LULEÅ
TEKNISKA
UNIVERSITET

2000:146

MASTER'S THESIS

# Metrics in Ad Hoc Networks

Andree Jacobson

**Civilingenjörsprogrammet**

Institutionen för Systemteknik
Avdelningen för Datorkommunikation

2000:146 • ISSN: 1402-1617 • ISRN: LTU-EX--00/146--SE

Master's Thesis

# Metrics in Ad Hoc Networks

Andree Jacobson
andree@sm.luth.se

Luleå University of Technology

May 19, 2000

Supervisor and Examiner:
Mikael Degermark
Department of computer science and electrical engineering
Division of computer communication
Luleå University of Technology

# Abstract

An ad hoc network, is a network that can form wherever two or more nodes using the same preferences to communicate are present. Such a network need no centralized control organ and is perfectly suited for many applications, disaster scenarios, search-and-rescue operations and police matters to mention some. This is currently an interesting research area and it is desirable to find good communication standards.

Part of this standardization procedure is the evaluation of suggested routing protocols. To compare these routing protocols, useful protocol independent metrics are needed. This thesis presents two new metrics. One describes the density of the network and another describes the direct connectivity rate for the nodes in the network.

Using a simulation study, it is shown that the density metric is useful when comparing ad hoc routing protocols. Direct connectivity and its close relationship to density is discussed. It is shown that the two new metrics are proportional. The conclusion is that direct connectivity is the preferable metric since no routing can take place if nodes are not connected.

# Contents

# List of Figures

# List of Tables

# Preface

This work has been carried out at Luleå University of Technology over the past 10 months. To be more exact, at the Networking division of the department of Computer Science and Electrical Engineering. The thesis idea was suggested by my supervisor, Mikael Degermark. I found the idea interesting and it was also a continuation of a project I did with a fellow student last year.

The reason I chose to do my thesis work at the university is because I find it to be a very stimulating work environment. In parallel to the thesis work I have had the opportunity to help teaching two courses and participate in a couple of graduate courses. The opportunity to do this as an undergraduate student has inspired me tremendously and has cleared the way for my career as a Ph.D. student.

This document is produced using VIM, LaTeX $2_\varepsilon$, Xfig and Matlab. These are all extraordinary tools that I now appreciate more than anything else available.

Luleå, May 2000

Andree Jacobson

# Acknowledgments

There are a number of people that I wish to express my gratitude to. Without your support, finishing this work would have been very difficult!

First of all, I would like to thank my fellow students here at the networking division. You have all always been there when I needed someone to discuss the problems that arose during my work here. In no particular order Andreas Jonsson, Pierre Fransson, Lars-Åke Larzon and Björn Nordgren. *Thank you!*

Next I would like to thank Olov Shelén, head of the networking division, for your support, understanding and encouragements to fulfill my wishes. You know what I'm talking about! *Thank you, Olov!*

I would also like to thank Mikael Degermark, my supervisor, for your valuable, though sparse, comments and your help to make me work independently. *Thank you!*

A special *thanks* to my dear friend and work-out buddy, Dennis Akos, for your always encouraging comments and never-ending inspiration. It is truly appreciated!

I have many dear friends all around me, who have all helped out in your special way. No one mentioned, no one forgotten. You know who you are! *Thanks!*

Finally I would like to direct a sincere *Thank you for everything* to my family. You have always been there, always supported me with whatever means you have had available, always encouraged me. The past 5 years of my undergraduate studies is not an exception. I dedicate this work to all of you, especially my wonderful parents! You're the best anyone could ever wish for! I know that I have not always showed you the appreciation you deserve. This is my way to make it up to you...

*To my family*

# Chapter 1

# Thesis Introduction

## 1.1 Background

IP-based computer communication over wireless networks is a research subject that has become ever more interesting over the last couple of years. The many possible applications for networks of this type, for example search-and-rescue operations, disaster scenarios and police matters, have created a need for efficient routing protocols. The IETF has therefore assigned a working group called MANET the task of developing routing protocol standards for evaluation. So far, a number of protocols have been suggested and comparisons between them is now an interesting topic.

When comparing routing performance, the metrics used play an important role. One problem is that some routing metrics are based on simulation results rather than the fixed input data. This could imply that a certain metric is biased towards a group of protocols. Furthermore, different metrics are used in different studies, making it hard to compare the results.

## 1.2 Goals of this thesis

The goal of this thesis was to find some new routing metrics for ad hoc networks. These metrics should be independent of the routing algorithms so that they can be used to show strengths and weaknesses in the different routing protocols. This was to be achieved by discussing the existing metrics, find new metrics and finally to evaluate the new metrics with a simulation study.

## 1.3 Research contributions

This thesis defines two new metrics, density and direct connectivity, that can be used to evaluate routing protocol performance in ad hoc networks. These are two new mathematical definitions and metrics.

**Density** – This metric is used to describe the denseness of an ad hoc network. The density is defined as the weighted number of overlapping radio transmitter areas over time. This number is then normalized to yield a percentage of maximum density in the network.

**Direct connectivity** – The direct connectivity is similar to the density. The difference is that in order for a node to be connected, it has to be within another nodes transmitter range so that communication can take place. This metric is defined and discussed in this thesis. However, there are no simulations regarding this metric.

## 1.4 Simulations & Results

A number of simulations are performed for four different routing protocols and ten different density values. It is shown that routing protocol performance can be evaluated using the density metric. And the results obtained corresponds well to the results presented in earlier studies.

## 1.5 Document outline

**Chapter 2** provides a theoretical background. Descriptions of basic ad hoc routing strategies and the protocols using them are presented. In addition to this, routing protocol evaluation methods are also discussed.

**Chapter 3** presents the existing routing metrics that have been used in other comparative studies, regarding ad hoc routing protocols.

**Chapter 4** introduces two new routing metrics. The metric definitions and a discussion about their validity are given.

**Chapter 5** describes the simulation setup and process in detail.

**Chapter 6** presents the results obtained from the density simulations.

**Chapter 7** discusses the results presented in chapter 6.

**Chapter 8** points out some refinements that could be done to the work presented in this thesis and suggest some work for future projects.

# Chapter 2

# Background

## 2.1 What is an ad hoc network?

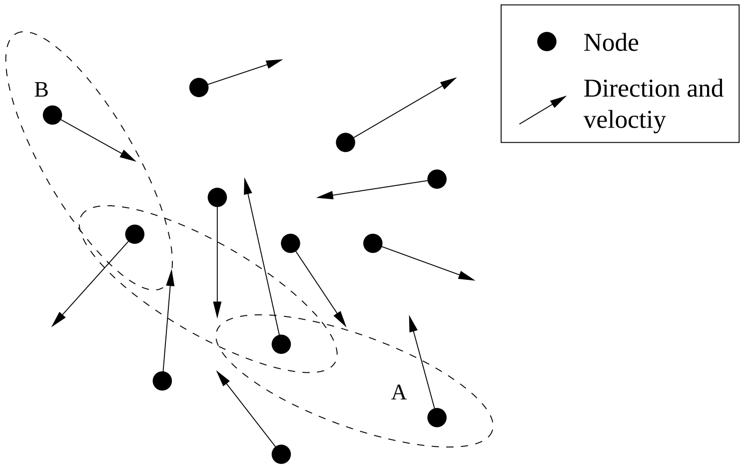


Figure 2.1: An ad hoc network where nodes are moving in arbitrary directions with arbitrary speed. Here node A is communicating with node B via two other nodes.

An ad hoc network (see example in figure 2.1) is a wireless network without centralized control where every node acts as a router, forwarding packets for other nodes as necessary. This makes it possible for this kind of network to emerge wherever there is a need for it. As soon as two nodes are within range, a network connection can be established. This type of network has many advantages over traditional wired networks, for example that it is possible to use very small and common devices as nodes. Examples are PDAs (Personal Digital Assistants), laptops and cellular phones. But the greatest advantage is that there is no need for an existing infrastructure in order for a network to form. Setting up a mobile network is very fast, efficient and can be done practically anywhere. This type of network is preferable in many situations when people wish to share information quickly, such as search-and-rescue operations, meetings or conferences, various military operations and police matters.

## 2.2 MANET

There is an IETF working group called MANET (Mobile Ad hoc Networks) that is assigned the task of developing routing protocol specifications for ad hoc networks [IET]. So far only one general RFC has been presented [CM99]. This RFC outlines suggestions and thoughts about protocol performance issues and evaluation. Furthermore 12 different protocol specifications are suggested as internet drafts, for example [HP99, JLT99, Toh97, BJM99, PRD00, LSG00, GK97, SSB99, PC97, $C^{+}97$, MJLA96].

In this work, four of these protocols (DSDV, DSR, AODV and TORA) are used to perform the metric evaluation. These four are described in greater detail. A few of the other protocols are also mentioned briefly. It was decided not to put that much weight in explaining the functionality of the protocols here since they are already well documented in a number of articles prior to this work.

## 2.3 Desired protocol properties

In a wired network, topology changes are rather infrequent. Most hosts and other nodes in the network have their given position. This is the natural behavior that we expect from a wired network. Link breakages will only occur when there is a physical disruption, such as a failing host or a cable was physically damaged. For this type of wired infrastructure a classic routing protocol functions very well. In order to maintain updated routing tables, routers exchange information by periodically sending update messages to each other. In case of a link failure, the routes have to be recalculated and once again propagated through the network. This process may take a couple of minutes, and this is the normal behavior in a wired network.

Obviously this approach will not work well in an ad hoc network. In an ad hoc network, rather frequent link changes are expected since the nodes are constantly moving. Consider, for example, the case where two nodes are communicating while moving away from each other. As long as they are both within transmitter range, communication can take place. When the distance between the nodes grow too large the communication will fail. When more and more nodes become involved in such a scenario, more links will form and new routes to the destinations may have to be computed.

These differences between wired and wireless networks make it quite obvious that an ad hoc routing protocol need to address some additional problems not present in a wired network. Below is a list of things that a routing protocol should take into account. The more of these properties the protocol can provide the better. Some of these properties are more important than others though. In the initial stages, power conservativeness was less important than functionality. However, these days, when functionality is already achieved, conserving the power in order to make batteries last longer is becoming increasingly important. The reason for this is that mobile units are constantly decreasing in size and hence battery sizes also decreases. Even though the size/power/efficiency ratio for batteries is constantly improved, the energy source for mobile units is still a limiting factor. A routing protocol should not add up more to the total energy consumption than necessary.

The goal of routing protocol design in general is to make the protocol;

- Scale as the network topology grows
- Respond quickly to topology changes
- Provide loop free routes
- Minimize delay (short routes)
- Present multiple routes to avoid congestion

In an ad hoc network, the routing protocol design should also strive to make the protocol:

- Have decentralized execution
- Be bandwidth efficient (minimize routing overhead)
- Utilize both unidirectional and bidirectional links
- Act power conservative

## 2.4 Routing protocol strategies

There are two basic ad hoc routing strategies. One is derived from the old and well known routing protocols that have been used for wired networks for a long time. These protocols are called *table-driven* or *proactive.* The other routing strategy is called *source-initiated, on-demand-driven* or *reactive.* These terms will be used interchangeably in the text from now on. The differences between the two strategies are explained in the next few sections. In addition to these two basic methods there is also a hybrid approach that utilize some of the functionality from both the proactive and reactive strategies.

This report focuses on four different protocols belonging to different protocol groups. These four will be given a deeper explanation while other protocols are just briefly mentioned.

### 2.4.1 Proactive strategy

The classic routing strategies for wired networks – link state, distance vector and source routing [PD96] are all well documented and thoroughly tested. Why not use them in wireless networks as well?

As mentioned earlier there is one significant difference between wired and wireless networks – the nodes are moving! In a wireless scenario where the mobility is negligible a conventional protocol would probably function very well. However, as soon as nodes start moving to any greater extent these protocols would fail to stabilize due to the frequently occurring route changes. The algorithms are simply not fast and efficient enough to handle the many events that will occur. In addition to this, conventional protocols also assume that all links are bidirectional, this may not always be the case in a wireless scenario since two nodes may have different transmitter ranges. Proactive protocols for wireless networks are therefore modified to address these problems. The proactive protocols discussed in this thesis are based on the distance vector routing algorithm.

In short, distance vector routing means that nodes keep track of the cost for its outgoing links. With constant intervals the expected values of the node's shortest distance to every other node in the network is broadcasted to all neighboring nodes who update their routing tables accordingly.

