertisement (LSA) and is passed through the network by flooding. All nodes receive the message and update their maps accordingly. If you compare this strategy with the strategy used in distance vector, it makes link state routing more reliable, easier to detect errors and consume less bandwidth. This is because link state routing uses event-triggered updates instead of periodic updates as in distance vector.

Source routing

There are two types of Source Routing, strict and loose. If strict routing is used then the sender decides the next exact route in which to pass the packet through the network. The route information is passed in a header that is added to the packet. This technique is rarely used. The more common variant is instead loose source record route (LSRR), in which the sender only determines a few hops that a packet must take to reach its destination. Source routing demands that every node knows the whole network topology. This can be solved in several ways, for instance by keeping a table over the network.

1.3.2 Routing protocol characteristics

In a MANET there are several factors and issues to be addressed. These factors are of big importance and they all depend on scenario we are targeting. For example there are not the same constraints and demands on a MANET for educational purposes. This is also the reason why there is not a single routing protocol to this date that is suitable for all scenarios. We have to take into account the factors that are important in the scenario we are working on. Now some routing protocol characteristics will briefly be explained. The whole chapter will then be concluded with suitable choices of routing protocol characteristics for the scenario.

1.3.2.1 Flat vs. Hierarchical

When deciding between the two architectures, the choice can be made by looking at the key aspects of different approaches. The flat characteristic has the following advantages over the hierarchical:

- Increased reliability and survivability
  - No single point of failure
  - Alternative routes in the network
- More "optimal routing"
- Better coverage, i.e. reduced use of the wireless resources
- Route diversity, i.e. better load balancing property
- All nodes have one type of equipment

No single point of failure means that if one node goes down, rest of the network will still function properly. In the hierarchical approach, if one of the cluster head goes down, that section of the network would not be able to send or receive message to other section for the duration of the downtime of the cluster head.

One thing that ought to be mentioned about the flat algorithm is that it doesn't scale very well. When the network becomes larger than the routing overhead will increase rapidly.

Using a hierarchical instead of a flat has a couple of advantages as well:

- Easier mobility management procedures (just ask the cluster head)
- Better manageability

1.3.2.2 Proactive vs. Reactive

In a proactive routing protocol all the routes to each destination is kept in an up-to date table. Changes in the network topology are continuously updated as they occur. The differences between the protocols are how the changes are spread through the network and how many tables each node maintains.

In the reactive approach a connection between two nodes is only created when it is asked by the source. When a route is found it is kept by a route maintenance procedure until the destination no longer exists or is needed.

Proactive protocol:

Advantages

- A route can be selected immediately without delay

Disadvantages

- Produce more control traffic
- Takes a lot of bandwidth
- Produce network congestion

Reactive protocol:

Advantages

- Lower bandwidth is used for maintaining routing tables
- More energy-efficient
- Effective route maintenance

Disadvantages

- Have higher latencies when it comes to route discovery

Reactive protocol face scaling problem when the number of nodes are larger and have many "active nodes". But how big this problem depends on the protocol we use and which scenario we have.

1.3.2.3 Unicast vs. Multicast

When using multicast routing a single packet is sent simultaneously to multiple recipients, while in unicast routing only a single packet is sent to one recipient in every transmission. Thus the multicast method is very efficient and a useful way to support group communication when bandwidth is limited and energy is constrained.

Due to the broadcast characteristic of the multicast protocol it is better suited for MANET then the unicast protocol.

1.3.2.4 Unipath vs. Multipath

In a routing protocol that has multipath capabilities the packet can be sent on multiple paths between the source and the destination. For multiple paths there is a higher chance that there will be a correct end-to-end transmission for a longer period of time between source and destination, then in a network with a unipath routing protocol. This means that the frequency of finding new routes is not as high, which leads to lesser route discovery traffic.

There are several ways to use the paths. Some protocols only use one path at a time. This means that when a first path is broken the other path is used and when both paths are broken a mutipath discovery procedure is instantiated. Another approach is to make use of both paths at once. Then packets will be sent on both paths at the same time, as in disparity routing.

1.3.2.5 Tree-based vs. Mesh-based

When categorizing multicast routing protocol this can be done into two different categories based on the network structure, the tree-based approach and the mesh-based approach. The two types have different strengths and weaknesses.

The tree-based approach is more bandwidth efficient due to the fact that it uses minimum number of packets for spreading packets to multicast group. It is also more energy efficient since when there is only one source, minimum number of nodes are involved in routing packets. The tree-based approach is good when the mobility rate is low and the tree structure is stable because then the path optimality makes it efficient.

The disadvantages are that link failures cause a reconfiguration of the entire tree since there is only a single path established between two nodes. This makes the tree vulnerable when the mobility is high because there will be more link failures due to constantly changing topology. Also, it is necessary to monitor every branch state information.

The mesh-based approach has several advantages. For instance it provides multiple paths between nodes, which makes it resilient to link failure. This feature also provides good performance when the mobility is high, i.e. it scales well with changing topology. The chance for a packet to reach its destination is very high in the mesh-based approach, i.e. it has a throughput. Multiple paths results in some disadvantages as well. Much bandwidth is wasted because of the fact that every packet is duplicated and sent on many different paths between the nodes.

The multiple paths also result in an increased overhead in order to maintain the forwarding group. The consequences are that a tree-based approach is favored when you want an energy efficient network. A mesh-based approach is better suited in a network that favors high packet delivery ratio and needs robustness to mobility.

1.3.2.6 Quality of services (QoS)

Quality of service is a measurement of how good the routes in the network are. The routes should guarantee a set of pre specified services attributes, such as delivery, bandwidth and delay variance (jitter). For a protocol to provide good QoS it must determine new routes rapidly and with minimal bandwidth consumption. There are several metrics that directly affect the QoS of every protocol. Packet delivery ratio, control packet overhead, average hop count, end-to-end latency of service will greatly affect the MANETs performance.

1.4 Discussion

The size of the network and the number of the nodes participating in the network is of great importance and affects almost every aspect of the choice of protocol characteristics. Our network is a small size network and does not have a lot of participating nodes. The choices of protocol characteristics heavily rely on these parameters and if they are changed the function of the MANET may be jeopardized.

We have found that flat architecture advantaged is better suited for our scenario than the advantages of the hierarchical. *The no single point of failure* is something that can't be accepted in our environment because every message should be able to reach every node at all times. We don't feel that the scalability issue of the flat architecture is a problem for us because our proposal is for a network that isn't especially large. To summarize, we have come to conclusion that the overall performance of the flat architecture makes it the best choice for our scenario.

When deciding how to maintain routing information we favor a Reactive approach even though it produces more congestion. This is because we want to have a protocol that is robust to high mobility and reactive protocols are well suited for mobile as hoc networks, especially when the mobility rate is high.

In our scenario with a highly mobile network with high demands on data delivery we want a protocol with **multiple paths** to ensure high throughput in the network.

The rest of this dissertation is organized as follows. The next chapter evaluates the routing protocol characteristics and traditional routing algorithm (Distributed Bellman-Ford) in Ad Hoc networks, compares it with On- demand protocols with different route selection metrics. It also describes an overview of Ad Hoc network. We extend this work by simulating three protocols, each from various routing approaches. This chapter studies the Ad Hoc routing protocol scalability. SMR is explained in Chapter 3. Chapter 4 conducts the simulation performance evaluation of different simulators and it focuses on our choice of protocol. Chapter 5 concludes the dissertation.

CHAPTER 2

LITERATURE REVIEW

2. A REVIEW OF EARLY ROUTING PROTOCOLS

Bandwidth and power constraints are the main concerns in current wireless networks because multihop, ad hoc mobile wireless networks rely on each node in the network to act as a router and packet forwarder. This dependency places bandwidth, power, and consumption demands on mobile hosts, which must be taken into account when choosing the best routing protocol. In recent years, protocols that built roots based "on-demand" have been proposed. The major goal of on-demand routing protocols is to minimize control traffic overhead. In this section, we perform a simulation and performance study on some routing protocols for ad hoc networks. Distributed Bellman-Ford, a traditional table-driven routing algorithm, is simulated to evaluate its performance in multihop wireless networks. In addition, to on-demand routing protocols Dynamic Source Routing (DSR) [7] and Associativity Based Routing (ABR) with distinctive route selection algorithms are simulated in a common environment to quantitatively measured and compare their performance. We have chosen these three protocols for the following reasons: (i) to evaluate the performance of a conventional table-driven routing scheme (DBF) in multihop wireless networks, and (ii) to study the performance of different routing metrics in dynamic ad hoc networks. The final selection of an appropriate protocol will depend on variety of factors, which are discussed in the section 2.

2.1 Routing Protocols Review

2.1.1 Distributed Bellman-Ford

It is a table-driven routing protocol, i.e. each router constantly maintains an up-to-date routing table with information on how to reach all possible destinations in the network. For each entry, the next router to reach the destination and a metric to the destination is recorded. The metric can be hop distance, total delay, or cost of sending the message. Each node in the network begins by informing its neighbors about its distance from other nodes. The receiving node extracts this information and modifies their routing table if any route measure has changed. For instance, a different path has been chosen as the best route or the metric to the destination may have been altered.

This protocol does not scale well to large networks due to a number of reasons. One problem is the so-called "count-to-infinity" problem. In unfavorable circumstances, it takes up to N iterations to detect the fact that a node is disconnected, where N is the number of nodes in the network. Another problem is the increase of route update overhead with mobility. Mobility can be expressed as rate of link changes and/or router failures. In a mobile network environment, event-triggered routing updates tend to out number the time-triggered updates, leading to excessive overhead an inefficient usage of the limited wireless bandwidth.

2.1.2 Dynamic Source Routing

Dynamic Source Routing (DSR) [7] was developed at Carnegie Mellon University. It is a direct descendant of the source routing scheme used in bridged LANs. It uses source routing instead of hop-by-hop packet routing. Each data packet carries the list of routers in the path. The main benefit of source routing is that intermediate node need not keep route information because the path is explicitly specified in the data packet. DSR does not require any kind of periodic message to be sent, supports uni-directional and asymmetric links, and sets up routes based on demand by the source. DSR consists of two phases: (a) route discovery and (b) route maintenance, which are explained in the following sections.

- *Route Discovery*

When the source has a data packet to send but does not have any routing information to the destination, the source initiates a route discovery. To establish a route, the source floods a ROUTE REQUEST message with the unique request ID. When this request message reaches the destination of a node that has the route information to the destination, it sends a ROUTE REPLY message containing path information back to the source. The "route cache" is maintained at each node. The node records the routes in order to reduce the overheads that are generated by a route discovery phase. When a node receives a ROUTE REQUEST packet, this message is forwarded only if all of the following conditions are met; (a) the node is not the target (destination) of the ROUTE REQUEST packet, (b) the node is not listed in source route, (c) the packet is not a duplicate, and (d) no route information to the target node is available in its route cache. If all are satisfied, it appends its identification to the source route and broadcasts the packet to its neighbors. If condition (b) or (c) is not met, it is simply discards the packet. If a node is the destination of the packet or has route information to the destination, it builds and sends a ROUTE REPLY to the source, as described above.

- *Route Maintenance*

  The main innovation of DSR with respect to bridged LAN routing is in route monitoring and maintenance in the presence of mobility. DSR monitors the validity of existing route based on the acknowledgements of data packets transmitted to neighboring node. This monitoring is achieved by passively listening for the transmission of the neighbor to the next hop or by setting a bit in a packet to request and explicit acknowledgement. When a node fails to receive an acknowledgement, a ROUTE ERROR packet is sent to the original sender, to invoke a new route discovery phase. Nodes that receive a ROUTE ERROR message delete any route entry (from their route cache), which uses the broken link. Note that a ROUTE ERROR message is propagated only when a node has a problem-sending packet through that link. Although this selective propagation reduces control overhead (no packets traversed a link), it yields a long delay when a packet needs to go through a new link.

- *Information Stored in Each Node*

  ➢ **Route Cache:** Each node stores routing information it has learned and overheard in its route cache. Routing information can be obtained while processing ROUTE REPLY messages and the source route list of a data packet header. More than one route for each destination can be stored in the cache. When a ROUTE ERROR message is received or overheard, routes that use the broken link specified in the ROUTE ERROR are removed from the route cache.

- **Route Request Table:** nodes producing a ROUTE REQUEST packet store information in the route request table. Recorded information includes the destination node of a ROUTE REQUEST, the time when the node last sent a ROUTE REQUEST to the destination and the time the node has to wait until it can send a next ROUTE REQUEST to the destination. The purpose of maintaining this table is to restrict frequent ROUTE REQUEST transmissions to the same destination.

2.2 STUDY OF AD HOC NETWORK ROUTING PROTOCOL

We have compiles a list of every routing protocol we have found. The detailed list can be found in Appendix. We cannot say that we have found every protocol because there are many new and different variations of protocols being developed all the time. We spent four weeks compiling this list and reading about the various protocols to get a clear picture about their function, perks and shortcomings. We have divided the protocols into different categories based upon their characteristics [8]. Protocols that we could not put into a specific category are put in the category of "other".

In the remainder of the section, we concentrate on ad hoc routing protocols that have the routing parts as their primary goal.

