Design and Implementation of an Anycast Protocol for Wireless Mobile Ad Hoc Networks

Reza Azizi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
July 2011
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Gürcü Öz
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Işık Aybay
2. Assoc. Prof. Dr. Muhammed Salamah
3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

This thesis conducts a detailed simulation study of stateless anycast routing in a mobile wireless ad hoc network. The developed model covers all the fundamental aspects of such networks with a routing mechanism using a scheme of orientation-dependent inter-node communication links. Using a flooding anycast mechanism, the thesis addresses another issue of locating the nearest server from a group of contents-equivalent servers in the network. The simulation model was developed in terms of a class of extended Petri nets and the simulation system Winsim is used in development and simulation to explicitly represent parallelism of events and processes in the network. The purpose of these simulations is to investigate the effect of node's probability of changing direction, maximum speed of the node, and different TTL over the network performance under three different scenarios.

In addition, the thesis provides extensive real-world experimental investigation of wireless ad hoc networks with stationary nodes in outdoor environments. The performance of wireless ad hoc networks is measured under various scenarios.

For the experimental investigations, more than one network configuration and different parameters were used in real-world outdoor environment. Different sets of experiments was done to investigate the effect of inter-packet transmission time and position of laptop from the ground level to the network performance. Conducting such experiments and gathering information will provide very valuable information about wireless ad hoc networks.

Thesis investigates five practically important performance metrics of a wireless mobile ad hoc network and shows the dependence of this metrics on the transmission radius, link availability, maximal possible node speed and different mobility models.

Keywords: Mobile wireless ad hoc networks, anycast, simulation, extended Petri nets, outdoor experimental study, performance evaluation.

ÖZ

Bu tezde kablosuz ve özel amaca yönelik ağlarda noktalar üzerinde durum bilgisi gerektirmeyen herhangi bir noktaya yönlendirme modeli önerilmiştır ve detaylı bir şekilde çalışılmıştır. Önerilen model devre arası iletişim taslağı ile birlikte bu tip ağların tüm temel beklentilerini karşılamaktadır. Bu yöntem dağılımcının herhangi bir noktaya yönlendirme mekanizması ile ağda eşit maksatlı sunuculardan birini belirleme işlemini öne çıkarmaktadır. Simulasyon için kullanılan Winsim sistemi, genişletilmiş Petri-net cinsinden yapılmış modelin geliştirilmesinde ve simule edilmesinde kullanılmıştır. Bu simulasyonlarda kablosuz özel amaca yönelik ağ noktalarının alan içerisinde yön değiştirme olasılıkları, noktaların değişik hızları ve kablosuz ağda yaratılan trafiğin iyileştirme yöntemleri çalışılmıştır.

Bunlara ek olarak tez kablosuz ve özel amaca yönelik ağlar üzerinde yapılan geniş kapsamlı deneysel çalışmaları da kapsamaktadır. Bu tezde, sabit noktalar kullanılarak açık alanda değişik ağ seneryoları kurularak yapılan deneyler de anlatılmıştır.

Deneysel çalışmalarda farklı ağ sistemleri ve değişkenler kullanılarak gerçek dünya ölçümleri yapılmıştır. Yapılan bir gurup deneyde paketlerin gönderim sıklığının ve laptopların yerden yüksekliğinin kablosuz ağın performansına etkisi tespit edilmiştir. Yapılan deneyler ve elde edilen sonuçlar kablosuz ağlar hakkında değerli bilgiler sağlamıştır.

Bu tezde kablosuz özel amaca yönelik ağlarda beş önemli performans ölçüm birimlerini araştırılmış ve bu ölçüm birimlerinin gönderim alanına, bağlantı mevcudiyetine, noktaların hızlarına ve farklı haraketlilik modellerine göre bağlılıkları tespit edilmiştir.

Anahtar Kelimeler: Hareketli kablosuz özel amaca yönelik ağlar, “anycast” gönderim, simulasyon, genişletilmiş Petri-netler, açıkalan deneysel çalışmalar, performans ölçümleri.

to

my beLoved family,

my Parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Any computer network, which is not connected by the cables and in which data is transmitted by using radio waves between nodes of the network is called a wireless network. Wireless networks support mobility, so users have access to network anywhere within the range. Also installing a wireless network is simpler and faster due to the elimination of cables.

Wireless ad hoc network [1] is a type of wireless network that does not need any existing infrastructure such as wireless router or access point. An ad hoc network consists of multiple nodes that are connected through wireless links. Since the transmission range for each node is limited, if receiver node is not inside the coverage area of sender, each node should participate in routing as intermediate node by forwarding data to other nodes until it reaches the receiver. It means every node can work as a router in network to establish a multi-hop wireless link between sender and receiver.

A mobile ad hoc network [2] (MANET) is a decentralized wireless ad hoc network in which nodes can move arbitrarily in any direction; therefore it results in frequent changing of links to other nodes. Like in other wireless ad hoc networks, every node should forward the data which is not related to it, and accordingly act as a router.

As mentioned above each node must be able to work as source, destination, or router and decide which way to route packets. The act of selecting paths to direct the packets or generally network traffic is called routing. Routing protocol is the tool used to control all the transmissions inside the network. It also should be able to handle the topology changes as a result of node's random movement through the wireless network. Some other issues that we should concern when drafting a routing protocol is: power consumption, limited bandwidth, mobility, and expense. An efficient routing protocol should balance between the issues cited above to have the ideal performance.

Flooding [3] is an algorithm in which every incoming message is sent to all reachable parts of the network. It is easy to implement and is used as a part of some routing protocols. Anycasting algorithm is used to choose the topologically nearest node in a group of possible receivers and forward data toward it.

Mobile ad hoc networks don't need presence of any infrastructures to be established [1]. Due to their dynamic nature, they have wide usage in military scenarios to disaster relief operations or sensor networks. Beside they are also used increasingly in our everyday life for transferring the data between wireless devices, and mainly to share internet in home networks or public places like airports, restaurants.

A detailed simulation study of stateless anycast routing in a mobile wireless ad hoc network is conducted. The proposed scheme enables representation of reliability aspects of wireless communication in a general and flexible way. Using a flooding anycast mechanism, the paper addresses issue of locating the nearest server from a group of contents-equivalent servers in the network. The simulation model was developed in terms of a class of extended Petri nets to explicitly represent parallelism of events and processes in the network. The goal of this thesis is to investigate an anycast routing protocol characteristics in wireless ad hoc network under different conditions with use of some performance metrics. In simulation, the behavior of five fundamental performance metrics - response ratio, average number of hops, relative network traffic, average response time and duplicate ratio - was investigated with varying distance of transmission and different combinations of model parameters.

The rest of the thesis is organized in the following way. Chapter 2 presents a classification for routing algorithms in wireless ad hoc networks and their characteristics. Chapter 3 provides specification of system assumptions and the chosen mobility model in simulation modelling. Chapter 4 explains the the application-layer program which was used in our experiments, and the organization of our conducted experiments. At the end of Chapters 3 and 4, results and discussions are mentioned. Finally Chapter 5 concludes the thesis.

Chapter 2

ROUTING PROTOCOLS IN AD HOC WIRELESS NETWORKS

2.1.Main Approaches to Investigate Wireless Ad Hoc Networks

There are two main approaches to investigate the performance of wireless ad hoc networks: Simulation modeling and real-world experiments.

First way is simulation modeling which needs less time and resource in comparison to real-world experiments, which requires a huge amount of resources and much more time. Physical and environmental conditions can affect the behavior of wireless ad hoc networks in reality, and simulations cannot put all of them in account. As a result, some of the hypothesis which used to simplify the modeling may lead us to invalid results in simulation. In real-world experiments, however, very precise and precious information about characteristics of ad hoc networks is obtained in exchange for larger resources and longer test times.

2.2.Survey of Routing Protocols

An ad hoc mobile network is an autonomous system consisting of mobile hosts that do not rely on the presence of any fixed network infrastructure [1]. In ad hoc networks, nodes are free to move in an arbitrary manner and in cases that mobile nodes cannot reach to the destination directly will relay their messages through other nodes. In comparison with wired networks, in ad hoc networks all the nodes must participate in the routing procedure. The basic characteristics and performance of the wireless ad hoc network is relevant to its underlying routing protocol. Most routing algorithms use only one network path, while multipath routing techniques allow us to use alternative paths as well.

Routing algorithms can be categorized by difference in delivery semantics: unicast, broadcast, multicast, and anycast [4]. When destination of all the packets which send from source through the network is a single specific node, it means we use unicast algorithm. In contrast to one-to-one distribution of unicast, broadcast is one-to-all, which mean delivering the packet from source node to every single node inside the network. Similarly, multicast messages are delivered to a group of nodes in network, which represents one-to-many relationship. Lastly, anycast is a transmission methodology in which packets from a source node are routed to the nearest server or to best localized server in a group of potential receivers all identified by same destination address, so it can be described as one-to-one-of-many relationship.

In ad hoc networks, nodes are not familiar with topology of network from the beginning, so they need to discover it. By this fact, we can divide ad hoc routing protocols into three categories: proactive protocols, reactive protocols and hybrid protocols. In proactive routing protocols, every node prepares one or more table which contains routing information from itself to every other node as destinations inside the network. These tables are updated regularly by exchanging information between nodes, in order to maintain latest network topology. Proactive or table-driven protocols results in a high overhead on the network. Also network shows slow reaction to failures or restructuring. Unlike proactive routing protocol, reactive routing protocol finds a route to a destination by route discovery process on demand.

For this reason, reactive or on-demand protocols have higher latency time, but lower overhead. Hybrid routing protocol is a combination of proactive and reactive routing protocol, in order to combine their advantages. First scenario is that network can be divided into zones, and use one protocols within the zone, and another outside it. Usually proactive protocol is used for the nodes which are close to the destination, and other nodes which assumed far work under a reactive protocol. Also there is a second scenario which a primitive, basic routing is provided through some proactively discovered routs and then uses reactive flooding to serve the demand on adjunct nodes.