### 2.4.2 Reactive strategy

The reactive approach to the problem works differently because routes are requested when needed, and it is the sender that initiates this route request (hence the name source-initiated on-demand). If a source wants to send a packet to the receiver, but does not have a route to the destination it will need to acquire this route from other nodes in the network. The source sends out a route request packet to its neighbors, asking for a route to the destination. This route request is then propagated through the network until it reaches a node that either has a route to the destination or is the destination itself. A route reply packet is then sent back over the same path as the request came from and a connection between the two nodes can be established. In case of a link failure, a route reconstruction phase is deployed in order to suggest an alternative route for the packet stream. The outdated route will be purged from the routing structure if it does not become valid again within a certain time. This scheme provides high connectivity in a dynamic scenario. As node mobility increases, so will the number of link changes, as well as the amount of overhead routing traffic.

### 2.4.3 Hybrid strategy

There are some protocols that combine the two different strategies. These protocols divide the network into zones (clusters) and run a proactive protocol within the zone and a reactive approach in order to perform routing between the different zones. This approach is better suited for large networks where clustering and partitioning of the network often occur.

## 2.5 Routing protocols

This section gives a more detailed description of the four[1] routing protocols used in the simulations for the work in this report. The first protocol described, DSDV, is the only proactive protocol. The other three are reactive.

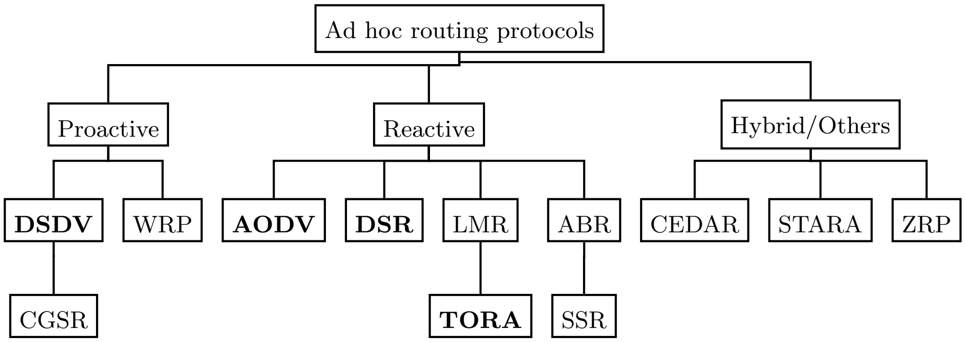


Figure 2.2: Categorization of ad hoc routing protocols

### 2.5.1 Destination Sequenced Distance Vector – DSDV

DSDV is an entirely proactive protocol, i.e., DSDV does not attempt to find a route for a packet if none is available in the node's routing table when the packet arrives. The advantage of this approach is that a packet can be forwarded immediately if there is an entry for its destination in the routing table. However, if the period between successive update messages is too long compared to the time between topology changes, DSDV will not be able to converge.

Since there is no mechanism in DSDV to explicitly query the network for a particular route when needed, all nodes keep a routing table which holds the routes for all reachable nodes. A node broadcasts updates of its routing table regularly to its neighbors, which set their routing tables accordingly.

DSDV avoids loops by using a quite simple technique. Each route is labeled with a sequence number, which determines its age. Newer routes have higher sequence numbers, and if a node receives an update which contains a route with a higher sequence number than the corresponding one in its routing table, or if the routes have equal sequence numbers but the new one is shorter, it updates its routing table with the new route.

---

[1] A fifth protocol, IMEP, is described in chapter 2.5.5. This protocol can not be used by itself but is rather a help protocol that TORA uses and is therefore included.

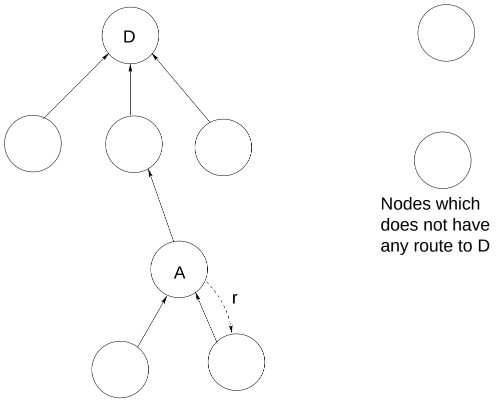


Figure 2.3: Network with stable routes. When A selects a new route, it bases its decision on the sequence number and the metric of the route. r must either have lower sequence number or higher hop-count. Thus, A will not select r as a route to D avoiding the routing loop.

Note that the routes will eventually stabilize if there is no movement in the network. Consider each node a destination. For each destination D, the network can be modeled as a set of trees with the routes to D as edges. These trees either has D or a node without a route to D as root. Hence, it contains no loops. To form a loop, a node A needs to select a route which goes through a node in one of its subtrees. However, routing updates propagate through the network starting at D. Thus all nodes on the path between A and the destination contain routes to D with higher or equal sequence number (higher if an update has not yet reached A). Conversely, all nodes below A in the tree, contain routes with lower or equal sequence numbers. Obviously, nodes below A is also further away from D. Consequently, if A closes a loop when selecting a new route, it must either have a lower sequence number or a higher hop count. Thus, if all nodes always pick routes with higher sequence numbers, or equal sequence numbers and lower hop counts, loops can never be formed. See figure 2.3.

There are two weaknesses with DSDV that can be identified. First, there is some redundant routing overhead. Many routes which are discovered will never be used and the bandwidth consumed for announcing them is therefore wasted. Second, the delay from the time a new connection is established to the time a new route is known is relatively long. Consequently, there can be substantial difficulties to find a route when mobility is high.

These two factors can be compromised to tune performance. Frequent routing updates find routes faster but increase the overhead.

For further information about DSDV, refer to [PB94].

### 2.5.2 Dynamic Source Routing – DSR

DSR was designed by Broch et. al. at Carnegie Mellon University [CMU] and is a strictly reactive protocol. It does not make queries for routes until they are needed. The protocol is based on source routing, which allows intermediate nodes to forward packets without having a fresh route in the cache. However, since every packet carries the complete route, there will be some extra overhead in each packet. The packet size depend on the distance between the communicating nodes.

When node A wants to send a packet to node B, it searches its cache for a route to B. If a route is found, it is inserted into the header and sent. If somewhere along the way a link is broken, possibly due to two nodes which have moved out of range of each other, an error message will be returned to A. A then searches its cache for additional routes.

If A does not have a route to B, it broadcasts a query to its neighbors. Each neighbor records its address in the query message and forwards it in a controlled manner to its neighbor. This process is repeated until B is reached. B then sends a reply along the reversed recorded route. It is also possible to create asymmetrical routes where B replies by sending out a query for a route to A, piggy-backing the recorded route to the query.

A possible optimization is that each node receiving the query, searches its cache for a path to B, and if found, the intermediate node replies with the cached route appended to the recorded one. Hence, the route discovery is shortened and routing overhead is reduced.

A DSR node must provide hop-by-hop reliability. Each node attempts to verify that the packet is received at the next hop. If it is unable to do so within a certain amount of time, an error message is sent back to the originator and the link is assumed to be broken. Consequently, the route will be considered invalid and deleted from the cache.

DSR provides limited support for multi-casting. By piggy-backing data to the route query for a multicast address, it will propagate through the network to all nodes interested in the multicast group. This scheme does not scale well and does not provide all of the qualities a multicast routing protocol should have.

### 2.5.3 Ad hoc On demand Distance Vector – AODV

AODV is as DSDV (chapter 2.5.1, using a sequence number to avoid routing loops, and periodic updates to keep routing tables at each node [PRD00]. However, it has been altered to provide routes on demand for better performance in ad hoc networks.

The sequence numbering is similar to the one used in DSDV. It states when the route was created. A higher sequence number indicates a fresher route, which should be used in favor of older ones. For this purpose, each node keeps the sequence number of the last generated route. This will be increased for each new route.

Route queries are done as follows; An arbitrary node A, wants a route to B. A then broadcasts a route request to its neighbors. Then it waits for a reply. The request may be re-broadcasted a limited number of times if no reply is received. It propagates through the network until it reaches a node which has a valid route to B.

A list of neighbors that are actively forwarding packets is also kept at each node. It contains all neighbors that have been positively acknowledged forwarding packets within a certain time limit. The link layer may indicate that links are down and stale routes could then be removed. Hence, a link breakage will be detected relatively fast.

Unlike the other routing protocols described in this report, AODV supports multicasting. There are no technical limitations in the design which makes multicasting awkward, and support for it has been developed. This is probably the main issue that favors the usage of AODV.

### 2.5.4 Temporally Ordered Routing Algorithm – TORA

TORA is a distributed routing protocol, i.e., each node need only maintain information about its closest neighbors (on a one hop basis). Furthermore it provides multipath, loop-free routing and is designed to minimize reaction to topological changes. The route establishment procedure may be done either proactively or reactively. It maintains per-destination states in a manner similar to the other distance-vector routing protocols. The design also allows it to find routes on-demand since it might not be desirable to maintain information about all possible routes at all times. Selected destinations may initiate proactive operation, similar to the traditional table-driven protocols. Invalid routes that might result from a network partition are discovered and erased. [PC97]

By assigning *upstream* or *downstream* directions to the links between routers, TORA creates a directed, multi-path routing structure where the destination is downstream from the source. This structure can be described as a directed acyclic graph (see figure 2.4). Each router keeps a value that can be thought of as the router's *height* in the routing structure. Routers may only forward packets to a downstream destination.

It should also be pointed out that TORA only performs routing and rely on Internet MANET Encapsulation Protocol (IMEP) to perform the underlying functionality. This introduces some overhead to the routing scheme.

### 2.5.5 Internet MANET Encapsulation Protocol – IMEP

The idea with IMEP[2] is to have a common general protocol that other routing protocols can make use of. It incorporates many common mechanisms that other protocols may need. These include:

- Link status sensing

---

[2] The IMEP internet draft is outdated and no new draft has been presented. This specification was found in [LH98] and is included since IMEP is implemented in the simulator. No information about why there was never a new IMEP draft has been found.

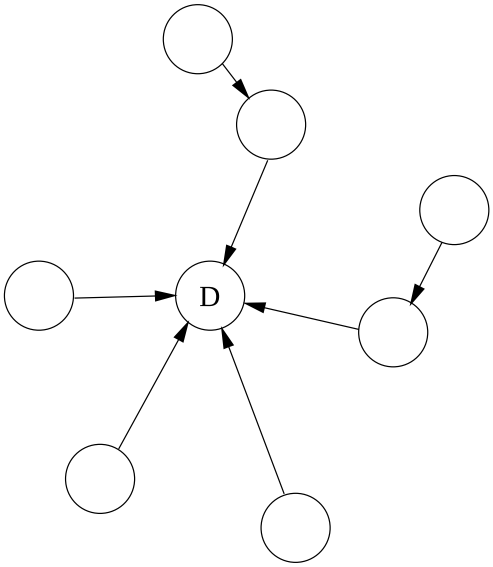


Figure 2.4: A Directed Acyclic Graph rooted at the destination

- Control message aggregation and encapsulation
- Broadcast reliability
- Network layer address resolution
- Hooks for inter-router security authentication procedures

IMEP also provides an architecture for MANET router identification, interface identification and addressing. IMEP's purpose is to improve overall performance by reducing the number of control messages and to put common functionality into one unified, generic protocol useful to all upper-level routing protocols.

IMEP was designed to support many ad hoc routing protocols, however of the proposed protocols only TORA and one other protocol (OLSR - Optimized Link State Routing Protocol) use it. It can be used by other protocols to provide some security and authentication. It should also be pointed out that both IMEP and TORA were designed by the same author.

The basic idea is good, but from a performance point of view it is not such a good idea. The work performed by the CMU monarch project [CMU] has shown that IMEP produces a lot of overhead, mainly because of IMEP's neighbor discovery mechanism that generates at least one hello message per second, but also because of the reliable in-order delivery of the packets that IMEP provides.

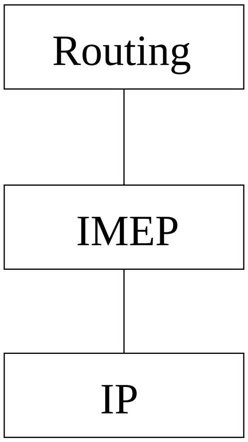


Figure 2.5: IMEP in the protocol stack

### 2.5.6 Other routing protocols

DSDV is a proactive protocol, other proactive protocols worth mentioning is Wireless Routing Protocol, WRP [MJLA96] and Clusterhead Gateway Swith Routing, CGSR [$C^+97$]. Examples of hybrid protocols are Zone Routing Protocol, ZRP [HP99] Cluster Based Routing Protocol, CBRP [JLT99].

