Studienarbeit
Contacts in MANETs

Studiengang:Informatik
Prüfer:Prof. Dr. rer. nat. Kurt Rothermel
Betreuer:Dipl. Ing. Abdelmajid Khelil
begonnen am: 01.05.2003
beendet am:
Mohamed Foued Krichene

31st October 2003

Abstract

Mobile ad hoc networks (MANET) are constitued by mobile devices equiped with short range radio. Communication is possible between devices within each other's radio range. The mobility leads to frequent topology changes in such networks, which harden typical networking tasks, such as routing or information diffusion. Hyper-Gossiping is an information diffusion protocol that uses mobility to distribute information even in partitioned MANET.

In analogy to spreading of infectious diseases, where contact patterns strongly impact the characteristics of spreading, we investigated contacts between mobile devices and defined new mobility metrics, such as contact rate, duration and frequency. Using these mobility metrics, nodes are now able to better understand underlying mobility patterns and to discover new partitions.

We used a contact-based metric, i.e. contact-rate, to improve repetitive behavior of Hyper-Gossiping in highly partitioned networks. Simulation results show that we could reduce up to 16% of needed rebroadcasts to disseminate an information.

Acknowledgements

My personal thanks to my advisor Abdelmajid Khelil for his direction and kind assistance during last months. Further on i thank Illya Stepanov for his Perl-scripts to support simulation and evaluation of simulation results.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

This chapter motivates our work and gives an overview of our objectives and our approach.

1.1 Motivation

Many flooding (and routing) protocols have been developed for mobile ad hoc networks in last few years. Most of them have recognized the importance of considering node mobility as key component for information diffusion. Since MANETs are infrastructureless and nodes lacks a global view of the whole network, it becomes a challenge to ensure best reachability with minimal overhead. Redundancy and collision of packets are not only resource wasting (e.g. batterie lifetime) but they also cause congestion of wireless medium. This problem is refered to as the broadcast storm.

Enabling adaptive protocols requires continuous feedback from the network, which can be realized using mobility metrics. In fact, mobility metrics gained an increasing interest in many protocol solutions. Especially link-based metrics have been concretely used in some protocol implementations, because they give a clear feedback about node mobility, and they are simple to compute. Unfortunately, these metrics fail to recognize movement pattern of single identifiable nodes. In many real world scenarios, indeed, nodes should be identifiable in order to understand the mobility pattern correctly, e.g. the link between two cars following each other in a highway may be broken and built again within a short time span. That is why we believe that mobile nodes can save unnecessary rebroadcasts and important resources if they have a more accurate overview of current network mobility.

1.2 Objectives

Objective of current work is to investigate contacts between nodes in MANETs and provide new mobility metrics based on these contacts. We believe that these metrics are protocol independent and give a better comprehension of the network topology and mobility without higher overhead than that needed for ordinary metrics. These metrics based on collecting and analyzing link information of identifiable nodes we refer to as *contact-based mobility metrics*. Contact-based metrics differ from link-based primarily in that, that single nodes are identifiable. Collected information about contacts during a predefined time period (*called* history or experience of node) should help mobile nodes to better adapt their protocols (e.g. parameters and thresholds, or switching criteria between different broadcasting schemes) to the dynamic network topology.

Further goal of this study work is to use contact-based metrics to improve the Hyper-Gossiping protocol, which uses node mobility to discover new partitions.

1.3 Approach

Outgoing from simple epidemiological analysis, the notion of *Contacts* in MANETs is introduced and defined. Contacts in MANETs are namely a certain view of encounters between individuals in real world scenarios. Proposed contact-based mobility metrics should satisfy a variety of requirements to be useful for different network topologies and protocol implementations.

Information about encountered nodes will be collected by every node using a simple mechanism (e.g. hello-beacons). So we let each node maintain a local database of time, duration of its last encounters with all other nodes. Network protocols (here Hyper-Gossiping) consult database to make *conclusions* about the *current mobility pattern* in the network, and eventually adapt their behaviour. Finally, we evaluate the enhanced Hyper-Gossiping in ns-2 and present a detailed analysis of its performance. We will add some new conditions and constraints for a better partition recognition. The new protocol should minimize unnecessary rebroadcasts without worsing the reachability.

Chapter 2

RELATED WORK

MANETs have been extensively studied in last years. However efficient routing and flooding in remains always an actual challenge. This chapter presents in Section 2.1 the notion of *contacts*, like it was introduced in other works. Section 2.2 deals with improvements on routing protocols using link-based mobility metrics; and finally Section 2.3 gives a global review on existing flooding techniques.

2.1 Contacts

The word *"Contact"* has a big influence in a variety of real-world domains: Business, sociology, epidemiology etc. We are here interested in the spreading of information in MANETs, which are constitued by mobile nodes that are equiped with a short range radio and can communicate if they are in each other communication range. These nodes are uniquely identified (e.g. using MAC-Adress). Given an adequate contact between two mobile nodes (happens if these nodes are in transmission range of each other), information can be exchanged. Outgoing from these reflections we realise that there is a big similarity between spreading of information in a mobile network and spreading of infectious diseases. Let's start now by describing some main characteristics of epidemic contacts.

2.1.1 Epidemic Contacts

Modeling infectious diseases started before 1900, the first dynamical system appeared around 1927 (Kermack and McKendrick). It is of interest in our work to study the containment strategies which in turn depend from the individual contact patterns.

Cohen et. al found out that the magnitude and spread of an epidemic is basically affected by the geography, the state und nature of the population affected and the temporal introduction and amplification of the virus (Cohen, et. al, 1996). In fact, Contact patterns play in epidemic diseases a major role. So, some diseases can only exist if people live very closely together (measles, the Black Death). Many vary with the way the society is organised (day-care homes, schools, family size). Some spread mainly in groups with special contact patterns (HIV)... (Johan Giesecke 2002)

The classical epidemic SIR model describes the infection and recovery process, and divides the nodes in three categories: *susceptibles* $S = u$, *infected* $I = v$ and *recovered* $R = w$ *[Hadel96]*. Let's try to make some comparison with spreading information between mobile nodes. The infected nodes are those that received the information, have stored it and are still able to forward it to other nodes[1]; while the recovered ones were infected in the past, but have *lost* interest about the information, i.e. they deleted the information from their cache and will not send it in the future to other neighbors. Susceptibles nodes represent all the rest. This system depends only on two parameters, the infection rate β and the recovery rate α. It can be described as follows (in its simplest form) [Hadel96]:

$$\begin{aligned}\dot{u} &= -\beta uv,\\ \dot{v} &= \beta uv - \alpha v,\\ \dot{w} &= \alpha v,\end{aligned}$$

Notice that in our model the population is assumed to be constant, which means that $\frac{d(u+v+w)}{dt} = 0$. The total population size $P = u + v + w$ is constant.

2.1.2 An Epidemic Model for Information Diffusion in MANETs

In [Khel02] authors found out that there is a strong similarity between Flooding-based approaches like diffusion and epidemic spreading of diseases [Khel02]. They prooved -using some simulations on mobile nodes- that the node density strongly influences the performance of flooding-based information dissemination. Therefore they proposed an adaptive approach based on the local density around a node in order to cope with the varying MANET charateristics [Khel02]. The application requirement in this work was resumed in reaching a certain ratio of nodes in a given time intervall.

Because there is a similarity between the dissemination of information among mobile devices and the transmission of infectious diseases between the individuals [Khel02], they tried to include existing mathematical models that describe epidemic processes in the description of the broadcasting process. Some epidemic models should then be used to describe and to adapt the information dissemination in MANETs. And in order to enable adaptation to the node density, an analytical expression for the infection rate of the MANET depending on node density was determined.

The developed analytical model for information diffusion allows a precise description of information diffusion in MANETs using few parameters. Now, assuming a node can perceive its environment (e.g. local node density and population size), it can process the infection rate of a given epidemic algorithm, using some analytical expressions. The node will be able to switch between different algorithms to adapt information dissemination to the needs of the application and to the current environmental situation in the MANET, e.g. the local node density[Khel02].

Node density was considered as feedback from the local environment to adapt the information diffusion algorithm to the current situation. In fact, node density is a good metric for different environments of MANETs; but node density doesn't describe the *mobility* in the mobile network. Being aware that mobility can also have a great influence on the spreading of information, we should search for other metrics that almost describe this phenomenon. To find this metric, we should first speak about *contacts* and *contact changes* within mobile nodes environments.

Contacts are for both transmission of information and diseases of big interest, because only when contact between nodes or individuals is granted, then communication is possible. In addition, contacts reflect movement of nodes and therefore provide potential metrics for mobility. Getting new contacts is important for the spreading process, otherwise the disease will be only transmitted in the local region and the information will not be able to cross the partition it belongs to initially. Only mobility makes it possible for nodes to encounter new nodes and to transmit the information to other parts of the network. In this context, it is important to determine which contacts are relevant for information diffusion, and how locally gained knowledge about contacts and contact changes can be exploited to ensure a better reachability.

---

[1] means that these node may send the information to their neighbors in the future

Next Section introduces *Small Worlds*, which are important for an efficient diffusion of information in sparse and large mobile network. It is just like the earth, where we live, which is formed of countries and continents. Infectious deseases can spread out dangerously if we take into account the *Small-World-Effect*. In the same Section is a protocol presented, that tried to take profit of the contacts-information collected in a mobile network in order to improve routing.

2.1.3 Contact-based Architecture and Small-Worlds (Zone Routing Protocol) [H-MARQ03, Helmy02, Bai03]

For resource discovery we may use simple protocols like *broadcasting* or *flooding.* This idea can be acceptable in small networks, but in very large networks with thousands of nodes, we should solve the problem efficiently. Let's make also some reflections about mobile networks!

MANETs can be described as regular networks, where *Clustering Coefficient* and *Path Length (average degree of separation)* are high, which means that two randomly chosen nodes will have many hops between them, and that neighbors of one node are with a high probability neighboors of each other. Adding a short number of *Short-Cuts* drastically reduces the average *path length* of the graph. *Short path length* and *high clustering coefficient* are typical characteristics of *small world graph* [2] *(six degrees of separation).* (see Figure 2.1)

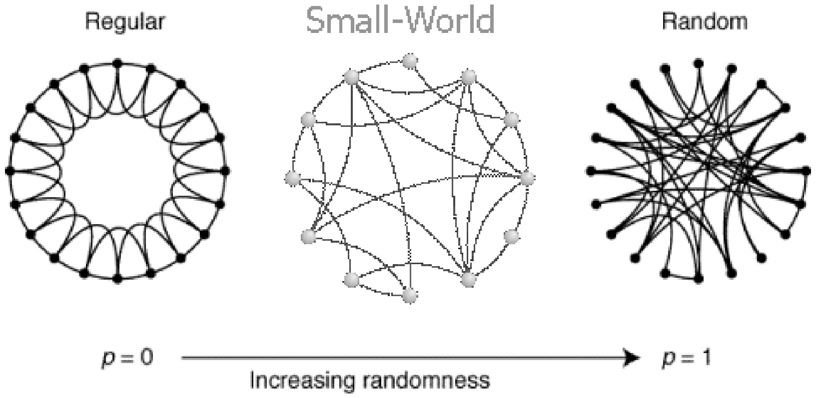


Figure 2.1: *Small World* graph: Between *regular* and *random* graph [Watts98]

Helmy et. al [H-MARQ03] used the concept of Small-World-Networks in order to improve Routing in mobile networks. The idea is to construct a Small-World using information collected about other nodes in the network. The question is: "How to construct a small world without having *global information* about the network; and how shall we choose *Contacts*?" Choosing random nodes will lead to unpredictable overhead. Every node has to choose *contacts* from his neighborhood (up to R hops away, called *edge* nodes), so it knows their characteristics. When they move, these nodes will have a network view with less overlap. If they are still maintaining contact with each other (r hops away, where $R < r < r_{max}$), we will get a better network coverage, and this graph will tend to a *small world graph. Contact* can be maintained using *periodic polling*, where each node sends a *validation* message to each contact, containing the respective path. This path will be updated, until the contact is too far away or cannot be reached, and then it is considered to be lost.

In the *mobility-assisted resolution of queries (MARQ)* architecture, each node uses a proactive protocol to maintain information about other nodes in its *zone,* up to R hops away. The protocol

---

[2] *Small world networks* may appear in differents domains like biology, sociology, www, etc. But they are not the goal of our study.

has to choose and maintain useful contacts with reasonable overhead. During a query, instead of using flooding to search for an answer, only the contacts are queried for information they maintain about their zones. The contacts may in turn query their contacts, and so on, until the answer is found. A salient feature of this architecture is that it takes advantage of mobility to select far away contacts to increase the efficiency of query resolution [H-MARQ03]. The idea behind zone-based routing using the notion of *contacts* based on the *small world graphs* phenomenon and metrics of stability and mobility [Helmy02] can be shown on Figure 2.2.

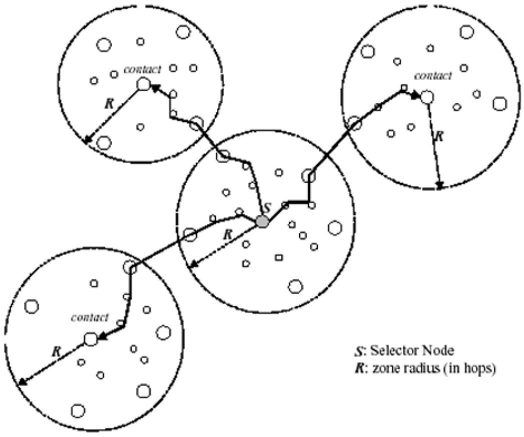


Figure 2.2: A node in the contact-based architecture chooses a small number of contacts outside its zone to increase its network view during query resolution or resource discovery [H-MARQ03]

Major contributions of this work lie in the introduction of the concept of *contacts* as short cuts in the wireless network (that attempt to construct a small world). It differs though from our study, because the goal of [H-MARQ03] is to avoid global flooding, while we aim to enable efficient global flooding. Furthermore our definition of contacts differ from their.

2.2 Improvement of MANET Routing Protocols Using Link-Based Mobility Metrics

In previous years, many works have coped with the design of routing protocols on MANETs. To send a given message from A to B, a *route* must be chosen. First important requirement for this route is its "minimal" length[3]. And because the transmission of data can take some moment of time, it is important to take the *Link Stability* into consideration. Next works considered *nodes mobility* as main reason for link instability (especially for routing or route discovery). Using mobility metrics to measure the mobility aspects of the local environment should be helpful to adapt protocol parameters to the estimated mobility models.

2.2.1 Efficient Route Discovery Using Encounter Ages [EASE03]

When a source node first wishes to establish a route to a destination it has to search in the whole network, until it finds either the destination or a node, who *recently* has seen the destination. The simplest way is to use flooding mechanism, where a packet is flooded across the network, possibly

---

[3] Route length should tend to be minimal, but for efficiency reasons the choose of minimal routes cannot be granted.

using an *expanding ring search* to "grow" the flood until the destination is found. And because the source doesn't have any information about the possible location of the destination, it has to send the packet in all directions.

Contacts are deployed in this work to enhance *packet routing* (using efficient location lookups)and to present a simple algorithm for *efficient route discovery* (flood-based) in mobile ad hoc networks. Basic idea is that each node has to keep a record (*history*) of their most recent encounter times with all other nodes in the network. Instead of searching for the destination, the source node searches for any intermediate node that encountered the destination *more recently than did the source node itself*, iterately. So this algorithm replaces the single network-wide search of current proposals (i.e. using global flooding) with a succesion of smaller searches, resulting in a cheaper route discovery. (see Figure 2.3)

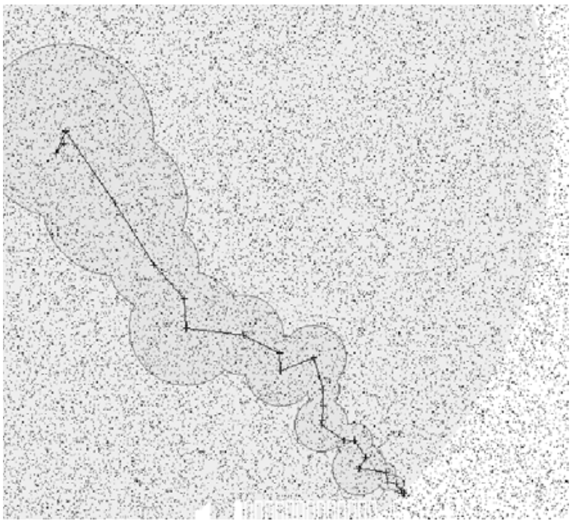


Figure 2.3: An example route for N=3200 nodes, with a random walk mobility process. The destination is in the lower right side. The darker surface covers the union of the minimal search disks that are necessary at each anchor point to find the nearest node which has encountered the destination more recently. The lighter circle, centered at the source, covers the minimal search disk that would be used in a single-step flood [EASE03].