2.2.1 An Overview of Ad Hoc Routing Protocols

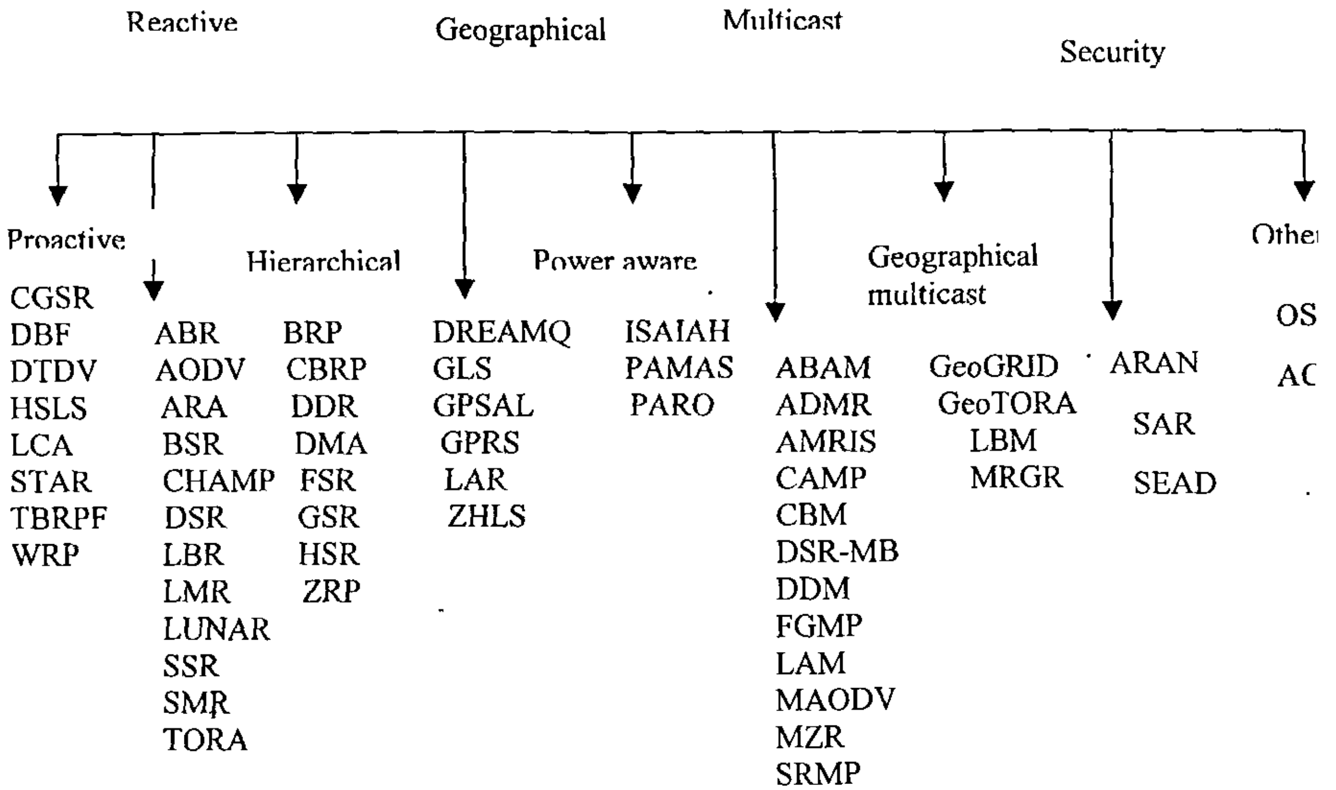


Figure 2.1: Overview of Ad-hoc routing protocols

2.2.2 TCP

One option would be to look at the possibility of using ordinary TCP to handle the routing procedure. The problem that arises when using TCP is that it can't distinguish between packet losses as a result of mobility or lossy channel from packet losses due to network congestion. When a packet loss is detected TCP assumes that it is due to congestion and lower its transmission rate, it can wait longer for the ACK of the current packet being transmitted. If it still doesn't get any ACK it will lower its Retransmission Timeout (RTO) even more. This is called ***exponential back off*** and is one of the reasons why TCP perform so poorly in a mobile ad-hoc network. Thus, in order to use TCP to work in a MANET it has to be adapted so it can detect what causes which loss.

Parameters	TBRPF	WRP	AODV	DSR	ZRP	DREAM	ODMRP	MAODV
Routing Approach	Flat	Flat	Flat	Flat	Hierarchical	Flat/ Geographical	Flat/Mesh based	Flat-Tree based
Routing Scheme	Proactive	Proactive	Reactive	Reactive	Hybrid	Proactive	Reactive	Reactive
Delivery Structure	The next Hop routing	Source routing	The next hop routing	Source routing	The next hop and source routing	Location based flooding or location based next hop routing	Group-based routing	Core-based tree
Loop free	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiple paths	No	No	No	Yes	Yes	Yes	Yes	No
Routing metric	Shortest path	Shortest path	Freshest and shortest path	Shortest path	Local shortest path	Shortest path	Shortest path	Shortest path
Frequency of update	Periodically and as needed	Periodically and as needed	As needed	As needed	Periodically and as needed	Periodically	Periodically and as needed	As needed
Multicast capabilities	No	No	Yes	No	Yes	No	Yes	yes

Table 2.1: Characteristics of chosen protocols

Even if there are protocols with different approaches there are two things that are almost identical. All protocols are loop free and they all use shortest path as their routing metric, except for AODV that uses freshest path in addition to shortest path. There are certain qualities that a routing protocol of our scenario should posses. We want a protocol that uses a reactive approach with multipath capabilities. The pure proactive protocols

TBRPF and WRP fall short in all three cases. Further WRP have high demands on the memory capacity and TBRPF is targeted for a large size network.

The two protocols based on mainly a reactive approach are AODV and DSR. These are all very good routing protocols with support from the IETF (AODV and DSR). But unfortunately AODV suffers from lack of multi path capabilities that we want our protocol should posses.

In our table, ZRP and ODMRP have very similar properties. One thing that separates them is their routing approach. ZRP is hierarchical and ODMRO is flat. We have come to the conclusion that we favor a flat approach in front of a hierarchical. Therefore if the choice stands between SMR and DSR we will choose SMR.

DREAM has an overall good performance except when the nodes are highly mobile. Them there can be problems in having an up to date location. MAODV is a competitive routing protocol that is based on AODV, but we have not chosen it due to the fact that it is a tree-based protocol. Another thing is that MAODV doesn't make use of multiple paths, which makes it less interesting.

2.3 PERFORMANCE EVALUATION OF ADVANCED ROUTING STRATEGIES

In this section, we investigate the performance of routing strategies in Ad Hoc Networks. Routing protocols for Ad Hoc Networks have adopted a variety of approaches. These protocols can be generally classified as: (a) distance vector based; (b) link state based; (c) on-demand; and (d) location based. The first two categories modify a traditional table-driven scheme to adapt to ad hoc networks. On-demand or reactive, routing protocols are proposed specifically for ad hoc networks. These protocols do not maintain permanent route tables. Instead, routes are build by the source on demand. With the advent of GPS (Global Positioning System), protocols that utilize location information to establish routes have been proposed. In this section, we conduct a performance study of routing protocols that represent each routing category. The distance vector based protocol WRP [8], the link state based protocol FSR [10], the on demand routing protocol DSR [7], the location based reactive protocol LAR [11], and the location based proactive protocol DREAM [12] are simulated in a common wireless network simulation platform.

2.3.1 Protocols Review

In this section first we will study different routing protocols used in the previous researches.

2.3.1.1 Wireless Routing Protocol

Wireless Routing Protocol (WRP) [8] is a distance vector based protocol designed for ad hoc networks. WRP modifies and enhances distance vector routing in the following three ways. First, when there are no link changes. Second, to improve reliability in delivering update messages, every neighbor is required to send acknowledgements for updated packets received. Retransmissions are sent if no positive acknowledgements are received within the time out period.

HELLO interval Max allowed HELLO miss	1 sec 4
Update acknowledgement timeout interval	1 sec
Retransmission counter	4
Retransmission counter	1 sec

Table 2.2: Parameter values for WRP

Third, the predecessor node ID information allows the protocol recursively calculate the entire path from source to destination. With this information, WRP substantially reduces looping situations, speeds up the convergence, and is less prone to the "count-to-infinity" problem. Still, temporary loops do exist and update messages are triggered frequently in networks with highly mobile hosts.

Table 2.2 shows the WRP parameter values used in experiments. Values suggested by the designers of WRP [8].

2.3.1.2 Fisheye State Routing

Fisheye state routing (FSR) [10] is a link state type protocol, which maintains the topology map at each node. To reduce the overhead incurred by control packets, FSR modifies the link state algorithm in the following three ways.

First, link state packets are not flooded. Instead, only neighboring nodes exchange the link state information. Second, the link state exchange is only time-triggered, not event-triggered. Third, instead of transmitting the entire link state information at each iteration, the FSR uses different exchange intervals for different entries in the table. As a result, FSR scales well to large network size since link state exchange overhead is kept low. As mobility increases, however, routes to remote destinations may become less accurate. Simulation parameter values for FSR are shown in table 2.3.

Scope		1hop
Hello interval	speed ≤3.5 km/hr	5 sec
	speed > 3.5 km/hr	1 sec
Max allowed Hello miss		3
INTRASCOPE UPDATE INTERVAL	speed ≤3.5 km/hr	5 sec
	speed > 3.5 km/hr	1 sec
INTERSCOPE UPDATE interval	speed ≤3.5 km/hr	15 sec
	speed > 3.5 km/hr	3 sec

Table 2.3: Parameter values for FSR

2.3.1.3 Dynamic Source Routing

Dynamic Source Routing (DSR) [7] is an on-demand routing protocol that builds routes only when necessary. A source floods a ROUTE REQUEST if data to send exist but no route to its destination is known. The ROUTE REQUEST packet records in its header the IDs of the node it traverses. When the destination or a node that knows a route to the destination receives the ROUTE REQUEST, a ROUTE REPLY is sent to the source via the recorded route. Each node in the network maintains a route cache storing routes it has learned over time. DSR uses source routing instead of hop-by-hop routing; the source node appends the list of node ID s that comprises the route in the data header. When a node learns the route is obsolete due to topology changes, it builds and sends ROUTE ERROR to the source. The source then invokes a route recovery process to construct a new route. No periodic message of any kind is required in DSR.

Table 2.4 shows the DSR parameter values used in the implementation. We have implemented some optimization features of DSR.

Time between retransmitted ROUTE REQUESTS	500 msecs
Max time where the same requests can be sent	10 secs
Non propagating ROUTE REQUESTS timeout	30 msecs

Table 2.4: Parameter values for DSR

2.3.2 Routing Protocols Summary

Table 2.5 summarizes key characteristics and properties of the protocols.

Protocols	WRP	FSR	DSR
Routing Strategy	Distance Vector	Link state	On-Demand
Selection Metric	Shortest Path	Shortest Path	Shortest Path
Loop-free	No	Yes	Yes
Periodic Messages	HELLOS	HELLOS, Route Entries	None
Updates Triggered by	Event, Time	Time	Event
Flooding Packets	None	None	RREQs
Routes in Data	No	No	Source Route
Promiscuous Mode	No	No	Yes
Need for GPS	No	No	No

Table 2.5: Summary of protocols

2.4 Ad Hoc Routing Protocol Scalability

As mobile networking continues to experience increasing popularity, the need to connect large number of wireless devices will become more prevalent. Many recent proposals for ad hoc routing have certain characteristics, which may limit their scalability to large networks. This section proposes 4 different combinations of enhancements, which may be incorporated into virtually any on-demand protocol in order to improve its scalability. The scalability of current on-demand routing protocol is evaluated through the selection of a representative from this class of protocols. The performance of un-modified on-demand protocol is compared with each the scalability option. Based on the observations, conclusion is drawn as to the expected scalability improvement, which can be achieved by each enhancement.

2.4.1 Background and Motivation

Recent advances in the portability, power, and capability of wireless devices and applications have resulted in the proliferation and increased popularity of these devices. As a number of users continue to grow, wireless routing protocols will be required to scale to increasingly larger populations of nodes. Conference networking scenarios can require the formation of networks on the order of tens to hundreds of nodes. Furthermore as the deployment of wireless networks becomes more widespread, new applications may encourage the formation of large ad hoc networks. Many of the proposed protocols for ad hoc networks [13,7,14,15,16,17,18] use a broadcast route discovery mechanism whereby route request is flooded across the entire network. While the impact of such route discovery flood may be limited in small networks, the impact will be significantly larger for larger networks. When a link break occurs in an active route, many of these protocols [7,14,16] require an error notification to be sent to nodes that were using that link. Again, for small networks with limited network diameters, this route error message can be propagated back to the source node relatively quick and some repair action is taken. However, as a network diameter and average path length increases, the error message may have to propagate across tons of hop to reach the source node. For such large networks or even smaller networks with rapidly moving nodes, it is likely that the source node will be unable to make a repair before another link in the breaks. Hence, this mechanism may prove ineffective for more stressful scenarios.

This section evaluates the scaling potential of on-demand ad hoc routing protocol by comparing a base routing protocol. With the performance of it combine with various enhancements. The scalability of AODV is investigated by evaluating its performance in networks as large as 10,000 nodes. Three methods for improving the scalability of ad hoc routing protocols are described and integrated into the AODV protocol for their evaluation. The enhancements include an expanding ring search for route discoveries initiated by a source node, a query localization protocol that also attempts to prevent the flooding of route request, and the local repair of link breaks in active routes. Further, the methods for preventing discovery floods are each in turn combined with the local repair mechanism, to yield a total of five possible improvement algorithms.

2.4.2 Enhancements

The scalability of many on-demand routing protocols may be limited due to a couple of important factor. The first is the need for flooding each RREQ. In small networks, flooding the RREQ across the network has a limited impact due to the small number of nodes in the network. As path lengths increase and as node mobility speed rise, the chances of active route breakage increase more frequently. Requiring an error message to be sent to the source node for each link break may result in an overwhelming number of route repairs by the source node. Particularly for high mobility and/or long path lengths, it may be true that the source node barely has time to rediscover a route before that route suffers from another link break.

2.4.3 Observations

In the previous sections, we have studied the scalability characteristics of on demand routing protocols, which are known to generally perform best in mobile multihop networks. We have studied that routing in ad hoc networks of tens of thousands of nodes is extremely difficult. In large networks, path lengths are longer compared with those in small networks (i.e. 50 or 100 nodes). Because network hosts are capable of mobility, longer routes are more prone to disconnection since a single link failure results in a route break.