Core Extraction Distributed Ad hoc Routing algorithm, CEDAR [SSB99] and Associativity Based Routing, ABR [Toh97] are both demand-driven protocols but their functionality is slightly modified as described below. Both CEDAR and ABR are protocols designed for smaller networks of tens or possibly hundreds of nodes making them appropriate for conferences and similar scenarios.

The main contribution of CEDAR is the addition of QoS into ad hoc routing. However, the route computations are performed by the network *core* nodes on behalf of all the other nodes in the core node's domain. The core node also keeps track of its domain topology. The different core nodes communicate with each other and via local state and local computations the core is kept intact and reacts quickly to link changes. Still, robustness rather than optimal performance is the primary concern of CEDAR [SSB99].

C-K Toh introduces ABR and the concept of associativity in [Toh97]. The associativity property describes the amount of time that a mobile node stays dormant before it starts moving again. It is similar to the *pause time* metric that is described in chapter 3.2.2. Routes are demanded by the source, but all route decisions are made by the destination that can choose from the different routes found. The associativity property is supposed to allow the routing protocol to choose long-lived routes,as opposed to shortest-path routes, resulting in fewer route reconstructions than for other protocols.

## 2.6 Protocol testing

Before employing a routing protocol in a real network, it has to be thoroughly simulated in order to find bugs etc. There seem to be a few different simulators in use. The Network Simulator from Berkeley [UCB], is used in a couple of the protocol suggestions and comparisons made. Some designers have their own simulators.

### 2.6.1 Scenarios

The most common approach for an ad hoc scenario is a randomized movement pattern within a constantly sized area. As far as the author knows, only two-dimensional simulations have been made, even though a three dimensional approach would be better since it would correspond better to the reality (radio signals do propagate through walls and floors to some extent).

The two-dimensional scenarios are typically based on a couple of input variables. *Pause time* and *velocity* are the two most significant variables for the movement model. Nodes are initially randomly distributed inside a rectangular area. When the simulation commences each node pauses at its current position for *pause time* seconds. The next step is to pick a new arbitrary location and start moving towards it. As with the pause time the velocity with which the node will start moving is randomly chosen from an interval of max and min velocity. When the node reaches its new position it will pause once again for *pause time* seconds and then the process will repeat itself until the end of the simulation is reached. All nodes behave in the same way.

#### Realistic scenarios

Even though random movement may be well suited for some simulations there is also a need for realistic scenarios. It turns out that humans are not as randomly distributed as we might think. This idea was foreseen in [LHJ+99] where a couple of realistic scenarios were presented. It is a very interesting approach and will therefore be a bit further explained.

**Conference** — This is a model of a conference or something similar. Few nodes are moving and there is one "speaker" which moves back and forth in the front of the room and transmits data. All movement in this scenario is fairly slow. Figure 2.6 shows the conference scenario. In the picture, the upper part is referred to as the *front*.

**Event coverage** — In this scenario there is more movement.The idea is that two temporary clusters are forming in the network and communication within these clusters take place. However information cannot be exchanged between the clusters. Figure 2.7 shows the event coverage scenario.

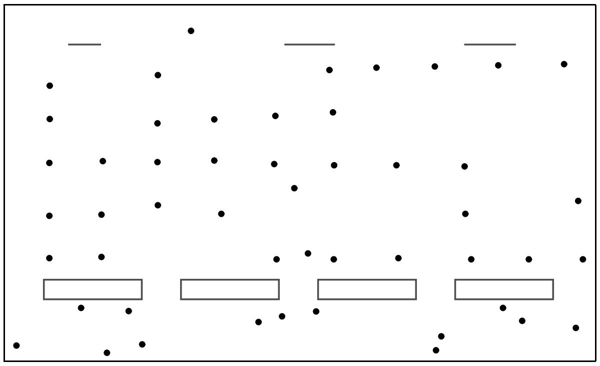


Figure 2.6: Conference scenario

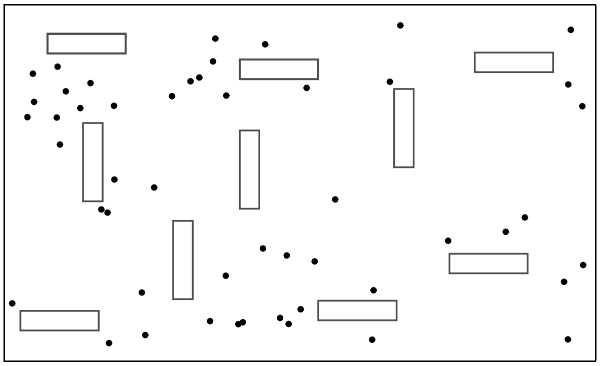


Figure 2.7: Event coverage scenario

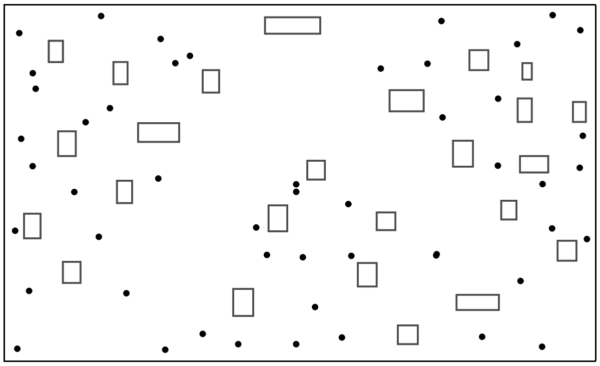


Figure 2.8: Disaster scenario

**Disaster area** — The third scenario models a disaster area where a couple of partitioned networks are characterized by high mobility. The three clusters are sometimes bound together by vehicles moving back and forth between different locations. It is the authors opinion that this scenario might not be as realistic as Larsson and Hedman suggests since in a situation like this, it would be more realistic to use a few base stations enabling the different networks to communicate. The point is that it is not very realistic to only rely on two moving vehicles to temporarily form a link between the clusters in a disaster scenario since information would most likely have to be exchanged all the time. Figure 2.8 show the disaster scenario. The two vehicles are moving diagonally over the scenario.

This work is the only work done on realistic scenarios so far. It is an interesting approach since these scenarios better model a realistic movement pattern of humans compared to the randomly created scenarios.

### 2.6.2 Communication patterns

In addition to the movement patterns, communication patterns are also used in a simulation. These communication models could be generated in a random fashion or if the communication is known, exact communication specifications can be set up. In the random case connections are set up between a predetermined number of node pairs..

When creating these patterns some input variables can be altered, such as sending data rate, packet size, traffic type and the number of sending nodes.

# Chapter 3

# Existing routing metrics

As mentioned in chapter 2.2, a number of ad hoc routing protocols have been suggested. These all try to solve the problems and different aspects that a wireless environment presents. But how is it decided which one is the best? This depends on the structure and properties of the network. The nodes might be moving fast or slow, they might be highly concentrated into a small area or widely spread out over a large area. There are undoubtably many questions that a designer of a protocol have to take into account and most likely a single protocol will not be able to have all suggested properties.

There are a few comparative studies written on the subject [BMJ+96, RT99, LHJ+99], The comparisons performed show that there are major performance differences between the protocols. In general, since weaknesses of earlier protocols are known, new protocol designs try to address those problems as well. Then it has to be proved that the new protocol is actually better than the older ones. In order to quantify the differences, some kind of performance metrics have to be used. The following sections will mention most of the metrics that have been used up until now. Where appropriate, their strengths and weaknesses will also be discussed.

There are two main categories of routing metrics, the first category, *performance metrics*, describes the outcome of a simulation, or a set of simulations. Scenario metrics is the other class of metrics. These describe the simulation input parameters.

## 3.1 Performance metrics

These metrics are interesting because they can be used to point out what really happened during the simulation and provide valuable information about the routing protocol. In the following sections some metrics of this type are described.

### 3.1.1 Packet delivery ratio

The packet delivery ratio presents the ratio between the number of packets sent from the application layer and the number of packets actually received at the destination nodes. It is desirable that a routing protocol keep this rate at a high level since efficient bandwidth utilization is important in wireless networks where available bandwidth is a limiting factor.

This is an important metric because it reveals the loss rate seen by the transport protocols and also characterizes the completeness and correctness of the routing protocol.

### 3.1.2 Routing overhead

Routing overhead is of course an interesting metric. In some way it reveals how bandwidth efficient the routing protocol is. The routing overhead metric simply shows how much of the bandwidth (which often is one of the limiting factors in a wireless system) that is consumed by the routing messages, i.e., the amount of bandwidth available to the data packets.

An interesting observation is that for all protocols there is a theoretical limit where some properties of the scenario force the data rate down to zero because all the bandwidth is used for routing messages. The ideal case is naturally no overhead at all i.e., only data packets traverse the network. An ideal routing protocol can be implemented in a simulator but a routing protocol without routing messages is a contradiction and can not be implemented in a real network.

The routing overhead is typically much larger for a proactive protocol since it periodically floods the network with update messages. As mobility in the network increases reactive protocols will of course have to send more routing messages too. This is where the real strength or weaknesses of the routing protocol can be revealed. On the other hand

In DSR another type of overhead presents itself even though it is easily overlooked in the previously described packet delivery ratio metric. DSR works by finding source routes to the destination on-demand. By storing information about all intermediate nodes in the packet header as the the route discovery packet traverses the network it knows the full route once the route discovery packet returns. These source routes cause the packet headers to grow and hence produce more routing overhead [BJM99]. Considering this, the traditional metric, packets sent versus packets delivered, might give the impression that DSR is able to deliver more packets than other protocols. Looking at the ratio payload bytes sent versus payload bytes received instead could result in a different performance for DSR. This would be most obvious in a network with long routes (many hops).

### 3.1.3 End-to-end delay

The term end-to-end is used to an average measure of performance between nodes in a network. It is the sources and the receivers that are involved. The end-to-end delay is therefore the total delay that a data packet experiences as it is traveling through a network. This delay is built up by several smaller delays in the network that adds together. These delays might be time spent in packet queues, forwarding delays, propagation delay (the time it takes for the packet to travel through the medium) and time needed to make retransmissions if a packet got lost etc.

Typically, in a packet based radio network without QoS (Quality of Service) the delay could vary much depending on the routing protocol. One parameter that is critical is the time a packet is kept in a buffer before it is dropped if there is no route for its destination. This buffering time is controlled by a timer in each node. If this timer is set to a high value it could imply that packets are delayed in a network for this rather long period of time. A high value would probably decrease the number of dropped packets but it would also result in a somewhat higher average delay. Of course this is a question of what is important in a particular network, low delay or few dropped packets. It is a tradeoff that the system designer need to do, and as stated earlier, this will have an impact on the end-to-end delay.

### 3.1.4 End-to-end throughput

Since the available bandwidth in a network is fairly well known, it is interesting to see what the actual throughput achieved in a simulation is. If a good estimation of this value can be extracted it would be possible to see how efficient the routing protocol is. The higher the average throughput, the less routing overhead consuming the bandwidth.

### 3.1.5 Path optimality

Traditionally this measurement compares the optimal path usually defined as the shortest path between two nodes in the simulator at the sending moment with the length of the path that the packet actually travelled. If the average actual path length is close to the shortest path, the protocol is said to be good. However, it is hard to know what the actual optimal path is (consider figure 3.1). Just settling with the shortest path does not address queuing and congestion in the network, or high latency links.

STARA (System and Traffic Dependent Adaptive Routing Algorithm) as presented by Gupta and Kumar in [GK97] is one of the few protocol suggestions that consider other optimal paths than the shortest path. STARA uses the mean delay as distance measure instead. ABR (discussed in chapter 2.5.6) use the expected longevity of a route when it make the routing decisions.

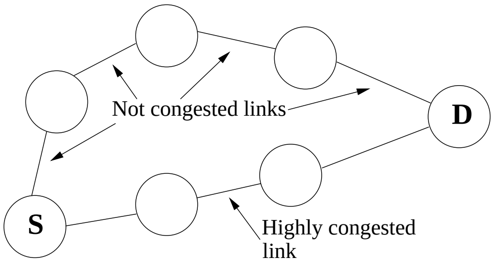


Figure 3.1: The shortest path from S to D is congested and is therefore not the optimal path, instead the optimal case here would be the longer but uncongested route.