An encounter between two nodes happens when those nodes are one-hop neighbors. The *encounter age* of two nodes is the time elapsed since they meet each other the last time. The algorithm uses a simple idea, which can be described in: "a node that was my neighbor 5 minutes ago is probably closer to me than a node that was my neighbor 5 hours ago".

The performance of this algorithm depends on node mobility process, but more concretely it depends on the *homogeneity* of the mobility processes. Specifically, if node velocities are very heteregeneous, then the relationship between encounter age and distance become very noisy. *Rescaling* of the node velocities does not affect the performance, because this will correspond to only rescaling the time, and therefore the encounter age.

2.2.2 Link Stability in Mobile Wireless Ad Hoc Networks [Gerh02]

Many protocols in ad hoc networks suffers from the frequently changing network topology, leading to many instable connections between different nodes. In fact, *Rerouting* is high costly in infrastructureless networks, since it results mostly in flooding the network. However, many applications require stable connections to guarantee a certain degree of QoS[4] [Gerh02]. On this account, many

---

[4] Quality of Service

studies was especially dedicated to develop methods that help choosing -from a set of links- the link that may have the highest residual lifetime.

- Previous Work

  Some approaches are based on measuring signal strength, distances or relative speed (by exchanging hello messages) between mobile devices, which makes it possible to predict the link availability for the next time. Some methods does require GPS-System (e.g. to get own position in order to calculate the distance to other nodes), which is in many situations like indoor rooms not possible, furthermore it is resource wasting. Other approaches measure the actual lifetime of a link using hello messages which are periodically broadcast. The assumption is that stable links are links that have existed in the past for a minimum time t_{thresh}. This threshold may depend on the relative speed of both devices or simply fixed to a certain time. Longer links may indicate that nodes are moving with the same velocity[5], which suggest that they will still keep neighbors for a relatively long period of time. In this case, links that are suggested to be stable will be chosen for building of routes.

- New Approach and Improvement [Gerh02]

  The simulations with many mobility models and the analysis of link lifetime have shown that the key to stable link selection is much more complex than just selecting the oldest link. The distribution of link durations (which can be observed in Figure 3.3 on page 30) shows that older links occurs relatively seldom. It can be even concluded that the residual lifetime of a link is ruled by its current age [Gerh02]. In fact, the average residual link lifetimes for young links decreases constantly with every second the link becomes older. In other words, the remaining lifetime decreases with increasing age. Some simulations on Manhattan Grid scenario have shown that links between 50s and 75s old, have a high probability (75%) to disappear almost immediately. It may be correct that some of the remaining links may stay a very long time alive; but real-world actions (which are of short duration) are not interested in how long a link *might* live, but for link stability und reliability for a rather short period of time. Those links are to be chosen, that have good chances of being available for a certain period of time.

  The objective now is to enable each node to estimate residual lifetime based only on observations of link lifetimes in the past. Gerharz et. al developed an equation that enables every mobile device to compute the probability for a link to stay available for a well-known time span s. As a consequence nodes have better chances to choose links, that are more stable for next time. These suggestions have been confirmed by many simulation results and comparisons with other techniques.

2.3 Broadcasting or Flooding Techniques (Categorization of Protocols)

Broadcasting is the process where one node sends a packet to all other nodes in the network. Broadcasting is frequently used in MANET routing protocols. For example, many unicast routing protocols such as Dynamic Source Routing (DSR), Ad Hoc On Demand Distance Vector (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR) use broadcasting or a derivation of it to establish routes [Will02]. A simple way for Broadcasting is *Flooding.*

Problems can be faced if two neighbors try to broadcast at the same time using the same channel; this will result probably in packets collision. Collision avoidance is inherently difficult in MANETs [Will02]. And because broadcasting nodes have no way to know if neighbors received successifully the packet, the efficiency in such protocols can be gained by trying to limit the probability of

---

[5] Same speed and same direction

collisions by limiting the number of rebroadcasts in the network [Will02]. Problems in broadcasting in MANETs can not only be caused by a high density of nodes, but also in MANET with a low node density or in case of network partitioning, where nodes can not communicate since they are likely to be out of each other's radio range. Hence, a flooding phase is likely to stop before all nodes are reached. Repeating the message transmission over a period of time, which is called hyper-flooding, can help to cope with such network partitioning [Khel02].

Because of the highly dynamic characteristics of MANETs [Sas03], it becomes a challenge to find new mechanism, which may solve the waste of resources, time and efficiency, caused by *simple* flooding mechanism. So, many studies tried to optimize flooding. In the next Section we describe *plain flooding*; and in following sections we present some optimizations for flooding protocols.

2.3.1 Simple Flooding (or Plain Flooding)

Broadcasting is an essential communication primitive in MANETs, for example for route discovery. The usual approach for broadcasting is flooding. Flooding requires no topological knowledge, which suits well for MANETs. It consists in that, that each node rebroadcasts a message to its neighbors, upon receiving it for the first time [Sas03].

The algorithm starts with a source node broadcasting one packet to all its neighbors. Each receiver node rebroadcasts the packet exactly one time until all reachable nodes are informed. Simple (or plain) flooding algorithm provokes a high number of unnecessary packet rebroadcasts, causing contention, packet collisions and ultimately wasting precious limited bandwidth [Sas03].

Previous work has demonstrated that Simple Flooding is relatively insensitive to mobility [Will02].

2.3.2 Probability-Based Methods

- Probabilistic Scheme (ro Gossiping)

  Similar to simple flooding, except that nodes only rebroadcast with a predetermined probability. In order to flood, a node in the network broadcasts the message with probability *p,* and takes no action with probability *1-p* [Sas03]. Having some nodes that do not rebroadcast, saves node and network resources, while delivery effectiveness is maintained. When p is 100% the algorithm behaves the same way like flooding.

- Counter-Based Scheme

  Algorithm is based on a study made by Ni et al (See [Will02]), where they prove an inverse relationship between the number of times a packet is received at a node and the probability that this node is able to reach additional area on a rebroadcast. After receiving a previously unseen packet, a node starts counting how much redundant packets will be delivered until a certain time limit T is reached. If the counter is less than a certain threshold value after time expires, then the node will rebroadcast the received packet; otherwise not. The idea is efficient and simply based on the fact, that in dense areas of the network, some node won't rebroadcast, while in sparse areas of the network all nodes rebroadcast [Will02].

2.3.3 Area-Based Methods

Algorithms using this method consider the coverage area of a transmission. So consider two nodes separated by only 1 meter. One of them is broadcasting a packet. If the receiver rebroadcasts the same packet, he will probably reach no new nodes. But if the rebroadcasting node is located at the boundary of the sender nodes's transmission distance, then a rebroadcast would reach significant additional area, 61% to be precise [Will02].

- Distance-Based Scheme

  Before rebroadcasting, a node compares the distance between itself and all neighbors that have previously rebroadcast a given packet[6].To do that, a node starts a time counter after receiving a new packet. While counter is still living, the node caches all redundant packets and calculate his distance to all senders. If any node is closer than a distance threshold, then there is no need to rebroadcast, otherwise rebroadcast [Will02].

- Location-Based Scheme

  Estimates the expected additional coverage area before deciding to rebroadcast using e.g. a Global Positioning System (GPS). This is done by adding own location to the header of a packet when sending it [Will02].

2.3.4 Neighbor-Knowledge Methods

These protocols can be classified by whether a node makes a local decision to retransmit a broadcast protocol [Will02]. In all following protocols either a node makes itself this local decision, or it is told whether it needs to retransmit a broadcast packet or not.

- Flooding with Self Pruning

  A simple protocol, which requires that every node has knwoledge about its 1-hop neighbors, which is obtained via periodic "hello"- messages. A node includes the list of his known neighbors in the header of the packet, before broadcasting it. Receivers compare their list with the sender's list; and decide by a high analogy of these lists not to rebroadcast, otherwise to rebroadcast.

- Scalable Broadcast Algorithm (SBA)

  Algorithm requires that all nodes have knowledge about their 2-hops neighbors, which is obtained in the same way via periodic "hello"- messages, which contains in their header a list of all sender's neighbors. After a node receives "hello"- packets from all its neighbors, it gets a list of his neighbors and the neighbors of his neighbors, which are 2-hop neighbors.

  When a node receives a broadcast data packet from one of his neighbors, it determines if he can reach new nodes using his knowledge of sender's neighbors. If yes, then it schedules the packet for a time intervall T[7]. If it then receives a redundant packet from another neighbor, it determines again if it can reach new nodes by rebroadcasting. This process continues until either time expires and the packet is sent, or the packet is dropped [Will02].

- Dominant Pruning

  Algorithm uses 2-hop neighbor knowledge too. Unlike SBA, the sender chooses -using certain algorithm- which receiver will have to rebroadcast the packet in order to cover 2-hop neighborhood. When a node receives a broadcast packet, it checks the header to see if its address is part of the list. If so it uses the same algorithm to determine which neighbors should rebroadcast; then it rebroadcasts the packet with the corresponding list [Will02]. A variant of this algorithm is "Multipoint Relaying".

- Ad Hoc Broadcast Protocol (AHBP)

  Analog to previous protocol, only nodes which are designated as a Broadcast Relay Gateway (BRG) within a broadcast packet header are allowed to rebroadcast the packet. BRGs are proactively chosen from each upstream sender which is a BRG itself [Will02]. The Algorithm for a BRG to choose its BRG set differs a little from *Multipoint Relaying* in aiming to minimize the overlapping of broadcasting areas.

---

[6] Using signal strength to determine the distance between sender and recipient[Will02].

[7] Time intervall to wait is called in the literature "Random Assessment Delay" (RAD), which is randomly chosen from a uniform distribution between 0 and T_{max} seconds[Will02].

Other variants of the Neighbor Knowledge Method are *CDS-Based Broadcast Algorithm* and *LENWB,* which will not be considered here. (See [Will02])

Conclusion:

In sparse networks, the protocols are expected to perform similarly to flooding, as each node may have to rebroadcast to reach isolated neighbors. As density increases, proportionally fewer nodes should rebroadcast [Will02].

2.3.5 Hyper-Flooding

In reality, there is another categorization we should mention here. That is, *flooding protocol* does require from every node that it rebroadcasts the packet just one time. You sure have noticed, that all protocol optimizations seen above require that a node rebroadcasts a packet less than once with the goal to save resources and to minimize overhead. We now will see a third variant which allows the network nodes to rebroadcast a packet more than once. This is called *Hyper-Flooding Protocol.*

Because Hyper-Flooding scheme allows more than one rebroadcast per node, the number of rebroadcasts per node will increase. Regardless of this fact, this protocol performs well in highly mobile scenarios where high reliability is a prime importance [Khel02]. Nodes periodically send "Hello"-messages to all neighbors, which allow them to construct a list of "all" current neighbors. Upon reception of a new packet, a node retransmits it and queues it in a packet cache. It can later rebroadcast the same packet, if it receives any packet (broadcast packet or "Hello"-message) from a new node not listed in his current neighboring list [Khel02].

This scheme may be effective in highly mobile networks and during special cases, like network partitioning or a car in the overtaking lane in a high way.

2.3.6 Adaptive Schemes

Adaptive approachs allow nodes to switch among different broadcast protocols on the fly. They use a trade-off between reachability and efficiency relying on current network situation. Main goal is to solve the broadcast storm problem by minimizing redundant rebroadcasts. Two approachs will be presented here:

1. Adaptive Flooding Protocol

   Nodes using this protocol can switch between scoped-, plain- and hyper flooding depending on the current network conditions perceived by each node. In ... *relative velocity* and *network load* have been used as criteria to switch between different flooding techniques.

   - Scoped Flooding Mode is used in environments where nodes are not likely to move too much (e.g. conference). It is based on a comparison of sender's and own lists of neighbors (using hello-messages). If first list is a subset of the second, then no rebroadcasting is needed (For simulation purposes only 85% overlap was considered sufficient).
   - Hyper Flooding Mode is used in highly mobile scenarios
   - Plain Flooding

2. Adaptive Counter-Based Scheme

   This scheme is based on origin idea of counter-based protocols (See 2.3.2). Problems that faces old version in sparse networks should be solved here. The idea is that each node can adjust its threshold C based on its neighboring information. Depending of its current number of neighbors n each node has to determine if to rebroadcast or not. Threshold is then a function of n: $C(n)$.

2.3.7 Hyper-Gossiping

This is the protocol we will investigate and improve using contact-based mobility metrics[8]. The protocol is based on a similar idea like Hyper-Flooding. In fact, Hyper-Gossiping allows nodes to rebroadcast a packet more than once. It is based on two major strategies for diffusion in MANETs, which are:

- Restrictive: To overcome *broadcast storm problem*[9] (**Gossiping** [Haas02]).
- Repetitive: To overcome *partitioning* (**hyper** flooding).

The repetitive strategy can be resumed that each node has to compare current neighbors with the potential receivers that the node thinks about they shall have received that packet. Are the sets different, then the node needs to rebroadcast the cached packet again. Each entry in the cache table contains the delivered message and the set of potential receivers S_i (ancien neighbors). Let S be the current neighbor list of the node.

The diffusion algorithm Hyper-Gossiping HG(p) operates in the following way:

```
on_receive (new_msg), do restrictive broadcast: if (0<=random()<=p) {broadcast(new_msg);}
on_changes_on_current_neighbor_list: if (($S_i \cap S = \emptyset$)&&(0<=random()<=p))
{$S_i \leftarrow S$; rebroadcast(cached_msg);}
```

We gave hier a brief overview of this protocol. For more details see Section 5.1.1 on page 51.

---

[8] The protocol is being developped by Abdelmajid Khelil in IPVS/VS department.

[9] The *broadcast storm problem* can occurs when the node density is too high and many nodes are repeating their incoming messages (like in Flooding), which results in radio transmission blocking out messages [Khel02].

Chapter 3

MOBILITY METRICS

Mobile ad hoc networks face many challenges to guarantee successful communication between mobile nodes. Because all nodes are autonomous (apart from e.g. geographical constraints) in choosing their velocities, they often may encounter new nodes, or may leave groups and join other groups. The primary challenges in MANETs are mobility and constraint resources (e.g. bandwidth, energy, etc.). But the major challenge which distinguishes MANETs from all other network models is *mobility* [Bol02]. Protocols that are able to adapt themselves to the continuous network changes, are also needed. This is only possible if mobile nodes are able to *measure mobility*. Thus, we focus our interest on *feedback* of the current *local* mobility characteristics of the network. *Measuring mobility and providing feedback* were at the center of many earlier works in order to enable adaptive protocols. In this chapter iwe study existing *mobility metrics*, and then present some new metrics based on what we call contact.

3.1 Introduction

To enable easy comparison or real-time adaptation of diffusion protocols, a mobility metric, that quantifies the effect of node movement is needed. This metric should be able to accurately estimate diffusion protocol performance and should be easily computable on real mobile devices without global network knowledge. Previous works have defined mobility metrics which paritally satisfy these requirements (e.g. mobility model parameters, average or relative node speed, link change rate, link duration, minimal route change metric).

A *mobility metric* provides feedback by quantifying the effect of node movement which signals the communication potential of the network[1] [Nav02]. This metric should be computed in real time by network nodes, and *must be independent* of movement patterns and network model. Mobility model independence of the metric is very important as regards the generality of the protocol, and that node does not have any knowledge about the mobility model.

As you will see, various mobility metrics have been proposed and new ones should be proposed in this work. Not all existing metrics do satisfy our requirement, that we list in following Section and which must be respected by every *mobility metric*, in order to be really useful in real-time implementations. After that a detailed overview about most existing mobility metrics is given. We will also show, that these metrics are insufficient to model mobility and to adapt *Hyper-Gossoping*. New mobility metrics that use *contact-based* information will be presented in the last Section of this chapter.

---

[1] A Metric should indicate *performance* and enable *adaptive protocols*.

3.2 Terminology

Many words that we currently use in this work have been used in earlier studies, sometimes with different and ambigous meanings. So we first begin by defining the most important key words, and then explain briefly the utility of using contacts for providing *mobility feedback*.

Neighbor

> In a mobile network, we should distiguish between physical and logical neighbors. Where physical neighborhood describes the fact that two nodes enter in the sending range of each other, the logical neighborhood can be first detected when a node delivers a message from another node in its range.
> In our simulations we will use physical neighborhood. Two nodes are also neighbors if they enter in the sending range of each other. For simulation purposes, there will be no restriction in form of *minimum time* used for neighborhood. In real network implementation a *minimum time* for neighborhood should be introduced to avoid oscillating status (Neighbor / Not Neighbor) for some nodes while entering transmission areas of other nodes.

Link

> A link between two nodes is considered to be established when the nodes come within each others transmission radius, and is considered to be broken when their distance exceed the transmission radius [Gerh02]. This assumption is based on ideal radio conditions. In other words, a link is considered to be established for the whole time, in which both nodes are neighbors. A link is always determined by its two involved nodes, its *time of incidence* and its *duration*.