Each route invalidation invokes a route recovery process and clogs the network with control messages. The unicast RREP packet may not reach to the source due to link break during route discovery. Even when the RREP packet survives to reach the source, the route may break shortly after and the source will need to initiate another route discovery. Therefore, maintaining routes with many hops in mobile ad hoc networks is a difficult challenge.

CHAPTER 3

DESIGN

3. SPLIT MULTIPATH ROUTING WITH MAXIMALLY DISJOINT PATHS

In recent years, routing has been the most focused area in ad hoc networks research. On Demand routing in particular, is widely developed in bandwidth constraint mobile wireless ad hoc networks because of its effectiveness and efficiency. Most proposed on-demand routing protocols however, build and rely on single route for each data session. Whenever there is link disconnection on the active route, the routing protocol must perform a route recovery process. Multiple paths can be useful in improving the effective bandwidth of communication pairs, responding to congestion and bursty traffic, and increasing delivery reliability. In QoS routing in wired networks, multipath routing has been widely developed [19,20,9,21,22,23,24,25]. These protocols used table-driven algorithms (link state [29] of distance vector [30]) to compute multiple routes. Studies show however, that proactive protocols perform poorly because of excessive routing overhead [26,27,28]. Multipath routing in ad hoc networks has been proposed in [3,31,14,32], including the one to be introduced in the previous section. Although these protocols build multiple routes on demands, the traffic is not distributed into multipaths; only one route is primarily used and alternate paths are utilized only when the primary route is broken.

We proposed a routing scheme called Split Multipath routing (SMR) that establishes and utilizes multiple routes of maximally disjoint paths. Multiple routes, of which one is the shortest delay path, are discovered on demand. Established routes are not necessarily of equal length. Providing multiple routes helps minimizing route recovery process and control message overhead. We believe utilizing multiple routes is beneficial in network communications, particularly in mobile wireless network where routes are disconnected frequently because of mobility and poor wireless link quality. Our protocol uses a per-packet allocation scheme to distribute data packets into multiple paths of active sessions. This traffic distribution efficiently utilizes available network resources and prevents nodes of the route from being congested. We evaluate the performance of our scheme by extensive simulation.

3.1 Route Discovery

Split Multipath Routing (SMR) is an on-demand routing protocol that builds multiple routes using request/reply cycle. When the source needs a route to the destination but no route information is known, it floods the ROUTE REQUEST (RREQ) message to the entire network. Because this packet is flooded, several duplicates that traversed through different routes reached the destination. The destination node selects multiple disjoint routes and sends ROUTE REPLY (RREP) packets back to the source via the chosen route.

3.1.1 RREQ Propagation

The main goal of SMR is to build *maximally disjoint multiple pa*ths. We want to construct maximally disjoint routes to prevent certain nodes from being congested and to utilize the available network resources efficiently.

To achieve this goal in on-demand routing schemes, the destination must know the entire path of all available routes. Therefore, we use the source routing approach, which contains the information about route at each node. The RREQ packet contains this route information. Additionally, intermediate nodes are not allowed to send RREPs back to the source even they have route information to the destination. If nodes reply from cache as in DSR [7] and AODV [16], it is difficult to establish maximally disjoint multiple routes because not enough RREQ packets will reach the destination and the destination node will not know the information of the route that is from the cache of intermediate node. When the source has data packets to send but does not have route information to the destination, it transmits RREQ packet. The packet contains the source ID and the sequence number that uniquely identify the packet. When a node other than the destination receives RREQ that is not a duplicate, it appends its ID and re-broadcasts the packet.

3.1.2 Route Selection Method

In our algorithm, the destination selects two routes which are maximally disjoint. More than two routes it can be chosen but we limit the number of routes to two in the study. We have used the shortest delay path as one of the two routes to minimize the route acquisition latency required by on-demand routing protocols. When receiving this first RREQ, the destination records the entire path and sends RREP to the source via this route.

The nodes IDs of the entire path are recorded in the RREP, and hence the intermediate nodes can forward this packet using this information. After this process, the destination waits for certain duration of time to receive more RREQs and learn all possible routes. It then selects a route that is maximally disjoint to the route that is already responded. The maximally disjoint route can be selected because the destination knows the entire path information of the first route and all other candidate routes. If there is more than one route that is maximally disjoint with the first route, the one with the shortest hop distance is chosen. If there still remains multiple routes that meat the condition, the path which delivers the RREQ to the destination more quickly is selected. The destination then sends another RREP to the source via the second route selected. Note that two routes of the session are not necessarily of equal length.

Our protocol uses the source routing scheme in which intermediate nodes do not reply from cache. Only the source node maintains route information to destinations. Each node hence uses less memory but packet header size is larger.

3.2 Route Maintenance

A link of a route can be disconnected because of mobility, congestion, and packet collisions. It is important to recover broken routes immediately to do effective routing. In SMR, when a node fails to deliver the data packet to the next hop of the route (by receiving a link layer feedback from IEEE 802.11 or not receiving passive acknowledgements [33]), it considers the link to be disconnected and sends a ROUTE ERROR (RERR) packet to the upstream direction of the route. The RERR message contains the route to the source and the immediate upstream and down stream nodes of the broken link. Upon receiving this RERR packet, the source removes every entry in its route table that uses the broken link (regardless of the destination). If only one of the two routes of the session is invalidated, the source uses the remaining valid route to deliver data packet.

When the source is informed of a route disconnection and the session is still active, it may use one of the two policies in re-discovering routes:

- Initiates the route recovery process when any route of the session is broken, or
- Initiates the route recovery process only when both routes of the session are broken.

The first scheme reconstructs the routes more often and produces more control overhead than the second scheme, but the former provides multiple routes most of the time and be robust to route break.

3.3 Allocation Granularity

When the source receives a RREP after flooding the RREQ, it uses the first discovered route to send buffered data packets. When the second RREP is received, the source has two routes to the destination, and can split traffic into two routes. We use a simple *per-packet allocation* scheme when there is more than one available route to the destination. One drawback of this scheme is out of order delivery and re-sequencing burden on the destination. We believe, however, that cost-effective reordering buffers are easily implemented. We decided to use the per-packet allocation approach because it is known to work well in most networks, and most of all it is fairly difficult to obtain network condition (such as available bandwidth) in ad hoc networks to apply more sophisticated schemes.

3.4 Simulation Environment

We evaluate and compare the performance of the following protocols:

- **SMR-1**: SMR which performs the route recovery when any route to the destination is invalidated
- **SMR-2**: SMR which performs the route recovery when both routes to the destination are invalidated
- **DSR**: Dynamic Source Routing [7] which uses single path

We have implemented the simulator within Network Simulator (NS-2). Our simulation modeled a network on 10 mobile hosts placed randomly within a 500 meter * 500 meter area. Each node has a radio propagation range of 250 meters and channel capacity was 1Mbps. Each run executed for ten seconds of simulation times.

3.5 The Multipath On-demand Routing Protocol

3.5.1 Network Model

The network is modeled as a set of nodes, each with an address, a sequence number, a cost, a routing table, and a queue. Every node is uniquely identified by an address. Sequence numbers are used to detect duplicate packets, and are useful for controlling network floods. As in distance-vector routing, the cost of the node is added to the cost field of each packet a node receives. The total cost of a route is then measured by adding the cost of all the nodes, which received the route message. The current implementation of MOR uses a fixed cost of 1 for each node. The cost of nodes could be set based on network conditions such as a node being low on energy and/or the observation of congestion.

3.6 Protocol

This section defines the MOR protocol.

3.6.1 Packet Headers and Organization

All headers have a common prefix with the type of the packet and a sequence number,

$$CM\ N \equiv (type, seq).$$

The valid types are

- gradient forming (GRAD),
- gradient return (RET),
- generic route control (RT),
- no route (NR), and
- data (DATA).

3.6.2 Sequence Number Generation

Each node has a current sequence number ($node_{seq}$), and subsequent numbers are generated by incrementing the current sequence number by 1. The purpose of the sequence number is to detect if a received packet is new, old, or duplicate, with higher numbers corresponding to newer packets. The use of sequence numbers is discussed later in route discovery.

3.6.3 Node Cost

Costs are associated with nodes rather than links in our model. Each node keeps a cost ($node_{cost}$) as part of its state information. The cost of a node is added to route packet's cost field while being forwarded by that node. This cost is 1 in experiments, but could be used to increase the cost of a node in case of low energy or congestion.

3.6.4 Route Discovery

MOR discovers routes in two ways, actively through route control messages, and passively by observing traffic passing by.

3.6.4.1 Active Route Discovery

Routes in on-demand routing protocols are typically discovered via a network flood. While MOR also makes use of a network flood to discover routes, it takes measures to minimize the number of network floods necessary for packet delivery.

Gradient Construction

If node A wants to communicate with node B, and node A does not have a route to node B, then A will broadcast a route control (RC) message where

RC ≡ (type, seq, ret Cost, cost, src, dest, prevHop),

- type=GRAD
- seq=++A_{seq} I
- retCost=0
- cost=0
- src=A

- dest=B
- prevHop=A

This initial packet and any forwarded version 2 of the initial packet will be referred to as A_{GRAD} . Each node {x|x = B} which receives A_{GRAD} message will forward a RC with

- cost=A_{GRAD}.cost+x_{cost},
- prevHop=x, and
- other fields have their values set to values in the received A_{GRAD},

Providing there are no entries in the routing table of x with destination A_{GRAD}.src. Otherwise, multipath logic is applied.

Route Entry

Each node x which does not ignore a received (RC) message constructs a route entry (RE) for the route back to RC.src:

RE ≡ (dest, seq, cost, nextHop, lastUsed),

- dest=RC.src
- seq=RC.seq
- cost=RC.cost+x_{cost}
- nextHop=RC.prevHop
- LastUsed=t_{now}

The lastUsed field is used for *least recently used* routing, and as an age indicator for when an unused route is removed after $T_{maxRouteAge}$.

Multipath Logic

If a node receives RC with src=A and already has a route entry RE for A, it will apply the following multipath logic

- ignore it if the cost is higher (RC.cost > RE.cost),
- ignore it if the message is old (RC.seq < RE.seq),
- delete the old REs and treat the message as new if the cost is lower (RC.cost < RE.cost),
- construct another RE if the cost is same and nextHop is different

(∀RE : RC.seq = RE.seq and RC.cost = RE.cost and RC.prevHop ≠RE.nextHop).
The exact multipath logic can be represented by the matrix in table 3.1, where action (1) is to replace all the existing routes to A (by deleting them) with this route, action (2) is to drop the packet and ignore the RC, and action (3) is to add this route to the routing table if it is a multipath (no other RE has RE.nextHop = RC.prevHop and RE.dest = RC.src).

If previous routes were replaced, as in action (1), then forwarding will apply depending on the type. See the *gradient construction* and *back trace reply* sections for forwarding of GRAD and RET packets respectively.

Table 3.1: Multipath Logic

Cost/Age	RC.seq> RE.seq	RC.seq = RE.seq	RC.seq < RE.seq
RC.cost < RE.cost	purge/add (1)	purge/add (1)	ignore/drop (2)
RC.cost = RE.cost	purge/add(1)	add if multipath (3)	ignore/drop (2)
RC.cost > RE.cost	purge/add (1)	ignore/drop (2)	ignore/drop (2)

Backtrace Reply

After the A_{GRAD} message reaches node B, B will send a route control B_{RET} as a reply to A, with RC ≡ (type, seq, retCost, cost, src, dest, prevHop),

- type=RET,
- seq=++Bseq,
- retCost= A_{GRAD}.cost,
- cost=0,
- src=B,
- dest=A, and
- prevhop=B.

This message propagates back along the gradient using the return cost (retCost) to restrict itself to the shortest return paths, and create routes to B. More precisely, nodes {x|x = A} which receive B_{RET} and for which B_{RET}.

retCost > RE_A.cost will broadcast RC with

- retCost= RE_A .cost

- cost= B_{RET} . cost+x_{cost},
- prevHop=x, and
- other fields have their values set to values in the received B_{RET}, where RE_A is any route entry in the routing table of x with destination A. If B_{RET}.retCost $\leq$ RE_A.cost, B_{RET} is ignored. In place of forwarding, multipath logic is applied if there exist any route entries of x with destination B. A route entry with destination B will be created if B_{RET} was not ignored by any of the above.

3.6.4.2 Passive Route Discovery

Besides network floods, MOR also discovers routes by observing traffic. The data header (DH) in MOR provides useful routing information:

$$DH \equiv (\text{type, seq, cost, src, dest, prevHop}).$$

A data packet from node A, forwarded by neighbor node B, can be used to construct a route to A if the route does not exist. The route entry can be constructed similarly as with route control,

- dest=DH.src
- seq=DH.seq
- retCost=DH. cost+x_{cost}
- nextHop=DH.prevHop
- lastUsed=t_{now},

where x is the receiving node. Return paths to nodes which initiated a network flood could be used if a route is later required to those nodes, provided the routes have not yet timed out. While this mechanism is limited to discovering routes taken from data packets passing through the node, it does not require any significant cost such as *promiscuous mode* 3. Passive route discovery discovers routes without the need to broadcast control packets, and is one of the features of MOR which minimize the use of network floods.

Where packets are not filtered based on their MAC destination address. In non-promiscuous mode, the network interface card, after examining the packet header, will normally halts the receiver if the packet is not addressed to the host. Promiscuous mode **results** in an increase in energy usage, as reception energy as well as energy required to process the packet by the CPU.

3.6.5 Routing and Load Balance

To route a data packet M, a node looks for all REs with dest = M.dest. The data packet could be forwarded to any of these REs. For load balancing, different REs should be chosen for each consecutive data packet routed to a certain destination. Any number of schemes may be used: round robin, random RE, or least recently used. MOR currently uses the *least recently used* scheme. Assuming RE_i has the smallest lastUsed value, the data packet is then *unicast* to RE_i .NextHop and lastUsed is updated to the current time.