## 3.2 Scenario metrics

A scenario metric is calculated from the input data to the simulation, or might even be an input variable (such as the *pause time* described in chapter 3.2.2). These metrics are interesting since their value will not be dependent of the routing protocol or the simulation process, as the performance metrics (described in chapter 3.1) might be. It is crucial that non-biased metrics exists in order to provide a truthful comparison between the different routing protocols.

### 3.2.1 Mobility

The mobility metric was introduced by Larsson, Hedman *et al.* in [LHJ$^+$99]. It is an attempt to measure the mobility in the network by calculating the relative node movement between all pairs of nodes in the network. The mobility metric is proportional to the number of link changes in a model where nodes move in a random fashion as described in chapter 2.6.

In some sence the associativity property as described in [Toh97] tries to catch the same property as this mobility metric with the difference that it incorporates it in the routing decisions.

### 3.2.2 Pause time

As described in chapter 2.6, pause time is also a simulation input variable. When used as a metric, the mean pause time of all the nodes throughout the simulation is used as a measure similar to the mobility metric (chapter 3.2.1). The longer the average pause time is, the less node movement within the network. However measuring mobility in this way may be very misleading since the relative movement between the nodes is left out. Even though nodes are pausing for extended periods at one spot they could be moving very rapidly in the next moment, causing many link breakages.

Still, pause time is a realistic description of human behavior as described in [Toh97] where an experiment was conducted by letting employees wear badges from the Active

| **Dormant time (minutes)** | | | | | |
|---|---|---|---|---|---|
| Distributions | Day of the week | | | | |
| | Mon | Tue | Wed | Thu | Fri |
| Minimum | 5.08 | 5.06 | 5.10 | 5.01 | 5.02 |
| Maximum | 299.15 | 277.00 | 281.68 | 223.06 | 297.64 |
| Mean | 35.79 | 36.26 | 41.08 | 40.84 | 47.99 |
| Standard deviation | 46.63 | 50.88 | 50.55 | 55.40 | 62.81 |

Table 3.1: Dormant time distribution of 52 badge wearers in a week at the Cambridge Computer Laboratory

Badge System [ICOC] that reported the bearers location at constant intervals and hence giving a rather truthful images of the movement behavior of humans in an office environment. It turned out that the pause time was indeed a realistic measure, because it showed that the average human, in an office environment tend to move periodically, pausing in between movements. The results achieved are presented in table 3.1.

# Chapter 4

# New routing metrics

Routing metrics that are only dependent of the current *physical* conditions of the network are the most interesting ones. In this section two new routing metrics will be presented. The first one is called *density* and the second one *direct connectivity.*

## 4.1 Density

Density is the first new metric introduced in this work. The idea with this metric was that if it should prove useful, there would be a metric, solely dependent on the scenario input variables. Such a metric would not be dependent of the actual performance of the routing protocols or any metrics calculated from the simulation output. This means that *density* belong to the class of scenario metrics described in chapter 3.2.

The question is whether or not the *denseness* of the nodes in an ad hoc network would influence the performance of the routing protocols used in the network. If so, it should be expected that an increased density of nodes in the network would decrease the routing protocols performance as a direct effect of less available bandwidth and hence higher congestion.

### 4.1.1 Definition

First a way to describe a mobile network has to be adopted. In this case the author chose to model a node in the network as a circle with the radius symbolizing the radio transmitter radius. A simple abstraction, although more complex than the commonly used *cell* concept [Rap96, Hil95] where each cell is modeled as a hexagon. This measure assumes that all nodes in the network have the same transmitter range.

Keeping this simple abstraction in mind, the next problem is to find a way to define the density as a mathematical expression. The goal is to be able to define a metric, a number, that can be used to describe the average state of the network over the course of the whole simulation.

The idea is to calculate the density at a given point as the total overlapping area of the circles at that moment. The most intuitive way to do this is to calculate the area between every pair of nodes and then add them all together.

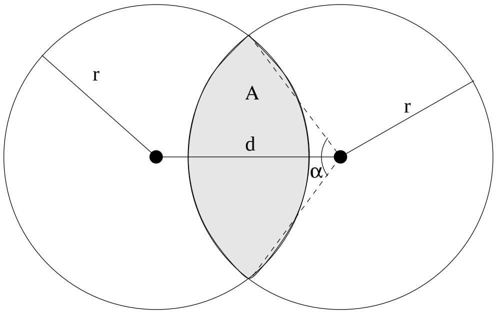


Figure 4.1: Area of two intersecting circles

The basic problem is to find the area of two intersecting circles shown as the shaded area in figure 4.1. Using some elementary trigonometry it is an easy task to derive the mathematical expressions 4.2 and 4.1.

$$\alpha = \arccos\left(\frac{d^2}{2r^2} - 1\right) \qquad 0 \leq d \leq 2r \tag{4.1}$$

Equation 4.1 is used to calculate the angle $\alpha$ between the intersection points on the two circles in figure 4.1. $d$ is the distance between the two nodes (which of course is a function of time) and $r$ is the transmitter radius. Once $\alpha$ is calculated the only thing left to find the area $A$ is to apply equation 4.2.

$$A = (\alpha - \sin\alpha)r^2 \tag{4.2}$$

Now it is possible to find the total normalized area for a given point in time, but how should the density metric itself be defined. The easiest way to consider the full scenario duration in one metric is to integrate the area over time. However, the primitive function to the area equation 4.1 is complex so it was decided that the integration would be performed numerically instead. The scenarios are divided up in a number of smaller time fractions (the time between two events in the movement pattern) that represents the different integral limits.

Romberg integration [RW90] was chosen as the integration method because it is fairly easy to implement and yield high precision results although requiring some processing time. For a definition and implementation of the integration method, please refer to appendix B.

To simplify the calculations, the radius of each circle is normalized to 1 length unit. In addition the area of each circle is also divided by $\pi$ and the full simulation time, resulting in a total area of 1 for each overlapping circle over the entire simulation.

This normalization help to make the interpretation of the metric more intuitive. If the density is 1 it means that we have 1 overlapping area (i.e., both nodes at the same location in a two node scenario).

### 4.1.2 Calculating the maximum overlapping area

In order to have something to relate the achieved area to a need to find the maximum area emerges. Since the area of each (normalized) circle is 1 this is a fairly easy task. With one node there would be no overlapping area at all. With two nodes, the maximum overlapping area would be 1, when the two nodes are placed on exactly the same location. For three nodes at the same spot, there would be 3 overlapping areas (Between node 1 and 2, node 1 and 3 and finally between node 2 and 3). Expanding this reasoning to an arbitrary number of nodes yields the following equation:

$$maxarea = \sum_{n=1}^{N-1} n \tag{4.3}$$

where N is the number of nodes in the network. Performing a division between the calculated area with the maximum area yields a number ranging from 0 to 1, providing a density percentage of the maximum density possible.

## 4.2 Direct connectivity

The second metric that is introduced here is called *direct connectivity*. This metric is used to show how many other nodes the average node are in contact with over the simulation time.

### 4.2.1 Definition

The definition of the direct connectivity metric is the following: A node which is able to communicate with another node directly (i.e. is located within another node's transmission range) is said to be connected. See figure 4.2.

Whenever two nodes are connected they contribute to each others direct connectivity. Summing up connections like this over the whole scenario gives us a value consisting of the average number of nodes that each node have been in contact with over the course of the simulation.

Since the nodes are all moving in one way or another, two nodes may meet and then become separated again. If they are in contact with each other half the time between two events they are said to have direct connectivity 0.5 during that interval.

An event might be that one node in the network changes direction. These events are all calculated in the beginning so that it is possible to divide the scenario into smaller time frames. This is possible since all motion in the network is of constant

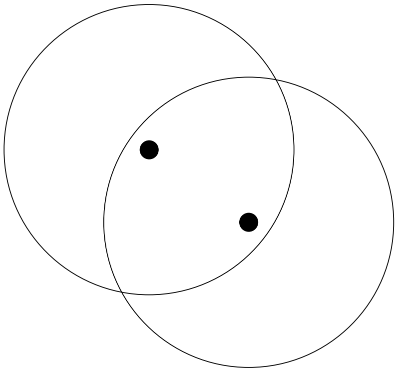


Figure 4.2: Nodes that are connected

speed. In other words, the time between two events is an interval in which all movement in the network is constant.

In order to make this somewhat more intuitive and a little more general the achieved value is normalized by dividing it with the maximum value possible yielding a percentage number.

For example, consider a network with 50 nodes. Each and every node's maximum direct connectivity would be 49 (or 1 if normalized). This value could only occur if this node was constantly within transmitter range of every other node in the network.

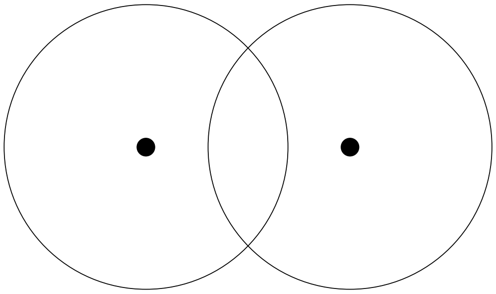


Figure 4.3: Nodes without direct connectivity, but with some density

This metric emerges naturally when working with the density metric since there can be a quite substantial density in a network even though no nodes are in communication range with each other, i.e., no direct connectivity. This is a fact since a circle (node) has to be in communication range with another node in order for the two to accumulate direct connectivity. The density though is accumulating as soon as there are some radio overlapping, even if the nodes cannot are not connected and hence can not communicate (see figure 4.3).

## 4.3 Metric evaluation & discussion

It would be desirable to show that the metrics are useful too. In [LHJ+99] it was shown that the mobility metric was almost proportional to the number of link changes that the network experienced. Performing the same comparison for the density metrics does not result in a proportional relationship. The logarithmic curves in figure 4.4 show that when the density is high (and the direct connectivity is almost 1, such as in the rightmost data points) link changes are rather infrequent, as expected.

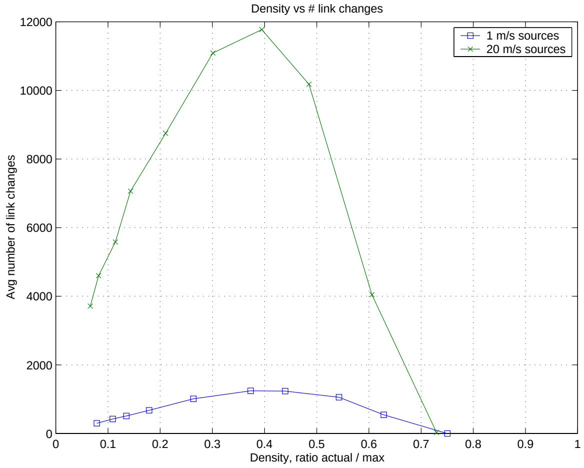


Figure 4.4: Comparing density to number of link changes

Generally when nodes are moving fast the number of link changes increase. Because of the more rapid movement there are more link breakages and communication disruptions. This is clearly shown in the graph. The curves both have a local maximum at a density of approximately 40% which is not that strange since a density of 40% allows the 250m transmitters to move out of range of each other and there are a lot of space to traverse with high speed.

However, this can not be considered to be a clear relationship between the density and the number of link changes. A hypothetical explanation to the graph becomes clear when the extremes are studied.

One extreme for the density is that all nodes stand still at the same location over the entire simulation. This is the maximum possible density. In such a scenario all links formed would be persistent through the whole simulation. A link change is very unlikely, the only reason could be a hardware failure. In that case a link failure would be experienced. Hence at this extreme the number of link changes would be 0.

At the other extreme, the scenario is so large that no links will ever form. In such a network, link changes would never be experienced since there are no links that can change. Hence at this extreme the number of link changes would also be 0.

Somewhere in between these two extreme points at least one maximum value would be expected. This reasoning corresponds well with the data shown in figure 4.4 and this is the evidence offered to show that density is a useful metric.

# Chapter 5

# Density Simulations

The purpose of the simulations carried out in this work is to provide data for the evaluation of the usefulness of the density (chapter 4.1) and direct connectivity (chapter 4.2) metrics.