Contact/Encounter

> A contact between two nodes begins with the first link established between them (first encounter), and ends with last one (last encounter). Consequently, a contact includes all links between both nodes, that have been registered in the history. This is only possible if nodes are identifiable with a unique ID-Number. Contacts are dependent from length, begin and end of the statistik-time interval. Two nodes encounter (=have contact) if they are 1-hop neighbor.

Speed

> Speed is a measure of how fast an object is moving. Speed is the scalar value of the velocity vector.

Velocity

> Velocity represents an object's speed and direaction at a single moment of time. Velocity is a vector (It includes therefore an absolute value and a direction).

Statistical time interval

> The time duration used to average the link connections [Nav02]. For our simulation purposes this may be equal to the simulation time. For some metrics it is important to choose a convenient value.

3.3 General Requirements for Mobility Metric

We list here some requirements that a mobility metric must meet in order to enable adaptation of MANET protocols at run-time (See [Nav02]). A mobility metric should be:

1. Computable in a distributed environment and in real network implementations wihtout global network knowledge: Metric should be locally computable without asking for information from other network nodes. Otherwise required communication could explode drastically, which is a big problem because of the limited resources. Simulation specific parameters should also not be used in calculating metrics (Like pause time in Random Waypoint Moobility Model).

2. A good indicator of protocol performance: A mobility metric must be able to indicate or predict the protocol's performance. For example, as the mobility metric's value change, this would indicate a corresponding change in delay. Thereby the protocol parameters customization will be facilitated.

3. Easy to compute (in terms of node resources): Because mobile nodes may have limited energy (battery power), limited processing ability or limited memory availibility.

4. Independent of any specific protocol: We call it a general purpose mobility metric. That makes it possible to be used by any protocol as a feedback mechanism for adaptive operation.

3.4 Overview of *Mobility Metrics*

In this Section, we will present a variety of metrics that have been used in previous works and explain why new ones should be introduced. These metrics are based on the neighboring information collected by every node in its *encounters list.* This presentation should allow us to easily classify the various metrics, so that we later can choose the convenient metric for the problem or protocol in question. Most of those metrics do conform to the requirements mentioned above. The few metrics that do not fulfill these requirements will be presented too in order to give a larger overview for the user, but they will not be used in our work.

Most of the mobility metrics have been used in adjusting routing protocols. Routing protocols are very much affected by node mobility. A higher arbitrary mobility in the network means that nodes change their neighbors constantly, and thus existing routes can only be maintained for smaller time of periods. The mobility metric *link duration* has shown its efficiency in many protocol implementations. This and other efficient implementations have been shown in Chapter 2 on page 15.

Figure 3.1 shows a typical movement scenario that we will principally use in our simulation tests. Every node in the network has to maintain a detailed list of all its encounters in the "past".

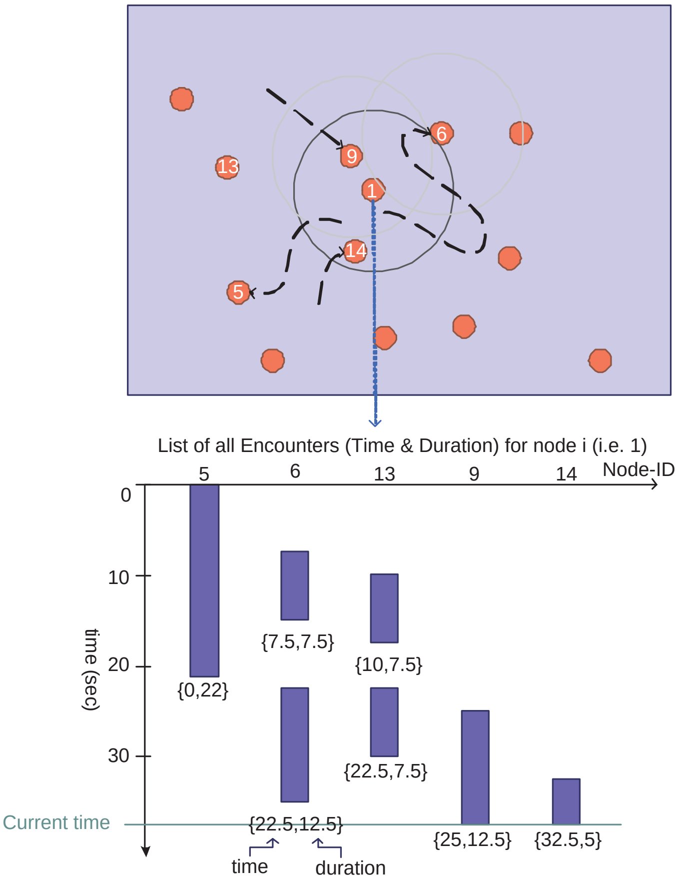


Figure 3.1: A typical simulation environment of a randon waypoint model; and a visualisation of the informations saved by a node about its last encounters.

So we first begin by defining some key words, and then list most well-known *mobility metrics,* that can be of interest for providing mobility feedback, which is important for adaptive protocols.

3.4.1 Existing Metrics

Many metrics have been used in the past with the aim to adapt routing protocols to the possible changes in the network topology due to node mobility. Those metrics do not only reflect the mobility of the network but also provide feedback about the structure of the network, and can help measuring the performance of protocols.

1. Speed-based Metrics:

   - *Node max Speed* or *Pause Time*[2] [Nav02]: Both metrics fail requirement 1 (See 3.3 above), because they require simulation specific parameters, which may not be accessible in a real network.
   - *Average relative speed between all nodes* ($\overline{RS}$) [Nav02][Bai03]: The average magnitude of relative speed of two nodes over all neighorhood pairs and all time. It requires that the nodes exchange their absolute speed, which violates requirement 1 and 3.

---

[2] for the random way point mobility model

$$RS = \frac{1}{count}\sum_{t=0}^{T}\sum_{i=1}^{N}\sum_{j=1;j\neq i}^{N} |v(i,t) - v(j,t)|\; if\; dist((x_i,y_i),(x_j,y_j)) \leq R \text{ (R is transmission range)}$$

- *Relative speed only among neighborhood*: Does not contradict requirement 3. But has the problem that the node must know its absolute speed. This may be possible in some *real world scenarios* (like *Cartalk*[3]), which make it possible to exchange the known absolute speed in order to calculate the relative speed. But in other scenarios (like people walking on the street), this is not possible[4]. This restriction makes this metric not useful for a wide range of real world scenarios.
- *Average* or *Maximum Speed*: Like the previous metric above, it requires that a node knows its own speed, which is not possible in MANETs without using infrastructure. Another disadvantage of this metric is that it doesn't reflect the relative mobility [Kwak03]. In case of using the same average speed while changing the mobility model will lead to the same metric results, and therefore to the same chosen decision.
- *Spatial Dependence* [Bai03]: The average function value of angle of relative velocity of two nodes over all neighborhood pairs and all time

$$C = \frac{1}{count}\sum_{t=0}^{T}\sum_{i=1}^{N}\sum_{j=1;j\neq i}^{N} \frac{min(v(i,t),v(j,t))}{max(v(i,t),v(j,t))} \times \frac{v(i,t) \bullet v(j,t)}{|v(i,t) \parallel v(j,t)|}\; if\; dist((x_i,y_i),(x_j,y_j)) \leq R$$

2. Link-based (Single-Hop Connectivity) Metrics:

   The idea is based on using link informations experienced by every node. There exists a link between two nodes, if they are in each other transmission range.

   Starting point is a list of encounter entries; and each *encounter entry* is a pair of *encounter-time* and *encounter-duration* $\{t_i, d_i\}$. This builds a distribution of link durations. Figure 3.2 shows the distribution of the link duration based on the graphical example shown in Figure 3.1.

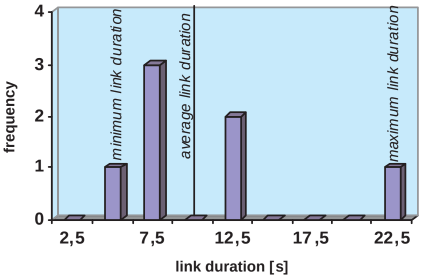


Figure 3.2: Distribution of Link Duration. This shows an example of the distribution for node 1 based on the example presented in Figure 3.1.

Average Link Duration is a typical metric, that can be gained from this distribution. Here a list of most used link-based metrics:

---

[3] I refer here to the project *CarTALK 2000*, whose goal to increase transport safety using cooperative driving between cars.

[4] Assuming that GPS or other positioning methods can not be used; which is downright correct in MANETs.

- Link Rate:
  - *"Actual"*[5] *(New) Link Rate*: Rate of getting new links during a short predefined time interval. It shows if the local environment is changing or being stable (If we consider the last 10 seconds, we get $Actual\,New\,Links\,Rate = \frac{1}{10} = 0,1\,links/sec$).
  - *Average (New) Link Rate (ALR)* [Nav02]: Number of all new links during a longer predefined interval averaged over the interval length. This is a good metric of how dynamic is the mobility of the nodes. ($ALR = \frac{\#new\,links}{Time} = \frac{7}{37{,}5} = 0,186\,links/sec$)
  - *Actual Link Change Rate* (Same like *"Single-Hop Connectivity Change Rate "* [Song02]): Rate of all link changes (getting or loosing links) during a short predefined time interval. Changes occur, when nodes either move into or out of transmission range from each other
    (If we consider the last 10 seconds, we get $Actual\,Link\,Change\,Rate = \frac{3}{10} = 0,3\,changes/sec$).
  - *Average Link Change Rate* (*ALCR*): Number of all link changes in the past averaged over history time. ($ALCR = \frac{\#link\,changes}{Time} = \frac{12}{37{,}5} = 0,32\,changes/sec$)

  There is no need to notice the *average link loss rate*, because it can be easily computed using above metrics (by subtracting ALR from ALCR).

- Link Duration (or *average duration of single-hop connection*)**:**

  The (actual) duration of one link is calculated as the time that two nodes are within transmission range of one another [Nav02]. In real implementations each node has to calculate this metric independently. The determination of the appropriate statistical time interval is also important for averaging link durations. Choosing a long statistical time interval leads to past link durations influencing actual link duration heavily. Choosing a short time interval may leads to an instable, high varying feedback. The determination of the statistical time interval is of high importance [Nav02].

  Figure 3.3 shows a typical link duration distribution for the random waypoint mobility model.

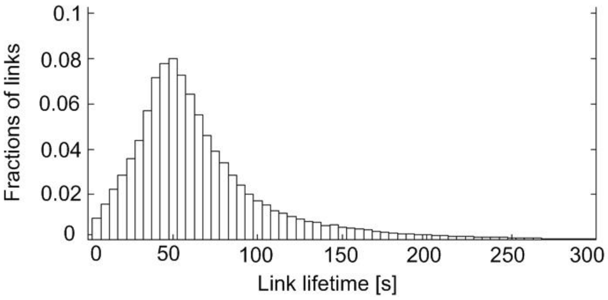


Figure 3.3: Distribution of Link Duration. This shows the result of a simulation serie on a Random Way-point scenario done by a related work (See [Gerh02]).

  - *Minimum Link Duration*: measured in seconds, it represents the minimum duration registered for a link (in Figure 3.1 $Minimum\,Link\,Duration = 5sec$).
  - *Maximum Link Duration*: In real world networks a link can take a long time, especially if nodes are not moving or moving in a group (in Figure 3.1 $Max\,Link\,Duration = 22sec$).
  - *Average Link Duration (ALD)*: Is calculated on a per node basis by averaging the individual link durations experienced with all neighbors [Nav02]. ($ALD = 10,64sec$)

---

[5] Normally we consider a very small interval!

    - *"Actual" Link Duration*: This can be the duration of the last finished link, or an averaging of a few number of last links (If we consider the last 3 finished links we get $Actual\,Link\,Duration = \frac{7{,}5+7{,}5+12{,}5}{3} = 9{,}16sec$ for example of Figure 3.1).
    - *Standard Deviation of Link Duration*: The standard deviation is the square root of the variance. The variance is a measure of how spread out is a distribution. It is computed as the average squared deviation of each number from its mean. $\sigma^2 = \frac{\sum_{j=1}^{p_i}(d_{i_j}-ACD)^2}{\sum_{i=1}^{n} p_i}$($n$: # encountered nodes; p_i: how often node i was encountered; d_{i_j}: duration of link j for node i)

  Link Duration is defined in next Section in a different way. In this work we will refer to the definition above, that defines this metric as *mobility metric,* that can be calculated in real networks wihtout needing global knowledge.
  Both *link change rate* and *link duration* have been studied in many previous works. In [Bol02], while applied on a MANET routing protocol, it was shown that first metric *link change rate* does not reliably indicate the performance of the protocol (data packet delivery ratio); although *average link duration* metric shows a smooth and predictable relationship between link duration and protocol performance: Longer lasting links create more network stability, while shorter duration links create less network stability [Nav02]. This shows that *link duration* can be successfully used as an indicator of protocol performance for routing protocols, because longer lived links create a more stable network, which in turn allows for a higher delivery ratio of data packets [Bol02].

3. Route-Based (Multiple-Hop connectivity) Metrics:

   The metrics listed below are used to measure routing protocol performance, since routing protocols are affected by the network topology dynamics. These metrics can analyse the effect of mobility on the connectivity graph between mobile nodes [Bai03].

   - *Number of Link Changes* $LC(i,j)$: Number of link changes for a pair of nodes i and j is the number of times the link between them transitions from "down" to "up" [Bai03].

     Average Number of Link Changes: This is the number of link changes $LC(i,j)$ averaged over all node pairs [Bai03].
   - *Route Change Rate* (experienced by a node): In MANET Routing Protocols, nodes tend to construct routes to destination nodes. This metric measures the route change rate experienced by a node. It is very much dependent on the mobility and the mobility model of the nodes.

     Simulation results on [Nav02] shows that the performance of routing protocol (see performance metrics below) seems to decrease as the number of route changes each node experiences increase.
   - *Route Duration* (In the literature also called *"link duration"*!) $RD(i,j)$: It is the average duration of the route existing between two nodes i and j. It is a measure of the stability of the route between these nodes [Bai03].

     $$D = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1;j\neq i}^{N} \frac{\sum duration\,for\,route\,(i,j)}{\sum route\,changes\,for\,route\,(i,j)}$$

     Average Route Duration: It is the Route Duration $RD(i,j)$ averaged over all node pairs. It is a measure of the network stability.
   - Path Availability $PA(i,j)$: It is the fraction of time during which a path is available between two nodes i and j. The nodes pairs of interest are the nodes that have communication traffic between them [Bai03].

     Average Path Availability: It is the value $PA(i,j)$ averaged over all node pairs.

4. Density-Based Metrics:

   These metrics are "mobility independent", but does give an important feedback about the local environment. They are to be combined with other mobility metrics to give the right feedback.

   - *Actual Node Degree*: Node Degree is the number of neighbors seen by the considered node. A node of degree $d = 0$ is isolated, i.e., it has no neighbors. The *node degree* is considered as feedback from the local environment. It depends on the node density and the transmission power of nodes in the network.
   - *Node Degree* (*Average Number of Neighbors* or *Average Neighborhood*)

5. Using Global-View Information:

   Metrics based on global information violate our general requirements for mobility metrics because they are not easy to compute and need a high communication overhead. In real world MANETs, global information is only possible by exchanging messages between nodes. This should be avoided because it affects the communication resources very negatively. In addition, exchanging such information will allow nodes to get more precise information about position, distant neighbors and mobility schemes in other parts of the network, which will make the use of these mobility metrics superfluous.

3.4.2 Are These Mobility Metrics Sufficient?

The above proposed mobility metrics have been used in many routing protocols, and have caused a significant performance enhancement. But there is a variety of protocols and a variety of network situations, where those metrics don't help much. Because we currently work on a variant of flooding protocols, i.e. *Hyper-Gossiping* (See Section 2.3.7), we will try to adjust our researches to this goal. In fact, The Hyper-Gossiping Protocol in its current version does use some techniques to adapt itself to the network status. Problems can be encountered in many situations, like partitioning. Using this protocol, every node upon delivering a message has to rebroadcast it with a *probability* p, that depends on its local node density. Each node has the possibility later to rebroadcast the message, if this is judged to be util. In the case of partitioning, the node joining the new partition should rebroadcast the message. Such situation and many others cannot be detected without using mobility metrics. This again shows the importance of these metrics. The problem is that mobility patterns are so diverse that the concerned node, based on results of metrics, may think of a certain situation, that in reality has not happened. Following figures give some examples for this.

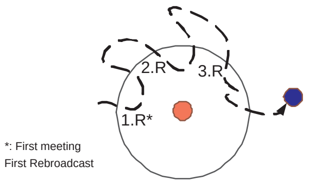


Figure 3.4: Shall the node rebroadcast the meassage 3 times? Is it not possible to recognize, that all three neighborhoods are originated from the same node?