Table 1 shows classification of some existing ad hoc network routing protocols. Some of the protocols are described in more details in the next section of this chapter. Also you can refer to citations to find more information about them.

Table 1: Classification of ad hoc network routing protocols

	Unicast	Multicast	Anycast
Reactive Protocols	AODV DSR	MAODV ODMRP	A-AODV ARDSR A-DSR
Proactive Protocols	DSDV OLSR	MOLSR	Route-Count Based Anycast Routing Protocol
Hybrid Protocols	ZRP	ZMAODV ZODMRP	Hybrid Anycast Routing Protocol

2.3. Wireless Ad Hoc Networks Routing Protocols

Dynamic source routing (DSR) [5] is a reactive protocol for ad hoc wireless networks. DSR is using route discovery protocol –broadcasting the route request packet and waiting for route reply packet which contains a sequence of network hops which establish route to the destination- to dynamically find a route to any other node in the network. Also a unique request id is assigned to each route request packet to discover the duplication in route requests which received. Route maintenance procedure controls the success of operation, in the case any problem happens in between current route.

Highly dynamic destination-sequenced distance vector (DSDV) [6] routing is a proactive protocol in which up-to-date routing tables are broadcasting periodically by each node to index which nodes are achievable from it. A sequence number, destination's address and number of hops needed to reach the destination is the content of data broadcasted by each node. Routes that have latest sequence number are selected when decision making is required to forward the packets. The receiver increase the metric since incoming packets will need one more hop to reach the destination before advertise it to its neighbors. If there is no broadcast from link for a specific time, it considers as broken link and will disclose in routing packets.

Ad-hoc on-demand distance vector (AODV) routing [7] is an on demand (reactive) routing protocol with little or no reliance on periodic advertisement which is compatible with dynamic self-starting networks. Nodes which are not placed on an active path don't take part in periodic routing table exchanges. Even it's not necessary for them to discover and maintain a route to another node unless prior node is forwarding packets as an intermediate station. AODV uses a modified version of broadcast route discovery mechanism which is based on DSR algorithm, which dynamically builds route tables at intermediate nodes. Destination sequence numbering method is also inspired from DSDV to acquire fresh routing information between nodes and ensure a loop-free routing. As we mentioned above AODV protocol is a combination of DSDV and DSR which decrease the network traffic, and can handle topology changes better.

Optimized link state routing (OLSR) [8] protocol is a proactive protocol which shows good performance over large population ad hoc networks. Each node *N* in the network selects a set of neighbor nodes with a bi-directional link which are called multipoint relays of *N* to retransmit *N*'s packet, and other neighbor nodes that are not a member of multipoint relays of *N* just will receive the packet and process it but will not broadcast it. Every node choose its multipoint relay group from its one hop neighbors in the way that group covers all the nodes that are two hops away and broadcast information about its group periodically so that OLSR protocol can use these groups to reach destinations inside the network. It results in smaller size of control packets and also reduces the flooding of control messages in OLSR protocol. Performance of exciting ad hoc algorithms such as AODV, DSDV, DSR, and OLSR is compared together in a lot of references [9, 10, 11, 12, and 13].

ARDSR [14] is a DSR based on-demand anycast routing protocol. Routing discovery is used whenever a node requires a route. Source node floods an anycast request packet to its neighbors which contains a sequence number, node list, and address of source node and anycast address. Node list is used to keep addresses of all nodes which packet traverse on its way to destination. So every node will add its address to node list before broadcasting it again. If the source node didn’t receive any reply packet during the specific period, it will try again. After reaching the maximum number of retransmission all the data packet for that destination will drop from buffer. When destination node in anycast group or a node which has the route to destination receives the request packet, it will generate a reply packet and send it back to source node. Source node will select the reply packet with smaller hop count as its route when receives multiple reply packets. Route maintenance checks the correctness of anycast routes to prohibit source node from sending data over an invalid path. Every time a node receive same packet which already forwarded to its next hop along the anycast route, it can make sure that next node received it correctly. This method is called passive acknowledgment. Link will be consider broken, and an error packet will be send backward along the transmission path until inform the source node from link failure, if still there is no receiving confirmation after a number of retransmissions.

A-AODV [15] is a reactive routing protocol which adopts AODV protocol to work as an anycast routing protocol. The first 5 bits of reserved field in AODV message format is employed for implementing anycast protocol. The first bit which is named Anycast flag decides whether packets are sending in unicast or anycast manner. If flag A=0, message is send as unicast and next 4bits are equal to 0000. If flag A=1, the next 4bits presents the anycast group ID. Anycast group ID needs to update every time a node joins or leaves an anycast group. All other fields work same as in unicast. A-AODV and ARDSR, two anycast based reactive routing protocols are compared together in [16], by the terms of delivery ratio, end-to-end delay and energy consumption.

A-DSR [17] is a DSR based on demand anycast ad hoc network routing protocol. First of all, every node requires maintaining an extra anycast group table to support anycasting. This table contains following fields: destination IP address, anycast group ID and lifetime. Lifetime shows the time when record was created or last time that it was updated. To distinguish between unicast or anycast packets, a flag is added into DSR header. If flag is not 0, it is used for anycast services. Anycast group ID is a 4bit integer, and individual for each anycast group. Creating a new anycast group, joining and leaving an anycast group are the operations that should be done to produce the anycast group table.

MAODV [18] is a multicast version of AODV which is capable of multicasting as well as unicast. Each node maintains two tables. Route table is first one which is used for collecting information for routes to other nodes in network. Second table which is named multicast route table contains necessary data about multicast groups and its leader. Group leader is the first node which demand membership in a that group. Also a multicast tree is created when a node join the multicast group. There is a third table named request table which is maintained only for the nodes that supports multicast routing. In [19], two new hybrid multicast routing protocols named ZMAODV and ZODMPR are proposed, and their performance is compared with their original counterparts.

In [20] an anycast routing protocol is proposed which is a table-driven protocol. Anycast hybrid routing protocol [21] also exist. This hybrid protocol is based on AODV routing protocol.

2.4.Survey of Experimental Studies in Wireless Ad Hoc Networks

Real-world experimental investigations can be categorized as indoor, fixed outdoor and mobile outdoor setups [22].

In [23] a set of common assumptions that are largely used in ad hoc network simulation studies was mentioned, and the weakness of these assumptions were proved by conducting outdoor experiments with 33 laptops. It also explains the difference between simulation and experimental results and be a guide for MANET researchers.

Comparison between four different routing algorithms is done in [24]. APRL, AODV ODMRP, and STARA were selected as routing algorithm for these outdoor experiments. They used one laptop to control the experiment and 33 laptops were moving randomly through a rectangular field to conduct the experiments. Each laptop had Wi-Fi 802.11b wireless card and also GPS service. GPS service was used to record the position of laptops every three seconds and used these position traces in indoor experiments.

An experimental comparison between AODV and SAODV routing protocols is done in [25]. For experiments they used ten 802.11-enabled laptops within a 250m by 100m field. First initialization of laptops were random for all experiments and laptops movement was by random waypoint mobility model with maximum node’s speed 2m/s. Later result of the experiments was compared with results they obtained via simulation.

In [26] a multithread program is used to investigate the data transmission in a wireless ad hoc network in outdoor fixed environment. Program is based on the prototype program presented in [27] and is developed under Windows OS and tested on a group of laptops with 802.11 a/b/g Wi-Fi wireless interface. Two different sets of experiment were done by changing the inter-node distance, and number of intermediate nodes, with varied application data size as a parameter for each set.

In [28], they extended the program presented in [27] to support more than one destination nodes in ad hoc network. In first network configuration in this fixed outdoor network, source node was positioned at center and three destination nodes were placed equally on different inter-node distance from source node to observe its effect on network performance.

In [29], they used application layer program which explained already in [28] over similar laptop computers with 802.11 b/g Wi-Fi wireless interface adaptors. Network consists of a source node, a destination node and eight intermediate nodes. Source node and destination node are placed out of each other’s coverage area in such a manner that packets are transmitted through the intermediate nodes which were placed randomly in the field. The average number of hops metric is used to present the efficiency of transmitting packets through intermediate nodes.

Chapter 3

MODELING AND SIMULATION OF ANYCASTING IN WIRELESS AD HOC NETWORKS

3.1.Extended Petri-Nets

A Petri net is a graphical and mathematical tool for modeling, and one of the most common methods. It is possible to collect significant information about the structure and the dynamic manner of a system by using Petri net's analysis. It is also easy to edit or adjust models when it is needed. Petri nets are specially used for concurrent systems [30].

The structure of a Petri consists of three elements: places, transitions and directed arcs. Places can be divided into simple and queue places. Simple places can have only one token at a time, unlike the queue places which have unlimited number of token at once. Simple places are shown by circles and queue places presented by ovals. Arcs run from a place to a transition or contrariwise. It is not possible to have an arc between two places or two transitions. Input places of the transition are places where an arc runs to a transition, and the places to which arcs run from a transition are called the output places of the transition. When there is a token at the start of input arcs, transitions can fire and token will be used eventually. The behavior of Petri nets can be present through their transitions [31].

An Extension of Petri nets was defined to cover the problems of original Petri nets and use it as simulation tool [32]. Evaluation nets or E-nets are a class of extended Petri nets which are suitable for modeling of simulation systems. E-nets have five kinds of elementary nets with particular number inputs and outputs. Minimal, functional and complete component of extended Petri nets are elementary nets, which consist of the minimal structural elements. An elementary net *E(t)* of a transition *t* can be defined with the following expression [32]:

$$E(t) \; = < \; C, P1, P2, r1, r2, d \; , m \; > \qquad (3.1)$$

Where *C* is a necessary (but not sufficient) condition to fire transition *t; P1* and *P2* are finite sets of inputs and output places for *t*, with $P1 \cap \; P2 = \emptyset \; and \; P1 \cup P2 \neq \emptyset$; *r1* and *r2* are functions of input and output selection respectively; *d* is delay function; and *m* is a data transformation function [32].