When a node A initiates a network flood to B, multiple routes are formed, so node A will not need to execute another network flood to find B unless all the paths breaks. Since the nodes forwarding the network flood now have one or more routes to A, they will not need to execute a network flood to find A. If A is the base station in a sensor network, none of the nodes sending data to A will need a network flood. In the best case scenario in which all data gathered are destined for the base station, a single network flood can set up all necessary routes.

3.6.6 Reliability

Another benefit of having multiple routes at each node is increased hop-by-hop reliability. Should a packet fail to transmit with a route entry RE_i, and another entry RE_j exists such that RE_i.dest = RE_j .dest, then the packet could be retransmitted using RE_j. RE_i is then put on pro- bation, and dropped from the table if it causes a number of failures. In this way, congested nodes do not immediately break routes, and routes break less often than for other protocols, leading to longer lasting routes and therefore fewer network floods. Also packet delivery is overall more reliable, giving higher performance for the protocol. Retransmission to an alternate route is observed to be helpful when certain nodes drop packets due to congestion. Other multipath protocols, such as

AOMDV[34], use disjoint paths. Immediate retransmission to congested nodes is a bad idea, since they may still be congested, but in a disjoint path, intermediate nodes do not have a choice of where to forward a given packet. In MOR, each node in a path may have a choice of next hops.

3.6.7 Route Maintenance

While routing data packets, each containing a data header (DH), failure to transmit may eventually remove all routes to DH.dest. MOR will advertise this loss of all routes by use of no route (NR) packets,

$$N R \equiv (type, seq, src, dest)$$

If a node A has removed its last route to destination B due to failure to transmit, A will Broadcast 5 no route packet A_{NR}, with

- type=NR,
- seq=A_{seq},
- src=A,
- dest=DH.dest.

Each node {x | x = B} which receives A_{NR} will search its routing table for a RE_i with

- dest= A_{NR}.dest, and
- nextHop= A_{NR}.src If RE_i does not exist, A_{NR} will be ignored and dropped.

Otherwise, RE_i is removed and x will search its routing table for E_j with the destination A_{NR}.dest. If RE_j exists, then the route is given to node A via unicasting a route control (RC) with RC ≡ (type, seq, retCost, cost, src, dest, prevHop),

- type=RT,
- seq=RE.seq,
- retCost=0,
- cost=RE.cost,
- src=RE.dest,
- dest= A_{NR}.src, and
- prevHop=x,

otherwise the node x has no other route to A_{NR} .dest and x will re-broadcast A_{NR} with A_{NR}.nextHop=x. If node B receives A_{NR} , B will respond with a "I am here" message, which is an RC with

- type=RT,
- seq=Bseq ,
- retCost=0,
- cost=0,

- src=B,
- dest= A_{NR}.src, and
- prevHop=B.

The RC used in route maintenance is a non-propagating route message (type=RT). Since no forwarding is done on RT packets, route entry construction is applied if there are no route en- tries with dest=RC.dest, or multipath logic applies.

3.7 Comparison

Table 3.2 is a summary of the energy and time comparison between DSR, AODV, and MOR. A look at the energy comparison without idle energy involvement may be worthwhile, since idle energy could just be a direct function of the time performance.

Table 3.2: Protocol Comparison to MOR

	MOR	DSR	AODV
Dense (% compared to MOR)			
Completion time	100 %	714 %	216 %
Energy usage	100%	207 %	134%
Sparse (% compared to MOR)			
Completion time	100 %	877 %	287%
Energy usage	100%	255 %	153 %

3.7.1 Idle Energy

If the routing protocol fails to deliver a packet, TCP may back off, resulting in idle time. TCP back offs result in increased energy usage and decreased time performance. It seems obvious that the idle energy usage involved would be at least partly dependent on the time performance, as for more time the system is running, the more idle time would result.

Table 3.3: Protocol Comparison to MOR without Idle Energy

	MOR	DSR	AODV
Dense			
Receive (rx) + transmit (tx) energy Rx + tx (% compared to MOR)	10704 % 100%	13836 % 130 %	13026 % 122%
Sparse			
Completion time	10252 %	13900 %	13041%
Energy usage	100%	136 %	127 %

3.7.2 Energy Without Idle

Table 3.3 shows energy comparisons between MOR, AODV, and DSR without idle energy involvement. With just transmit (tx) and receive (rx) energies considered, all three protocols used approximately the same amount of energy between sparse and dense topologies. This similarity in energy use between the dense and sparse topologies is probably just a coincidence, since one should also note that the transmit energy is higher for the sparse topology while the receive energy is higher for the dense topology. Since the sparse topologies contain longer routes, more transmissions would be required to deliver the data, resulting in higher transmission energy usage. Receive energy may be higher in the dense topology because transmissions reach a larger number of neighbors. While the increase in idle energy usage is a direct consequence of the amount of time needed to complete the *task*, the receive and transmit energies are due to the actions of the routing protocol involved. The tx and rx energies are still partly affected by TCP because if TCP retransmit packets, tx+rx energy would increase. However, retransmissions are due to packet drops, and packet drop frequency depends on the routing protocol used. Another reliable transport protocol with back offs geared toward wireless communication would still need to retransmit lost data. The point is, tx+rx energy is independent of TCP backoff, and is generally due to actions taken by the routing protocol used.

CHAPTER 4

DEVELOPMENT

4. SIMULATION ATTEMPT

In order to test and validate our results the idea is to perform simulations that compare different aspects of the performance of SMR in comparison with some security protocol like DSR. The performance aspects that we want to evaluate are packet delivery ratio, route acquisition time, latency, traffic byte overhead and other QoS metrics. When conducting these simulations we expect how much the security protocols would affect the overall performance of SMR. Will it generate a large amount of extra overhead and congestion that the QoS is severely degraded? If that's the case, is this degradation of QoS acceptable in comparison with the gained security.

Due to different factors a simulation that test and validate our choice of protocols are not conducted. Instead it handles our survey of possible simulations environments for simulating our routing protocols and then addresses different factors that make the simulation impossible.

4.1 Simulation programs

Several simulation-programs are available. We have performed a survey of the commonly used simulators ns2, GloMoSim, QualNet and OPNET in order to determine the most suitable simulator for our scenario. The study is based on the following criteria's:

- Which protocols does it support?
- How well is it documented?
- Is it complicated to install?
- How frequent is the simulator used in research-papers regarding MANETs?
- Can the existing code be extended in any way?
- How user-friendly is it?

4.1.1 NS2

This simulator is probably the most commonly used software of the four. NS2 stands for the Network Simulator 2 and is developed by ISI, the Information Sciences Institute at the USC School of engineering. The source code can be downloaded, free of charge, and is compiled on different platforms, e.g. Unix and Windows. An extensive manual for the installation and use of the software on the ns2 homepage is also available. Other people have also put tutorials for this program on the Internet.

The software is text-based and might therefore be a bit complicated to use if you are not familiar to Unix-commands. Some parts are managed with GUIs, which makes it easier to understand what is happening.

Many different extensions to this software are developed by various researchers. Many wireless extensions have been contributed by the UCB Daedelus, the CMU Monarch projects and Sun Microsystems. The documentation to these extensions is not always extensive and the developers of ns do not always support them. In NS2 it is possible to alter and write our own code to make it more suitable for our own scenarios.

The most recent version of ns2 is ns-2.26 which is released on the 26 of February 2003 and supports AODV, DSDV, DSR and TORA. If we want to simulate on other protocols there are extensions that support ADMR, AODV+, AODV-UU, Ariadne, MAODV, ODMRP, SEAD and ZRP.

4.1.2 GloMoSim

GloMoSim stand for Global Mobile information systems Simulation library and supports protocols for a purely wireless network. It was developed at UCLA Parallel Computing Laboratory (UCLA PCL) and is intended for academic institutions for research purposes. It is only possible to download the current version, GloMoSim 2.0 (December 2000), from the GloMoSim homepage if you are within the edu domain. If commercial users want to use GloMoSim they have to obtain the commercial version called QualNet. This version is extended in some areas. In order to get GloMoSim to work you have to install Parsec, which is a C-based simulation language developed by PCL at UCLA. There is very little documentation of the installation procedure. Either it is very easy to install or it is poorly documented.

In any case, if we would run into trouble while installing we would not get much help from the documentation. Also the documentation of how to use the software is poorly described.

GloMoSim support some protocols, which lies in our interest. These are AODV, DSR, Fisheye, LAR, ODMRP and WRP. If we want to develop our own protocols in GloMoSim, it is possible. But to do so we should have some familiarity with Parsec. Although the code to the protocols will be written purely in C code, with some Parsec functions for time management, we will need to use the Parsec compiler.

Although we have read some papers in which GloMoSim have been used, it's not as frequently used as ns2.

4.1.3 QualNet

QualNet is developed by SNT (Scalable Network Technologies) and is network simulation software. SNT claims that we can use QualNet when we design a network or network device to optimize, saving time and money.

The QualNet software consists of five tools plus integration modules and model libraries. QualNet Animator allows for graphically designing the network library (using a wide library of components) and can be used to display the simulation as it runs or later on. QualNet Designer is for streamline code development. We can generate code for our own protocol from scratch and make special statistic reports. We can also make adjustments to the already made protocol models. QualNet Analyzer is a graphic tool that presents statistics of different experiments in graphs.

4.2 Choice of Simulator

When choosing a simulation program the question of what we want to simulate and which resources we have to conduct these simulators are of great importance. What we want to simulate should work in any of the four simulators, if we have the right tools and knowledge. But due to our restrictions and the available back up provide for us the choice is quite limited.

QualNet and OPNET are well-developed commercial software products and should be easier to use than the other two. The problem is that they cost a lot! We haven't got the financial support to buy these products. This is an important factor since we haven't got the time and knowledge to implement the protocol or write the simulation code on our own.

GloMoSim is available for downloading only if our IP address resolves to an academic domain name. The documentation is not very good and it seems to be hard to get any kind of support. Even though some papers have used GloMoSim to simulate MANET protocols, the questions how to validate and compare our results with other works are an issue of concern. Developing of new software for GloMoSim also seems to be quite sparse.

Ns2 is free to download and researchers for simulating mobile ad hoc networks commonly use it. It has an extensive manual and some support in the mailing list. New features are developed continuously and added functions for protocols are available for downloading.

ODMRP is available, either in the base-installation or with some extension, in ns2, GloMoSim and QualNet. OPNET on the other hand has no known support for ODMRP and no extensions for it has been developed.

None of the four simulation programs are currently including the possibility to simulate the security protocols. If we want to simulate them we have to implement them on our own, writing our own source code. There are projects that have used the simulation programs for simulating their security routing protocol, like the Monarch Project that has implemented Ariadne in ns2.

Altogether the choice of simulation program is quite clear, ns2 provides the best overall solution for our purpose.

4.3 Problems with the simulation

We were not at the present date able to get the source code from any of the security protocols LHAP, SMT, SRP or TESLA. To try and write our own source code for any of these protocols is out of question? We did not have the kind of knowledge or time to acquire that knowledge.

The problems do not stop there. Now comes the part of installing ns2 with different extensions that are necessary. The different extensions that we had to install were not as compatible with ns2 as we would have liked. Not only that but different extensions were supported by different versions of ns2. But after much work we got a fairly good installation to work with. The problem with the installation probably arises because there are so many different things to install to get ns2 up and running. Also when the authors of the extensions developed their code they probably altered the original ns2 code in order to get their extensions to work. Then when they presented their work they might not include some of these changes, which make it almost impossible to install their extensions in a problem free manner.

The next step is to simulate the routing protocols. This provided to be an equally challenging task. How to accurately simulate the protocols is greatly affected by different parameters that we define in our simulation code. To know how to accurately define these parameters, we must have extensive knowledge about both the routing protocol we want to simulate and ns2 in general. One way to get around this could have been to take the parameters from already done simulators and use them in our work. This also is not possible because even though other people have performed detailed simulations, how they define the parameters in their simulation code is not well documented.

To perform a simulation with a result that can provide something to our paper proved to be an impossible task with our circumstances. The problems that we have to deal with can be concluded in some points:

- The difficulty installing it.
- Documentation exists for ns2 but it is not very thorough when it comes to more challenging simulations.
- The amount of parameters that could affect the simulation is defined wrongly.
- Researchers tend to make the simulation they've done to work in just their special case, it is hard to reproduce or use their results due to lack of documentation.
- How to parse the simulation output to get correct data for our metrics?
- How to validate our results?

The simulation could probably be done, but not without spending a lot of time in studying source codes for ns2 and the routing protocols. However, it was not possible for us due to our time frame.

4.4 PROTOCOL USED – Split Multipath Routing (SMR)

With multiple streams, more advanced usages are possible. One of the examples is the layered coded video streaming. The video content consists of the base layer and the enhancement layers. The base layer data without which reconstruction fails completely is transmitted through the most reliable and robust path and the enhancement layer data is transmitted through the other path. In this case, the receiver which has a cellular network access and a wireless LAN access can always receive the base layer through the cellular network. It also receives the enhancement layer when being within wireless LAN coverage areas and thus improves the video quality.

To split data to multiple links correctly, multi-stream senders need much more knowledge about the receiver's links as well as their own links. However there is no protocol which allows multi-stream senders to realize the characteristics of receiver's links sufficiently. Without such link information, the ability of multi-link transport can be limited.

This section presents some example multi-link transport applications and the scenarios.

Bandwidth Aggregation

A simple advantage of multipath transport is bandwidth aggregation. It is achieved by striping data across the multiple interfaces of the mobile host. The simplest striping scheme is round robin striping, where the sender sends packets in round robin order on the paths. However, since different paths have different characteristics in terms of bandwidth, delay, and loss rates, most of data striping schemes are performed in consideration of these parameters.