Suppose that the node in Figure 3.4 want to send an important message, and is told from its protocol to broadcast the message every time a new node is coming in its transmission area. Using link-based metrics like *link change rate,* the node cannot recognize that it is encountering the same node all three time. He will probably decide to rebroadcast the same message three times, leading to big loss of node and communication resources.

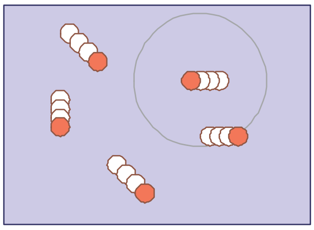


Figure 3.5: Five nodes are moving fast in a closed room. Is it possible for each node to recognize the situation?

Figure 3.5 shows 5 nodes moving fast in a closed room. Metrics results will show a very high *link change rate* and a very low *link duration*. Upon these results, every node will suggest being very mobile and encountering many new nodes. For a flooding job, every node will probably rebroadcast the same messages many times, although we see, that one rebroadcast per node is largely sufficient.

Is it really difficult to recognize these situations? We have thought about introducing new mobility metrics based on contact informations. Only one contact can exists between two specific nodes, even if they meet themselves many times in the past. Contacts have been defined in Subsection 3.2. In next Section, we will present a set of new metrics based on *contacts*.

3.5 Contact-Based Mobility Metrics

During simulation or in real networks, each node has to collect information about its last encounters with other nodes in the network, which we call *Contact Experience* or *History* of a node. Following metrics use these information or a part of them. We will also present some *new mobility metrics* based on contact information, which will be very helpful for our work, and can also open some new work domains for other studies.

While neighborhood begins when any two nodes enter in each other transmission range, and ends when they leave it, a contact begins when nodes encounter each other for the first time, and ends after they have encounter each other for the last time. On this account, one contact may include many links. High attention should be paid to the fact that contact-based metrics are very dependent on history time (resp. acquisition time interval).

- Contact Duration:

  By adding all links durations of a specific node, we get its contact duration. Figure 3.6 shows the distribution of the contact durations. Please note that the example in Figure shows only 5 contacts, although there was 7 links registered.

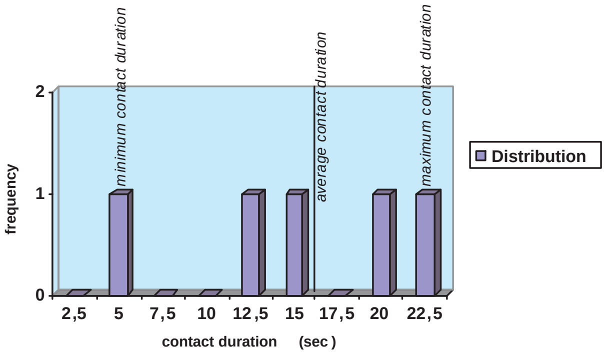


Figure 3.6: Distribution of contact duration (based on the exemple above). Because #contacts and #links may be different, we get a different distribution than of link duration.

- *Minimum Contact Duration*: The minimum contact duration must be -per definition- equal or greater than the minimum link duration. (In the considered example: $minimum\,contact\,duration$ = 5sec)
- *Maximum Contact Duration (*$maximum\,contact\,duration = 22sec$*)*
- *Average Contact Duration (*ACD*)*: Averaging all contact durations over all encountered nodes. ($ACD = 14,9sec$)
  $ACD = \frac{total\,links\,time}{number\,of\,contacts} = \frac{\sum_{j=1}^{n}(\sum_{i=1}^{p} link\,duration(actual\,node, node\,j))}{n}$ in seconds. (n: #encounters; p: #links for each encounter)
- *Standard Deviation of Contact Duration*

- Contact Rate:

  Contact rate is a function over time. We show in Figure 3.7 two examples of contact rate functions for two real world scenarios: small closed room, and wide opened space (highway).

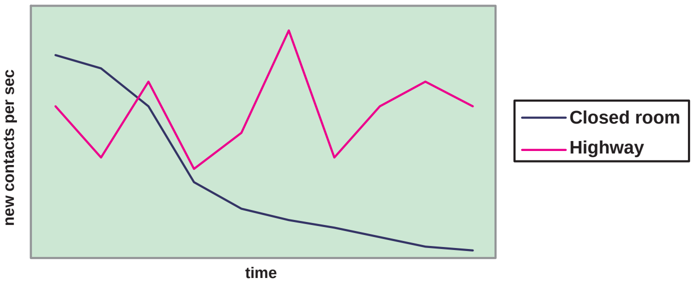


Figure 3.7: Typical curve progression of contact rates experienced by mobile nodes in a small closed room, and in a highway. These two scenarios haven't been simulated; and these results are just intuitively expected.
The average contact rate is calculated every 10 seconds, and is plotted in the graphic. We suggest, that in a closed room, where mobile nodes knows each others after a certain time, the rate of getting new contacts will be potentially reduced along the time axis, although still getting new links. In a highway, overtaking cars are always meeting new cars, that are never seen before, that's why the average contact rate keeps high.

- *"Actual" Contact Rate*: Rate of getting new contacts during a short predefined time interval. (If we consider the last 10 seconds, we get $Actual\,Contacts\,Rate = \frac{1}{10} = 0,1\,contacts/sec$)

  - *Average Contact Rate (ACR)*: Number of all new contacts during node history averaged over the history time. ($ACR = \frac{\#new\,contacts}{Time} = \frac{5}{37{,}5} = 0,133\,contacts/sec$)
  - *Derivation of Contact Rate (ACR')*: By calculating the average contact rate on small time intervals and comparing them, we get the derivation of contact rate over the whole history time. This gives a good indication about changes in mobility dynamic or may explains some scenarios like partitioning (While joining the new partition, the Contact Rate Function shows a positive kick).

- Contact Frequency:

  A node may be encountered many times during the history. It is interesting to know how often each node has been encountered.

  - *Minimum Contact Frequency*: A contact in the history must be encountered at least one time. ($Min\,Contact\,Frequency = 1link/contact$)
  - *Maximum Contact Frequency*: ($Max\,Contact\,Frequency = 2link/contact$)
  - *Average Contact Frequency*: Number of links per contact, or all links averaged over all contacts. ($Aver\,Contact\,Frequency = 1,4link/contact$)

- Combining Metrics:

  - *ACD / ALD*: Must be greater than 1, because every node may be encountered more than one time, so that a contact duration with one node may accumulate many link durations. ($\frac{ACD}{ALD} = \frac{14{,}9}{10{,}64} = 1,4003$)
  - *ACR / ALR*: Must be lower than 1, because some nodes may be encountered n times, so that they may be considered as *new* link only the first time. ($\frac{ACR}{ALR} = \frac{0{,}133}{0{,}32} = 0,415$)

  Following metrics show possible changes in the network mobility patterns. They compare Actual metric feedbacks with averaged feedback on the whole simulation time:

  - *Actual Link Duration / Average Link Duration*
  - *Actual Link Rate / Average Link Rate*
  - *Actual Contact Rate / Average Contact Rate*

We now list the most important mentioned *Metrics* in a table in order to compare them. This table is based on some reflexions and the results of different simulations. It shows that mobility metrics may depend on many model parameters and factors:

mobility metric	environment[6]	mob. model[7]	protocol[8]	mob. indicator[9]
node degree	yes	no	no	no
average link duration	no	no	no	yes
average link rate	yes	no	no	yes
average contact duration	yes	yes	no	yes
average contact rate	yes	yes	no	yes
average contact frequency	yes	yes	no	yes

---

[6] = dependent of simulation environment (i.e. density, room area, history time length)

[7] = dependent of mobility model

[8] = dependent of network protocol

[9] = good indicator of mobility

3.5.1 Concept Problem: Privacy

This is not a simulation problem, but it represents a difficulty for realizing the contact-based metrics in real-world scenarios, because *privacy* of mobile devices has to be transmitted to neighboring nodes.

3.5.2 Understanding Mobility Scenarios Using Contact-Based Mobility Metrics

Mobility metrics (presented in sections 3.4.1 and 3.5) can give good feedback about the current mobility scenario. We will present here some real world scenarios and try to find the right metrics results, that theoretically fit them. It is very interesting to know that these scenarios can not be recognized using ordinary metrics like *link change rate* or *link duration;* but only by using new metrics, and possibly combining them.

- Cars on the overtacking lane of a highway: Car A encounters new cars (with other IDs), that are never seen before (at least in the last history time). Some cars (fewer number) can be encountered many times or are just within A's radio range for a long time. These are cars that are driving with the same speed.

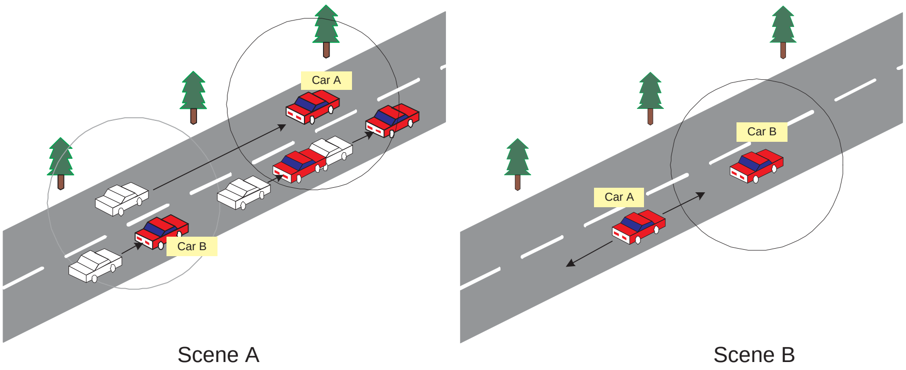


Figure 3.8: Cars on a highway: Many encounters with new and old contacts

In case of car A (Scene A), this is characterized by:

  - "low" average link duration: because overtaking cars should only take few seconds (not minutes or hours).
  - "high" average link rate: coming into transmission range of new cars within small time period.
  - "low" average contact frequency: cars that are overtaken (i.e. car B), should not be encountered later.
  - "low" ACD/ALD (resp. "high" ACR/ALR): most cars are encountered only one time.
  - "low" node degree: Because cars shall take distance security.

Scene B represents also a typical scenario in highways. Cars A and B are driving with similar speeds, but links between them are often broken and built when A comes in and out in radio range of car B. Only using contact-based mobility metrics both cars are able to recognize that is the same car.

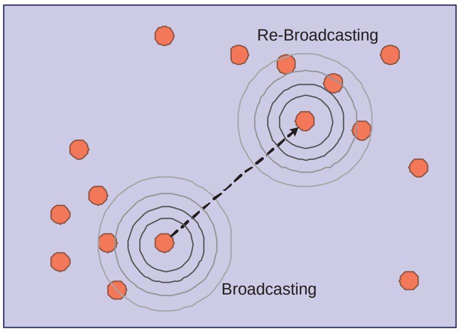


Figure 3.9: Partitioning and the role of mobile nodes to "join" between them

- People walking on the pedestrian zone "on a sunny day":

  In examples with people we assume that they have small mobile devices, which can't have a great transmission range in order to conserve resources like batteries.

  - "high" node degree: Because of possible crowd.
  - "low" average link duration: Because in the crowd, many people will be encountered, and left.
  - "very high" average link rate: Too many new links to other people.
  - "low" average contact frequency: Because of the big open area, most people are not to be encountered more than one time in small time periods (history time).

- People attending an exhibition room:

  This is similar to the *random waypoint* mobility model, where nodes are moving with a certain speed, before they stop and wait for a certain pause time, they move again into another direction. Assuming that the room is big in comparison to the transmission range of individual nodes (in order to allow links to be broken und built again), we get following results (we show only differences to above example of pedestrian zone):

  - "high" average contact frequency: The same people may be encountered many times, even in small time periods.
  - "high" average contact duration ("high" ACD/ALD): Same reason.

- People leaving a group and joining a new group (*Partitioning*):

  Joining a new partition is always characterized by:

  - "high" *actual* contact rate: In the last time interval, there was many new nodes encountered, that are not listet in the history (never seen before).
  - "high" actual contact rate / average contact rate: This is typical for partitioning, because a high bounce in the contact rate function may be observed.

Nevertheless, much more important in adapting MANETs protocols is not to recognize the exact mobility scenario (if it is a highway, or a pedestrian zone), but to extract vital information describing the mobility pattern in the local environment, which allow every node independently from others to take a better decision, if it should i.e. rebroadcast or not.

We have seen in previous examples, that we still need "old" mobility metrics like *average link rate* to recognize the environment mobility; but we have shown too that proposed mobility metrics based on contact informations are very important in order to distinguish e.g. between mobile nodes moving in a small room, and mobile nodes in a wide area. By only using previous metrics, a protocol cannot distiguish between both scenarios, and will apply the same decision. Using contact-based metrics, the same protocol can better understand the mobility scenario, und thus adjust its response in a more efficient way.

3.5.3 Which Metric Fits for Hyper-Gossiping?

We can not really answer this question, if we don't fix improvements possibilities for Hyper-Gossiping. Because we assume that nodes are mostly *much more slower* than the diffusion of the flooding message in real networks, it will be really astonishing if the mobility could improve the *delivery delay* of the message to the network nodes. Delivery delay can be improved by avoiding packets collision and contentionfor example, which is considered in the initial version of our protocol using a probabilistic feature.

Now think of a real world scenario. Multiple flooding attempts can dissipate precious node resources. Messages may have different priorities where important messages will be diffused rapidly, whereas less important messages may tolerate larger delays. In such case, it is better to save resources, while ensuring the same delivery result. Moreover feedback -gained from mobility metrics- tells the node if it has to rebroadcast the message or not. Nodes that always meet new nodes for the first time or since long time ago will have -for example- to rebroadcast their message more often than nodes with a lower contact rate. Likewise, nodes that always have the "same neighbors" or don't encounter really new nodes, which may suggest, that they didn't move or just move in a group, will probably have to rebroadcast received messages only once. In partitions, nodes should at the right time to rebroadcast their messages to initiate their spreading in the new partition.

These reflections will be applied to an existing flooding protocol (Hyper-Gossiping) in order to improve its performance (See Chapter 5).

Chapter 4

SIMULATION RESULTS FOR CONTACT-BASED METRICS

In previous chapter, we presented some *contact-based* metrics. After judging these metrics to be very important for environment feedback and protocol improvement, we intend now to analyse simulation results of these metrics under different circumstances. Simulation environment and used mobility models will be presented in first Section. Simulation results will be shown and discussed in second Section. We will use these results to imporve ad-hoc protocols by example of *Hyper-Gossiping.*

4.1 Simulation Environment

A mobility scenario consists of choosing the mobility model, the physical dimensions of the network, the number of nodes and the communication model. It is important to simulate our protocol in many different network topologies. Unfortunately this will not be possible in a "student research project", and we will investigate the *Random Waypoint Mobility Model (RWP)* and the *Reference Point Group Mobility Model ("RPGM")* and typical scenarios, e.g. a *disaster rescue scenario.* For the RWP we will vary node density and mobility parameters (speed, pause).

For simulation purposes we will use the Network Simulator ns-2[1], which is a discrete event simulator frequently used by the MANET community. Movement files for the *Reference Point Group Mobility Model* are generated using a mobility scenario generation and analysis tool[2], and are then deployed in ns-2. The disaster scenario is generated by the Scenario Generator[3], that was developed by Li Qiming.

4.1.1 Simulation Parameters

Simulation parameters that have been chosen for the simulated network are listed below. They have been chosen in a way that they remain similar to further works, and so that simulation time is kept reasonable.

- Area=1000mx1000m (for disaster scenario: 700mx500m)

---

[1] Internet site for the Network Simulator ns-2 is http://isi.edu/nsnam/ns/

[2] BonnMotion: http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/

[3] The Scenario Generator: http://www.comp.nus.edu.sg/~liqm/scengen/

- Node number = 100 (for disaster scenario: 54 nodes)
- Radio range = 100m
- Simulation time=200s
- Vmax varies from 2m/s to 20m/s (for disaster scenario: 20m/s)

Justifying the choice of the paramters is really difficult. Our goal is to construct an almost real world scenario. This choices may affect some metrics results, which have to be considered in other works. A question that may legitimately be asked is, if it is better to choose Vmax high or not? Real world scenarios may vary from pedistrian-scenarios to highway-scenarios. But we agree that it is unlikely to have nodes moving too fast in a small area. However, mobility can be considered as a proxy for link change rate in general. Obstructions, interferences from other utilizations of the public bandwidth, or turning off node devices can cause a link to be broken [Will02]. Since this is not considered by the Random Waypoint Mobility Model, it is comprehensible to choose a high Vmax, even if only slow nodes are considered. We have fixed also Vmax to 20m/s which corresponds to 72km/h, which seems to be "reasonable".