## 5.1 The simulator

The simulator used for these simulations is the Network Simulator (version ns-2.1b6) [UCB]. Ns is written in C++ and uses OTcl as a command and configuration interface [FV99]. This means that the simulator is implemented in C++ and that the simulations are set up by using tcl scripts.

A typical simulation run from ns produces a trace file containing packet traces for routing and data packets, current node locations etc. that can later be analyzed to collect useful statistics..

Originally ns was designed for wired networks. As a result of the Monarch project at CMU (Carnegie Mellon University) a research group implemented wireless functionality into ns. A couple of ad hoc routing protocols were implemented and the functionality of the 802.11 physical layer was also implemented so that research could continue. This work was presented in [BMJ+96].

Later, this source code was made publicly available and currently the original CMU wireless extensions are being merged into the standard ns distribution.

## 5.2 Simulation environment

All simulations were performed on Intel PII 500MHz machines running FreeBSD 3.4. The main part of the simulations were run on a cluster of 30 machines in order to speed up the simulation process.

## 5.3 Simulation variables

As mentioned in chapter 2.6 there are many simulation parameters that need be varied in order to perform exhaustive simulations. In this chapter the simulation parameters used to produce the simulation suite for this work are presented and explained.

In order to have a starting point, density calculations were performed on the CMU test scenarios, available from their web page [CMU]. These scenarios were generated for their work in the article [BMJ$^+$96]. Some of the decisions made in the following sections are based on the results obtained in these calculations.

Table 5.1 shows a summary of the simulation parameters and the following sections provide a deeper explanation of the choices that were made.

| Parameter | Value |
|---|---|
| Transmitter range | 250m |
| Bandwidth | 2Mbit |
| Interface queue length | 50 packets |
| Simulation time | 900 s |
| Number of nodes | 50 |
| Pause time | 30 s |
| Traffic type | Constant Bit Rate |
| Packet rate | 4 packets/s |
| Packet size | 512, 1024 byte |
| Maximum number of flows | 10, 30, 50 |
| Scenario size | 200x200m - 1600x1600m |
| Nr of scenarios for each size | 3 |
| Density | 7% – 75% |

Table 5.1: Simulation parameters

Summing up these parameters results in 60 different scenarios, (2 packet sizes, 10 scenario sizes, 3 different randomly generated scenarios for each scenario size), 6 different transmission patterns to be simulated over 4 routing protocols; a substantial amount of simulations. The simulator is set to simulate the 914MHz Lucent WaveLAN$^{TM}$ DSSS radio interface, each node with an approximate transmitter range of 250 meters.

### 5.3.1 Routing protocols

It was decided to use the same protocol suite as in [BMJ$^+$96], that is DSR, DSDV, AODV and TORA/IMEP. The reason for this is that they were all implemented in the simulator and that there are previous papers on the topic that the results obtained can be compared to.

### 5.3.2 Pause time

In the CMU paper the pause time was one of the most important metrics used and they ran the simulations on a large number of values. However the density calculations showed that the maximum density in those scenarios emerged when the pause time was 30 seconds. For all the other scenarios it was less. Regarding this, the pause time in all simulations was set to 30 seconds. This was done in order to reduce the number of parameters that needed to be varied.

### 5.3.3 Node velocity

Even though the node velocity has no critical impact on the density of the scenarios, it certainly has a huge impact on the routing protocol's ability to deliver data packets and maintaining stable routes. Two different speed limits were set, 1 and 20 m/s respectively. When generating the scenarios the node speed chosen for each node is randomly distributed between 0 and this speed limit, resulting in an average speed of 0.5 and 10 m/s.

### 5.3.4 Density - Simulation area

The CMU scenarios were all 1500x300 meters and even though the maximum velocity and pause times were varied they are very similar regarding the density (all between 16 and 28 %). For the three realistic scenarios in [LHJ+99] the values are 6, 3 and 10 % respectively.

It was clear that a whole new set of scenarios had to be generated in order to vary the density so that the density values could be put on the axis of a graph and represent some illustrative values.

Varying the density is difficult since the density depend on most of the other input variables in a scenario. Still it is desirable to have some density values of different magnitude in order to vary the density while performing the simulations. Therefore the density was varied indirectly instead.

The easiest way to vary the density with the simulation variables available in the simulator is to keep the number of nodes constant and then change the simulation scenario area. For this purpose new scenarios have been generated. These are all quadratic with the side length varied from 200 to 1600 meters. The nodes all have a transmitter range of 250 meters, so the smallest scenarios have full direct connectivity. The actual density varies between 7% and approximately 75%, in 60 different scenarios, 3 for each scenario size as shown in table 5.2.

### 5.3.5 Number of nodes

As indicated in the density section above the number of nodes in the scenarios were kept constant. This constant value is fixed to 50 in all simulations. The reason to fix

| Scenario size | Average density |
|---|---:|
| 200x200m | 75 % |
| 300x300m | 60 % |
| 400x400m | 50 % |
| 500x500m | 40 % |
| 600x600m | 33 % |
| 800x800m | 23 % |
| 1000x1000m | 16 % |
| 1200x1200m | 12 % |
| 1400x1400m | 9 % |
| 1600x1600m | 7 % |

Table 5.2: Scenario size and corresponding density values, when keeping the number of nodes constant at 50 and varying the maximum speed between 1 and 20 m/s.

this value at 50 was that it is the default setting in the simulator and the fact that 50 nodes are quite a few, still not too many and is the value used in most comparative studies. More nodes would add extra complexity to the simulations and hence require more valuable time when simulating.

### 5.3.6 Data packet size

It is assumed in this thesis that the density metric would come to show its characteristics when the ad hoc network is congested. When congestion occurs, the nodes have to fight for the available bandwidth in the shared media. In [BMJ+96] it is stated that a packet size of 1024 bytes should be enough to encounter congestion for all four protocols. Therefore the packet size was chosen to be 512 and 1024 bytes.

The data packet size is very closely related to the variables described in chapter 5.3.7, since the data packet size, sending rate and the number of sending sources together make up the total load fed into the network.

### 5.3.7 Sending rate / Number of sending sources

In all simulations the sending rate was fixed at 4 packets per second. The number of sending sources was varied between 10, 30 and 50, resulting in a total maximum constant bit rate (CBR) transfer rate of 20kB/s to 200kB/s.

### 5.3.8 Traffic type

One of the main advantages of ns is that its traffic descriptions and implementations are well implemented and documented. Ns has one of the best TCP implementations, being a simulator. This is of course very valuable when performing simulations. However in this case, TCP's flow control mechanisms such as slow start, congestion avoidance and fast retransmissions are not desired. They introduce overhead traffic and make it harder to calculate the amount of lost packets and evaluate the performance of routing protocols. This is why constant bit rate (CBR) traffic is used in all simulations.

### 5.3.9 Interface queue length (IFQ)

This specifies the number of packets that can fit in an interface queue, (a separate queue for each mobile node). If this queue fills up, packets will be dropped until there is space available in the queue. In all simulations the IFQ was set to 50 packets.

## 5.4 Simulation process

When it was decided how all the input parameters should be varied it was time to start the simulations. The DSR simulation was rather straight forward. Everything worked and all simulations went through the simulator without any problems. As was the case with AODV. The simulations failed only for a few scenarios.

With DSDV only about 50% of the 360 scenarios went through. For TORA it was even less. Also somehow, almost all routing messages were missing from both the DSDV and TORA simulations making it very hard to calculate any routing overhead at all. This bug was never fixed and the simulations were not rerun since there were so many other problems with the simulator anyway.

# Chapter 6

# Simulation Results

This chapter presents the results[1] from the simulations described in chapter 5.3. Two simulation sets were run, one preliminary set to decide what input variables to vary and a more exhaustive set to obtain some higher precision simulation data.

## 6.1 Preliminary simulations

Initially a smaller set of simulations were performed in order to find an intelligent way to decide which simulation parameters to vary and which to keep constant. This was necessary since the number of possible parameters to vary is quite substantial, as described in chapter 2.6.

These simulations were run with a data packet size of 64 bytes and most other parameters were fixed to their default values. It turned out that all protocols functioned fairly well at this low network load, even though the simulations were far from exhaustive. In fact, with that version of the simulator, the TORA simulations were unsuccessful because of problems with the simulator. This is why TORA is missing on these first diagrams. It should be pointed out, once again, that the simulator used for these simulations is not the same version as the one used for the second set of simulations. The one used for this set is the original extended ns version from CMU.

Figure 6.1 shows that the ratio between received and sent packets is 100% when the density is high. This means that every packet sent reached its target, a desirable property for routing protocols. As the density decreases, i.e., the scenario size is increasing compared to the number of nodes in the network, the number of packet drops increase and fewer packets can be delivered. This behavior corresponds well to what is expected since the physical distance between the nodes eventually may grow

---

[1] In several of the graphs presented, data points for some protocols are missing. The reason for this is that some simulations failed and did not leave any useful data. This is especially true for scenarios with much traffic, a lot of movement and the protocols TORA and DSDV. Most of the strange values in the graphs are also an effect of this.

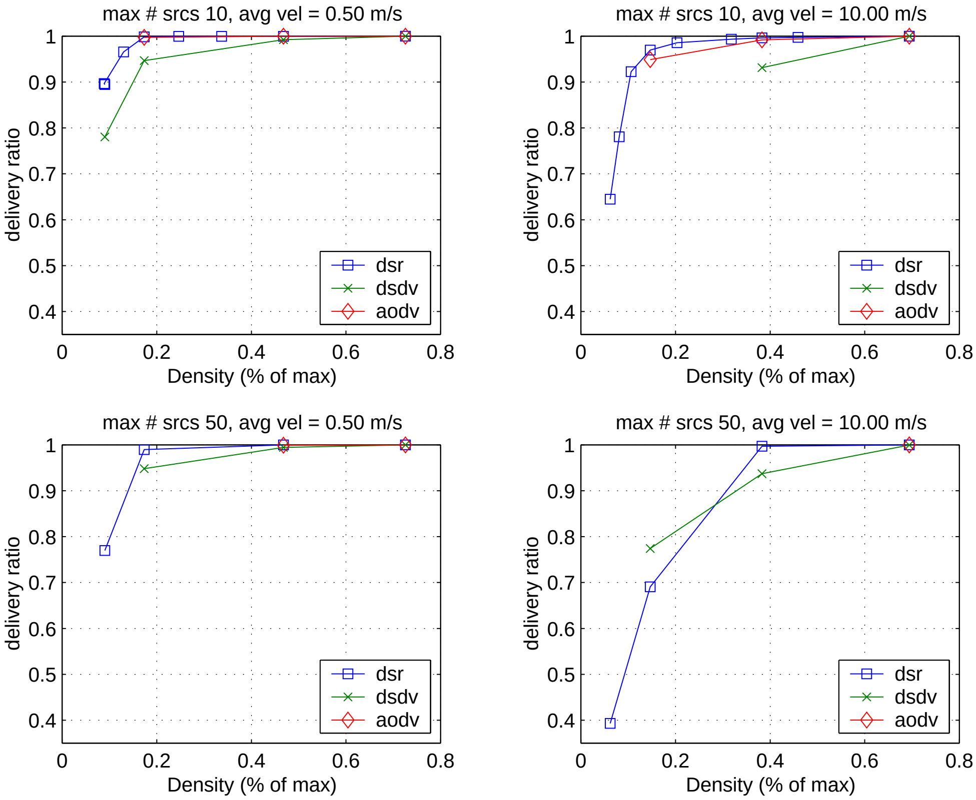


Figure 6.1: Packet delivery rate, 64 byte packets

too large for communication to take place.

In figure 6.2 the CBR overhead is shown. The number of overhead packets is defined as the number packets received at a node where the node itself is the destination for the packet. In other words the overhead is calculated as seen by the end point nodes. The corresponding number of bytes is calculated to include the DSR and AODV source route overhead[2]. It is clear that the relationship between density and the ratio *overhead packets/sent packets* is the same as in figure 6.1. The higher the density the less overhead traffic is generated. In the highest density scenarios the direct connectivity is very close to 100% (all nodes are within all the other nodes' transmitter range all the time). When this is the case the need for routing decreases since most links can be established on a point-to-point basis, meaning that the source can communicate directly with the destination. In a scenario like that, the need for routing messages diminishes.