Congestion control mechanisms based on the receiver's acknowledgement can adjust the appropriate data rate to the path condition and estimate the bandwidth. Striping data over multiple TCP connections needs the resequencing to manage the buffer bundling multiple connections. If one of the receiver's interfaces becomes disabled, the TCP connection using this interface may stall the other TCP connections. Although decoupling the loss recovery from congestion control avoid this state, the link up/down notification from the receiver would be useful also.

Requirements

This section describes the requirements of multi-stream transport. Multi-stream senders can obtain the characteristics of the correspondent's links involved in the connection. As described in the problem statement, efficient multi-link transport needs much more knowledge about correspondent's links. Multi-stream senders should note that depending on the last-hop characteristics is sub optimal or even may cause the negative impact adversely.

Data path can be selected based on the application data priority. Some multi-stream applications prioritize application data and intend to split high-prioritized data and low-prioritized data into different paths. Data path is determined by choosing a pair of source and destination addresses.

Multi-stream senders can detect status changes of the correspondent's links involved in the connection as soon as they happen. Wireless link states are time-variant and come under the influence of terminal movement. Status changes may affect the path selection and multi-stream senders have to readjust the data path for each application data.

Link Information

The link information describes that multi-stream_senders may require multi-link transport. There are two types of link information. One is link property which is static information and the other is link status which is dynamically changeable and time-variant. Using detail information specific to a particular link at the upper layer may be layer violation and reduce the flexibility of the layer_principle. Therefore, the link information has to be represented by abstracted parameters for the upper layers.

Link Property

- ***Link Bandwidth***

This information is necessary to realize the maximum data rate for each link. For striping data across multiple links, this information is primary hint for adjusting how much amount of data for each link.

- *Link Delay*

This information helps delay-constraint multi-stream applications. If some pieces of data are real-time constraints but others are not, real-time constraint data can be transmitted through the faster and the other data is transmitted through the slower link. For example, video and audio data is real-time constraint but timed-text data is not so real-time constraint. Another advantage is reordering the video frame to be transmitted.

- *Link Robustness*

This information indicates the robustness of the links against terminal movement. Wired links never suffer from the variation in quality as long as connected. Cellular links such as GPRS links cover large areas, so the short movement of the host does not really affect the link status. Small-cell wireless links such as wireless LAN links covers the small areas (usually inside buildings) and even short movement may affect the link status.

- *Link Reliability*

This information indicates whether the link layer provides reliability through the use of retransmission. It is relevant to signal strength. When the link layer does not provide reliability, a low signal link may suffer from the high loss rate. On the other hand, when a link layer provides reliability, a low signal link may suffer from additional link delay caused by retransmission.

Link Status

- *Signal Strength*

The signal strength parameter is specific to wireless link and mainly used for the handover mechanisms. For example, the mobile node which has two wireless LAN links chooses the receiving interface with higher signal strength. When both links have a weak signal strength, bicasting can be required.

Closing Statements

While this work is initially focusing on dual wireless terminals equipped with a cellular access link and a Wireless LAN access link, the draft does not limit the case and it needs solutions that can accommodate any combination of all kinds of links. After reading about various kinds of protocols we have chosen to look at SMR because it's the best match regarding our criteria. SMR provides multiple paths, which is beneficial when it comes to dealing with problems. Simulations indicate that protocols that provide multiple paths perform well under mobility. The protocol has been evaluated in several papers and the general conclusion is that SMR is the best routing protocol in the multipath area.

We present split multi path routing protocol that builds maximally disjoint paths. Multiple routes of which one is the shortest delay path are discovered on demand. Established routes are not necessarily of equal length. Data traffic is split into multiple routes to avoid congestion and to use network resources efficiently.

4.4.1 Characteristics

- Route discovery
    - RREQ propagation
    - Route selection method
- Route maintenance
- Allocation Granularity

SMR is an on-demand routing protocol that builds multiple routes using request/reply cycles. When source needs a route to the destination but no route information is known it floods the route request (RREQ) message to the entire network. Because this packet is flooded, several duplicates that traversed through different routes reach the destination. The destination node selects multiple disjoint routes and sends ROUTE REPLY (RREP) packets back to the source via the chosen route.

The main goal of SMR is to build maximally disjoint multiple paths. We want to construct maximally disjoint routes to prevent certain nodes from being congested and to utilize the available network resources efficiently.

In our algorithm, the destination selects two routes that are maximally disjoint. More than two routes can be selected but we limit the number of route to two in this study. One of the two routes is the shortest delay route; the path taken by the first RREQ the destination receives. We use the shortest delay path as one of the two routes to minimize the route acquisition latency required by on-demand routing protocols.

A link of a route can be disconnected because of mobility, congestion and packet collisions. It is important to recover broken routes immediately to do effective routing in SMR when a node fails to deliver the data packet to the next hop of the route.

4.5 MULTIPATH CLASSIFIER

This is an object, devise to support equal cost multipath forwarding, where the node has multiple equal cost routes to the same destination, and will like to use all of them simultaneously. This object does not look at any field in the packet. With every succeeding packet, it simply returns the next filled slot in round robin fashion. The definitions for this classifier are shown below.

CLASSIFIER HEADER FILE:

```
#ifndef ns_classifier_h
#define ns_classifier_h
#include "object.h"
class Packet;
class Classifier : public NsObject {
public:
    Classifier();
    virtual ~Classifier();
    inline int maxslot() const { return maxslot_; }
    inline NsObject* slot(int slot) {
        if ((slot >= 0) && (slot < nslot_))
        return slot_[slot];
        return 0;
}
    inline int mshift(int val) { return ((val >> shift_) & mask_); }
    inline void set_default_target(NsObject *obj) {
```

```
        default_target_ = obj;
        }
    virtual void recv(Packet* p, Handler* h);
    virtual NsObject* find(Packet*);
    virtual int classify(Packet *);
    virtual void clear(int slot);
    enum classify_ret {ONCE= -2, TWICE= -1};
    virtual void do_install(char* dst, NsObject *target) {
    int slot = atoi(dst);
    install(slot, target); }
    int install_next(NsObject *node);
    virtual void install(int slot, NsObject*);
    // function to set the rtg table size
    void set_table_size(int nn);
    // hierarchical specific
    virtual void set_table_size(int level, int nn) {}
    protected:
        virtual int getnxt(NsObject *);
        virtual int command(int argc, const char*const* argv);
        void alloc(int);
        NsObject** slot_;      /* table that maps slot number to a NsObject */
        int nslot_;
        int maxslot_;
        int offset_;           // offset for Packet::access()
        int shift_;
        int mask_;
        NsObject *default_target_;
        int nsize_;       //what size of nslot_ should be
        };

        #endif
```

CLASSIFIER _MPATH .CC FILE

```
#include "classifier.h"
class MultiPathForwarder : public Classifier {
public:
        MultiPathForwarder() : ns_(0) {}
        virtual int classify(Packet*) {
        int cl;
        int fail = ns_;
        do {
                cl = ns_++;
                ns_ %= (maxslot_ + 1);
            } while (slot_[cl] == 0 && ns_ != fail);
        return cl;
            }
        private:
            int ns_;
            };

        static class MultiPathClass : public TclClass {
        public:
            MultiPathClass() : TclClass("Classifier/MultiPath") {}
            TclObject* create(int, const char*const*) {
            return (new MultiPathForwarder());
            }
            } class_multipath;
```

4.4 MULTIPATH ROUTING USING NS-2

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows
$ns color 1 blue

Node set multiPath_ 1
$ns rtproto DV

#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
$ns trace-all [open all.tr w]

#Define a 'finish' procedure
proc finish {} {
global ns nf
global downTimes
$ns flush-trace
#Close the trace file
close $nf
#Execute nam on the trace file
exec nam out.nam &
exit 0
}
```

```
#Create ten nodes with color and shape
set n0 [$ns node]
$n0 shape "circle"
$n0 color "pink"
set n1 [$ns node]
$n1 shape "hexagon"
$n1 color "purple"
set n2 [$ns node]
$n2 shape "hexagon"
$n2 color "purple"
set n3 [$ns node]
$n3 shape "hexagon"
$n3 color "purple"
set n4 [$ns node]
$n4 shape "hexagon"
$n4 color "purple"
set n5 [$ns node]
$n5 shape "hexagon"
$n5 color "purple"
set n6 [$ns node]
$n6 shape "hexagon"
$n6 color "purple"
set n7 [$ns node]
$n7 shape "hexagon"
$n7 color "purple"
set n8 [$ns node]
$n8 shape "hexagon"
$n8 color "purple"
set n9 [$ns node]
 $n9 shape "circle"
$n9 color "pink"
```

```
#Create links between the nodes
$ns duplex-link $n0 $n3 1Mb 100ms DropTail
$ns duplex-link $n0 $n2 1Mb 100ms DropTail
$ns duplex-link $n1 $n3 1Mb 100ms DropTail
$ns duplex-link $n2 $n3 1Mb 100ms DropTail
$ns duplex-link $n2 $n4 1Mb 100ms DropTail
$ns duplex-link $n3 $n4 1Mb 100ms DropTail
$ns duplex-link $n4 $n5 1Mb 100ms DropTail
$ns duplex-link $n4 $n6 1Mb 100ms DropTail
$ns duplex-link $n5 $n6 1Mb 100ms DropTail
$ns duplex-link $n5 $n8 1Mb 100ms DropTail
$ns duplex-link $n6 $n7 1Mb 100ms DropTail
$ns duplex-link $n6 $n9 1Mb 100ms DropTail
$ns duplex-link $n7 $n8 1Mb 100ms DropTail
$ns duplex-link $n7 $n9 1Mb 100ms DropTail
$ns duplex-link $n8 $n9 1Mb 100ms DropTail

# setup TCP connections
set tcp1 [new Agent/TCP]
$tcp1 set class_ 1
$ns attach-agent $n0 $tcp1

set sink1 [new Agent/TCPSink]
$ns attach-agent $n9 $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set type_ FTP

#Schedule events for the FTP agents
$ns at 0.5 "$ftp1 start"
```

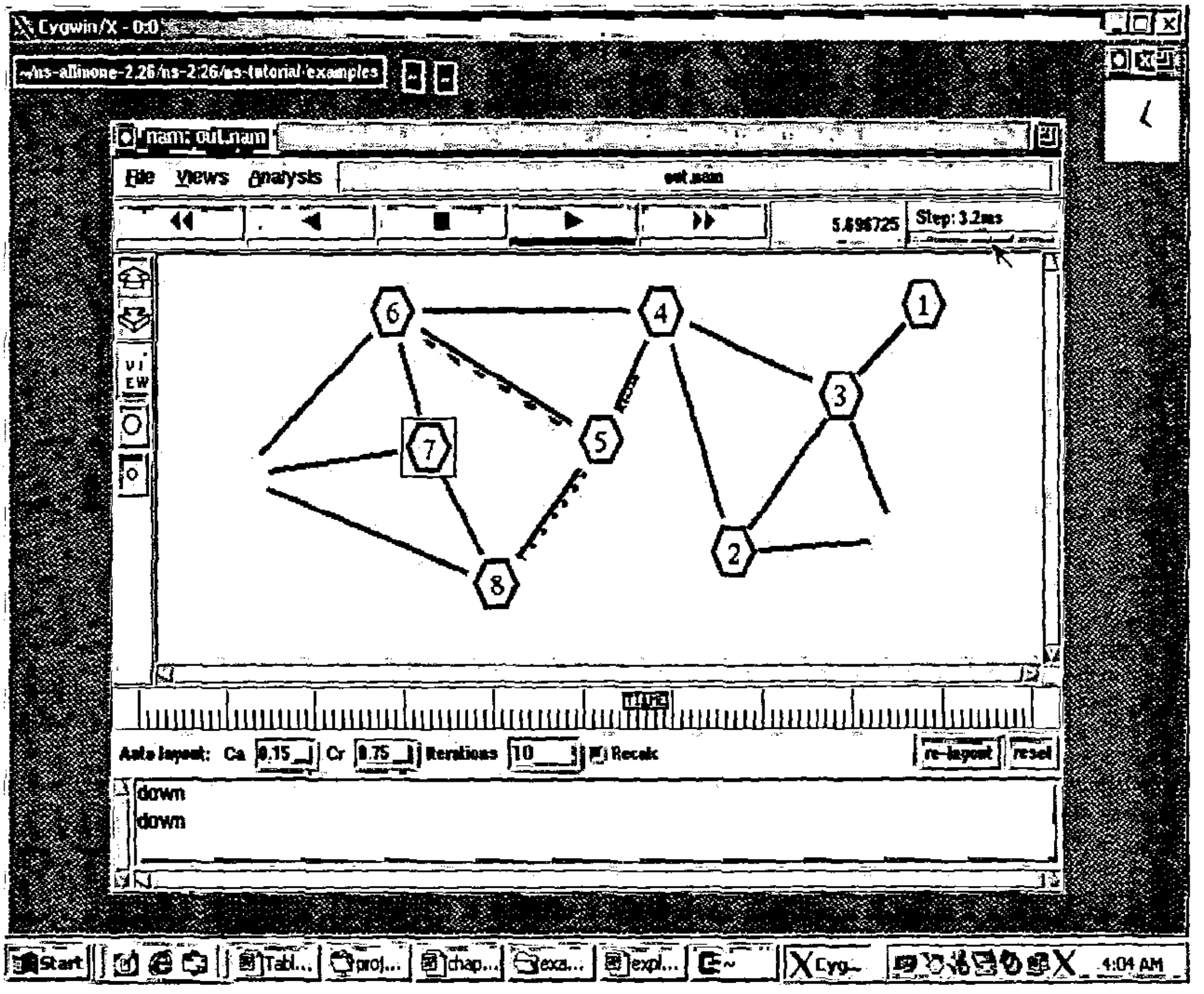
```
$ns rtmodel-at 2.0 down $n4 $n6
$ns rtmodel-at 6.0 up $n4 $n6
$ns rtmodel-at 6.5 down $n6 $n9
$ns rtmodel-at 9.0 up $n6 $n9