The other parameters have been chosen under consideration of further works, so that an easy comparison can be granted. It is also important not to enlarge the simulation environment in order to keep the computing time favorable for a high number of simulation trials. The area, number of nodes and the radio range are also constricted by the *node density*; in this case equals 1 node per 10.000m^2(disaster scenario: 1 node per appr.6.500m^2). In the average case, each node should have 3 up to 5 nodes in its neighborhood, which seems to be realistic. The simulation time was fixed to 200s in order to avoid following scenarios (for a single node): 1. all/most encountered neighbors are *new contacts* (short sim-time) 2. all/most encountered are *old contacts* (long sim-time).

4.1.2 Mobility Models and Scenarios

Many mobility models have been used for simulation purposes. For a random movement the random waypoint mobility model is used and for network partitioning both the reference point group mobility model and a predefined disaster-rescue scenario are chosen.

4.1.2.1 *Random Waypoint Mobility Model*

A correlated random walk is a stochastic process describing a particle that moves on the real line with constant speed γ and changes direction according to a Poisson process with parameter 2μ. The state of the particle is its position $x \in R$and its direction of motion $s \in \{+, -\}$ [Hadel96].

In Random waypoint model, each node chooses a random destination, and move towards it with a random velocity chosen from $[0, V_{max}]$, makes a pause; and then chooses another destination, and so on. Unlike other *Mobility Models,* the random waypoint model does not seem to capture the mobility characteristics of *spatial dependence, temporal dependence* and *geographic restrictions* (barriers or obstacles constraining mobility).

With this mobility model, there is a complex relationship between node speed and pause time. For example, a scenario with fast mobile nodes and long pause times actually may produce a more stable network (few link changes per mobile node) than a scenario with slower mobile nodes and shorter pause times [Bol02]. Thus, we choose to vary both pause time and speed along the axis in our simulations to get a higher variety of possible scenarios.

In [Bol02] authos presented an interesting simulation result. Following simulation parameter were set in network simulator ns-2: Node pause time equal to 10 seconds and speed ranging in [1,5,10,15,20]. Examination of the mobility scenarios reveals that with link durations shorter than 12 seconds network partitioning occurs with increasing frequency [Bol02]. This means that partitioning is an important characteristic of mobile networks, that may also occur in dense and very mobile networks. Delivering a message within a partitioned network is an important task, that may be considered by every serious protocol.

Because the Random waypoint model does not care about many mobility constraints, we have to take into account other mobility models like Reference Point Group Mobility (RPGM) Model, Freeway Mobility Model, Manhattan Mobility Model. In this work, we will investigate a typical manet-scenario, a disaster-rescue scenario, and finally we present some preliminiary results for the RPGM model.

4.1.2.2 Disaster-Rescue Scenario

The Scenario-Generator of Li Qiming can generate movement files for ns-2 for three different scenarios: conference, convention and disaster. The disaster-rescue scenario is chosen for this work's simulations. The program generates ns-2 movement files based on a specification file as input. For the disaster scenario the coordinates of all partitions have to be defined in addition to number of nodes in each partition. In order to join the partitions together some helicopters including their movement parameteres are defined.

4.1.2.3 *Reference Point Group Mobility Model* (RPGM)

The *Reference Point Group Mobility Model* ("RPGM") was developed by Hong et al. in [Hong99]. It uses a simple mechanism to make nodes walk in groups while being "free" in choosing their own velocities or changing their group dependency. The network is formed by many groups, whose motions are represented by "virtual centers", that have a location, a speed, a direction, an acceleration, etc (Motion of reference point $= \overrightarrow{GM}$). Nodes that belong to a certain group have always an indivudual movement which is the sum of $\overrightarrow{GM} + \overrightarrow{RM}$, where $\overrightarrow{RM}$ is the relative random motion around the reference point. The validity of the results is guaranteed by using a sequence of checkpoints along the path during fixed time interval. A group moves also from one checkpoint to the next continuously.

4.1.3 Problems Facing Simulation Results

1. First Problem: Links, that overlap with Acquisition Time Frame

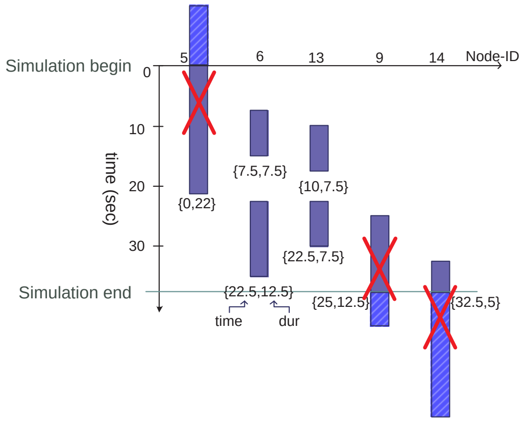


Figure 4.1: Problem with Links that begins at $t = t_0(= 0)$ sec or ends at $t = SimTime$.

Figure 4.1 shows the problem that we can face (in the simulation), if we begin to consider link durations at $t = t_0$ and end at $t = Simulation\,End$. This will result in many links, that are not covered totally, which may cause an important incertitude in our metrics results (especially for contact duration). To solve this problem we just take away from our calculations those links, who overlaps with the begin and the end of the simulation. This should give us a more precise result for *contact duration metric.*

2. Second Problem: Choosing History Acquisistion Time Length

   While encountering a new neighbor, each node decides according to actual content of the History Table, if this neighborhood represents a *new contact* or just a new encounter of an *old contact.* The differentiation is really very difficult, because it depends in first line from the length of the aquisition time for the Encounter-History-Table. Choosing this time-frame too long may lead to a very big history table with many hundreds or thousands of nodes. Encountering new contacts, that are not listed in the table, will be of low probability, especially in some special scenarios like closed rooms. Choosing it too short may lead to the fact that all (or most) new encountered nodes represent new contacts. To avoid this, a convenient time-frame has to be chosen. Many flooding simulations have shown that a broadcasting message should reach all or most of the nodes after few minutes. As a consequence, the interest on rebroadcasting an old message from the *cache* may be lost after a certain time. So we decide to fix the time-frame (i.e. simulation-time) to 200s. Depending on the purposes and the simulation parameters, this valus may be varied.

3. Third Problem: Choosing Simulation Parameters

   Table **??** on page ?? shows that all three main mobility metrics (contact -duration, -rate and -frequency) does depend heavily on the simulation environment (i.e. density, room area, history time length) and the mobility scenario. An example should gives us the random waypoint model for in first case with e.g. 25 nodes in 200x200m area and in second case with 500 nodes in 2000x2000m area. Having the same *node density* and the same *mobility model,* both examples should deliver the same average link rate. If using the same time-frame for the contact-history, mobile nodes in first example after a certain period of time will encounter all (or most) nodes in the simulation area. As a consequence, the contact rate converges to 0. Whereas, the second example works differently, because encountering all 500 (or most) nodes will take a very long time, which is not covered by the history-time.

   And because there is no best solution which can be justified completely, we intend to choose our simulation parameters far from this two extremas. Nodes should neither know most of nodes in the network, nor have too little knowledge about old contacts. We opted so the simulation paramters presented in Subsection 4.1.1.

4.2 Simulation Results

By using the random waypoint mobility model and by choosing the simulation parameters previously specified on Subsection 4.1.1, we show now the results of the first simulations set in next three paragraphs. By Varying V_{max} and $Pause_{max}$, we get 16 different scenarios. Some of these results are shown below. Defintions for mobility metrics -listed below- should be read on Subsection 3.5 before continuing with next sections.

Please note that most of scenarios have been simulated only for one run, which may explains some incertitudes in few charts. But as long as we do not intend to meet important conclusions from these charts, this may be sufficient for this work.

4.2.1 Random Waypoint

Most of our simulation results are based on the random waypoint mobility model because of its universality. Contact-based mobility metrics are studied in this Subsection and compared with results of link-based metrics.

4.2.1.1 Contact Duration

The distribution of the average contact duration for all 100 nodes are represented in Figure 4.2. For a better comparison, we displayed 12 scenarios for $Vmax\, in\, [2, 5, 10, 20] m/s$ and $Pause\, in\, [0, 20, 50]$. We first conclude that the contact duration metric is not really affected by the pause-time, but more heavily by max-speed. When the mobility increases the contact duration of nodes decreases; and when the mobility decreases the contact duration of nodes increases.

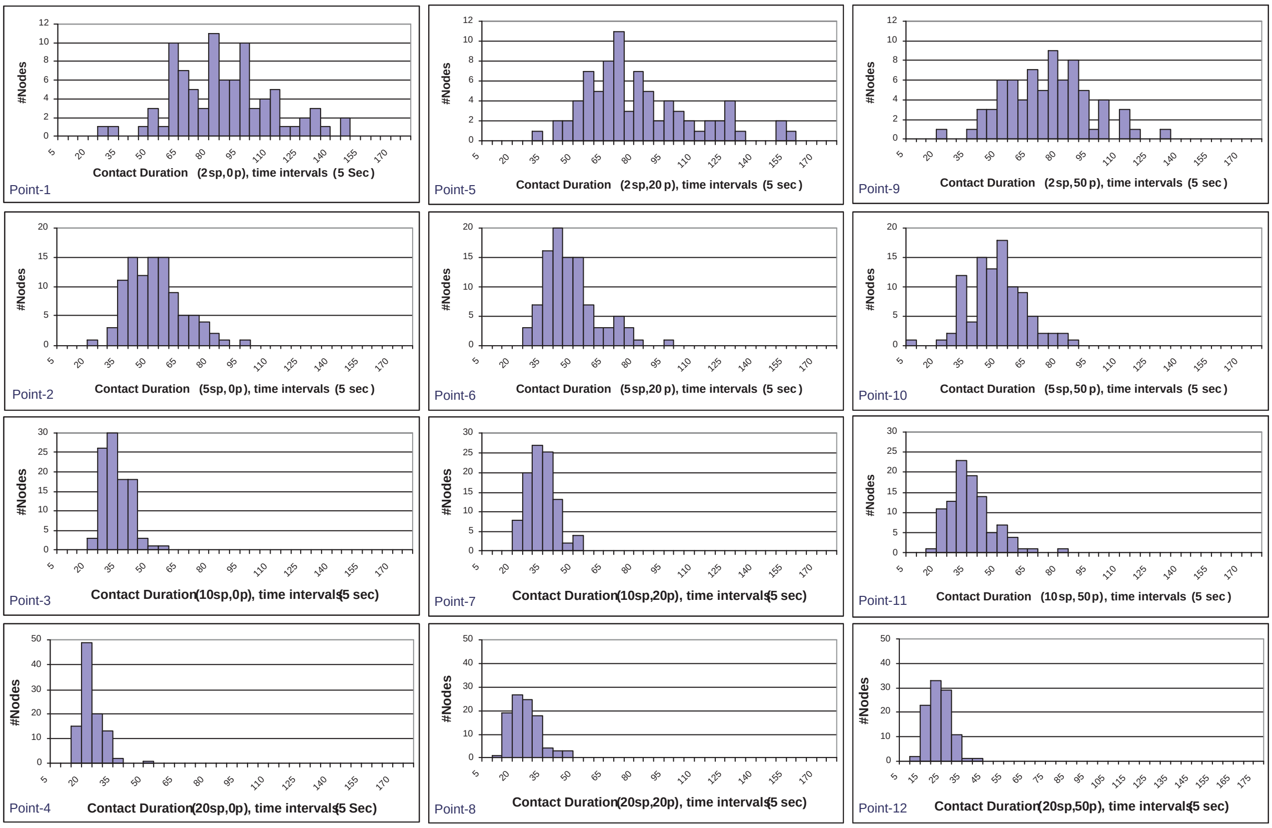


Figure 4.2: Contact Duration Distribution for 12 different scenarios: $V_{max}\, in\, [0, 20] m/s$ and $Pause\, in\, [0, 50] s$.

In Figure 4.3 the 3D-diagramm shows the variation of the mobility metric *contact duration* over node speed and pause. The results confirm the idea, that this mobility metric is more sensitive to speed variations, than to pause variations. This can be explained by the fact that nodes do not take their pause in the same moment. For example, some nodes do their pause, while most neighboring nodes are still moving. So links will be broken, and link- and contact- duration will not differ too much, whether the node take more or less pause.

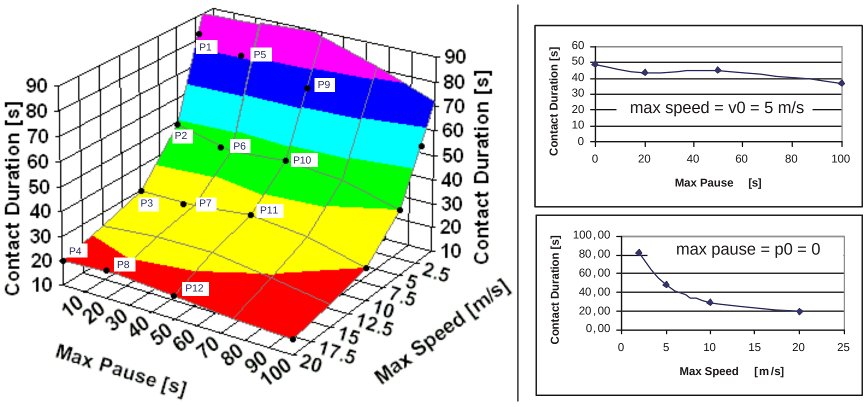
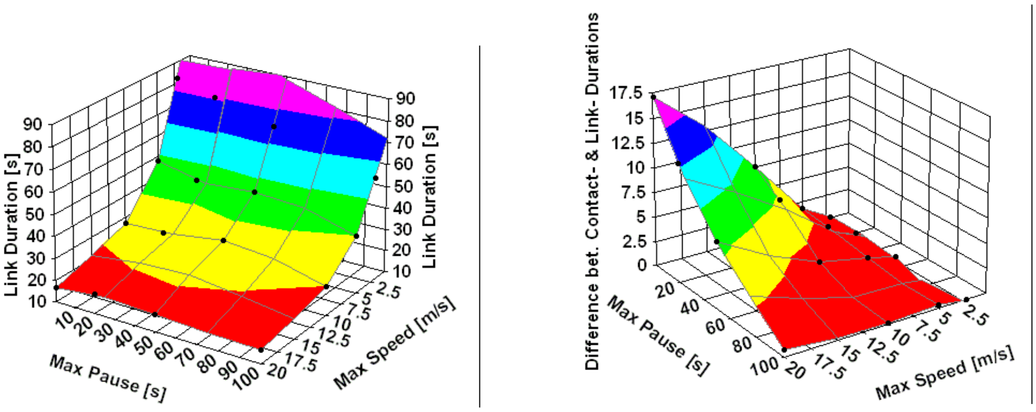


Figure 4.3: Average Contact Duration for the random way-point mobility model. Diagramms on the right show sections of the 3D-diagramm for resp. $v = 5m/s$ and $p = 0s$.

We compare now contact duration and link duration for the random waypoint model.



a) b)

Figure 4.4: Average Link Duration (l.) and the percentage of the difference between *contact duration* and *link duration* (r.)

Figure 4.4 a) shows the average link duration in dependency on speed and pause. We observe a great similarity with the contact duration diagramm on Figure 4.3. Figure 4.4 b) represents the difference between both metrics (contact- and link- duration). We see obviously, that the difference between both metrics is dependent from both mobility parameters: max speed and pause time.

4.2.1.2 Contact Change Rate

Since the lost of a link can not be considered as a lost of a contact, we again insist that the *contact change rate* means *rate of new contacts* (that means *lost contacts* will not be considered). Figure 4.5 shows that it may be difficult to find a clear similarity between the different distributions, when either speed or pause-time are varied (in contrast to *contact duration*).