---

[2] The validity of the DSDV and TORA/IMEP overhead graphs is therefore not confirmed

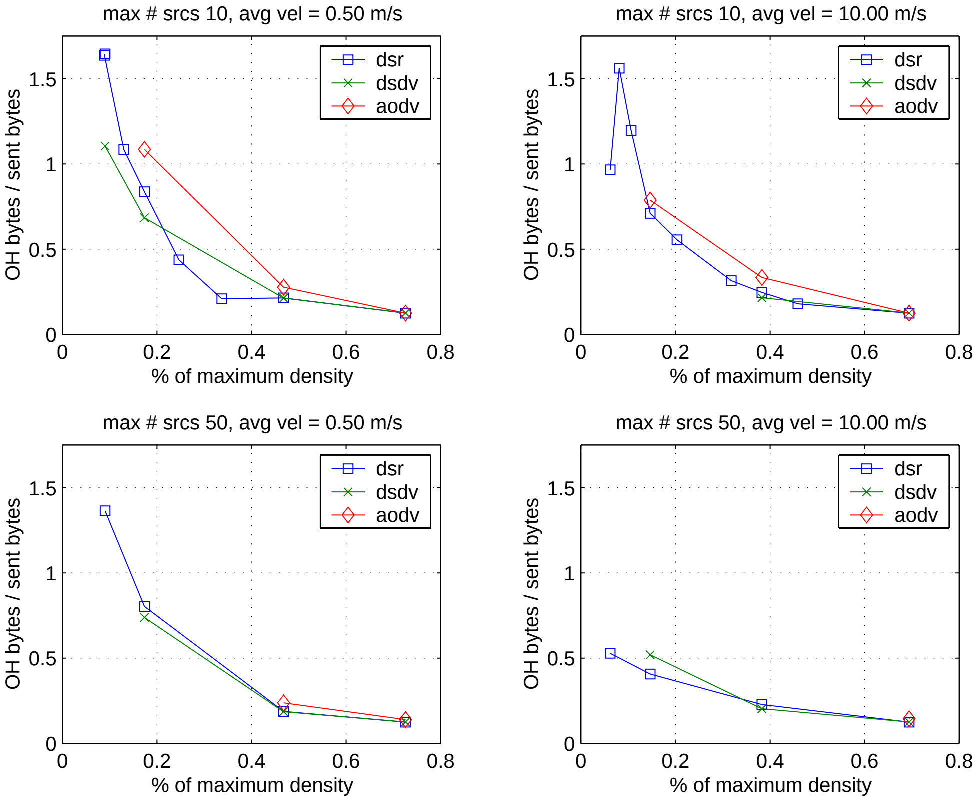


Figure 6.2: Overhead traffic, 64 byte packets

## 6.2 Simulations

As described in chapter 5.3, the second set of simulations were more exhaustive, varying more parameters and including three different runs for each scenario. Besides the more exhaustive testing scheme, the packet size was increased from 64 bytes to 512 bytes and 1024 bytes. This was expected to result in a decrease of the routing protocol performance since the network load was increased.

As the graphs on the following pages show, the expected results were achieved. No graphs for the 512 byte simulations are shown since 512 bytes was not really enough to congest the network to the desired level. In other words, the 512 byte graphs look very similar to the 1024 byte graphs where the maximum number of sending sources are limited to 10.

### 6.2.1 Graph explanations

#### Packet delivery rate graphs

The results in figure 6.3, show the CBR packet delivery rate for the 1024 byte communication patterns. The behavior is the same as in figure 6.1 (the 64-byte simulations); Lower density (larger scenario area) causes the delivery rate to drop. As the max number of transmitting nodes increase the network suffers from higher and higher congestion causing the delivery ratio to drop. In the two bottom graphs, where there are max 50 transmitting nodes, the network is so congested that even at 75% density not all packets were delivered. At 75% density all nodes are within transmitter range of each other in these scenarios.

From this graph it is seen that DSR and AODV have approximately the same packet delivery performance. In the few cases where DSDV and TORA data is present it clearly shows that the packet delivery performance of these protocols is worse than that of DSR and AODV.

#### Packet drop ratio graphs

The drop ratios displayed in figure 6.4 correspond well with the packet delivery ratio. In fact the values should be inverted since the packets that are not delivered are dropped somewhere.

#### Overhead byte graphs

Figure 6.5 show the amount of overhead bytes that are sent per byte useful data sent into the network. The overhead definition is the same as in the 64 byte simulation case. The higher the mobility and the greater the amount of data in the network, the lower the overhead. Where DSDV data is present it shows that DSDV has the least overhead data so this is a little different from the packet delivery rate where the DSDV and TORA performance was the worst.

It should also be pointed out that when comparing the amount of byte overhead from the 1024 byte simulations with the byte overhead from the 64 byte simulations, the relative overhead is the same in both cases. It looks as if the overhead in figure 6.2 is much more (approximately 1.5 byte per sent byte) but this is because the packet size is only 64 bytes. Since the relative number of packets sent was approximately the same, the routing overhead shows to be constant.

Another thing to notice is that as the number of sending sources increases, the overhead decreases. From this it can be concluded that the amount of overhead packets is constant. When the amount of data in the network is higher, the relative byte overhead is less.

## 6.3 Protocol performance

The graphs show a routing protocol behavior that is typical for these protocols. DSR and AODV has an overall performance that is good compared to that of DSDV and TORA/IMEP. This is also shown in other comparative studies [BMJ$^+$96, RT99, LHJ$^+$99] so the results were expected.

It is interesting to note that the routing protocol performance in general does not seem to suffer that much from an increased node speed. However, it should be noted, once again, that the two protocols that would really have problems with increased node speed, DSDV and TORA/IMEP, failed to simulate in most of the high speed scenarios and hence not much data is available to show this. Looking at the sporadic data points available in for example 6.3 (the top and bottom subgraphs) it shows that DSDV is not able to deliver as many packets as the other protocols, except in the low mobility scenarios (top left graph).

It is already documented by the CMU Monarch project that TORA produces a lot of overhead since it is using IMEP as an underlying protocol. The simulations performed in this work confirm this. Some of the TORA/IMEP simulations had to be interrupted because of the huge amount of output that the simulations resulted in. The largest trace file contained more than 2.4GB of uncompressed data...

## 6.4 Data acquisition

When the simulations finished, the result was a huge amount of trace files. All in all after the second round of simulations the resulting compressed trace files occupy more than 10 GB of disk. A short description of the trace file format is given here.

```
r 6.232315361 _7_ RTR --- 4 DSR 36 [a2 7 8 800] -------
[8:255 7:255 255 7] 2 [0 1] [1 1 2 7->8] [0 0 0 0->0]
```

The two lines above are one single line in the trace file. It describes a DSR routing packet being received by node 7, at the time 6.232315361. The packet size is 36 bytes. Node 7 is the originating node (it is a packet returning to the sender) but the packet is received from node 8. a2 is the expected time (in hex format, 162 s is the decimal notation) in seconds that it will take to send this packet over the wireless channel. The information in the next set of brackets is the IP addressing information for the MAC layer. The last two brackets holds the routing information that the route request resulted in. This is a typical route reply packet.

These next two lines:

```
s 6.232315361 _7_ RTR --- 0 cbr 1028 [0 0 0 0] ------- [7:1 8:1 32 8] [0] 0 1
```

is a packet trace for a cbr packet originating from node 7. It is very similar to the routing packet except that it does not contain any routing information.

Every trace file contains $\sim$500k – $\sim$45M lines like this. The graphs presented in this work were generated by parsing through all these files with perl scripts to collect statistics and then feeding Matlab with the results.