$ns at 9.5 "$ftp1 stop"

#Call the finish procedure after 5 seconds of simulation time
$ns at 10.0 "finish"

#Run the simulation
$ns run
```

SCREEN SHOT OF RUNNING NS PROGRAM:



CHAPTER 5

RESULTS & CONCLUSION

5. Result

To get a reliable group communication in mobile ad hoc network, the transmissions have very high demands on delivery rate and transmission time. This makes the function of the routing protocol critical. In mobile ad hoc network we also have the demand of high information security. To design a protocol that satisfies both demands fully is difficult, especially in a changing environment like a MANET.

From the extensive study it is concluded that SMR combined with some security protocols are the solution that best satisfies the qualities required for the mobile ad hoc network scenario.

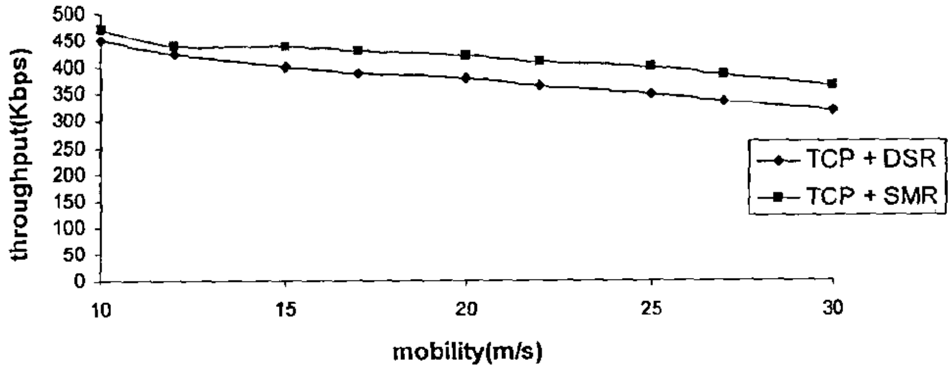
To perform a simulation of SMR with some security protocol is a challenging task. Some of the source codes are hard to obtain, if possible, and requires understanding of the source code if they are to be used. The simulation program ns2 has an extension that implements SMR among the security protocols. To conduct a simulation with SMR and another security protocol is an overwhelming task and out of the scope of this thesis. In order to conduct a simulation extensive familiarity with ns2 is needed. Also the implementations of the protocols require much problem solving in order to function. A simulation with useful results is due to these facts not completed.

Until now there is not a single solution that we have found for our scenario, which fully satisfies all the requirements for a reliable group communication in a mobile ad hoc network.

5.1 SIMULATION RESULTS

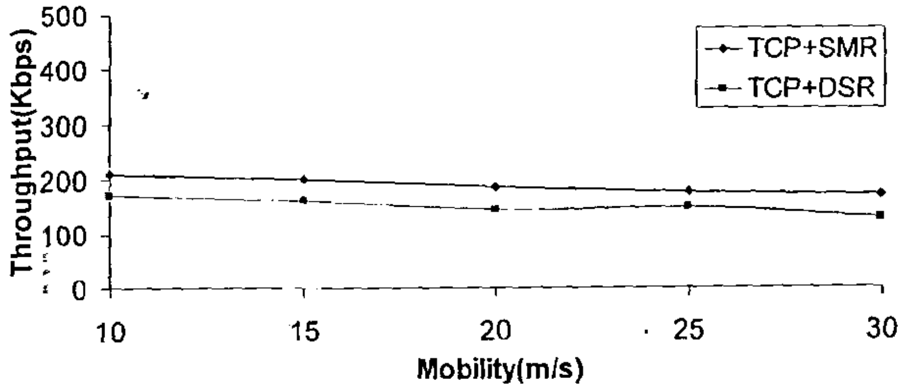
We simulate different scenarios with different traffic patterns. Our results show that when TCP is using multiple paths, it always behaves better than using only single path in all investigated scenarios. Study indicates that SMR outperforms DSR because multiple routes provide robustness to mobility. The performance difference becomes evident as the mobility degree increases.

TCP Performance while utilizing 2 paths simultaneously.

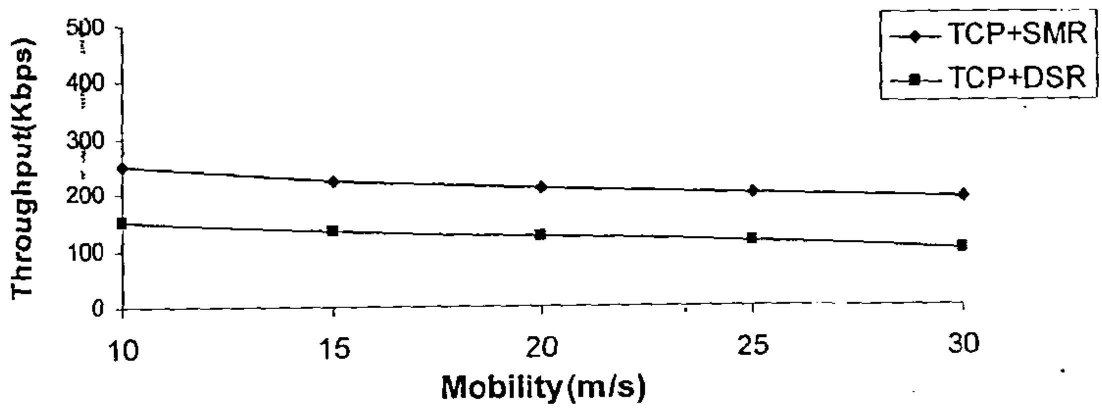


The overall throughput consistently implies that use of SMR over multiple paths provide prominent benefit to TCP performance. SMR also showed shorter end-to-end delay.

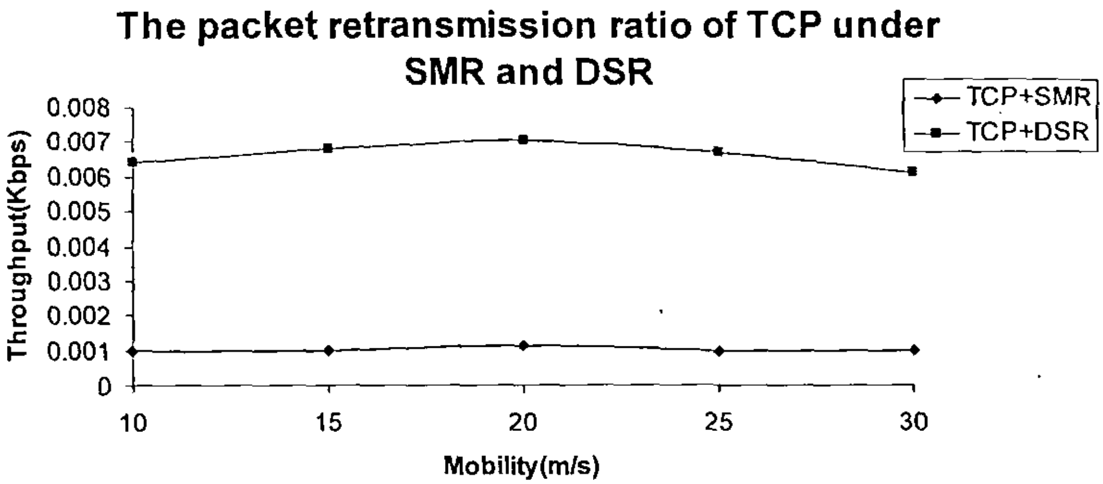
TCP Throughput when 1 FTP/TCP AND 4 CBR/UDP connections are used



Average TCP Throughput when 5 FTP/TCP connections are used



Clearly the packet retransmission of TCP over SMR is much lower than that of DSR. SMR had considerably fewer packet drops compared with DSR.



5.1.1 Packet Delivery Ratio

Packet delivery ratio is obtained by dividing the number of data packets correctly received from the destinations by the number of data packets originated from the sources. We can observe from the result that both SMR (i.e. SMR-1 and SMR-2) schemes outperform DSR, especially when the mobility increases (i.e., the pause time decreases). In DSR, only when route is used for cached session and when that route is invalidated the source uses each route that is learned from overhearing packets. If no such route is available it sends a RREQ to discover a new route. In the latter case, intermediate nodes that have cached routes to the destinations provide those routes to the source by sending RREPs. DSR however, does not apply any aging mechanism for cached route entries, and hence routes stored in the cache (either by the source or intermediate nodes) may be staled. After a route break source nodes will used these newly acquired but obsolete routes only to learn that they are also invalid, and will attempt another route recovery. Many data packets are dropped during this process and more delay is needed to discover correct routes. Between SMR protocols, SMR-2 delivers more packets than SMR-1. We can analyze that the control packets generated by route rediscovery processes of SMR-1 cause collision and contention with data packets. Even though SMR-2 will have only one available route to the destination after the other route is broken, it can still deliver the data packets without producing control traffic as long as the remaining route stays connected and that leads to a good throughput performance.

Both data and control packets are measured. The reasons for packet drops can be incorrect route information, mobility, collisions, and congestions. DSR cannot maintain precise routes and drops more packets as nodes moves more often (i.e., less pause time). The usage of state routes from cache is the major reason of DSR packets drops. Both SMR schemes have considerably fewer packet drops compared with DSR. SMR-2 has fewer packet drops because it invokes fewer route recovery processes and consequently, transmits less control messages.

5.1.2 Control Overhead

Normalized routing load is the ratio of the number of control packets propagated by every node in the network and the number of data packets received by the destination nodes. This value hence represents the protocol efficiency. When there is no mobility DSR has the smallest value. This result is expected because SMR protocols generate more control packets while building multiple routes. On the other hand, DSR builds single route for each session and minimizes flooding overhead by allowing intermediate nodes of replying from cache.

Cached routes are useful in static networks as they remain valid for the entire duration. As mobility is increased, however, SMR-2 shows better efficiency than DSR. DSR yields fewer overheads in initial route discovery process, but it invokes more route reconstruction procedures than SMR-2 since DSR intermediate nodes often reply with stale routes. Additionally, DSR transmits considerably more RERR packets than SMR schemes because the former has more route disconnections and route recoveries. Furthermore, DSR sends RERR packets whenever a unicast packet (data, RREP, and RERR) fails to be delivered to the next hop. SMR sends RERR only when the data packet is undeliverable. Therefore, DSR shows higher normalized routing load than SMR-2 when mobility is present. We can also observe that SMR-1 shows less efficiency than other protocols regardless of mobility. Since the source floods the network with RREQs when any route of a session is disconnected, more control packets are transmitted than

DSR and SMR-2. We can deduce from this result that excessive flooding makes the protocol inefficient.

5.1.3 Hop Length

DSR has the shortest hop distance when there is no mobility because SMR schemes' second routes may have longer distance than the first routes. With mobility however, the hop distance of DSR grows and become larger than those of SMR protocols. If the route is established directly from the destination, it can be the shortest route since it is built based on the most recent information and accounts for node locations after movements. DSR, however, uses cached routes from intermediate nodes. These routes may not be fresh enough and do not exploit the current network topology. DSR therefore builds longer routes than SMR protocols. Longer paths have better chance of having route breaks since one-link disconnection results in route invalidation.

5.1.4 Delay

DSR has the longest delay in mobile scenarios because it delivers data packets on routes longer than those of SMR. In addition, DSR yields longer delays in reconstructing routes and the period of time the data packets are buffered at the source node during route recovery results in larger end-to-end delays. SMR on the other hand, uses the remaining valid route when one of the multiple route is disconnected, and hence no route acquisition latency is required.

5.2 CONCLUSION

The size of the network and the number of nodes participating in the network is of great importance and affects almost every aspect of the choice of protocol characteristics. Our network is a small size network and doesn't have a lot participating nodes. The choices of protocol characteristics heavily rely on these parameters and if they are changed the function of the MANET may be jeopardized.

We have found that flat architecture advantaged is better suited for our scenario than the advantages of the hierarchical. *The no single point of failure* is something that can't be accepted in our environment because every message should be able to reach every node at all times. We don't feel that the scalability issue of the flat architecture is a problem for us because our proposal is for a network that isn't especially large. To summarize, we have come to conclusion that the overall performance of the flat architecture makes it the best choice for our scenario.

When deciding how to maintain routing information we favor a Reactive approach even though it produces more congestion. This is because we want to have a protocol that is robust to high mobility. The reactive protocols are well suited for mobile ad hoc networks, especially when the mobility rate is high.

In this scenario, a highly mobile network with high demands on data delivery, wants a protocol with **multiple paths** to ensure high throughput in the network. A **high QoS** is always desirable in a network but not always a priority.

For a small sized network, a reactive protocol with multipath capabilities is best approach. The protocol should be able to send information through multiple paths to ensure the throughput of the network. In situations that demand a high quality of service, a protocol that ensures the quality of the network is of great importance.

There are many categories and different aspects to take into account when we choose a protocol. But from our extensive list we narrow it down to eight routing protocols.

SMR (Split Multipath Routing Protocol) is one of the multipath protocols which we find the most suitable for our research. SMR is an extension of DSR.

The choices of different protocols were made accordingly:

- Proactive - TBRPF and WRP
- Reactive – AODV and DSR
- Hierarchical – ZRP
- Geographical – DREAM
- Multicast – MAODV and ODMRP

We do not just look at the properties in the table when we make our choices of protocol. Our choice is based on the overall functionality that the protocol can provide. Considering all aspects we come to the conclusion that SMR is best suited for our scenario.

We have conducted performance study of four protocols that represents various routing categories. Overall, all protocols perform much better with the group mobility model than with the random waypoint model. WRP and FSR, especially, are the main beneficiaries of the group movement model. Each protocol's performance degrade as mobility rates increase, but DREAM is the most robust to the speed of the network hosts. However, because of the data flooding, DREAM became less effective under heavy traffic scenarios. On-Demand protocols are highly effective and efficient in most scenarios. Extra delay in acquiring routes, though, make them less attractive in delivering real-time traffic.

In, summary there is no single routing strategy which is best for all network situations. Every protocol has its advantages and disadvantages in different situations. The choice of a routing protocol is made carefully after considering every aspect which we have provid in this chapter.