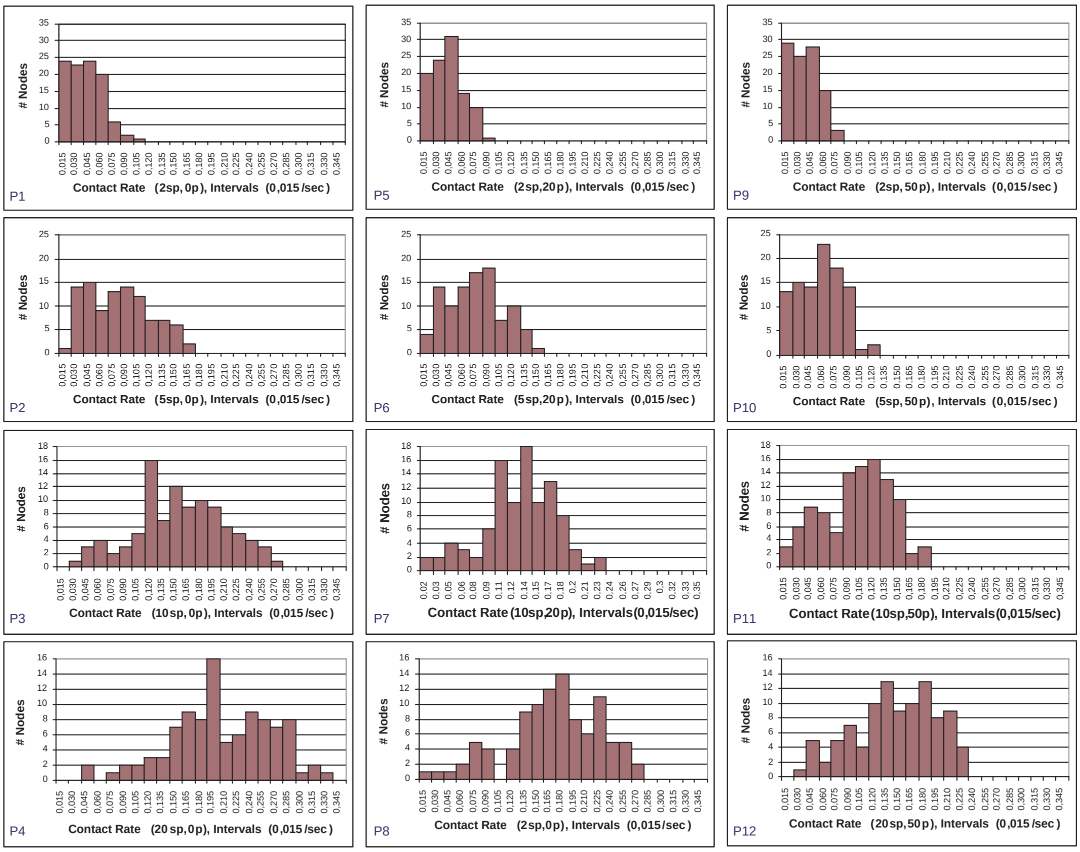


Figure 4.5: Contact Duration Distribution for 12 different Random Way-point configurations.

From Figure 4.6 we can easily deduce that both speed and pause-time influence the metric results.

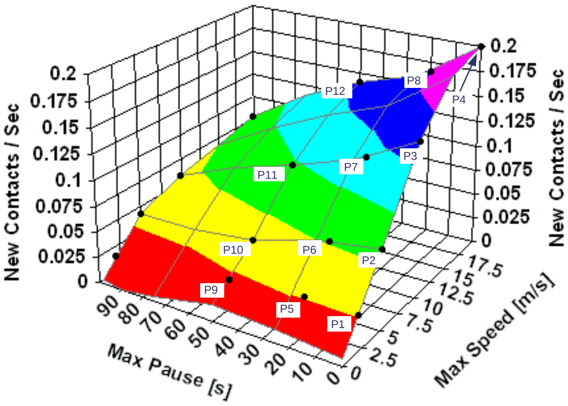


Figure 4.6: Average Contact Rate for the random way-point mobility model.

Figure 4.7 shows the mapping of the *average*[4] *contact/link -rate* over time. Please notice that at the beginning of simulation ($t = 0$), contacts history of each node is still empty. While encountering

---

[4] over all 100 nodes

	1. period	2. period	3. period	4. period
Contacts (per sec)	0.291	0.138	0.091	0.056
Links (per sec)	0.346	0.249	0.233	0.215
Difference(%)	15.96%	44.54%	60.82%	74.03%

Table 4.1: Difference between contact-rate and link-rate over 4 periods of time.

other nodes, the history table will be filled. This explains the difference between contact-rate and link-rate, that increases with increasing time. In Table 4.1 we divide the simulation time (500s) in 4 periods, each of them having approx. 125 seconds, and then we calculate the mean value for both link and contact change rate. This should shows the progression of both metrics clearly.

Table 4.1 shows that the average link rate is decreasing over time. This phenomenon is caused by the *border effect*, which is a charcteristic of the random waypoint model. In fourth period, we see that more than nearly 3/4 of nodes encountered represent old "contacts". This is a very important information, that can not be obtained from the *link metric.* Whereas, the information obtained from the *contact change rate metric* tell us, that some nodes are re-encountered within a short time period, i.e. history time. We also can "deduce" that nodes are located together in a closed room (Furthermore, there should be no explicit partitioning, because there is no clear peak in the contact-rate function, which can explain the joining of a new partition)

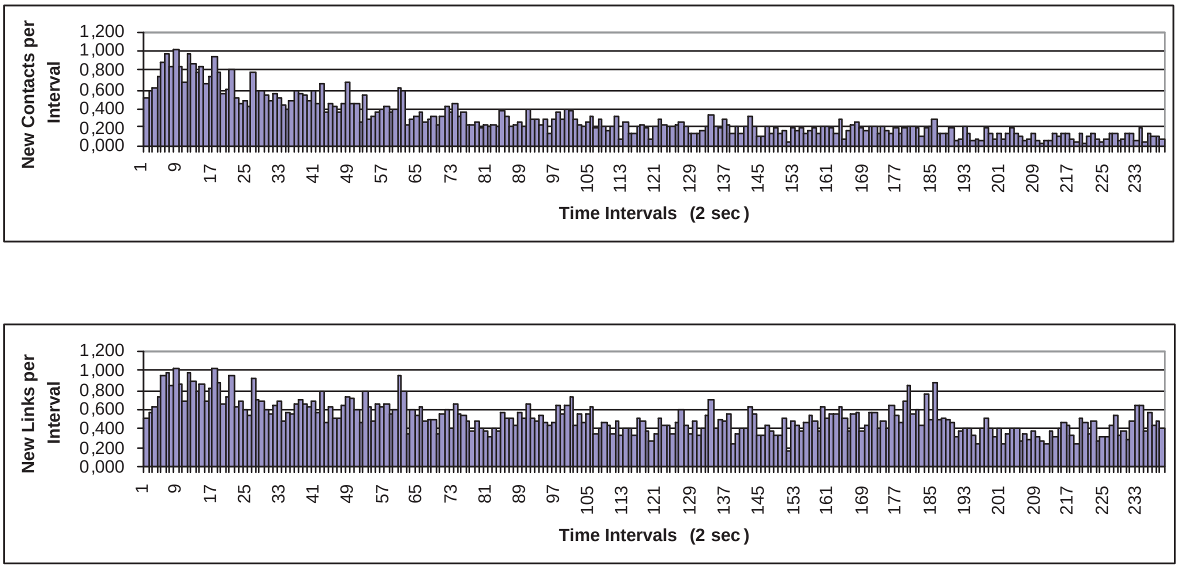
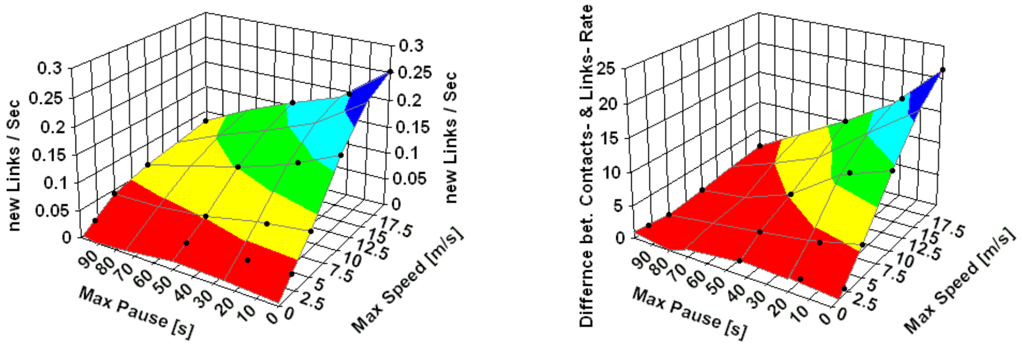


Figure 4.7: *New Contacts- vs. New Links- Rate* over time averaged over all 100 nodes (max speed=20m/s, pause=0s)

A comparison between *contact change rate* and *link change rate* shows a similar result to that of *duration metrics.* These results are shown in Figure 4.8. Again we observe that both metrics does depend from both node max speed and pause time.



a) b)

Figure 4.8: Average Link Change Rate (a) and the percentage (Calculated by dividing through the link change rate value) of the difference between *average contact change rate* and *average link change rate* (b)

4.2.1.3 Contact Frequency

The contact frequency metric is a good indicator for the redundancy of contacts; it indicates how often a given node is encountered within the history time. This metric may be used for the dynamic adjusting of the history acquisition time. Figure 4.9 shows the contact frequency over node max speed and pause time.The link frequency is always equal to 1.

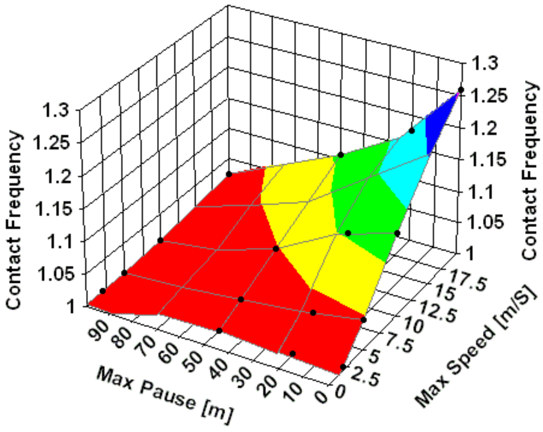


Figure 4.9: The Average Contact Frequency for the random way-point mobility model.

4.2.2 Disaster-Rescue Scenario

Now, we want to investigate above metrics for a typical manet scenario. We will concentrate on different group mobility models.

Using the Scenario-Generator we can generate three different environments for ns-2, which are *conference*, *convention* and *disaster*. We focus our interest on the disaster scenario, which consists in 2 disaster areas, that are overflown regulary from 2 helicopters. The helicopters start from the base area. All three areas (2 disasters + base) are populated by mobile nodes. The whole scenario is displayed in Figure 4.10.

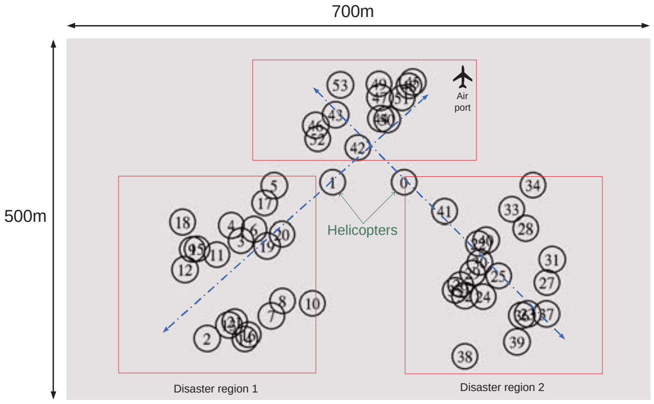


Figure 4.10: Disaster scenario (base, 2 disaster areas and 2 helicopters)

The type of disaster scenarios makes it easy to model partitioning, which is the "master key" of this work. The track of the helicopters can be compared to a node moving from one partition to an other one. So let's concentrate on one of both helicopters (helicopter-1). Due to the high mobility of the helicopter, many links are broken and new ones are created, and link duration is kept small and generally cannot exeed some value. Each helicopter can get *contact* with a maximum of 33 nodes in the presented simulation scenario (20 in disaster area, 12 in base area and the second helicopter). If we compare this evidence with the second diagramm in Figure 4.2.3, which represents the amount of new links for the helicopter-node over the time, we see that the heli-node has counted 126 new links during only 200 seconds, which represents a relative high *new link rate* of $0,63links/sec$. Just compare it with the *new link rate* of last example $0,261links/sec$(See table in 4.2.1.2), although being in a quite "dense" area. Using only this metric to understand the simulation environment may be really confusing for the node, especially if we consider the devolution of the link rate metric over the time. In fact, diagramm b) in Figure 4.2.3 shows many importants vacillations in the results. Because there is no clear disjunction between the phases in the diagramm, the node could not recognize, if it moves between partitions or just moves within a single partition. Even by combining these few perceptions with diagramm d) in Figure 4.2.3, which can be easily calculated without paying attention to *contacts*, the situation cannot be understood. Quite the contrary, by combining both diagramms (2. and 4.) the node may at most "recognize" a partitioning by $t \approx 100sec$, because both link rate and node degree converge to zero, before going "rapidly" up. This is a wrong realiziation, because these moments does just represent the pause-time of the helicopter in the base area!

Judging the above metrics and results to be insufficient, we orient us now to *contacts perception* and *contacts analysis*. Each mobile node has to count and save his own contacts. The complete contact-table for the heli-node is represented in the third diagramm in Figure 4.2.3. There, we recognize easily some trend: there exists two recurring phases. In the first one (approx. $[0sec, 11]$; $[71, 118]$; $[168, 200]$) only 13 contacts are -again and again- encountered, which clearly indicate a paritition with at least 13 nodes (these are 12 nodes of base area + second helicopter). The second partition is discovered in the other phases with 20 other nodes (disaster area 1). Using an algorithm to detect and extract this useful information from the *contacts table* makes it easy for each node to adapt itself (its protocol) to the changeable environment. Even by only using the first diagramm, which represents the *new contacts rate* in function of the time, in combination with the fourth one (*contact number*[5]) the node can recognize that almost no new contacts after time $t = 38s$ have been found, while the contact number is keeping high, which contests some assertions, that could not be recognized without using *contacts information* (by only using links information -like in first paragraph above-), e.g. the node is being very mobile and is encountering

---

[5] By constant time, the contact number is the same as the node degree

new nodes and new partitions all over the time. Assuming this false statement could cost the node a good deal higher overhead and resources.

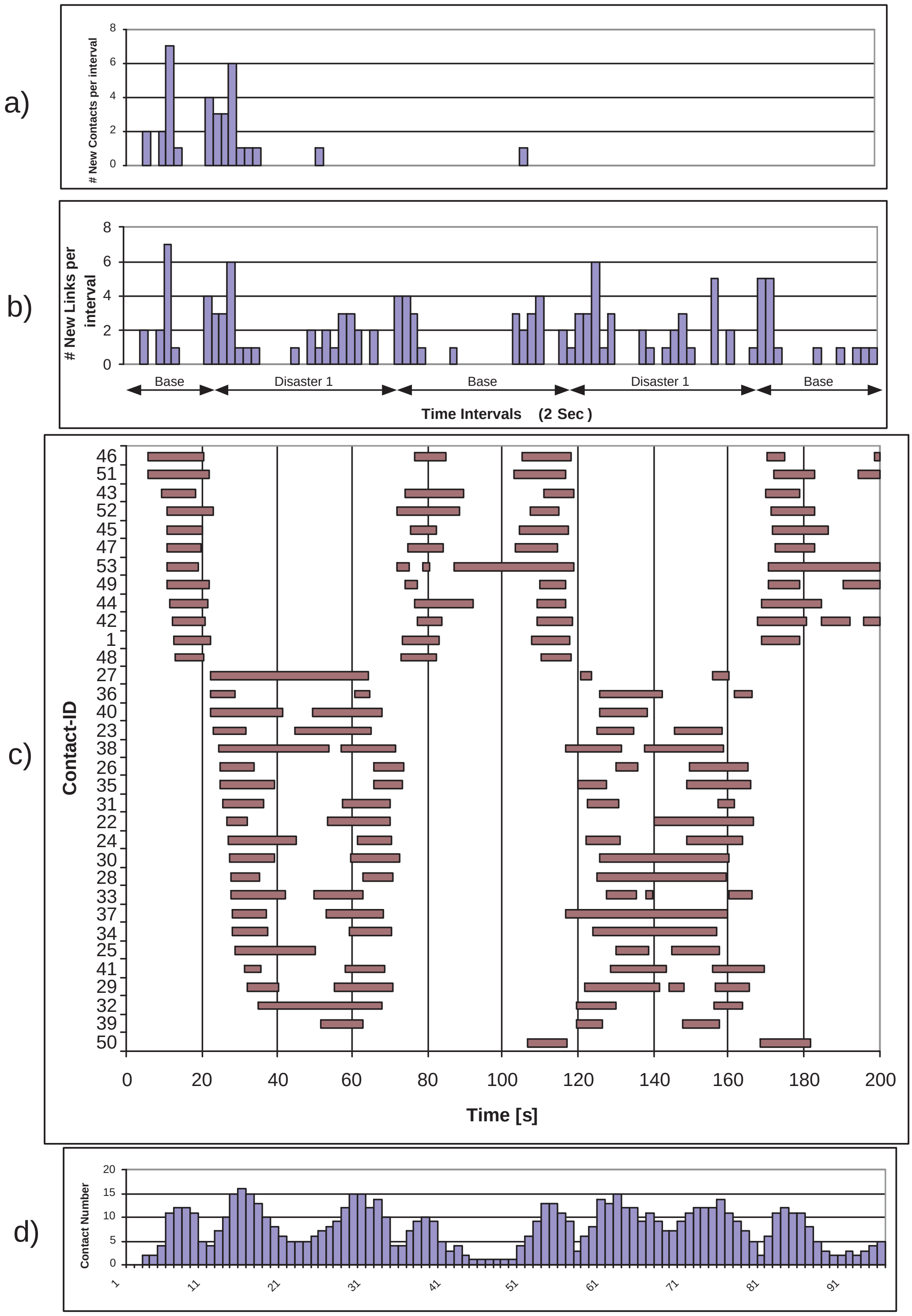


Figure 4.11: (a) contact rate, (b) link rate, (c) contact table and (d) contact number[7] of one helicopter in disaster scenario.