Among all available structures of elementary nets in extended Petri nets, it is proven that just 5 types of elementary nets are enough to model any data processing system [33]. In order to develop simulation model of wireless ad hoc network only three basic elementary nets types T, X, and Y shown in Figure 1 were used.

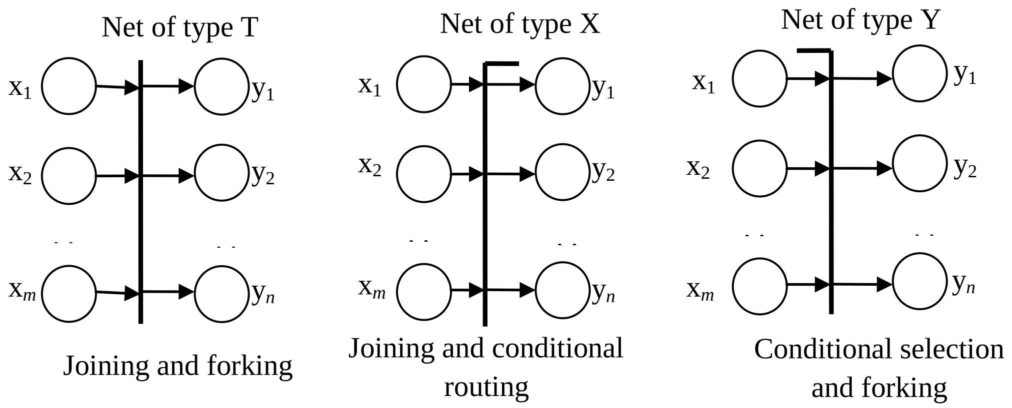

Figure 1: Elementary nets T, X and Y of extended Petri nets.

elementary net is described as follows for its functionality; associated time delay, and transformation of token attributes. To fire a T transition, it is required that all input places are with token and all output places are empty. Elementary net of type Y provides conditional selection of a token one of the input places of active transition. Existence of at least one token in one of input places and being empty at all output places are requisites to fire this net. Elementary net of type X supplies conditional selection of one of output places. It is required for all the input places to have tokens.

3.2.Simulation System Winsim

Winsim is the simulation system which used for modeling and simulation of wireless ad hoc network in this thesis. “The developed simulation system Winsim implements a class of extended Petri nets with attributed tokens and associated functions of time, data transformation and control in transitions. It is especially useful for modeling and simulation of parallel and distributed systems and the related algorithms.” [34]. Winsim Also has high level programming language possibilities for processing complex data, and provides quick simulation.

Model Description Language (MDL) and Modeling Control Language (MCL) are tools used by user to interact with the simulation system, during creation and execution of models. MDL used to input our model information as a set of segments during creation of the model. Model can consist of one or more segments, which will be linked to each other before execution of the model. For the models which have multiple segments which are exactly same, it is possible to use several copies of that segment. It will simplify the process of developing the model. MDL is implemented as an extension of Object Pascal Language. MCL can be used before or during the launch of ready model to manage and control the simulation run by initiate values to parameters used in the model. So there is no need to recompile the model, to run simulation with different parameters.

The model can be executed by desired MCL statements after compiling and creating the model in Winsim simulation system.

3.3. The System Architecture and Assumptions

In this part, firstly I will specify assumption and configuration of wireless ad hoc network. The area of network is assumed to be restricted to a rectangular shape with system configuration parameters x_{min} and x_{max} for horizontal axis, and y_{min} and y_{max} for vertical axis. Also the number of the nodes within this area is fixed and their primary distribution assumed to be random with uniform probability distribution within the (x_{min} , x_{max}) and (y_{min} , y_{max}) limited area.

Next assumption is that, nodes have capability to communicate with each other, by using of bidirectional wireless channels. The transmission radius is assumed to be same in different directions. Besides, even within this limited coverage area, inter-node connection is not reliable due to different reliability aspects of wireless communication, like interference, fading, or climate conditions. Each node has a unique identifier or address.

Another assumption is that, movement of the nodes in the given area is same in form with a chosen mobility model. A node will bounce and continue moving within the area in a new direction, if it reaches to the borders.

This model assumes that nodes change position alternatively at discrete steps. This time interval for each step, is another system configuration parameter, and is defined by τ. Therefore, if current location of node i at the time t is $(x_i(t), y_i(t))$, it will change to $(x_i(t+\tau), y_i(t+\tau))$ at time $t+\tau$. Correspondingly, $\Delta x_i = |x_i(t+\tau) - x_i(t)|$ and $\Delta y_i = |y_i(t+\tau) - y_i(t)|$ are distances that node i travels along the horizontal and vertical axes during each step. Maximum distance that node i can traverse during time interval τ are denoted by Δx_{max} and Δy_{max} in both axes. Δx_{max} and Δy_{max} are also system configuration parameters. As a result, by having the step duration τ, and maximal distances Δx_{max} and Δy_{max} maximal node speed in X and Y directions can be calculated as follows:

$$V_{max}(X) = \frac{\Delta x}{\tau} \tag{3.2}$$

$$V_{max}(Y) = \frac{\Delta y}{\tau} \tag{3.3}$$

Values of Δx_i and Δy_i are different for each node, and are selected from uniform probability distribution in the range $(0, \Delta x_{max})$ for X axis and $(0, \Delta y_{max})$ for Y axis.

Thus each node is moving with a different speed in the range of $(0, \Delta V_{max}(X))$ and $(0, \Delta V_{max}(Y))$.

3.4.Mobility Model

A mobility model controls the movement of mobile nodes, and change in their speed and location. Mobility models are used in simulation-based network evaluation in order to measure performance. Mobility models are divided into two major categories: traces and synthetic models. Traces collect its information from observing mobility patterns in real life systems. So, increasment in number of participates and observation interval eventuates to more accurate information. Thus, it is applicable for the network environments that traces already exist for them. For new network environments (e.g. ad hoc networks) we need to use synthetic models. Synthetic models try to present the realistic manner of mobile nodes without using traces [35].

The Random direction mobility model [36] is one of commonly used synthetic models. In this model, mobile nodes select a random direction and speed in which to travel to a destination at the border of network. By reaching the simulation borders, the mobile node becomes stationary for given pause time. Afterwards, mobile node chooses another angular direction (between 0 and 180 degrees) and continues the movement. Average hop count for Random Direction Mobility Model is higher than most other mobility models. More information about mobility models for ad hoc networks can be obtained from [37], [38], and [39].

In this thesis, the random direction mobility model is used, but with some modifications. In the modified version, mobile nodes still continue to select random directions but can change their direction of movement at the end of any step, with the probability *p*. So nodes are not forced anymore to reach the borders to choose a new direction. By using the *p* probability as another system configuration, we can demonstrate various motion patterns. The original random direction model can be reached, if value of p is set to zero in extended version.

A flooding-based [40] simulation system was developed relying on the chosen mobility model. System is used to localize an anycast server in wireless ad hoc network by employing anycast service. It is assumed that there are two types of nodes in the network. Simple nodes (clients) are the first type of nodes, they are sources of anycast requests. Simple nodes (intermediate) re-transmit anycast requests, which come from source nodes, in multicast mode inside the network area. Simple nodes are also capable of forwarding unicast replies generated by server nodes.

It is assumed that there is one group of anycast servers in the network with five identical mobile server nodes. These server nodes are distributed randomly in the network area.

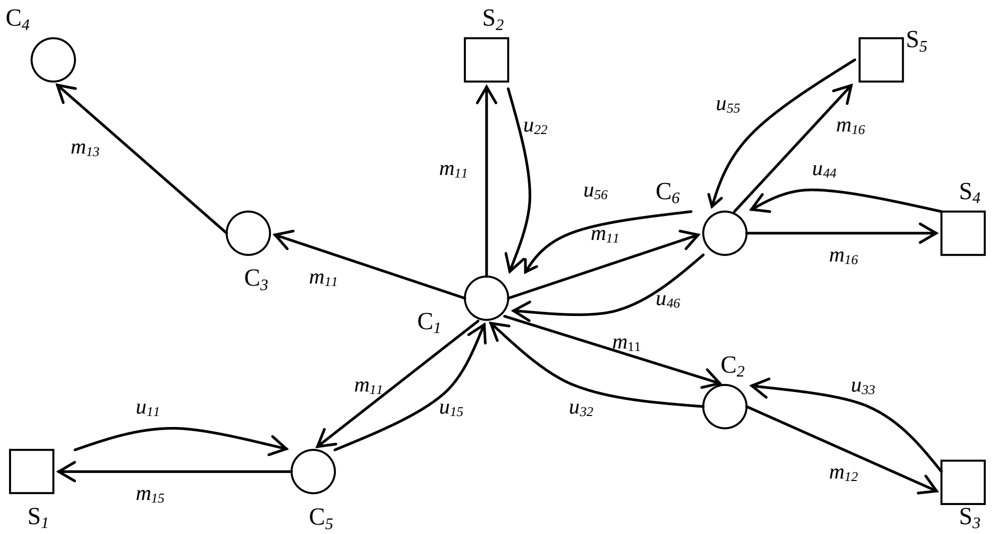

Figure 2: A scenario of the pure flooding for anycasting in an ad hoc WLAN.

Figure 2 presents a scenario of the pure flooding scheme. In the figure, m_{ij} is a multicast request to an anycast server generated by requesting node, *i* and transmitted most recently by node j. U_{ij} is an unicast reply of anycast server node *i* and transmitted most recently by node j, where, C_1, C_2, ... C_m represents the simple nodes (clients) and S_1, S_2,.... S_n represents the anycast server nodes.