We present the Split Multipath Routing (SMR) protocol for ad hoc networks. SMR is an on-demand protocol that builds maximally disjoint routes. Our scheme uses two routes for each session; the shortest delay route and the one that is maximally disjoint with the shortest delay route. We attempt to build maximally disjoint routes to avoid having certain links from being congested, and to efficiently utilize the available network resources. Providing multiple paths is useful in ad hoc networks because when one of the routes is disconnected, the source can simply use other available routes without performing the route recovery process.

We introduced two approaches in SMR route maintenance. The first scheme builds a new pair of routes when any existing route of the session is disconnected. The second scheme performs rerouting only when both routes are broken. Our study indicates that SMR outperforms DSR because multiple routes provide robustness to mobility. The performance difference becomes evident as the mobility degree increases. SMR has considerably fewer packet drops compared with DSR, Splitting the traffic into multiple routes helps distribute the load to the network hosts. SMR also showed shorter end-to-end delay because route acquisition latency is not required for all route disconnections. Between SMR protocols, the second scheme showed better efficiency as it performs fewer route recoveries and hence generates less control overhead.

REFERENCES AND BIBLIOGRAPHY

REFERENCES & BIBLIOGRAPHY

[1] S.J.Lee, J.Hsu, R.Hayashida, M.Gerla and R.Bagrodia, "Selecting Routing Strategies for Your Ad Hoc Networks", Technical Report, Computer Science Department, University of California, Los Angeles, October 1999.

[2] S.J.Lee, E.M.Royer and C.E.Perkins, "Ad Hoc Routing Protocol Scalability", Technical Report, Computer Science Department, University of California, Los Angeles, July 2000.

[3] S.J.Lee and M.Gerla, "AODV-BR: Backup Routing in Ad Hoc Networks", Proceedings of the IEEE Wireless Communications and Networking Conference, Chicago, IL, September 2000.

[4] S.J.Lee, M.Gerla and C.C.Chiang, "On Demand Multicast Routing Protocol", Proceedings of the IEEE Wireless Communications and Networking Conference, New Orleans, LA, September 1999.

[5] S.J.Lee, W.Su and M.Gerla, "Wireless Ad Hoc Multicast Routing with Mobility Prediction", ACM/Baltzer Mobile Networks and Applications, Special issue on Routing and Multicasting in Wireless Networks, 2000, to appear.

[6] S.J.Lee, W.Su and M.Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks", Internet Draft, January 2000, work in progress.

[7] D.B.Johnson and D.A.Maltz, "Dynamic Source Routing in Ad Hoc Wireless Network", In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996.

[8] S.Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", ACM/Baltzer Mobile Networks and Applications , special

issue on Routing in Mobile Communications Networks, vol.1, no.2, October 1996.

[9] S.Murthy and J.J. Garcia-Luna-Aceves, "Congestion-Oriented Shortest Multipath Routing", Proceedings of the IEEE Conference on Computer Communications, San Francisco, CA, March 1996.

[10] G. Pei, M.Gerla and T.W. Chen, "Fisheye State Routing : A Routing Scheme for Ad Hoc Wireless Networks, " Proceedings of the IEEE International Conference on Communications (ICC), June 2000.

[11] Y.B.Ko and N.H. vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks", Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Dallas, TX, October 1998.

[12] S.Basagni, I.Chlamtac, V.R.Syrotiuk and B.A. Woodward, "A Distance Routing Effect Algorithm For Mobility (DREAM), Proceedings of the ACM/IEEE International Conference On Mobile Computing And Networking, Dallas, TX, October 1998.

[13] G.Aggelou and R.Tafazolli "RDMR: A Bandwidth-Efficient Routing Protocol For Mobile Ad Hoc Networks", Proceedings of the International workshop on Wireless Mobile Multimedia, Seattle, WA, August 1999.

[14] V.D.Park and M.S.Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Proceedings of the IEEE Conference on Computer Communications, Kobe, Japan, April 1997.

[15] M.R.Pearlman and Z.J.Haas, "Determining the Optimal Configuration for the Zone Routing Protocol, "IEEE Journal On Selected Areas in Communications, special issue on Wireless Ad Hoc Networks, vol.17, no.8, August 1999.

[16] C.E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing", Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), February 1999.

[17] R.Siva Kumar, P.Sinha and V.Bharghavan, "CEDAR: a Core-Extraction Distributed Ad Hoc Routing Algorithm, IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks, vol.17, no.8, August 1999.

[18] C.K.Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks", Wireless Personal Communications Journal, special issue on Mobile Networking and Computing Systems, Kluwer Academic Publishers, vol.4, no.2, March 1997.

[19] J.Chen, P.Druschel and D.Subramanian, "An Efficient Multipath Forwarding Method", *Proceedings of IEEE* International Conference On Computer Communication (INFOCOM), March 1998.

[20] J.Cidon and R.Rom and Y.Shavitt, "Analysis of multipath Routing", IEEE/ACM Transactions on Networking, vol 7, December 1999.

[21] R.Ogier, V.Rutenburg and N.Shacham, "Distributed Algorithms for Computing Shortest Pairs of Disjoint Paths ", IEEE Transactions on Information Theory, vol.39, no.2, March 1993.

[22] D.Sidhu, R.Nair and S.Abdallah, "Finding Disjoint Paths in Networks", Proceedings of the ACM/SIGCOMM Symposium on Communications Architectures, Protocols and Applications, Zurich, Switzerland, September 1991.

[23] N.Taft-Plotkin, B.Bellur and R.Ogier, "Quality-of-Service Routing Using Maximally Disjoint Paths", Proceedings of the IEEE International Workshop on Quality of Service , London, UK, June 1999.

[24] S.Vutukury and J.J. Garcia-Luna-Aceves, "An Algorithm for Multipath Computation Using Distance-Vectors with Predecessor information", Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN) , Boston, MA, October 1999.

[25] W.T.Zaumen and J.J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations", Proceedings of the IEEE Conference on Computer Communications (INFOCOM) , San Francisco, CA, March 1998.

[26] J.Broch, D.A.Maltz and D.B.Johnson, Y.C.Hu, and J.Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Proceedings of the ACM/IEEE International Conference On Mobile Computing And Networking, Dallas, TX, October 1998.

[27] P.Johanson, T.Larsson, N.Hedman, B.Mielczarek, and M.Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks", Proceedings of the ACM/ IEEE International Conference on Mobile Computing and Networking (MOBICOM), August 1999.

[28] S.J.Lee and M.Gerla, and C.K.Toh, "Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks", IEEE Network, vol.13, no. 4, July/August 1999.

[29] J.M.McQuillan, I.Richer and E.C.Rosen, "The New Routing Algorithm for the ARPANET", IEEE Transactions on Communications, vol. COM-28, no.5, May 1980.

[30] G.S.Malkin and M.E.Steenstrup, "Distance Vector Routing", In Routing in Communications Networks, Prentice Hall, 1995.

[31] A.Nasipuri, R.Castaneda andS.R.Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks", ACM/Baltzer Mobile Networks and Applications, 2000.

[32] J.Raju and J.J. Garcia-Luna-Aceves, "A New Approach to On-demand Loop-Free Multipath Routing", Proceedings of the IEEE International Conference on Computer Communications and Networks, Boston, MA October 1999.

[33] J.Jubin and J.D.Tornow, "The DARPA Packet Radio Network Protocols", Proceedings of the IEEE, vol. 75, no.1, January 1987.

[34] Asis Nasipuri and Samir R.Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks", In 18th Annual IEEE International Conference on Computer Communications and Networks (ICCCN), Boston, 1998.

This message is not flagged. [Flag Message - Mark as Unread]

Date: Fri, 14 May 2004 09:54:12 GMT
To: stauseef@yahoo.com
Subject: Acceptance of Paper
From: "Dimitri Yoripov" <dimitri@oitj.com> | This is not spam | Add to Address Book

Dear Dr S.Tauseef Ur Rehman
After the evaluation by the committee, the paper described *below has been accepted. You are now required to submit* the camera ready copies. Please make sure that you meet the deadlines set for the tasks. The details of the paper are:

1. Analysis of Multipath Splitting for Transmission Control Protocol in Hybrid Environment, Adila Arif, Ummara Naseem, S.Tauseef Ur Rehman, Khalid Rashid

Naseem, S.Tauseef Ur Rehman, Khalid Rashid

In order that the notifications be published, please email the camera ready copies through your university email address.

Regards

Dr. Dimitri Yoripov
Technical Editor
Oxbridge Information Technology Journal

Delete | as attachment

Previous | Next | Back to Messages   Save Message Text

Check Mail Compose   Mail Upgrades - Search Mail - **Mail Options**

Mail - Address Book - Calendar - Notepad

Address Book · Auctions · Autos · Briefcase · Calendar · Careers · Chat · Classifieds · Finance · Games · Geocities · Greetings · Groups · Kids · Mail · Maps · Member Directory · Messenger · Mobile · Movies · Music · My Yahoo! · News · PayDirect · People Search · Personals · Photos · Radio · Shopping · Sports · TV · Travel · Warehouse · Weather · Yellow Pages · more...

ANALYSIS OF MULTIPATH SPLITTING FOR TRANSMISSION CONTROL PROTOCOL IN HYBRID ENVIRONMENT

Adila Arif
adilaarif@hotmail.com

Ummara Naseem
sabooha4@hotmail.com

Dr.Khalid Rashid
drkhalid@iiu.edu.pk

Dr.S.Tauseef-ur-Rehman
stauseef@yahoo.com

Department of Computer Science, International Islamic University Islamabad, Pakistan.

Abstract:

In this paper, we present a new and simple idea for evaluating TCP performance in MANET over multipath. Multipath is useful in improving the effective bandwidth of communication pairs, responding to congestion and bursty traffic, and increasing delivery reliability. Previous research mostly uses UDP traffic for performance evaluation. We focus on how to improve path availability to TCP connections, namely by using multipath routing. Multipath routing improves the path availability. We proposed an on-demand routing scheme called Split Multipath Routing that establishes and utilizes multiple routes of maximally disjoint paths. Our protocol uses a per-packet allocation scheme to distribute data packets into multiple paths of active sessions. We evaluate the performance of our scheme using extensive simulation.

Introduction:

Most cellular wireless systems operate in buildup areas where there is no direct line of sight and radio path between terminal, the transmitter, and the receiver. This may cause external interference, jamming, node mobility, unpredictable radio medium and multiple access contention. Among them, frequent link breakage is one of the major factor degrading TCP performance. TCP sender will encounter continuous packet losses over an extended period due to frequent link breakages. Route re-computation takes a finite amount of time. During this time no packet can reach the destination through the existing route. Packets and ACKs may get queued and possibly dropped. In turn leads to timeouts at the source, which is misinterpreted as congestion. A solution to this problem is to detect the link failures and freeze the TCP state until a new connection is established. Example schemes include Explicit Link Failure Notification (ELFN) [1] and TCP-F [2]. Both ELFN and TCP-F rely on intermediate nodes to report the link breakage. All the schemes mentioned above are targeting at preventing TCP from wrongly reacting to packet losses due to link failures. However, if link failures happen frequently, TCP will suffer performance degradation even when the above schemes are applied. To overcome this problem, another solution is to improve the path availability using multipath routing. Previous research was made using single path to the destination while multipath maintains several paths to the destination simultaneously. So far from the best of our knowledge, no detailed investigation of TCP performance over multipath routing protocols have been reported in the literature.