4.2.3 Reference Point Group Mobility Model (RPGM)

Using the software-tool *bonnMotion*, we generated some easy scenarios for the RPGM-model. Figure 4.12 shows some snapshots of the *Network Animator (nam)* from a scenario composed of 4 autonom groups (100 nodes totally). Results show that all nodes have very similar experiences. We focus our interest on the node number 45, whom contact rate function over time is presented in Figure 4.13.

First snapshot ($t = 0s$) shows the node of interest wihtin its group (partition). It can communicate with some of its neighbors, which explains the few contacts[8] registered at the beginning (Figure 4.13). Until approx. the 40-th second, there is no more contacts registered, even though links within the local group may break and be built again. In second snapshot ($t = 46s$), three groups merge, which explains both peaks just after the 40-th second in Figure 4.13. Third snapshot shows the "merging" of all groups together, which explains the peak by 80-th second. The few contacts registered before the peak, are probably caused from not yet recognized nodes from other groups. During the rest of the simulation the groups encounter each others in many situations, but we see clearly on Figure 4.13, that no new contacts are registered until at least the end of the simulation, when some few contacts are again registered, which corresponds to the fifth snapshot ($t = 195s$).

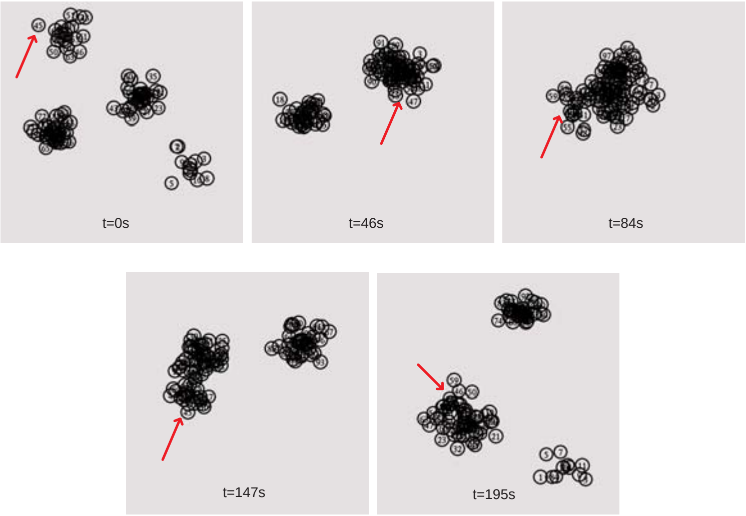


Figure 4.12: Snapshots from the *Network Animator* for one RPGM-Scenario (4 groups, 100nodes)

Figure 4.13 shows the contact rate over time. In contrast to Figure a), at least two peaks can be easily recognized, which corresponds to the merging of many partitions together.

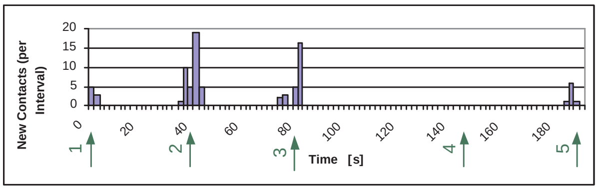


Figure 4.13: Contact rate of node 45 over time im RPGM scenario.

---

[8] Explains, why there is only few, but not more

Chapter 5

USING CONTACT-BASED MOBILITY METRICS TO IMPROVE HYPER-GOSSIPING

Chapter 3 and 4 reviewed mobility metrics in general, and introduced some new metrics that present the core of this work, and which we call contact-based mobility metrics. We believe that these new metrics can be deployed in a variety of MANET protocols and network environments, but this work can not qualify all possible improvements that can be gained by means of these metrics. Although we want to present in this chapter a concrete example of protocol improvement using contact-based metrics. Section 5.1 shortly describes the problem and introduces our concept. Section 5.2 shows implementation details of the *Contact-Based Metrics "Protocol" (CBM).* Simulation results using random waypoint model and a typical real-world scenario (i.e. disaster rescue scenario) will be presented in Section 5.3.

5.1 Concept

The main goal of this work is to imporve Hyper-Gossiping Protocol by adding to it some new constraints. In the same way, new mobility metrics have been proposed, since mobility is a main characteristic of mobile ad hoc networks in general and for Hyper-Gossiping specially. Now we want to couple our contact-based mobility metrics with the Hyper-Gossiping protocol. The Hyper-Gossiping protocol was presented on Subsection 2.3.7. Please refer to this description, before continuing with this chapter.

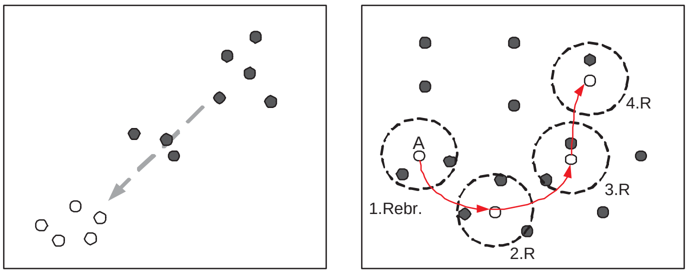
5.1.1 Hyper-Gossiping Protocol

The Hyper-Gossiping protocol tries to compromise between the number of rebroadcasts per node, that should be minimized, and network reachability, especially by network partitioning, where it seems as if we need an extra effort to deliver the diffused information to all nodes in all partitions. In fact, the number of rebroadcasts on receiving a new message are minimized by a proposen *density constraint*, so that not all nodes has to forward the message upon receiving it (See Sec. 2.3.7). Partition discovery is accomplished as follows: Each node has to maintain a list of potential receivers for every packet in its cache. Potential receivers are node's neighbors as it receives the packet for the first time and also redundant packets, its current neighbors every time it rebroadcasts the message, and naturally all packet's sender(s). Nodes in the network send periodically hello-messages. Upon receiving a hello-message from a neighbor, each node checks the lists of all cached packets and compare them with the set of actual neighbors. Neighbors are discovered using the *Neighboring Discovery Protocol*, which will be presented in next Section. If both sets (potential receivers for this packet and actual neighbors) are disjoint, then the node has to rebroadcast the corresponding packet.

The rational behind using the repretitive feature in Hyper-Gossiping is to enable partition discovery for a better network reachability. In the majority of cases, this goal is achieved; but in some cases partitions can not be discovered just by using aforementioned mechanism (e.g. overlapping partitions, when two partitions merge together for a short time). A side-effect is given when nodes move inside the partition that received the packet, what results in neighborhood changing, and consequently to many unnecessary rebroadcasts. Thus, we wish to introduce a new mechanism, that minimizes unnecessary rebroadcasts without worsening the partition discovery capability of current protocol. These reflections favorise the use of contact-based mobility metrics, which will be explained in next Section.

5.1.2 Problem Description

Without a global knowledge of the network topology, it is impossible for nodes to certainly discover partitions, especially when thinking about dynamic network and changeable partitioning. The Hyper-Gossiping protocol is a solid approach to solve this problem, but unnecessary rebroadcasts should be minimized, especially in the same partition, where the information easily attains most or all of its nodes. The idea is to add some new conditions upon receiving a hello-message either to rebroadcast the cached entries or not. Before continuing with this discussion, we give you an overview of weak points of the current version of Hyper-Gossiping. This is shown in Figure 5.1.



a) b)

Figure 5.1: Problems facing Hypergossiping: a) joining a new partition in a group and b) rebroadcasting the same packet in the same partition more than once.

The Figure 5.1 shows two typical scenarios, where Hyper-Gossiping breaks. Figure 5.1 a) shows a group of nodes leaving a partition towards a new partition. Unfilled nodes represent nodes of the new partition, that do not receive the packets from the old partition yet. Unfortunately, none of the moving group will be able to rebroadcast the cached packets, because nodes are moving in a group so that receiver's set and neighbor's set probably will never be disjoint (see the Hyper-Gossiping repretitive condition). The second example b) shows only one partition, where all nodes received the broadcasted packet. Node A is moving within its own partition and will probably rebroadcast its cached packets every time it encounters new neighbors that are not marked as potential receivers in its packet's lists. This may result in a big number of unnecessary broadcasts. In this paper we will try to improve the second problem b) without affecting the reachability. Moreover, this solution should be quite adequate to network partitioning in general so that it can be useful for later improvements and other protocols.

5.1.3 Enhancement in Partition Discovery

Nodes moving within one partition and nodes moving into a new partition have often similarities and differences in their experiences. We look for a new repretitive condition (using contact-based mobility metrics) that fits to network partitioning. While leaving an old partition and joining a new one, each node may experience 3 different phases:
1. Leaving old partition,
2. Moving into new partition and
3. Joining new partition. First phase is characterized mostly by a set of known neighbors.

Nodes in second phase can have either old neighbors or none. Scenario a) from last Section shows nodes moving in a group within a partitioned network. Nodes in last phase mostly experience a high rate of new encounters in small time intervals.

Existing mobility metrics fails mostly to recognize partitions. For example, "link number" (converging to zero) can be used to recognize second phase, but fails when nodes are moving in a group, which really often happens in real-world scenarios. "Link ratenew may be used to recognize second and third phase because link rate converges in second phase to zero but increases when entering new partition. This metric fails in recognizing moving out and joining the same partition that results in too many rebroadcasts.

Contact-based metrics have the advantage that they can recognize a new encounter as an old contact. For protocol purposes *new contact rate metric* will be used. Partitions should be discovered in a two-phase recognition mechanism:
1. When leaving the old partition and moving into new one, the contact rate metric *must* converge to zero
2. When joining new partition, the contact rate *should* clearly increase.

Used metric give an easy mechanism to recognize partitioning, but additionally should minimize the unnecessary rebroadcasting of cached packets in the same partition, especially by not huge partitions, where the probability that a new encounter is a known contact is somewaht high. Using contact-based metrics has surely many other advantages, that will be briefly introduced and discussed in next sections.

5.2 Implementation

Since the Hyper-Gossiping protocol is written in ns-2 and since this network simulator is really preferred by the MANET community we decide to continue using ns-2 for our simulation purposes. Network simulator ns-2 is a discrete event simulator, with a simple model. It supports both wired and wireless medias, and different protocols. ns-2 is also object-oriented, written in C++ (which is fast to run), with an OTcl interpreter (which is fast to write and change) as a frontend [NS-Manu]. This permits a seperation between data and control.

Current version of Hyper-Gossiping includes following c++ files:

- hypergossiping.cc (.h): includes protocol logic itself. It treats the sending and receiving of packets, instantiates NDP and starts Timer.
- hypergossping_header.cc (.h): defines the general Hyper-Gossiping header.
- ndp.cc (.h) (Neighbor Discovery Protocol): implements Hello-beaconing and maintains a list of all current neighbors based on hello-beacons.

For this work purpose a new "protocol" is added and this is the *Contact-Based Metrics Protocol (CBM).* The CBM protocol will be explained in Subsection 5.2.2.

5.2.1 Neighbor Discovery

Neighboring discovery is used in both NDP and CBM protocols with other purposes. The NDP protocol manages a list of neighbors for all nodes in the network. Neighbors are discovered using hello beacons, that are sent at intervals randomly chosen between MinHelloInterval and MaxHelloInterval. The neighborlist for every node is saved in a vector, that is called *neighbor_ vector.* Every node keeps its *cache_ table,* where the *neighbor_ vector* and a list of packets are cached.

The CBM protocol uses global information obtained directly from implemented classes in ns-2. At the beginning, God (general operations director) has been used. God is aware of whole network topology. It is used to store global information about the state of the environment, network, or nodes. God object stores the total number of mobile nodes and a table of shortest number of hops required to reach from one node to another. This is the information, that an omniscent observer would have, but that should not be made known to any participant in the simulation. The god object does not calculate this on the fly during simulation runs, since it can be quite time consuming [NS-Manu]. In tcl-script God is instanciated with:

```
create-god <num_nodes>
```

The number of mobile nodes is passed as argument which is used by God to create a matrix to store connectivity information of the topology [NS-Manu]. In order to obtain the requested nodes information from God, one of its many methods (defined in mobile/god.cc) has to be used; so we used following method: int hops(int i, int j) which returns the minimum number of hops needed to join nodes *i* and *j.* The method is then called in following way:

```
#include "god.h"
if (God::instance()->hops(node_i, node_j) != 1) ...
```

The single complaint for using God is that node information are saved in input files, so that these files sometimes become really huge and consequently the time for simulation increases too much. So we opted later for another possibility, which is to retrieve node positions directly from MobileNode class, and then to let the CBM protocol calculate neighborhood information himself. This alternative saved simulation time and memory. In a real world scenario, the CBM protocol has to retrieve encounter information using NDP.

5.2.2 Contact-Based Metrics Protocol (CBM)

Each mobile node has to maintain an *encounter table (list of contacts* or *conatct history).* This represents its *experience* with other nodes in the network. This table contains an entry for every node in the network that was encountered in the past. A table entry contains the node identification, first and last encounter time, and encounter number with this node. In order to save *memory volume* and *battery lifetime*, history table's size should be kept reasonable. This is achieved by e.g. fixing statistical-time interval to a given time.

The encounter table may look like the following: it contains one entry (2-columns list) for all encountered nodes (only encountered nodes, because it is impossible for the node in real world circumstances to have information about other nodes, before encountering them).

In this table, t_{i_j}represents the *encounter time* of node i for the j-time; and d_{i_j}represents the *encounter duration* of node i for the j-time, both in seconds.

The history table is filled by checking neighbors regulary. For this purpose a Timer has to be started after the CBM-instance is initialized:

node 1		node 2		node 3		node 4		. . .		node n	
t_{1_1}	d_{1_1}	t_{2_1}	d_{2_1}	t_{3_1}	d_{3_1}	t_{4_1}	d_{4_1}	. . .	. . .	t_{n_1}	d_{n_1}
t_{1_2}	d_{1_2}	t_{2_2}	d_{2_2}	. . .	. . .	t_{4_2}	d_{4_2}			t_{n_2}	d_{n_2}
t_{1_3}	d_{1_3}	t_{2_3}	d_{2_3}	t_{3_r}	d_{3_r}	. . .	. . .				
t_{1_4}	d_{1_4}	. . .	. . .			t_{4_s}	d_{4_s}				
. . .	. . .	t_{2_q}	d_{2_q}								
$t_{1_{p-1}}$	$d_{1_{p-1}}$										
t_{1_p}	d_{1_p}										

Table 5.1: Encounters table contains history of correcponding node.

```
cbm.htimer.handle((Event*) 0);
```

An instance of the current MobileNode has to be passed to CBM in order to extract position information of nodes:

```
cbm.mn_ = this.mn_;
```

The CBM instance then checks the neighborhood information periodically and updates its history table. Simultaneously, some important variables are computed. The CBM class offers, in addition, many methods to calculate statistical informations.

Upon receiving a Hello message (of type HYPERGOSSIPING_HELLO) each node compares its current neighbors set with potential receivers set (See 5.1.1). If both set are disjoint, a CBM-method to compare between "actual" and total *contact rate* is called. This method is called for every packet that satisfy last requirement and gets as argument the packet's delivery time. It goes then throw all entries in history table and checks following two conditions:

1. Compare "actual" wih total *new contact rate*: Starting from the packet's delivery time, all encounters in the history have to be considered. Encounters happening before are discarded as well as their contact information. From this moment until NOW[1] all new contacts (different from encounters) are added and then divided by the resulting time interval. The outcome of this is the *new contact rate over total time.* The *actual new contact rate* is calculated over a small time interval, that was fixed in our example to 5 seconds. Both metric results have to be compared. Is total contact rate higher than actual contact rate, then the packet's rebroadcasting is prohibited. This is reasonable if we consider that nodes joining some new partition will with higher probability have a higher contact rate than over their past experience. Otherwise a high contact rate alone can not indicate a new partition, e.g. in dense partitions. It is almost wise to compute the total contact rate from the packet's delivery time, because the node can not recognize from the history table if older contacts have received the packet or not, so not considering them is safer. In addition, mobility pattern before delivering the packet is not relevant for rebroadcasting.

2. Search for *new contact rate* equals zero: The CBM protocol looks only in last 10 seconds for a time interval (equals 2 sec), where the contact rate equals zero. Is this interval given, this may indicate a possible partitioning (even a temporary partitioning in a very dynamic mobile network). Otherwise rebroadcasting is prohibited. Chosing last 10 seconds is also resonable, because old partitions may not disturb the results.

Are above conditions fulfilled the node should rebroadcast the corresponding cached packet. While second condition just depends on current time, first condition depends in first line on packet's

---

[1] Time when the method is called

delivery time. The combination of both metrics should help nodes to recognize new partitions and to minimize unnecessary rebroadcasts in same partition. Please note that both conditions as well as fixed values are not severe, so that reachability should not be negatively affected, especially in this early phases of studying contact-based mobility metrics.