In this method, the requesting node (and each intermediate simple node) will transmit (re-transmit) a request message in multicasting mode, i.e. to all close neighbors (nodes in the coverage area). The server node will transmit its reply always in unicast mode, using the addresses of intermediate and source nodes in the received request. The server never re-transmits a reply from any other server node.

When a multicast request message is transmitted from a source client node, it stores the source address and addresses of all intermediate nodes as shown in Figure 3.

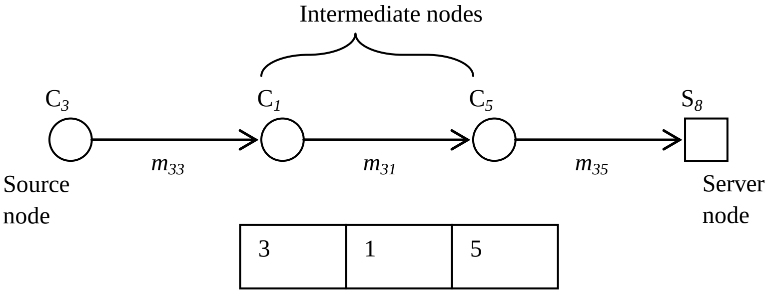

Figure 3: An addressing part of a multicast request message.

In the figure, request message m_{35} received by server node S_8 contains addresses 3, 1 and 5 of nodes C_3, C_1, C_5 with 3 as the address of the source node. As Figure 2 shows, the source node can receive replies from a few server nodes. In this case, the source node can choose the server from which the reply comes first since this server is probably the nearest server. Then the source node will discard reply messages from all other servers. They are considered as duplicated messages.

Figure 4 presents a possible scenario of request-reply messages in a network with six client nodes and three server nodes.

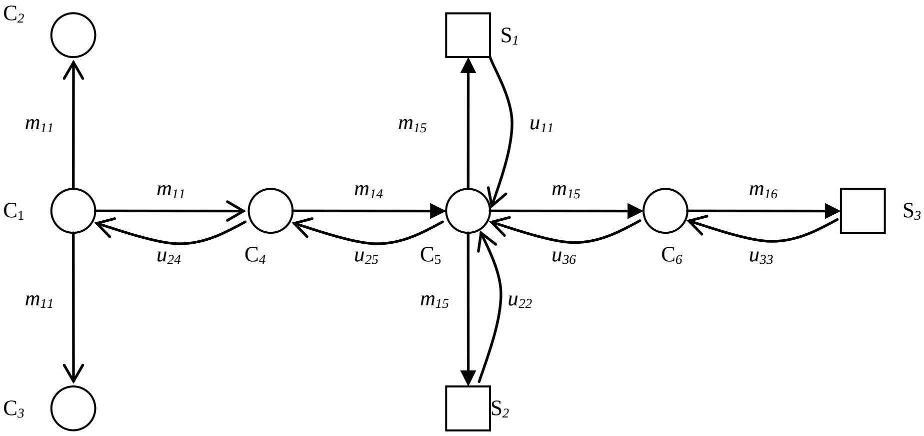

Figure 4: A possible scenario of request/reply messages in a network with 6 clients and 3 server nodes, where C_1 is the source node.

In Figure 4, only a reply from server S_2 is delivered to a client C_1 in the form unicast message u_{24}. Replies from servers S_1 and S_3 (see unicast messages u_{11} and u_{36}) are discarded by node C_5 after it forwarded a reply from server S_2. In the figure, m_{1i} is a multicast request message generated by client node 1 and transmitted (re-transmitted) by node i, and u_{1i} which is an unicast reply message initiated by server 1 and transmitted (re-transmitted) by node i.

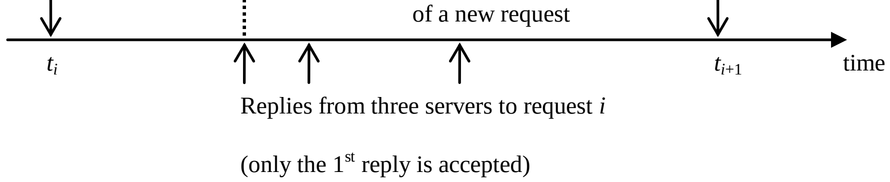

Figure 5: A timing diagram of requests and replies.

A timing diagram of multicast requests and unicast replies is shown in Figure 5. The interval of preparation of a new request should be sufficiently large to receive all reply messages (to accept only the first one and to discard all the subsequent replies). Each reply will contain a unicast address of the replying server. This address could be used in the subsequent point-to-point communication between the source node and the server.

In the current model, there is only one source of requests. Therefore, logically, the system is equivalent to a finite population queuing system [41] with one client and a few identical servers.

3.5. Structure of the Model

The suggested model of a wireless mobile ad hoc network model, with anycasting scheme is developed using extended Petri nets. Simulation system Winsim is used to implement this multi-module model. An inter-node communication scheme which is already implemented in [42] is used for this model.

The proposed model is based on a general model of WLAN in [30], and is composed of two types of modules as shown in Figure 6. The first module type, which is named “node module”, performs functionality of a node in the wireless network. Therefore, the total number of modules that is required is equal to number of nodes of the network.

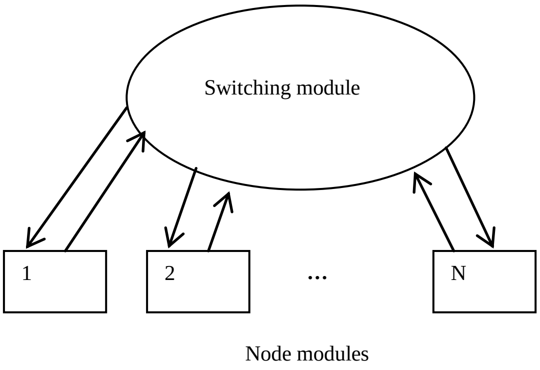

Figure 6: The General structure of the model.

The second type of module is called “switching module”. It is the duty of the switching module to recognize which nodes are able to receive data packets which another node transmits through the network. Switching module is also responsible to direct the random movement of each node according to the chosen mobility pattern. For this reason, the switching module creates and updates coordinate of each node in specified time intervals.

The switching module makes essential initialization of the node module at the beginning of the simulation run. The switching segment which represents switching module works as the main segment, and all other segments which represent all nodes are attached to it.

The block diagram of switching module is shown in Figure 7, and the Petri net scheme of switching module is shown in Figure 8. In this scheme, the switching module starts to work by firing of transition T4, which generates the random coordinates for first time. Then transitions Y1 and T1 as a loop, periodically update the coordinate of mobile nodes in network until end of simulation.

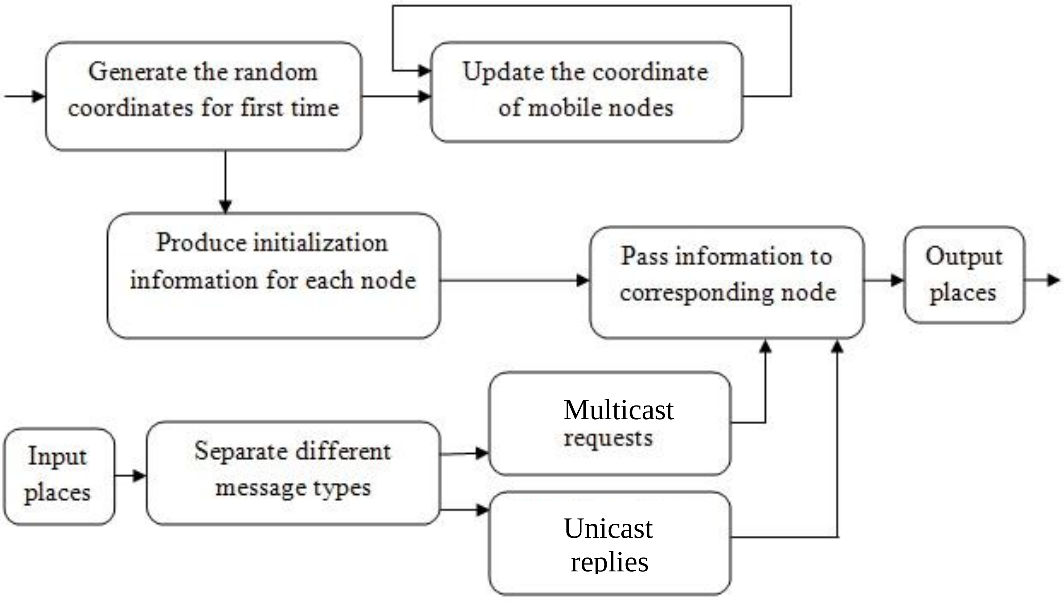

Figure 7: Block diagram of the switching module.

Transitions Y2, X2, T8 and T9 which form another loop, are used to produce and pass initialization information to each node module. Transition Y3 and X2000 are responsible to pass this initialization information which is in service message schema. In more detail, X2000 uses a particular place from output places S201, S202, … , S250 to hand over the service message to corresponding node module. This loop runs once at the beginning of the simulation.