In this paper, we study TCP performance with an on-demand multipath routing protocol named Split Multipath routing (SMR), which is an extension of Dynamic Source routing protocol (DSR).

RELATED WORK:

Recently, TCP performance in ad hoc wireless networks [4] has become an active research field. Link failures due to mobility have been identified as one of the major factors degrading TCP performance. To combat this problem, Holland and Vaidya proposed Explicit Link Failure Notification Scheme (ELFN) whereby the intermediate nodes notify the TCP sender when a link failure happens [1]. With the help of ELFN, TCP senders can tell whether a packet loss is due to link breakage or congestion. ELFN does yield higher throughput in most cases. In ELFN the TCP timer are frozen until the network layer informs TCP that a new route has been found to the destination. TCP sender receives ELFN it send probes. It leaves "stand by" mode on receiving an acknowledgement for a probe. If time interval between probe packets is greater then the route discovery is slower. If it is smaller, then it causes congestion. In TCP-F [2], Chandran and Prakash proposed a scheme, very similar to

ELFN by asking the intermediate node to notify the TCP sender about the network condition. TCP-F uses route failure notification (RFN) packet to inform the source when route is disruption and route re-establishment packet inform source when route is reestablished. In case of TCP-F protocol, intermediate node detects the route disruption. It explicitly sends RFN to source and records the event. Each intermediate node that receives the RFN invalidates the particular route. If it knows an alternate route, that route is used for further communication and RFN is discarded else RFN is propagated towards source. One of the intermediate nodes that had previously forwarded RFN learns about new route.

Another more serious problem that link failure may cause is exponential back off of retransmission timeout (RTO) interval. In the conventional TCP protocol, when a retransmission timeout happens, TCP sender retransmits the lost packet and doubles RTO. This procedure is repeated until lost packet is acknowledged. Such an exponential back off of RTO helps TCP react to congestion gracefully. Wrongly applied exponential back off significantly degrades the TCP performance.

Dyer and Boppana [6] proposed a mechanism called fixed-RTO to solve this problem. When the retransmission timeout happens consecutively, the authors think that it is mainly due to route break not congestion. Thus after retransmitting the lost packet, fixed RTO will freeze the RTO value until the route is reestablished. However ELFN requires support from intermediate nodes while fixed RTO is purely end-to-end mechanism.

The above related work mostly focuses on letting TCP detect route failures and react to them in a proper way. In this paper we focus on how to improve the path availability to TCP connections, namely by using multipath routing.

SIMULATION ENVIRONMENT AND PROTOCOL MODEL:

We proposed a routing scheme called Split Multipath routing (SMR) that establishes and utilizes multiple routes of maximally disjoint paths. SMR is an extension to DSR [3]. When the source needs a route to the destination but no route information is known, it floods the ROUTE REQUEST (RREQ) message to the entire network. Because this packet is flooded, several duplicates that traversed through different routes reached the destination. The destination node selects multiple disjoint routes and sends ROUTE REPLY (RREP) packets back to the source via the chosen route. We want to construct maximally disjoint routes to prevent certain nodes from being congested and to utilize the available network resources efficiently. When the source is informed of a route disconnection and the session is still active, it may use one of the two policies. In the first scheme, it initiates the route recovery process when any route of the session is broken. Another alternative way is to only perform the route recovery when both routes of the session are broken.

SIMULATION PLATFORM:

This paper is basically, a performance study based on simulations. The simulation platform used is NS2 [7]. In all simulation experiments, 10 mobile hosts are placed randomly within a 500m * 500m area. Each node has a radio propagation range of 250 meters and channel capacity was 1 Mb/s. Each run executed for 10 seconds of simulation time.

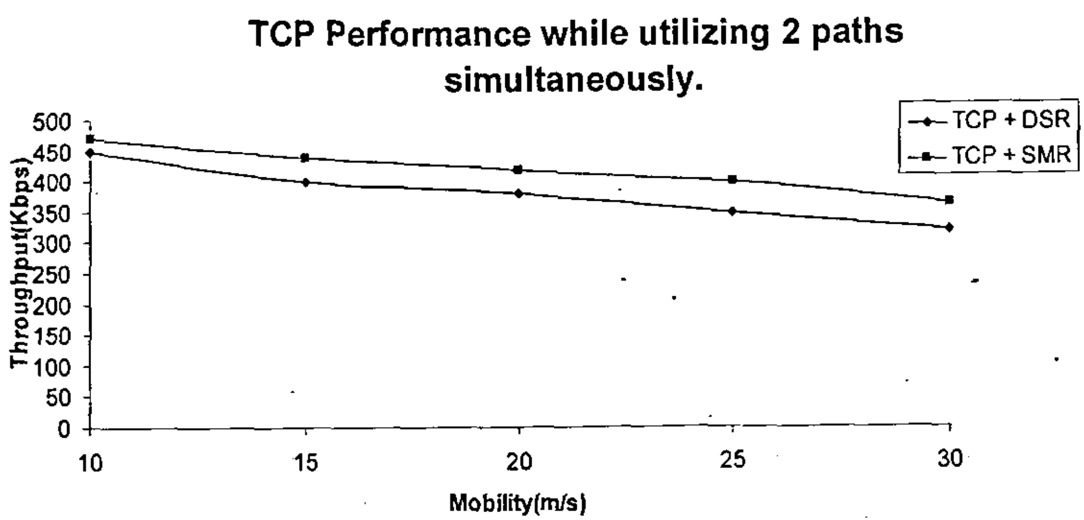


Fig 1 (a): TCP Performance while utilizing 2 paths simultaneously

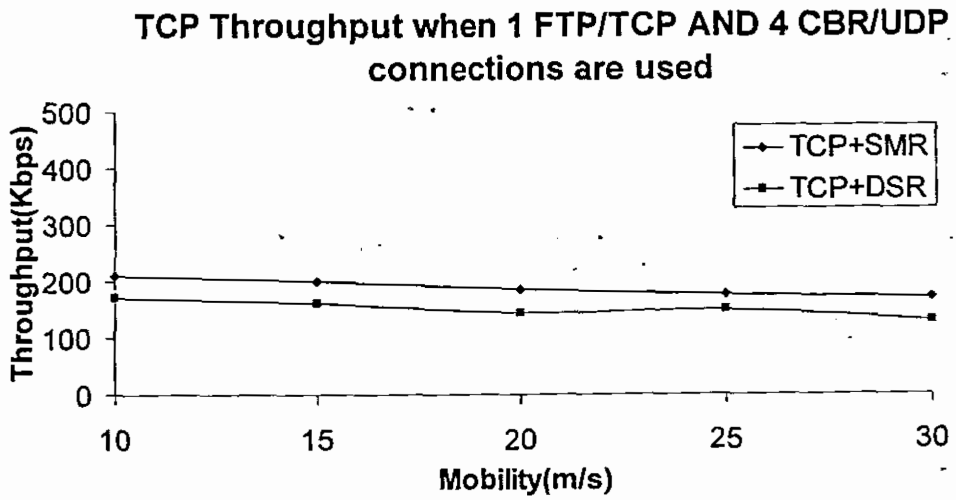

TCP Throughput when 1 FTP/TCP AND 4 CBR/UDP connections are used
500
400
300
200
100
0
Throughput(Kbps)
TCP+SMR
TCP+DSR
10
15
20
25
30
Mobility(m/s)

Fig 1(b):

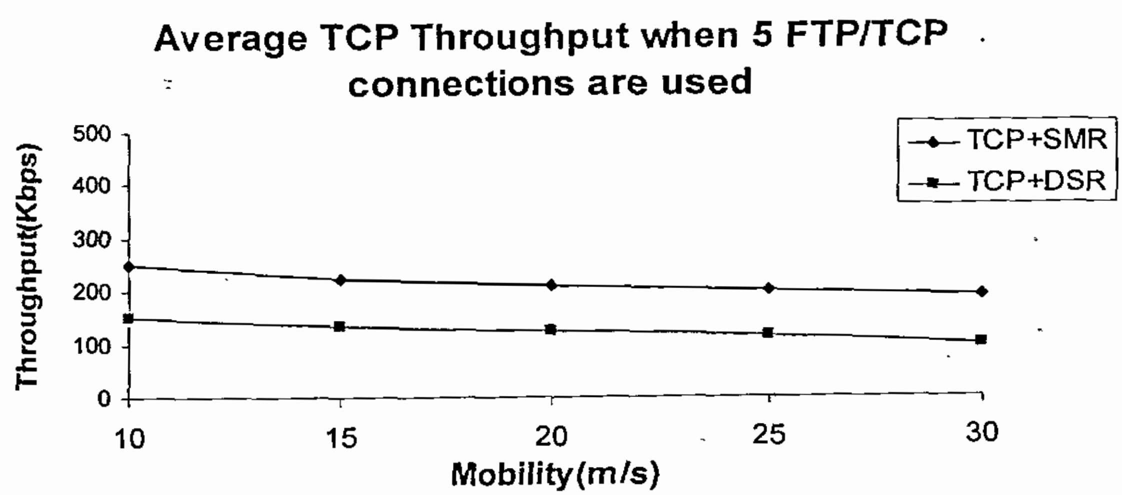

Average TCP Throughput when 5 FTP/TCP connections are used
500
400
300
200
100
0
Throughput(Kbps)
TCP+SMR
TCP+DSR
10
15
20
25
30
Mobility(m/s)

Fig 1 (c):

We simulate different scenarios with different traffic patterns such as 1 TCP connection, 1 TCP + 4 CBR/UDP connections and 5 TCP connections. The TCP Maximum Segment Size (MSS) is set to 1000 bytes and we use FTP application for generating TCP traffic

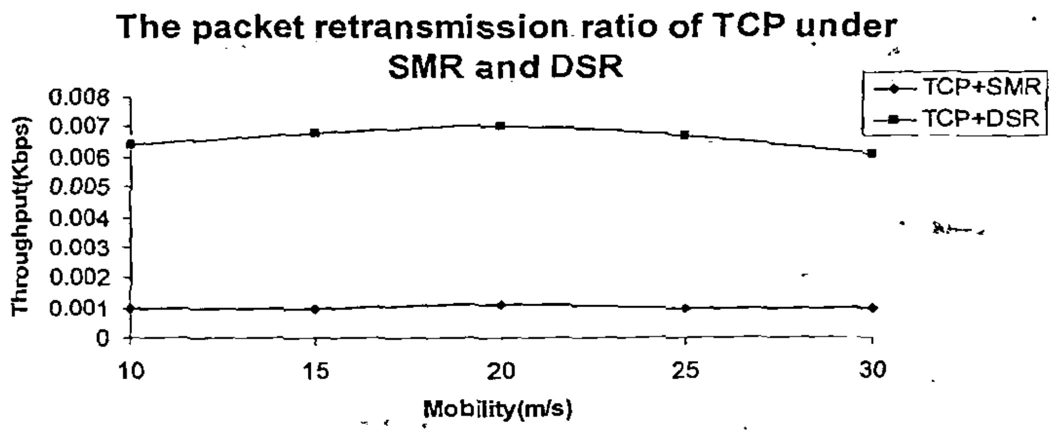


Fig 2:

TCP USING MULTIPATH CONCURRENTLY:

The multipath routing protocol, SMR, scatters TCP packets evenly on the multiple paths. For simplicity SMR uses two paths at the same time. We compare TCP performance over SMR and DSR. We simulate different scenarios with different traffic patterns. Fig 1 shows that when TCP is using multiple paths, it always behaves better than using only single path in all investigated scenarios. Study indicates that SMR outperforms DSR because multiple routes provide robustness to mobility. The performance difference becomes evident as the mobility degree increases. UDP based CBR traffic achieves poor performance over DSR. Fig 2 shows clearly that the packet retransmission of TCP over SMR is much lower than that of DSR. SMR had considerably fewer packet drops compared with DSR.

The overall throughput consistently implies that use of SMR over multiple paths provide prominent benefit to TCP performance. SMR also showed shorter end-to-end delay.

TCP USING SMR:

A. Route Selection

In our algorithm, the destination selects two routes that are maximally disjoint. More than two routes can be chosen, but we limit the number of routes to two in this study. One of the two routes is the shortest delay route; the path taken by the first RREQ the destination receives. We use the shortest hop path as one of the two routes to minimize the route acquisition latency required by on-demand routing protocols. When receiving the first RREQ the destination records the entire path and sends the RREP to source via this path. The node IDs of the entire path is recorded in this RREP, and hence the intermediate nodes can forward this packet using this information. After this process, the destination waits for a certain period of time for more RREQs and learns all possible routes. It then selects the route that is maximally disjoint route to the route that is already replied. The maximally disjoint route can be selected because the destination knows the entire path route information of the first route and all other candidate routes. If there is more than one route that is maximally disjoint with the first route, the one with the shortest hop distance is chosen. If there still remain multiple routes that meet the condition, the path that delivered the RREQ to the destination the quickest between them is selected. The source then sends another RREQ to the destination via the second route selected. Note that the two routes of the session are not necessarily of equal length.

Because our protocol uses the source routing and the intermediate nodes do not reply from cache, only the source nodes maintains the route information to the destinations. Each node uses less memory, but the header size is larger because we use source routing.

B. Route Maintenance

A link of a route can be disconnected due to mobility, congestion, and packet collisions. It is important to recover the route immediately to do effective routing. In SMR when a node fails to deliver the data packet to next hop of the route, it considers the link disconnected and sends route error packet (RERR) to the upstream direction of the route. The RERR message contains the route to the source, and the immediate upstream and downstream nodes of the broken link. Upon receiving the RERR message, the source removes every entry in the routing table that contains the broken link (regardless of the destination). If any of the two routes is invalidated, the source uses the remaining valid route to deliver the data packets.

When the source is informed of a route disconnection and the session is still active, it may use one of the two policies in rediscovering routes:

- Initiates the route recovery process when any of the route session is broken, or
- Initiates the route recovery process when both routes of the session are broken.

C. Allocation Granularity

When the source receives the RREP after flooding the RREQ, it uses the first discovered route to send buffered data packets. When the second RREP is received, the source has two routes to the destination, and can split traffic into two routes. We use a simple per packet allocation scheme [5] when there is more than one available route to the destination. We decided to use the per packet allocation approach because it is known to work well in most networks.

Conclusion:

This paper is an experimental study of TCP performance over multipath routing in Ad Hoc networks including Split Multipath Routing (SMR). We presented the Split Multipath Routing Protocol (SMR), which is an on-demand routing protocol that builds maximally disjoint routes. We attempt to build maximally disjoint routes to avoid having certain links from being congested and to efficiently utilize the available network resources. Providing multiple paths is useful in Ad Hoc networks because when one of the route is disconnected, the source can simply use other available routes without performing the route recovery process. Thus, our work will prove to be of value for future investigations in this direction.

REFERENCES

[1] G.Holland and N.H.Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Proceedings of ACM MobiCom '99*, Aug 1999.

[2] K.Chandran, S.Raghunathan, S.Venkatesan, and R.Prakash, "A feedback-based scheme for improving TCP performance in ad hoc wireless networks" *IEEE Personal Communications Magazine,* vol.8, no.1, Feb. 2001.

[3] D.B.Johnson and D.A.Maltz," Dynamic source routing in ad hoc wireless networks," *Mobile Computing, edited By T.Imielinski and H.Korth, Chapter 5,* 1996.

[4] M.Gerla, R.Bagrodia, L.Zhang, K.Tang, and L.Wang, "TCP over Wireless Multihop Protocols: Simulation and Experiments," Proceedings of the IEEE International Conference on Communications (ICC), June 1999.

[5] R.Krishnan and J.A.Silvester, "Choice of Allocation Granularity in Multipath Source Routing Schemes", Proceedings of the IEEE Conference on Computer Communications (INFOCOM), March 1993.

[6] T.D.Dyer and R.V.Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks", *Proceedings of ACM MobiHoc '01*, Oct.2001.

[7] "Network simulator 2", available at http://www.isi.edu/nsnam/ns/