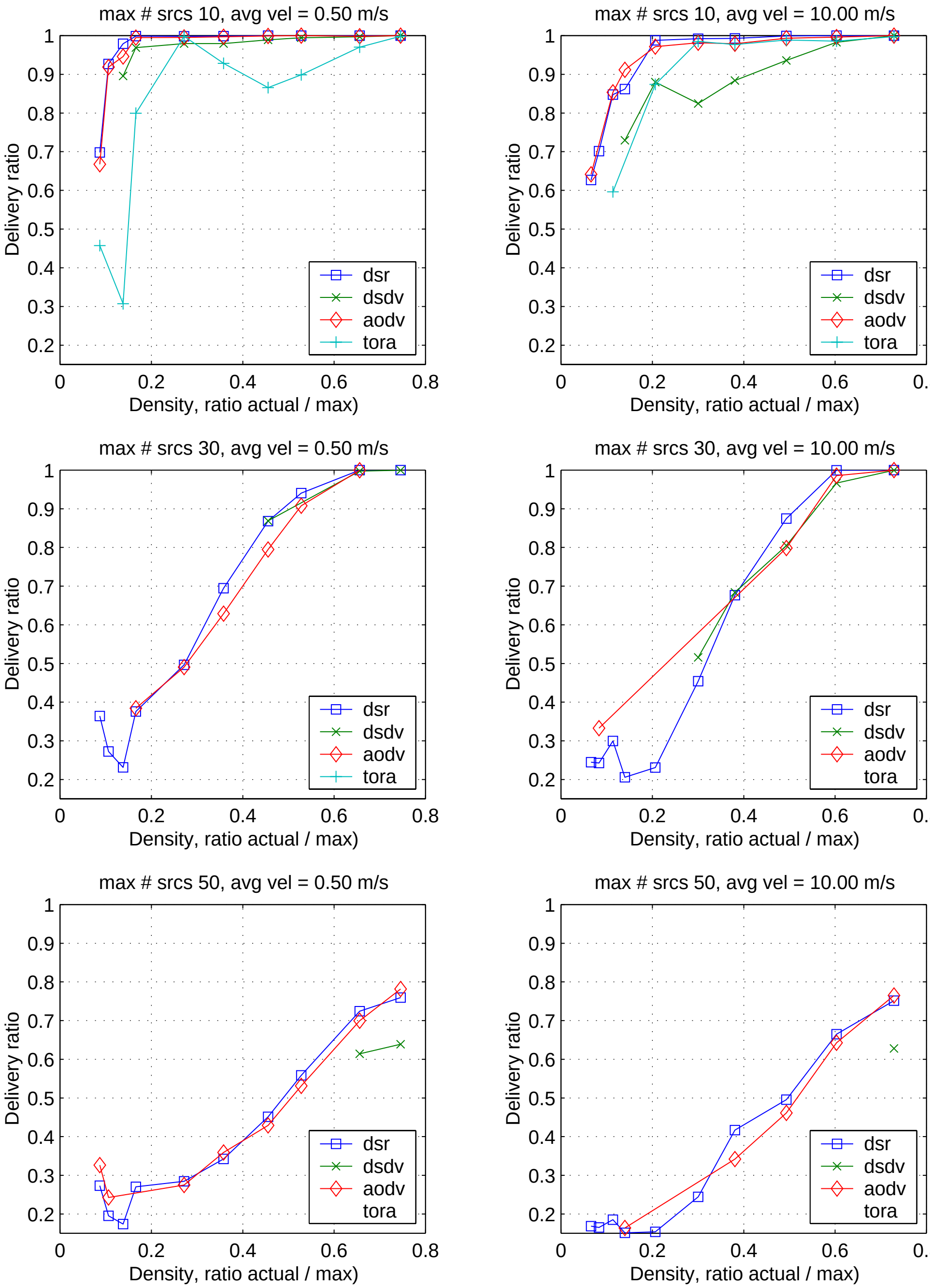


Figure 6.3: Packet delivery rate, 1024 byte packets

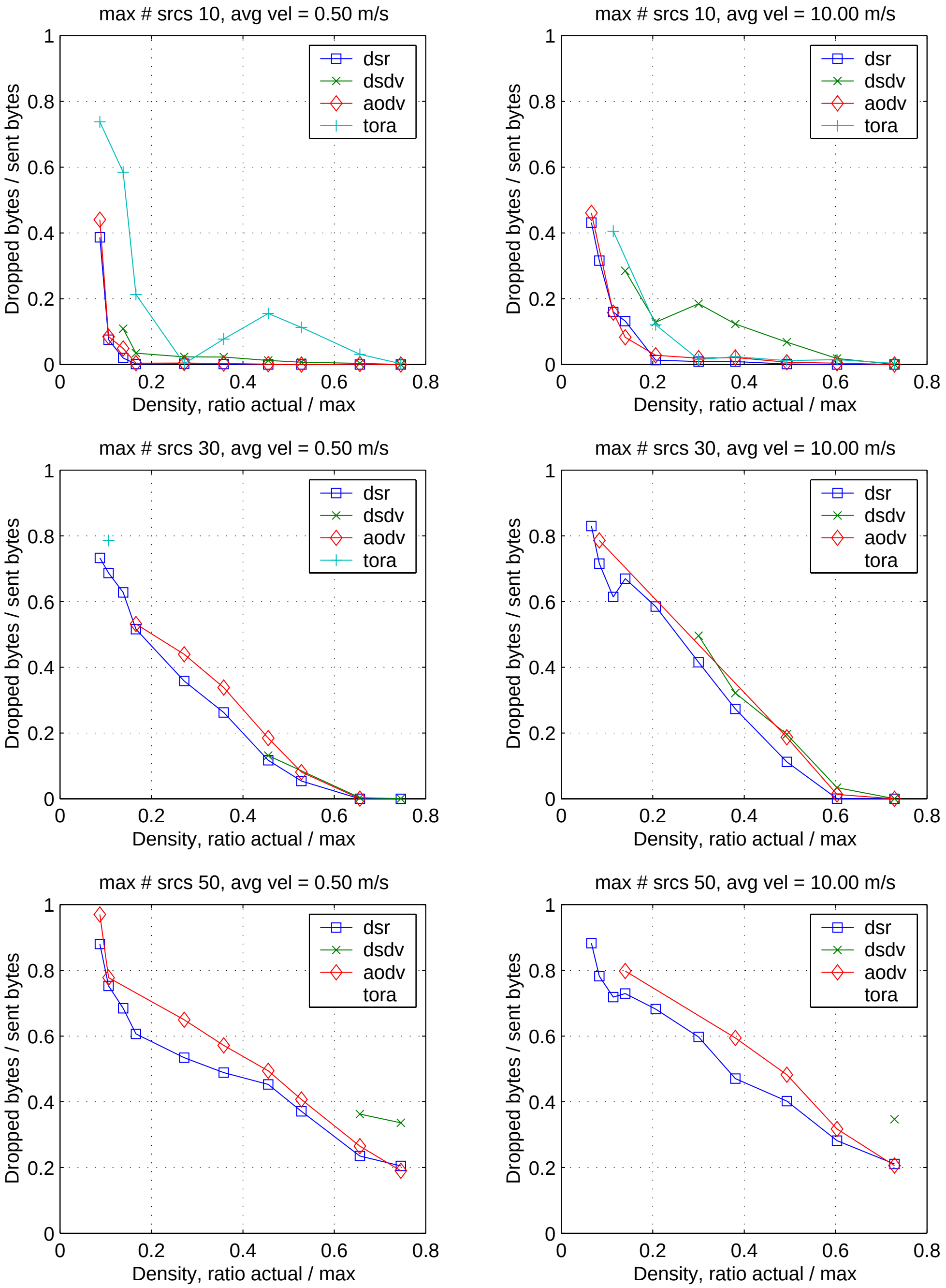
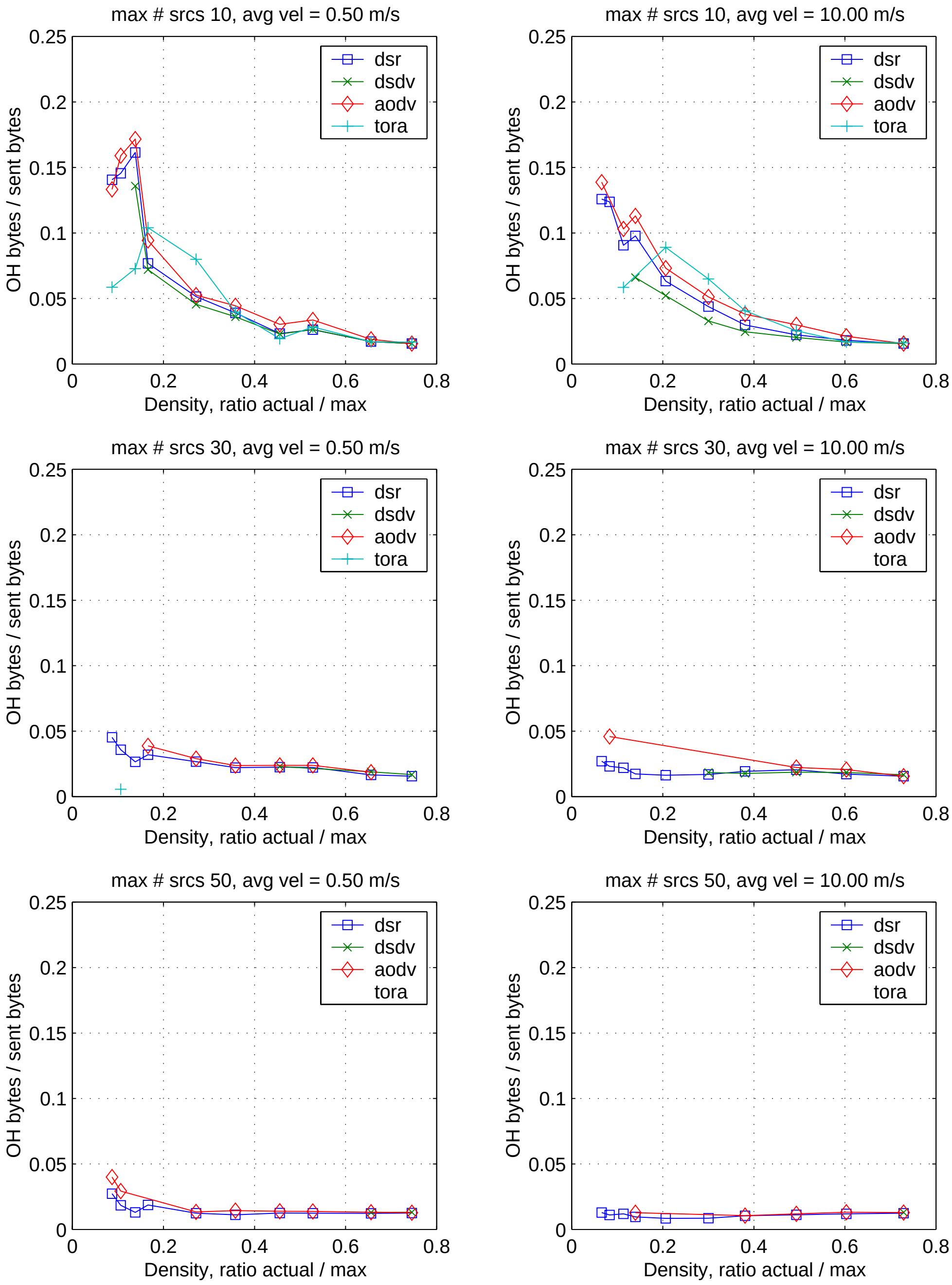


Figure 6.4: Data packet drop ratio, 1024 byte packets


max # srcs 10, avg vel = 0.50 m/s

max # srcs 10, avg vel = 10.00 m/s

max # srcs 30, avg vel = 0.50 m/s

max # srcs 30, avg vel = 10.00 m/s

max # srcs 50, avg vel = 0.50 m/s

max # srcs 50, avg vel = 10.00 m/s

Figure 6.5: Overhead traffic (bytes), 1024 byte packets

# Chapter 7

# Discussion and conclusions

## 7.1 The density metric

This thesis work was performed to find out if routing protocol performance would suffer in a very dense network. As it turned out, the simulations show the exact opposite.

One observation is true for all protocols simulated; The denser the network, the better the performance. The simulation results show that protocols are more negatively influenced by congested networks than dense networks. The fact that routing performance is decreased when a network is congested is not surprising since packets are dropped when the interface queues fills up as the network becomes congested.

However, the density metric fills a purpose anyway since it can be useful when comparing protocols. The validity of the density metric is discussed in chapter 4.3 and the simulation results presented in the previous chapter confirm this. It is certainly possible to see differences between the different routing protocols.

What would have happened if the density was changed by varying the number of nodes while keeping the scenario size constant? In the scenario with the highest density presented in this work (approximately 75%) the area is still very small, only 200x200 meters and 50 nodes. Since the nodes all have a transmitter range of 250m the nodes all have direct connection with each other all the time (at least 99.99% of the time to be exact). Achieving the same density in a larger network, say for example 1600x1600, would demand at least 3200 nodes in order to keep the same ratio of nodes/m$^2$. Even though this is not an unrealistic value it would certainly take *a lot* of time to simulate it. However, it would be interesting to see what happens when the density is really high but the nodes still need to send their data across the network in order for it to reach its destination. When the scenario size is 1600x1600 meters the 250m transmitter radius does not cover that much of the total area.

Another approach could be to modify the simulators parameters for the radio interface using the wave propagation model. That way the transmitter radius could be decreased. If the scenario size is then kept at the same size as before the change in transmitter range, the density would decrease. However, if the scenario is scaled down with the transmitter range, the density would be constant since it is normalized to a transmitter radius of 1 length unit. If it was intended to increase the density the number of nodes would have to be increased even further.

It is the authors opinion that the density metric introduces a new metric that is useful when comparing ad hoc routing protocols. This is shown by the simulations to be true for randomly generated scenarios. The metric include both a time concept and a mobility concept since both of these factors are considered when calculating the density value.

## 7.2 The direct connectivity metric

Since there are no simulations where the direct connectivity was varied it could be interesting to see what the average node connectivity is in the scenarios that were used for the second set of simulations.

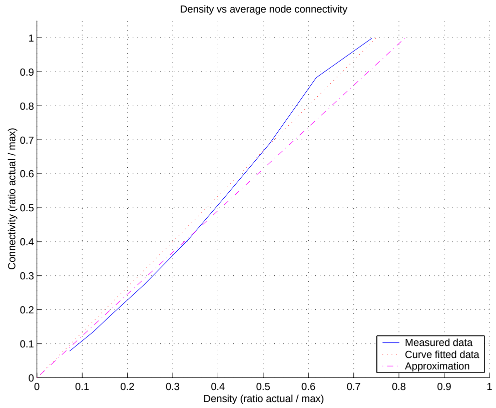


Figure 7.1: Comparison between density and direct connectivity

As figure 7.1 shows that when the density is varied from about 7% to 75%, the direct connectivity is varied from 7% to 100%. The direct connectivity in the scenarios generated here is obviously proportional to the density. The mathematical relationship can be expressed as:

$$conn = \frac{4}{3} density \tag{7.1}$$

Equation 7.1 is a result of linear curve fitting in the graph. The equation also corresponds well to the explanation given below. In figure 7.1, equation 7.1 is plotted as a dotted line for comparison.

When the density is low, the direct connectivity should also be low since the average distance between nodes would be longer than the communication range. It is also clear that at 100% density, the direct connectivity would also be 100%. When increasing the distance between the nodes, density will decrease as well, but the direct connectivity will stay at 100% as long as the nodes are not able to move further away from each other than the transmitter radius. The linear proportionality is a rational behavior since both the density and the direct connectivity is ultimately dependent on the distance between two nodes. The slope of the dotted line is the the one calculated in equation 7.1. This slope of the fitted curve is not far from the constant that equation 4.1 and 4.2 yield when the distance between the two nodes is set to one circle radius. The result is

$$\begin{aligned} cos\ \alpha &= -\frac{1}{2} \\ A &= \alpha - sin\ \alpha \end{aligned}$$

The dashed line in figure 7.1 show this slope. The measure calculated here is the area of two unit circles overlapping each other exactly with one circle radius, i.e., direct connectivity!

In theory, it is possible to create a scenario with density but without direct connectivity. Even though such a network would hardly be useful at all, it is worth mentioning since the relationship just described would not fit in. Hence, equation 7.1 is only applicable on randomly generated scenarios, such as those used in these simulations.

Generally speaking, the direct connectivity might be a better metric than density since it also includes the notion of communication ability. Even better would be, a combination of both.

## 7.3 The simulations

In retrospect it would have been smart to decrease the simulation time from 900s to 250s since the simulations would have run much smoother and statistically less problems would have occurred, since at 900s simulations, quite a few failed after about half the time. Also, since the second set of simulations basically just confirmed the results from the first set of simulations, the second set wouldn't have had to include so many simulations.

The differences in simulation results between 900 and 250 seconds would probably negligible since all the scenarios are generated to produce random motion.

## 7.4 The simulator

Ns is a widely used simulator within the networking community, however it is constantly under revision and there are parts, especially concerning the mobility extensions that still are not fully implemented nor bug free.

The people at CMU did a massive amount of simulations for their research presented in [BMJ+96] and they have made their source code publically available. Unfortunately the development of ns since then has made it almost impossible to use their old version, and some major changes have been done in the implementation to fit newer versions of ns. So about halfway through the thesis work the old simulator was exchanged to a newer version.