5.2.3 Trace Support

The trace support for wireless simulations use cmu-trace objects. Output is usually a file with separated lines. Each line has a similar format, which describe the event type (*s* for *send*, *r* for *receive, d* for *drop, f* for *forward*), node property tags (like *node id* ...), Next hop info, Packet information. Trace files are then analysed using scripts in Perl. Most important evaluation meetrics obtained from these scripts are *Reachability*, *Delay* and *Number of rebroadcasts per node.* These are described in detail in next Subsection.

5.3 Simulation Results

Most important results on Hyper-Gossiping protocol using contact-based metrics are presented in this section. All figures show a comparison between old version and new version of the protocol. Presented example is just a basic approach for using contact-based metrics in mobile networks. Both Hyper-Gossiping protocol version were tested using the random waypoint mobility model and a predefined disaster-rescue scenario. We run at least 10 simulations runs for every configuration.

5.3.1 Random Waypoint Mobility Model

First set of simulations have been tested using following simulation parameters:

Simulation Parameters	
Number of Nodes	{10, 20, 30, 50, 60, 65, 70, 80, 100, 120, 150, 200, 500}
Simulation Area Size	1000m x 1000m
Transmission Range	100m
Max Speed	$20m/s$
Pause Time	$2sec$
Packet Size	250 bytes
Simulator used	NS-2 (version 2.1b9a)

Please note that networks with a relative low number of mobile nodes are high partitioning, whereas dense networks are mostly free from partitions.

Simulation results show that *reachability* (means almost 100% reachability for all nodes) is not negatively affected by using the new version of Hyper-Gossiping. So first conclusion for the random waypoint model is that the given reachability is conserved; whereas *delay* and *mean number of rebroadcasts per node* (MNR) are changed. Results for both performance metrics are shown in Figure 5.2. First diagram a) shows the gain of mean number of rebroadcasts obtained by comparing new version to old one. The curve for 95% reachability shows a permanent amelioration up to 15% (for 50 nodes), but may have a small regression by 65 nodes and by more than 120 nodes. The curve for 100% delivery has a similar course, but the resgression is highly observable by 50 to 70 nodes and by 100 nodes. Anyway, at first view we can speak here about a balance of amelioration and regression for 100% reachability. In the light of second diagram b) of the same figure, the cause of this regression is easily identifiable. The new version for Hyper-Gossiping needs up to 50% more time (in some cases) for delivering the packet to all network nodes. Particulary by 50 to 70 nodes where the biggest regression has been registered, the delivery time difference reaches its maximum. In other words, the regression is the direct result of the different delivery delays; but in reality number of rebroadcasts in function of time was clearly regressing in most/all cases.

Second diagram b) in Figure 5.2 shows, in addition, a high convergence of both delivery delay's curves in case of 95% reachability (except for 65 nodes where difference is about 25%, which again explains the discussed regression). This means that the new version of Hyper-Gossiping shows a satisfying result; and troubles may only occur by very few nodes in the network.

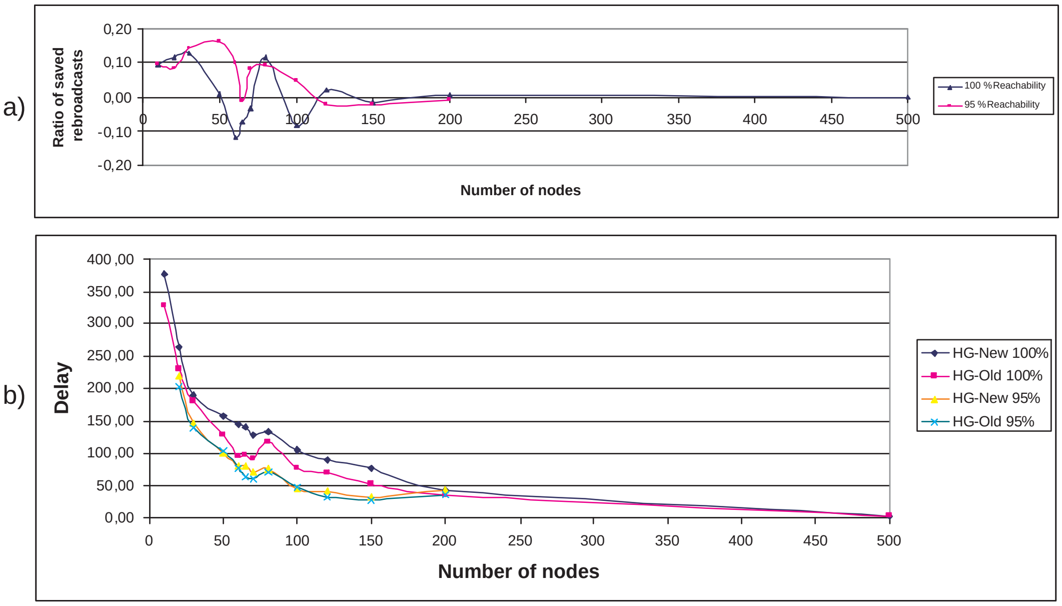


Figure 5.2: Both version of hypergossiping protocol using rwp-model while varying number of nodes: a) Mean number of rebroadcasts per node and b) Delay needed to reach 100% and 95% of all nodes.

To test the impact of mobility on Hyper-Gossiping, max speed has been varied within $[5, 30]m/s$ for 20 mobile nodes. Figure 5.3 shows that the new version of Hyper-Gossiping is not really affected by mobility, while the old one deteriorates by decreasing mobility. Highest difference of 22% is reached for $10m/s$. A possible explanation for this effect is that less mobile nodes need much more time in order to deliver the information to all nodes. During this long time interval the old version of Hyper-Gossiping loses much more resources in unnecessary rebroadcasts, while the new version is doing better.

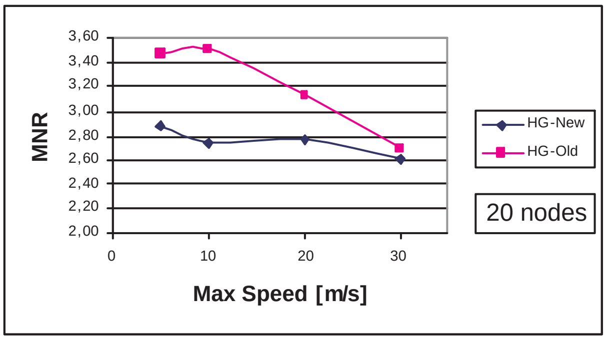


Figure 5.3: Mean number of rebroadcasts per node (MNR) over speed for both versions (20 nodes)

Figure 5.4 shows the comparison between the evolution of delivery rate for different network densities. The curves shows often a small lateness of new version in comparison of old one. However, one example c) (100 nodes) shows a faster devolution of the new version. This demonstrates that the delivery rate function can also be improved.

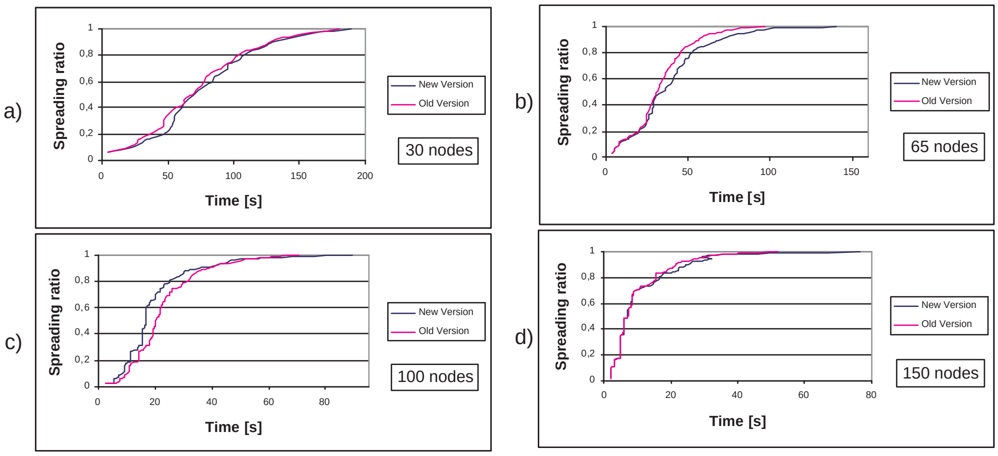


Figure 5.4: Comparing spreading ratio over time for both versions (30, 65, 100 and 150 nodes)

5.3.2 Disaster-Rescue Scenario

The disaster-rescue scenario used in this example is very similar to the scenario shown in Figure 4.10 on page 48. Modifications are here listed: 1000x1000m simulation area, 100 nodes divided into 3 groups (2x 33nodes, 1x 30nodes) and 4 helicopters. Every two helicopters are connecting two partitions together, and all helicopters have different speeds. They are responsible for carrying the information to all partitions.

A total number of 34 simulations have been realised for the disaster-rescue scenario. The packet's initiator has been randomly chosen from all three partitions, and then simulated 10 times. Also all 4 helicopters have been chosen as packet's initiator and then simulated only 1 time. The results are represented in Figure 5.5. All 4 diagrams show three breaks, which correspond to the information diffusion within all 3 partitions. A clear lateness of new HG-version comparing to old version is observed most diagrams especially by joining a new partition. This is a side-effect of both new conditions used in new version of Hyper-Gossiping. Already by encountering first node in the new partition, old version of Hyper-Gossiping will rebroadcast its cached packets, while the modified version waits in the majority of cases until the actual contact rate exceeds the total contact rate. The new mechanism has the advantage that number of potential receivers while rebroadcasting a packet is increased, but it may results in some latency.

Table 5.2 shows a comparison between all simulated scenarios. Although the delay of new version of Hyper-Gossiping is always higher than old one, there is a clear improvement of MNR up to 19%. This example demonstrates that the new version of Hyper-Gossiping works best with partitioned networks.

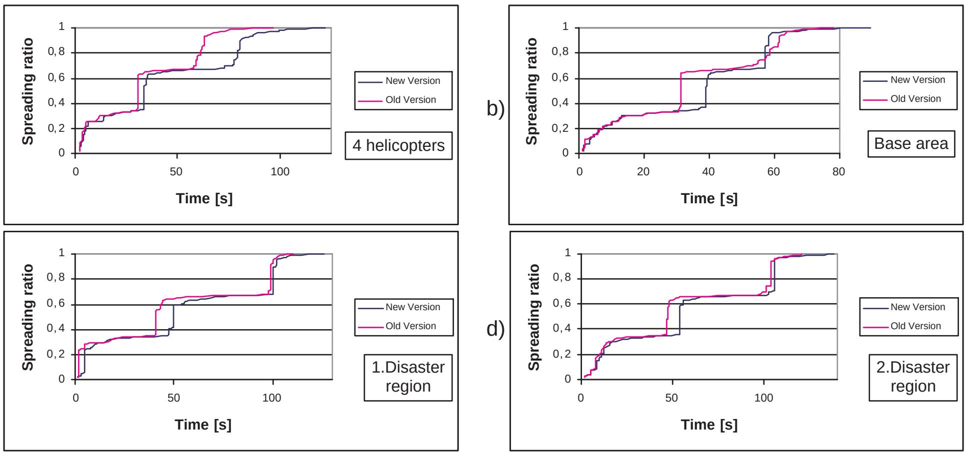


Figure 5.5: Comparing spreading ratio over time for all examples in the disaster-rescue scenario

	helicopters		Base area		Disaster reg. 1		Disaster reg. 2	
	old	new	old	new	old	new	old	new
Delay to reach 100% [sec]	96,02	118,34	78,22	91,56	110,41	125,88	128,05	141,95
MNR	1,67	1,53	1,49	1,37	1,76	1,44	1,94	1,57
ratio of saved broadcasts	8,4%		8,1%		18,2%		19,1%	

Table 5.2: Comparing delay and MNR of both Hyper-Gossiping version for disaster-rescue scenario

Chapter 6

SUMMARY AND FUTURE WORK

Based on knowledge about epidemic spreading of diseases, where contact patterns strongly influence the characteristics of spreading, we presented the notion of *contacts* for MANETs. We then defined new contact-based mobility metrics, e.g. contact duration, contact rate and contact frequency. These metrics have been profoundly simulated and analyzed using the random waypoint mobility model (RWP), where we varied maximum node speed and pause time. Contact-based metrics were compared with link-based metrics. It was observed that the difference between contacts and links in RWP is not really considerable but this was expected because of uniform distribution of nodes in RWP. Contact-based metrics are also strongly time dependent. We also investigated contact-based metrics with more realistic scenarios that are the reference point group mobility model (RPGM) and a typical real-world scenario (i.e. disaster-rescue). Last examples, where network partitioning is normal, showed a clear divergence between link-based and contact-based metrics. These results showed also the possibility of using these new metrics to adapt protocols to node mobility. However, we miss in this work the theoretical examination of contact-based metrics, which can be investigated in future work.

In addition, we showed the usefulness of these metrics for an example of diffusion protocol, i.e. Hyper-Gossiping. We could reduce the number of rebroadcasts up to 16%. As expected this profit is given for highly partitioned networks. We mention here that this improvement can not be attained using existing mobility metrics. We believe, indeed, that contact-based metrics will be very helpful if application is not delay critical, means it can tolerate delays in range of minutes or even hours! In that case Hyper-Gossiping could use mobility to increase the capacity of the network and to save limited resources. But how should Hyper-Gossiping use these contact-based metrics to minimize the number of rebroadcasts needed for dissemination?

Bibliography

[H-MARQ03] Helmy, A.:
Mobility-Assisted Resolution of Queries in Large-Scale Mobile Sensor Networks (MARQ).
Computer Networks Journal-Elsevier Science (Special Issue on Wireless Sensor Networks). August 2003

[Helmy02] Helmy, A.:
Architectural Framework for Large-Scale Multicast in Mobile Ad Hoc Networks.
IEEE International Conference on Communications (ICC 2002). New York, April 2002

[Bai03] Bai, F.; Sadagopan, N.; Helmy, A.:
IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols in Adhoc Networks.
IEEE INFOCOM (The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies). San Francisco, March/April 2003 [INFOCOM April 2nd, 2003]

[EASE03] Grossglauser, M.; Vetterli, M.:
Locating Nodes with EASE: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion.
IEEE INFOCOM. 2003

[Hadel96] Hadeler, K. P.:
In Proceedings of the Dundee Conference on Ordinary and Partial Differnetial Equations 1996

[Sas03] Sasson, Y.; Cavin, D.; Schiper, A.:
Probabilistic Broadcast for Flooding in Wireless Mobile Ad hoc Networks.
In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2003)

[Nav02] Boleng, J.; Navidi, W.; Camp, T.:
Metrics to enable Adaptive Protocols for Mobile Ad Hoc Networks.
ICWN'02, pp.293-298. 2002

[Will02] Williams, B.; Camp, T.:
Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks.
In Proceedings of ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc) 2002, pp. 194-205. July 2002

[Bol02] Boleng, J.:
Exploiting Location Information and Enabling Adaptive Mobile Ad Hoc Network Protocols.
Thesis submitted to the Faculty of the Colorado School of Mines. 2002

[Khel02] Khelil, A.; Becker C.; Tian, J.; Rothermel, K.:
Epidemic Model for Information Diffusion in MANETs.
Proceeding of MSWiM. Atlanta, 2002

[NS-Manu] Fall, K.;Varadhan, K.:
The *ns* Manual (formerly *ns* Notes and Documentation).
The VINT Project (http://www.isi.edu/nsnam/ns/doc/index.html). August 2002

[Kwak03] Kwak, B. J.; Song, N. O.; Miller, L. E.:
A Mobility Measure for Mobile Ad-Hoc Networks.
IEEE Communications Letters, vol. 7, pp.379-381. August 2003

[Song02] Song, J.; Miller, L. E.:
Empirical Analysis of the Mobility Factor for the Random Waypoint Model.
Presented at OPNETWORK. Washington, August 26-30, 2002

[Gerh02] Gerharz, M.; de Waal, C,; Frank, M.; Martini, P.:
Link Stability in Mobile Wireless Ad Hoc Networks.
In Prodeecings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02), pages 30-39. November 2002

[Hong99] Hong, X.; Gerla, M.; Bagrodia, R.; Pei, G.:
A Group Mobility Model for Ad Hoc Wireless Networks.
In Proceedings of ACM/IEEE MSWiM'99, pp.53-60. Seattle, WA, August 1999

[Haas02] Haas; Halpern; Li:
Gossip-based ad hoc Routing.
In IEEE INFOCOM. June 2002

[Vis02] Viswanath, K.; Obraczka, K.:
An Adaptive Approach to Group Communications in Multi-Hop Ad Hoc Networks.
International Conference on Networking (ICN 2002)

[Tseng03] Tseng, Y. C.; Ni, S. Y.; Shih, E. Y.:
Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network.
IEEE TRANSACTIONS ON COMPUTER, Vol. 52, No. 5. May 2003