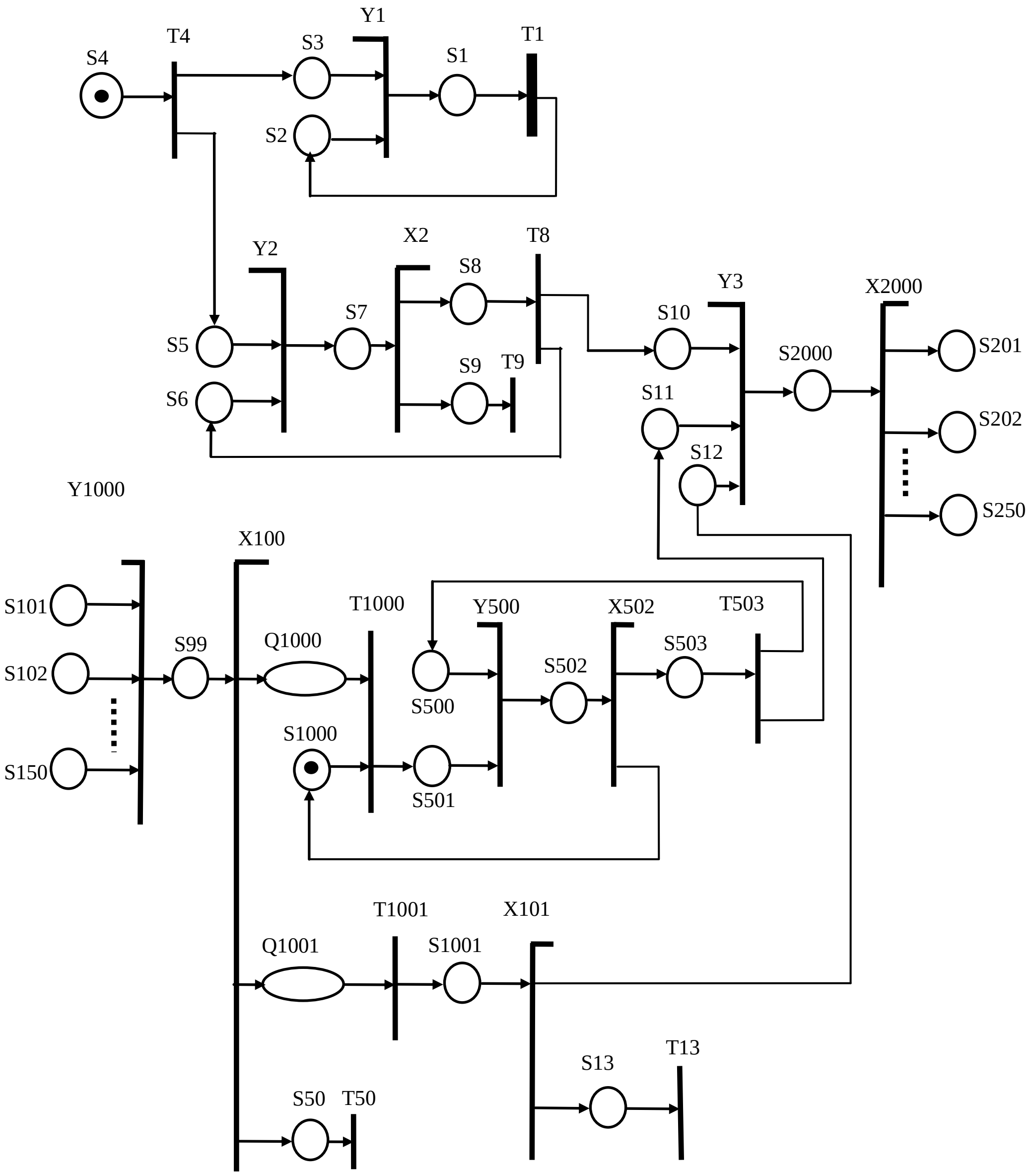

Figure 8: Petri net scheme of the switching module.

Next scope of switching module is in charge of controlling the communication links between senders and nodes that are able to receive it. Each node that wishes to transmit a packet, submits its request to switching module via one of input places S101, S102, … , S150 (for a network with N=50 nodes). Nodes can submit either a multicast message or a unicast reply.

To localize an anycast server, it is necessary to generate a multicast request. After a server receives this request, an acknowledgment will be sent back to the source node by using unicast reply. The switching module uses two different scopes for different message types. It is the task of the loop consisting of transitions T1000, Y500, X502 and T503 to handle multicast requests when unicast replies are handled by transitions T1001, X101 and T13 as a whole. T13 discards the unicast reply if the receiving node is not reachable. Packets which are sent by nodes waits in queue Q1000 until transition T1000 pass them to potentially reachable neighbors, according to transmission radius. Transition X100 is used to separate the two message types mentioned above, and will discard any other type of message.

As mentioned before, the network nodes are categorized into two sets: simple nodes or client nodes and anycast server nodes. In the developed model, it is assumed that there is one simple node N which generates requests to localize anycast server. The rest of simple nodes are responsible to forward requests from source node N to anycast servers and forward back replies from anycast servers to source node.

Although there are two variant sets of network nodes, both of them are implemented by the same type of module. The block diagrams of node module is shown in Figures 9 and 11. Also, the Petri net scheme of node module is given in Figures 10 and 12.

Initializing data from switching module, requests for multicast transmission from other nodes and replies from anycast servers are three different types of inputs for every node. Separating these three types of inputs from each other is duty of transition X1. Transition T1 sets initial state of each node once and only after receiving initialization data. Transition Y40 and T41 together establish a loop that activates by receiving a token through place S40. The goal of this loop is dynamic control of directionally dependent links for the given node.

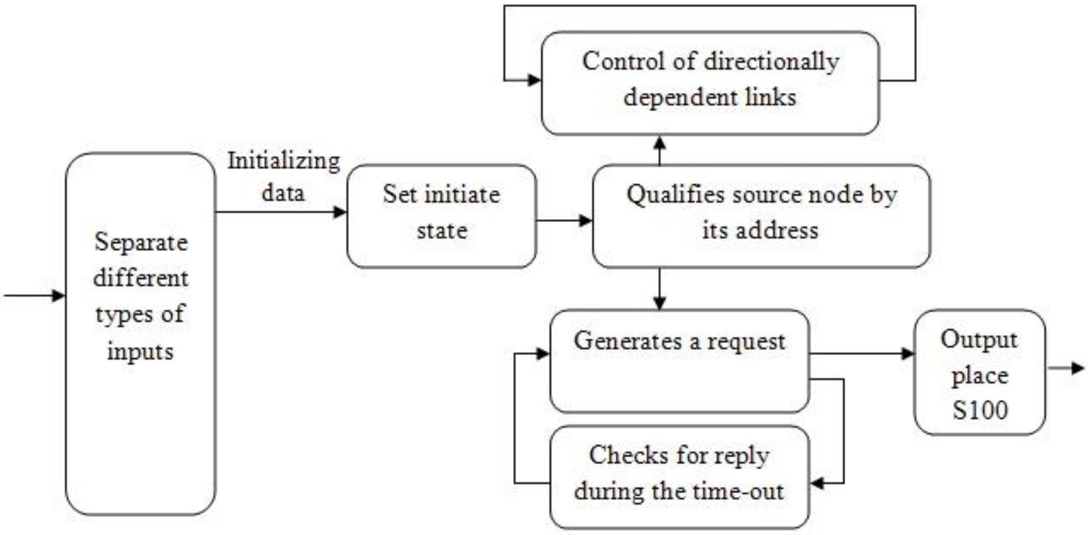

Figure 9: Block diagram of the node module (Part 1).

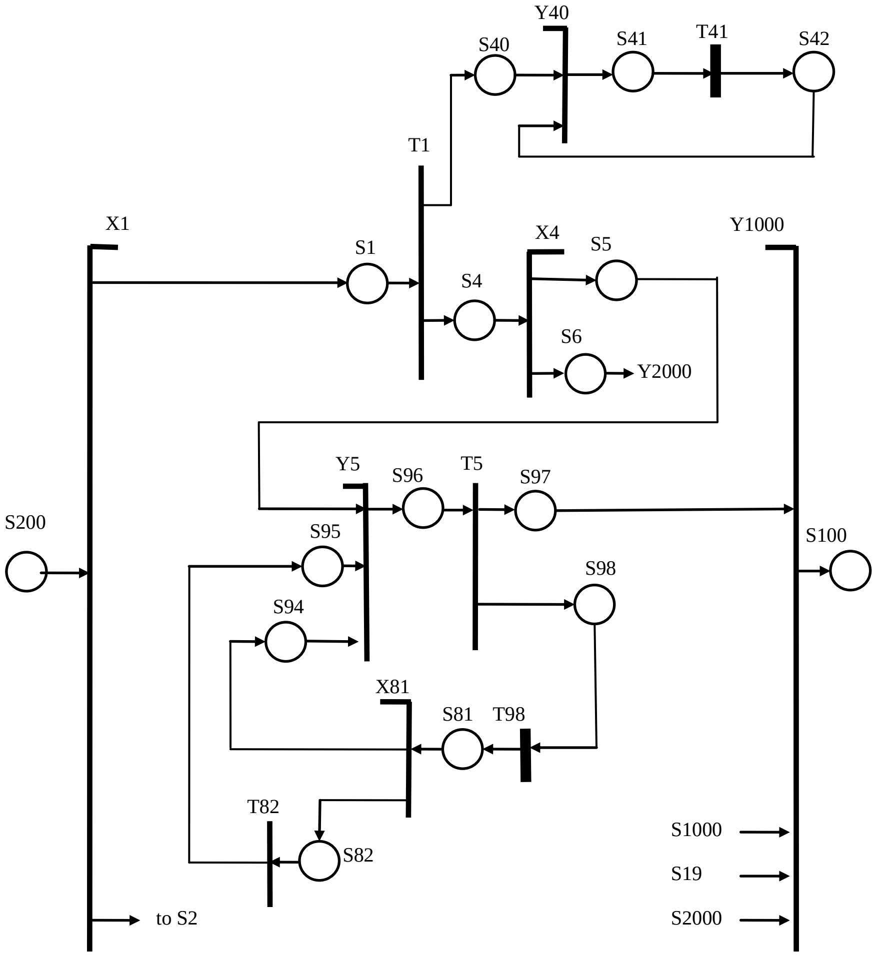

Figure 10: Petri net scheme of the node module (Part 1).

Transition X4 qualifies source node by its address and allows only to it to create multicast request to localize anycast servers. As mentioned before, this model has only one source node.