Working with the simulator in order to have it function has been a struggle. It took quite a while before it was functioning the way it was supposed to. Even though most of the work has consisted of setting up the correct script variables it was more of a hassle than expected. Finally, when the simulator seemed to work and the simulations were started it turned out that DSR was the only protocol that worked flawlessly.

It seems that the current status of the implementation is that all four routing protocols are implemented but most of them, especially DSDV and TORA seem to suffer from bugs. Some simulations just get stuck in endless loops. Sometimes, these loops include print statements causing the output file to grow uncontrollably. When this happens that simulation output is not much to keep, and it slows down the simulation process that the simulators, especially when they run on 30 different machines, need constant attention (someone killing the simulations that fail).

### 7.4.1 Modifications

Ns requires a certain amount of hacking from the user and I am not an exception to that rule. The modifications that needed to be done to complete this work was mostly modifications to scripts for scenario generation and statistics collection from the trace files. However, serious bug fixing in ns need to be done for TORA and DSDV. This was not a task included in this thesis and hence nothing was done about it.

## 7.5 Not a routing problem?

Looking at the simulation results of for example TORA, with the massive amount of overhead generated, it has to be pointed out that it might not be the best option to do routing in such small scenarios. Take a look at BlueTooth for example. It solves many of these issues already on the link layer, as soon as two nodes (a source and a destination) are in direct contact with each other, no routing has to be performed. At that point routing protocol comparisons would be superfluous.

# Chapter 8

# Future work

Since the simulations performed to compile this work were not exhaustive due to bugs in the simulator, filling the blanks in the simulation data would be desirable. In order to do this some bug-fixing in the ns implementation needs to be done, at least regarding the wireless extensions.

As this document was written, several routing protocol specifications were presented by MANET [IET] and it would be interesting to see how these new protocols would behave in simulations and what results they would have upon the work in this thesis.

Further testing to evaluate the density and connectivity metric could be done. For example trying to find more relationships between these new metrics and other metrics. In addition to this, comparative tests between density and connectivity should be performed to see if one of the metrics should be preferred over the other.

# Appendix A

# Glossary

**ABR** - Associativity Based Routing

**ad-hoc** - means *for a specialized purpose only.* Ad-hoc networks are networks without centralized control, meaning that they can emerge wherever there are nodes that can communicate with each other.

**AODV** - Ad hoc On demand Distance Vector (See chapter 2.5.3)

**CBR** - Constant Bit Rate. CBR traffic is a kind of traffic that keep the bit rate constant, say for example 64 bytes, 4 times per second or something similar.

**CBRP** - Cluster Based Routing Protocol

**CEDAR** - Core Extraction Distributed Ad hoc Routing algorithm

**CGSR** - Clusterhead Gateway Switch Routing

**DSDV** - Destination Sequenced Distance Vector (See chapter 2.5.1)

**DSR** - Dynamic Source Routing (See chapter 2.5.2)

**full connectivity** - means that all nodes have some way to communicate with each other during the whole scenario. Or one could also say that there are no partitions in the network.

**IMEP** - Internet MANET Encapsulation Protocol

**internet draft** - A working document of the IETF. They are valid for six months and should then be revised. Drafts should always be referred to as *work in progress.*

**LMR** - Lightweight Mobility Routing

**MAC** - Medium ACcess Layer - This is what the physical network layer is called with another name. Typical MAC layers can be 802.11 or BlueTooth.

**MANET** - Mobile Ad hoc NETworks

**multicast** - Sending data packets from one source to many destinations without having to set up a different data-path to each and every destination.

**multi-path routing** - A routing protocol that supplies multi-path routing will try to find different routes to its destination making it possible to chose another way to the target in case of link failure.

**node** - A device able to communicate with other devices in a network. Examples of nodes could be cell-phones, laptops or palm tops.

**OLSR** - Optimized Link State Routing

**PDA** - Personal Digital Assistant. A PDA is a small computerized notebook or organizer that will fit in your palm or pocket.

**piggy-backing** - Attaching a useful load of data onto a normally very small acknowledge packet.

**QoS** - Quality of Service - This is the general term used when talking about differentiated services in a network. In a QoS enabled network it is possible to achieve lower delay or a guaranteed bandwidth, if the user is prepared to pay extra for it.

**RFC** - Request For Comments - Is a series of notes used to document the development of the Internet, mostly communication issues, protocol standards etc. It was started in 1969.

**SSR** - Signal Stability Routing

**STARA** - System and Traffic dependent Adaptive Routing Algorithm

**route** - A path from a source to a destination.

**TORA** - Temporally Ordered Routing Algorithm (See chapter 2.5.4)

**WRP** - Wireless Routing Protocol

# Appendix B

# Romberg integration

## B.1 Definition

This method[1] to approximate $\int_a^b f(x)\,\mathrm{d}x$ uses Richardson extrapolation, starting with the trapezoidal rule

$$T(h) = \frac{h}{2}(f(x_0) + sf(x_1) + \ldots + 2f(x_{n-1}) + f(x_n)), h = \frac{b-a}{n}$$

with truncation error $\approx kh^2$

$$T_1(h) = \frac{4T(h) - T(2h)}{3} \qquad\qquad T_2(h) = \frac{16T_1(h) - T_1(2h)}{15}$$

The calculations can be organized according to the following scheme

$$\begin{array}{lllll}
T(8h) & & & & \\
T(4h) & T_1(4h) & & & \\
T(2h) & T_1(2h) & T_2(2h) & & \\
T(h) & T_1(h) & T_2(h) & T_3(h) & \\
T(h/2) & T_1(h/2) & T_2(h/2) & T_3(h) & T_4(h/2) \\
\quad\vdots & \quad\vdots & \quad\vdots & \quad\vdots & \quad\vdots \\
\xrightarrow[(2^2–1)-\text{rule}]{} & \xrightarrow[(2^4–1)-\text{rule}]{} & \xrightarrow[(2^6–1)-\text{rule}]{} & \xrightarrow[(2^8–1)-\text{rule}]{} &
\end{array}$$

---

[1] Definition is based on the definition given in [RW90].

## B.2 Implementation details

For anyone interested, the c implementation of the integration is included below. Several previously declared variables and functions are used in the implementation without any further explanation.

```
/* Calculate the Romberg numbers in the first column*/
for (i=0;i<RM;i++) {
  if (i==0) {
    d1 = distance(n1,n2,a);
    d2 = distance(n1,n2,b);
    d_result = (d1+d2)/2;
    R[i * RM +0] = (h[i]/2.0)*(area(d1)+area(d2));
  }
  else {
    part = 0.0;
    for (j=1;j<=(1 << (i-1));j++) {
      d1 = distance(n1,n2,a+(j*2-1)*h[i-1]/2);
      part += area(d1);
    }
    R[i*RM+0] = 0.5*(R[(i-1)*RM+0]+h[i-1]*part);
  }
}

/* Calculate the rest of the romberg numbers */
for (i=1;i<RM;i++) {
  for (j=1;j<=i;j++) {
    /*
     *        j-1
     *       4   * R       - R
     *             k,j-1    k-1, j-1
     * R   = -------------------------
     *  k,j        j-1
     *            4     -   1
     *
     */

    /* 4^(j-1) becomes 2^2j since we need to do 4^((j+1)-1)
     * This is all becuase we use 0-based indexes here
     * for the Romberg numbers */

    R[i*RM+j]=((1<<(2*j))*R[i*RM+j-1]-R[(i-1)*RM+j-1])/((1<<(2*j))-1);
  }
}
```

# Bibliography

[BJM99] Josh Broch, David B. Johnson, and David A. Maltz. The Dynamic Source Routing protocol for mobile ad hoc networks. Internet draft, Internet Engineering Task Force, October 1999. Work in progress.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt*

[BMJ+96] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. *IEEE Personal Communications*, 3(1), February 1996.

[C+97] Ching-Chuan Chiang et al. Routing in clustered multihop, mobile wireless networks with fading channel. In *The IEEE Singapore International Conferens on Networks (SICON'97)*, pages 197–211, Singapore, April 1997. IEEE.

[CM99] S. Corson and J. Macker. RFC2501 - Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations. Request For Comments, January 1999.
**URL:** *http://www.ietf.org/rfc/rfc2501.txt*

[CMU] CMU. Cmu monarch project: Mobile and wireless networking. Web page. 1999-11-12.
**URL:** *http://www.monarch.cs.cmu.edu/*

[FV99] Kevin Fall and Kannan Varadhan. *ns Notes and Documentation.* UC Berkeley and Xerox PARC, 1999.

[GK97] Piyush Gupta and P.R Kumar. System and Traffic dependent Adaptive Routing Algorithm for ad hoc networks. In *Proceedings of the IEEE Conference on Decision and Control*, volume 3. IEEE, December 1997.

[Hil95] Stefan G. Hild. A brief history of mobile telephony. Technical report, University of Cambridge, Computer Laboratory, Jan 1995.

[HP99] Zygmunt J. Haas and Marc R. Pearlman. The Zone Routing Protocol (ZRP) for ad hoc networks. Internet draft, Internet Engineering Task Force, June 1999. Work in progress.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-02.txt*

[ICOC] S.p.A. Ing. C. Olivetti & C. Information about active badges. Web page. 2000-03-07.
**URL:** *http://www.cl.cam.ac.uk/abadge/documentation/abwayin.html*

[IET] IETF. Mobile ad hoc networks (manet) charter. Web page. 1999-09-11.
**URL:** *http://www.ietf.org/html.charters/manet-charter.html*

[JLT99] Mingliang Jiang, Jinyang Li, and Yong Chiang Tay. Cluster Based Routing Protocol (CBRP). Internet draft, Internet Engineering Task Force, August 1999. Work in progress.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-cbrp-spec-01.txt*

[LH98] Tony Larsson and Niklas Hedman. Routing protocols in wireless ad-hoc networks – a simulation study. Master's thesis, Luleå University of Tecknology, 1998.
**URL:** *http://epubl.luth.se/1402-1617/1998/362/index.html*

[LHJ+99] Tony Larsson, Niklas Hedman, Per Johansson, Bartosz Mielczarek, and Mikael Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of the fifth annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 195–206, Seattle, August 1999. ACM/IEEE.

[LSG00] Sung-Ju Lee, William Su, and Mario Gerla. On-Demand Multicast Routing Protocol (ODMRP) for ad hoc networks. Internet draft, Internet Engineering Task Force, January 2000. Work in progress.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-odmrp-01.txt*

[MJLA96] Shree Murthy and J.J.Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *ACM Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks*, pages 183–197, October 1996.

[PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of ACM SIGCOMM '94*, pages 234–244. ACM, August 1994. 9/9/99.
**URL:** *http://www.cs.umd.edu/projects/mcml/papers/Sigcomm94.ps*

[PC97] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM '97 Conference*, Kobe, Japan, April 1997. IEEE.

[PD96] Larry L. Peterson and Bruce S. Davie. *Computer Networks – A systems approach.* Morgan Kaufmann Publishers Inc., San Fransisco, 1996. ISBN: 1-55860-368-9

[PRD00] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Ad hoc On-demand Distance Vector (aodv) routing. Internet draft, Internet Engineering Task Force, March 2000. Work in progress.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-05.txt*

[Rap96] Theodore S. Rappaport. *Wireless Communications: Principles & Practice*. Prentice Hall, Inc., New Jersey, 1996. ISBN: 0-13-375536-3

[RT99] Elizabeth Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[RW90] Lennart Råde and Bertil Westergren. *Beta $\beta$ Mathematics Handbook*. Studentlitteratur, Lund, 2nd edition, 1990. ISBN: 91-44-25052-5

[SSB99] Prasun Sinha, Raghupathy Sivakumar, and Vaduvur Bharghavan. CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm. In *Proceedings of the IEEE INFOCOM Conference*, New York, March 1999.
**URL:** *http://www.cs.columbia.edu/~hgs/papers/infocom99/239-438162214.ps.gz*

[Toh97] Chai-Keong Toh. Associativity Based Routing for ad hoc mobile networks. *Wireless Personal Communications Journal - Special Issue on Mobile Networking & Computing Systems*, 4(2), March 1997.

[UCB] UCB/LBNL/VINT. UCB/LBNL/VINT network simulator - ns (version 2). Web page. 1999-09-11.
**URL:** *http://www-mash.cs.berkeley.edu/ns/*