Requests are produced periodically by the loop which contains transitions Y5, T5, T98, X81 and T82. When transition T5 generates a request, it is handed over to the switching module through place S97, transition Y1000 and output place S100. Also transition T98 starts a defined time-out for a copy of this request via place S98. After time-out finishes, transition X81 checks if a reply is received during this period or not. If a reply is received during the time-out, transition T82 will record its characteristics. If no reply was received, loop will start to repeat by a token via place S94.

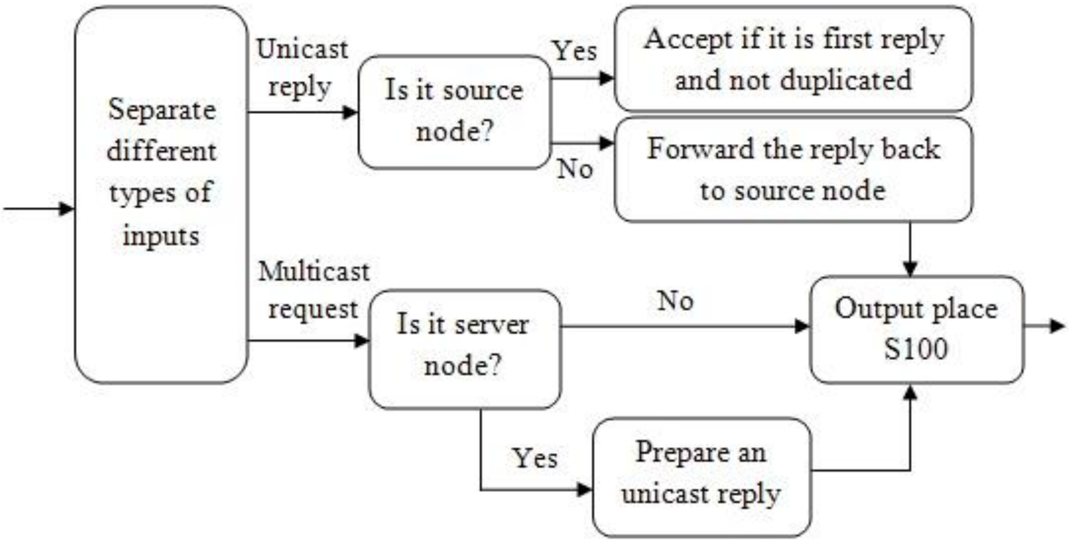

Figure 11: Block diagram of the node module (Part 2).

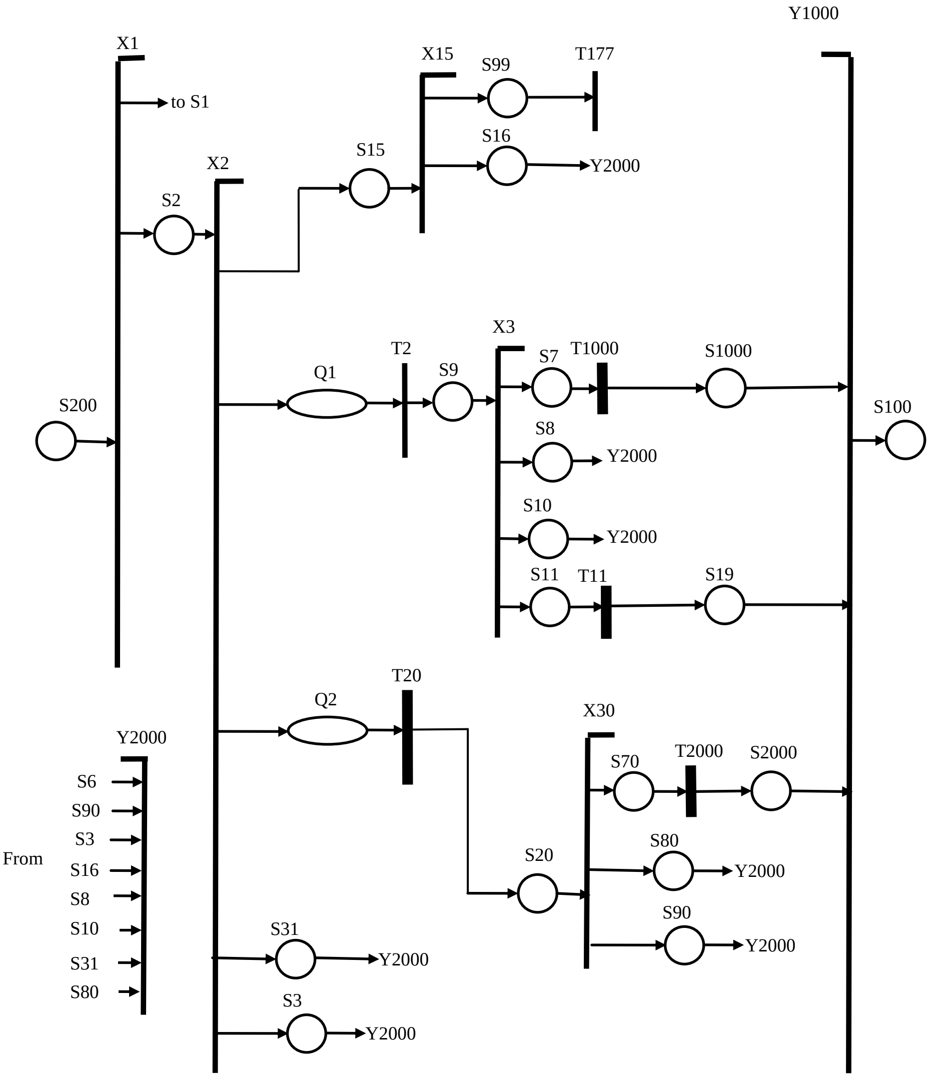

Figure 12: Petri net scheme of the node module (Part 2).

Each request and reply that received by any type of node passes through place S2 and after is divided by transition X2. In case source node receives a unicast reply for its request, transition X15 will be active by token that place S15 passes. Transition X15 will lead to place S99 if it is earliest reply or to place S16 if it is a duplicated reply.

If the node that receives request is not a source node, it will pass to transition X3 through transition T2 and place S9 after waiting in queue place Q1. Transition X3 will divide them into server or non-server subsets. For non-server subsets, the request will pass to the switching module by output place S100 via transition T1000, place

S1000, and transition Y1000. Transition T1000 is used to implement the propagation delay.

For server subsets, requests will be discarded via places S8 and S10 in the situation when the TTL becomes equal to 0 after a decrement, or when the request is a duplicate. Otherwise, the server will provide an unicast reply by transition T11 and pass it to the switching module. If a non-source node receives a unicast reply, transitions T20 and X30 will handle this reply. Then reply will be forwarded back to source node by transition T2000. Places S80 and S90 are used in order to discard the reply if it has TTL=0 or is a duplicate.

There are two more cases that transition X2 is responsible to separat. The first case is when a message (request or reply) is received at the period that link status is OFF, the message is discarded via place S31. The second case happens when a source node receives a request, so S3 will count it and then discard it since source node is the only node generates requests but never forwards them to other nodes.

3.6. Performance Metrics

Performance metrics are used to help researchers to investigate the wireless ad hoc networks. Delivery ratio and average number of hops per delivered packet are the most popular performance metrics used for this reason. The delivery ratio (referred as response ratio) characterizes how the network is effective in delivering packets from source nodes to server (destination) nodes. Average number of nodes that a packet traverses in its way to the source node is represented as the average number of hops. Both of these performance metrics have direct relation with the implemented routing algorithm, node mobility models and inter-node communication links specification [42].

Response time is another performance metric. Response time is the time interval between the moments the source node sends a request, until the reception of the reply message. This metric is important for some real-time applications which need small time interval.

Each packet that is transmitted by a source node will be usually retransmitted by some intermediate nodes until it is received by server nodes. Relative traffic is the performance metric that represents the number of times each packet is transmitted by other nodes. As matter of fact, it is necessary to keep relative traffic as low as possible to have less overloading in the network.

All received replies in source node for a request, after receiving the first reply are taken as duplicated replies. This characteristic of the network behavior is shown by duplication ratio performance metric. It has a direct relation with robustness and availability of the network, but should not be large to have less traffic in the network.

Assuming that there is only one source node in the network, the five performance metrics listed above, can be formally defined as follows:

The first performance metric, response ratio of packets, is defined with the expression:

$$n_s = \frac{N_r}{N_s} \tag{3.4}$$

where N_s is the number of request packets transmitted by the source node and N_r is the number of first replies received by the source node. In the developed simulation model, the number of firings of transition T5 in the Petri net scheme of a node module (source node) represents the N_s, and place S99 in the source node module represents N_r (see Figure 8). Accordingly, $0 \leq n_s \leq 1$, with the ideal (maximal) value of $n_s = 1$.

The formal definition of the second performance metric, the number of hops per early replies, is as follows. Let $L <= N$ be the number of nodes were delivered at least one packet each. Assume, without the loss of generality, that the server nodes have numbers 1, 2, …, L and the number of replies received from node i after h_{ij} hops be denoted by m_{ij}. Then, the average number of hops per early replies from each server node i is:

$$h_i = \frac{\sum_{j=1}^{k_i} m_{ij} h_{ij}}{\sum_{j=1}^{k_i} m_{ij}} \tag{3.5}$$

where k_i is the number of replies having the same hop counter at node i, $i = 1, 2, \ldots, L$. The proposed model computes these values for each node $i \in \{1, 2, \ldots, L\}$. The overall average number of hops per early received replies is:

$$h = \frac{\sum_{i=1}^{L} h_i m_i}{\sum_{i=1}^{L} m_i} \tag{3.6}$$

where

$$m_i = \sum_{j=1}^{k_j} m_{ij} \tag{3.7}$$

is the number of replies received from node i. The expression (3.6) is a general formula that is valid for multiple receivers. In the proposed model, where there is only one source node that calculates the number of hops, the histogram of place S99 of the source node module provides the average number of hops.

The third performance metric, the relative traffic, is estimated with the use of expression:

$$n_f = \frac{N_f}{N_s} \tag{3.8}$$

where N_f is the number of packets transmitted by all network nodes. Packets transmitted from the source node (N_s) and all other nodes are included in this number. Generally, $n_f \geq 1$, with ideal (minimal) value being equal to one, $n_f = 1$.

The number of firings of transition Y1000 in the scheme of the switching module of the model represents N_f (see Figure 7).

The average response time, the fourth performance metric, measured at the source node, is calculated using the expression:

$$R = \frac{1}{N_r} \sum_{i=1}^{N_r} R_i \tag{3.9}$$

where N_r is the number of earliest replies at the source node and R_i is the round trip time for reply i, i = 1, 2, …, N_r. In the model, a data attribute of place S15 is used to calculate the average response time per received reply.

The last performance metric, the duplicate ratio, is estimated with the expression:

$$n_r = \frac{N_d}{N_r} \tag{3.10}$$

where N_d is the number of duplicated (discarded) replies and N_r is the number of earliest replies received by the source node. N_d is represented with place S16 in the Petri net scheme of the source node module (see Figure 9).

3.7. Simulation Setup

Simulation experiments were organized and conducted according to the following setup. It is assumed that the network area is a rectangular (square) of 500 m x 500 m. Such an area is quite realistic for small and medium-sized ad hoc wireless networks. The network area is populated by *N*=50 nodes, having numbers 1, 2, 3,…, *N*. The first m nodes are anycast servers with numbers 1, 2, …, *m* < *N*. The number of anycast servers, m, is specified in the file of parameters. The nodes with numbers *m+1*, *m+2*, …, *N* are simple nodes. It is assumed that *m* < *N-m,* i.e. the number of anycast server nodes is less than the number of simple nodes. For the sake of simplicity, simple node *N* is the source node that generates anycast requests.

Initial positions of the nodes (simple and servers) are random and different in different simulation runs with the uniform probability distribution [30] in given area.

That is, the network area with its nodes can be approximated as a point Poisson field [30]. All nodes move from their initial positions according to the chosen mobility model.

In the multicast and broadcast (or area restricted broadcast) mode of transmission, control packets RTS, CTS and ACK are not transmitted [43]. This considerably reduces the traffic of the network.

Any anycast server can receive more than one multicast request, but only the first received multicast request will be accepted and responded. Any simple node can receive more than one unicast reply from a few servers, but only the first unicast reply will be accepted and forwarded (if not the source node).

As was mentioned earlier, only one network node was used as a source of anycast request messages. All other network nodes work as message routers or servers of anycast messages transmitted by the source node. Correspondingly, the source node discards all requests that can be transmitted by other nodes since these requests are copies of messages initiated by the source node. The source node assigns a unique number to each generated packet. Interval between transmissions of requests by the source node is set to be 500ms. For a small sized or medium sized ad hoc network, this interval is sufficiently large to complete all activities in a network related to a request transmitted by the source node before it transmits the next request. As a result, at any moment of simulation time, the model will handle, at different nodes, messages with the same identifier. This considerably simplifies the model and its study.

To obtain sufficiently stable statistical results of simulation, the total number of requests transmitted by the source node is set as $Ns = 2000$ messages. With this number of messages and inter-message interval of 500 ms, the simulation interval of each run is 2000 × 500 ms + 1000 ms = 1001000 ms, where 1000 ms is a small margin to provide the clearance of the model at the end of each simulation run [31].

Starting from a chosen random position, each network node (including the source node) moves in a random direction with a constant random speed in the given area. The random speed of a node is set according to uniform probability distribution in the range $(0, V_{max})$, where V_{max} is the maximal speed set as a network configuration parameter.

As it is explained in [42], the inter-node communication is considered as very reliable for nodes, which are very close to each other. For this reason, in the model, the distance to very close nodes is assumed to be a random variable which has a lower bound equal to zero and upper bound being uniformly distributed from 5 to 10 meters. Also when probability of changing direction is equal to zero, nodes change the direction of their movement randomly at the border of the network area was used in this mobility model.

One more parameter of the inter-node communication scheme is the interval in which the states of oriented inter-node communication links are checked. This interval is the same for all simulation experiments and was set at 2000ms. At the end of this interval, the state of each link can change.

The simulation experiments were conducted for maximal transmission distances 30m, 60m, 90m, 120m, 150m, 180m and 210m. Obviously, with these distances, the message transmitted or forwarded by a node can reach only a subset of network nodes in the given area. This is true for real ad hoc wireless networks.

It is also assumed that each network node, intermediate or destination one, can loose any message, transmitted by another node, with some probability l. In the simulation experiments, as parameters, link availability l was used as the message loss probability in the range of $0< l <1$. The value of time-to-leave (TTL) field in generated packets was fixed at seven or four in each request message.

In simulation, three series of experiments were conducted. In the first series, the chosen performance metrics were studied for maximal transmission distances in the range (30, 210) with V_{max} = 5 Km/h; and link availability l=0.5. As parameters, six values of changing direction probability p = 0.0, 0.3, 0.5, 0.7, 0.9 and 1.0 were used.

In the second series of simulation experiments, the dependence of performance metrics on transmission radius was investigated with the maximal node speeds 5 Km/h, 30 Km/h and 50 Km/h. It should be noted that, with the given value of V_{max}, different network nodes will move with different speeds in the range (0, V_{max}). For these series of experiments link availability l=0.7 and probability of changing direction p= 0.0 were used. In both series of experiments, the value of TTL in each request message was fixed at seven.

In the third series of conducted experiments, the effect of TTL value on the performance metrics was investigated. For this reason, a set of experiments were performed by setting TTL to 4 and 7 with link availability l = 0.05, 0.1, 0.3, 0.5 and 0.7 when V_{max} was set to 5 Km/h, probability of changing directions p=0.0 and the maximum transmission distance is varied in the range (30-210) meters. All the parameters and setup of simulation setup are shown in Table 2.

Table 2: Parameters of simulation setup

Network area	500 m x 500 m
Number of nodes	50
Total number of requests	2000
Interval between transmission of requests	500ms
TTL (Time-to-leave)	4 and 7
Link availability (l)	$0 < l < 1$
Maximal transmission distances, m	30 to 210
Maximal node speed (V_{max})	5 Km/h, 30 Km/h and 50 Km/h
Changing direction probability (p)	$0 \leq p \leq 1$

3.8. Results of Simulations

The results of the first series of simulations are presented in Tables 3 – 8, and in Figures 13 – 17. The results of the second series of simulation are shown in Tables 9 – 11, and Figures 18 – 22. Tables 12 – 21, and Figures 23 – 32 represent the results of the third series of experiments. Afterwards, Tables 33 – 37 represent the comparison for third series of experiments.

Table 3: Simulation results for link availability l=0.5, probability of changing direction p= 0.0 and maximal node speed V= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.049	1.354	22.590	0.000	1.011
	2	0.051	1.348	16.139	0.000	1.000
	3	0.056	1.445	14.943	0.000	1.000
	4	0.047	1.213	15.415	0.000	1.000
60	1	0.108	3.033	25.325	0.069	1.290
	2	0.286	3.347	22.021	0.000	1.244
	3	0.195	3.678	18.745	0.000	1.133
	4	0.053	3.193	32.725	0.047	1.575
90	1	0.299	7.486	33.064	0.184	1.614
	2	0.182	5.882	34.234	0.121	1.646
	3	0.140	4.454	35.724	0.148	1.609
	4	0.179	6.753	32.962	0.214	1.663
120	1	0.684	21.578	43.872	0.256	2.003
	2	0.654	26.312	30.423	0.244	1.581
	3	0.431	21.948	36.209	0.330	1.749
	4	0.409	28.667	43.192	0.343	1.987
150	1	0.569	41.380	38.630	0.772	1.916
	2	0.783	45.296	29.737	0.440	1.574
	3	0.674	45.961	35.433	0.617	1.787
	4	0.560	43.083	36.195	0.499	1.767
180	1	0.728	51.383	29.888	0.802	1.682
	2	0.728	51.383	29.888	0.802	1.682
	3	0.783	50.458	27.058	0.615	1.493
	4	0.714	50.806	27.837	0.617	1.563
210	1	0.823	52.530	21.837	0.828	1.410
	2	0.937	52.125	18.517	1.054	1.267
	3	0.919	52.063	17.249	1.278	1.245
	4	0.816	53.448	23.847	0.865	1.483

Table 4: Simulation results for link availability *l*=0.5, probability of changing direction *p*= 0.3 and maximal node speed *V* = 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.000	1.552	0.000	0.000	0.000
	2	0.000	1.384	0.000	0.000	0.000
	3	0.000	1.253	0.000	0.000	0.000
	4	0.000	1.076	0.000	0.000	0.000
60	1	0.352	7.378	43.209	0.118	1.936
	2	0.455	5.990	31.043	0.063	1.526
	3	0.008	2.043	20.410	0.000	0.000
	4	0.553	5.054	22.332	0.000	1.181
90	1	0.771	14.274	42.425	0.482	1.284
	2	0.371	18.288	36.772	0.390	1.208
	3	0.669	17.242	30.143	0.149	1.517
	4	0.411	16.364	42.425	0.469	1.122
120	1	0.942	25.255	41.232	1.109	1.112
	2	0.742	36.077	29.601	0.090	1.562
	3	0.402	31.040	54.998	0.136	2.357
	4	0.808	34.880	63.867	0.199	1.899
150	1	0.965	44.910	14.436	1.463	1.111
	2	0.633	46.040	25.440	0.206	1.376
	3	0.531	42.687	44.927	0.206	2.156
	4	0.986	47.546	11.863	0.075	1.059
180	1	0.970	50.242	11.953	1.596	1.056
	2	0.973	52.105	14.734	1.392	1.141
	3	0.500	52.388	48.948	0.127	2.391
	4	0.982	51.779	13.235	0.180	1.097
210	1	0.975	53.523	19.985	1.918	1.081
	2	0.940	52.233	27.356	1.029	1.301
	3	0.762	53.251	26.996	0.452	1.502
	4	0.801	53.150	28.226	0.583	1.582

Table 5: Simulation results for link availability l=0.5, probability of changing direction p= 0.5 and maximal node speed V= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.000	1.406	0.000	0.000	0.000
	2	0.000	1.016	6.818	0.000	0.000
	3	0.000	1.000	0.000	0.000	0.000
	4	0.000	1.010	0.000	0.000	0.000
60	1	0.342	6.941	41.634	0.066	1.927
	2	0.342	6.941	0.000	0.000	0.000
	3	0.000	3.790	0.000	0.000	0.000
	4	0.000	1.925	0.000	0.000	0.000
90	1	0.810	17.315	24.010	0.000	1.331
	2	0.362	13.291	46.657	0.080	2.066
	3	0.607	12.398	23.438	0.079	1.289
	4	0.019	4.722	0.000	0.000	3.316
120	1	0.904	29.194	18.509	0.940	1.193
	2	0.425	18.793	68.106	0.176	3.012
	3	0.822	27.369	25.337	0.467	1.395
	4	0.040	38.021	14.339	0.056	3.407
150	1	0.946	45.603	15.582	0.843	1.275
	2	0.909	49.817	38.385	0.616	1.148
	3	0.748	47.909	42.782	0.212	1.919
	4	0.632	50.716	20.621	0.382	2.141
180	1	0.995	50.213	11.177	0.688	1.041
	2	0.923	52.345	19.620	1.052	1.290
	3	0.855	52.553	22.901	0.669	1.355
	4	0.803	52.928	32.574	0.502	1.789
210	1	0.985	52.081	11.728	1.895	1.064
	2	0.940	51.872	18.405	1.164	1.307
	3	0.948	51.348	14.715	1.270	1.141
	4	0.936	51.896	14.378	1.280	1.113

Table 6: Simulation results for link availability *l*=0.5, probability of changing direction *p*= 0.7 and maximal node speed *V*= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.021	1.351	46.977	0.000	1.971
	2	0.000	1.000	0.000	0.000	0.000
	3	0.000	1.565	0.000	0.000	0.000
	4	0.000	1.000	0.000	0.000	0.000
60	1	0.381	7.625	44.730	0.072	0.000
	2	0.000	1.492	0.000	0.000	0.000
	3	0.000	7.087	23.397	0.000	1.276
	4	0.800	3.967	0.000	0.403	1.036
90	1	0.794	16.440	22.504	0.423	1.272
	2	0.212	10.675	60.982	0.021	2.576
	3	0.094	6.557	60.982	0.000	0.000
	4	0.372	19.566	68.376	0.103	2.736
120	1	0.931	32.879	65.850	0.095	2.747
	2	0.875	32.338	18.266	0.660	1.179
	3	0.702	27.373	25.341	0.268	1.348
	4	0.401	31.951	55.554	0.128	2.415
150	1	0.969	45.213	56.070	0.111	1.097
	2	0.442	42.909	55.888	0.141	2.460
	3	0.908	46.497	20.498	0.731	1.244
	4	0.886	52.131	23.711	0.771	1.399
180	1	0.991	50.045	81.262	1.862	1.055
	2	0.929	52.051	18.880	0.998	1.241
	3	0.954	51.924	17.025	1.035	1.180
	4	0.914	50.785	17.869	0.847	1.182
210	1	0.983	53.235	43.773	1.933	1.078
	2	0.908	52.846	22.664	0.869	1.397
	3	0.954	53.268	46.164	0.343	1.010
	4	0.906	52.047	19.297	0.883	1.252

Table 7: Simulation results for link availability l=0.5, probability of changing direction p= 0.9 and maximal node speed V= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.000	1.648	10.519	0.000	0.000
	2	0.000	1.000	0.000	0.000	0.000
	3	0.000	1.393	0.000	0.000	0.000
	4	0.000	1.451	0.000	0.000	0.000
60	1	0.000	3.524	0.000	0.000	0.000
	2	0.000	2.716	45.752	0.153	2.084
	3	0.010	2.498	76.324	0.000	0.000
	4	0.286	7.253	51.799	0.049	2.318
90	1	0.821	16.831	22.978	0.495	1.297
	2	0.143	3.182	57.126	0.000	2.279
	3	0.048	4.151	63.760	0.000	2.552
	4	0.017	4.467	79.871	0.000	3.118
120	1	0.197	27.659	19.154	0.062	1.220
	2	0.903	29.699	93.897	0.013	3.797
	3	0.857	18.830	75.190	0.762	3.005
	4	0.203	42.556	27.068	0.665	1.480
150	1	0.976	44.425	13.996	1.266	1.090
	2	0.823	50.252	17.448	0.655	1.143
	3	0.584	47.910	49.366	0.334	2.329
	4	0.564	43.664	49.282	0.365	2.358
180	1	0.975	50.661	28.638	0.188	1.499
	2	0.765	52.013	28.014	0.558	1.530
	3	0.973	51.985	13.539	1.405	1.110
	4	0.903	51.074	19.380	0.774	1.242
210	1	0.984	52.851	11.889	1.984	1.080
	2	0.917	52.209	20.519	0.912	1.338
	3	0.892	53.125	18.633	1.013	1.246
	4	0.978	51.899	14.147	1.593	1.139

Table 8: Simulation results for link availability *l*=0.5, probability of changing direction *p*= 1.0 and maximal node speed *V*= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.000	1.000	0.000	0.000	0.000
	2	0.007	1.530	45.345	0.000	2.000
	3	0.000	1.000	0.000	0.000	0.000
	4	0.000	1.000	0.000	0.000	0.000
60	1	0.343	4.916	46.185	0.102	2.051
	2	0.481	3.592	25.775	0.000	1.337
	3	0.092	2.147	23.848	0.000	1.269
	4	0.000	3.892	0.000	0.000	0.000
90	1	0.532	18.396	18.180	0.251	1.162
	2	0.021	11.112	28.339	0.011	1.432
	3	0.463	18.616	25.881	0.000	1.385
	4	0.669	11.121	27.635	0.002	1.389
120	1	0.879	28.041	21.310	0.889	1.323
	2	0.562	36.107	55.900	0.314	2.420
	3	0.605	35.841	47.167	0.393	2.245
	4	0.955	42.009	20.344	0.063	1.232
150	1	0.948	45.853	29.770	1.362	1.563
	2	0.943	47.411	19.281	0.966	1.259
	3	0.917	48.043	27.172	1.029	1.456
	4	0.755	51.379	18.622	0.281	1.236
180	1	0.979	50.328	12.905	1.723	1.085
	2	0.971	51.542	13.494	1.669	1.114
	3	0.860	52.885	20.662	0.815	1.291
	4	0.932	51.564	19.137	1.081	1.271
210	1	0.980	51.042	12.897	1.862	1.102
	2	0.988	52.669	11.228	1.916	1.043
	3	0.957	52.711	16.757	1.436	1.216
	4	0.966	50.981	13.483	1.580	1.101

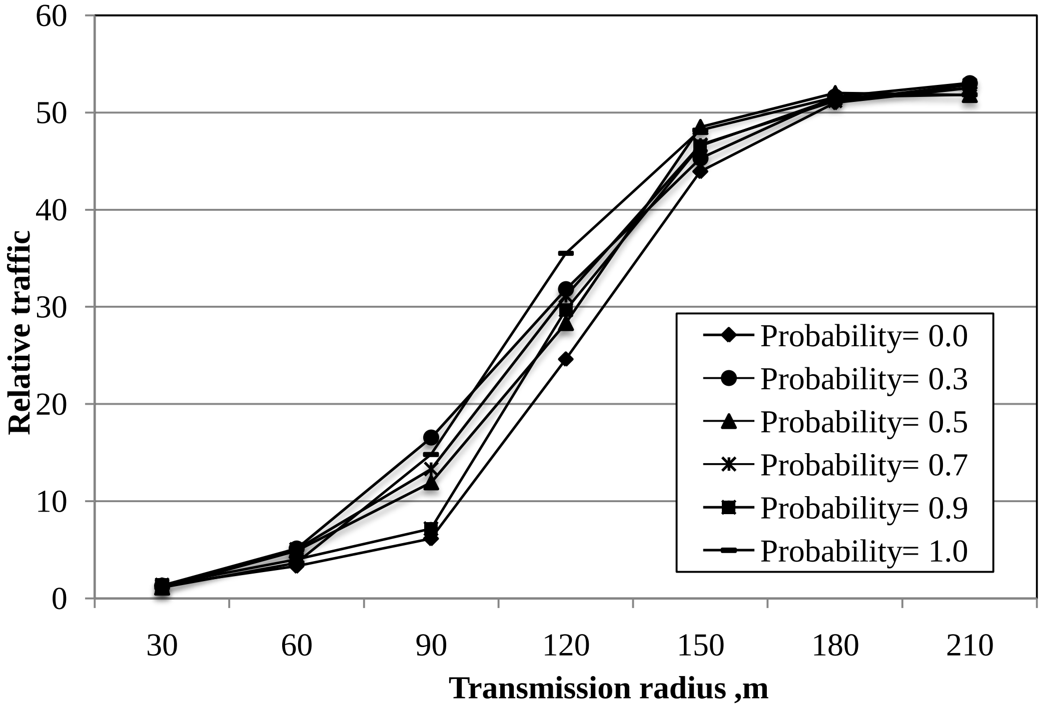

Figure 13: Relative traffic, n_f, versus transmission radius with link availability *l*=0.5 and maximal node speed *V*= 5 km/h.

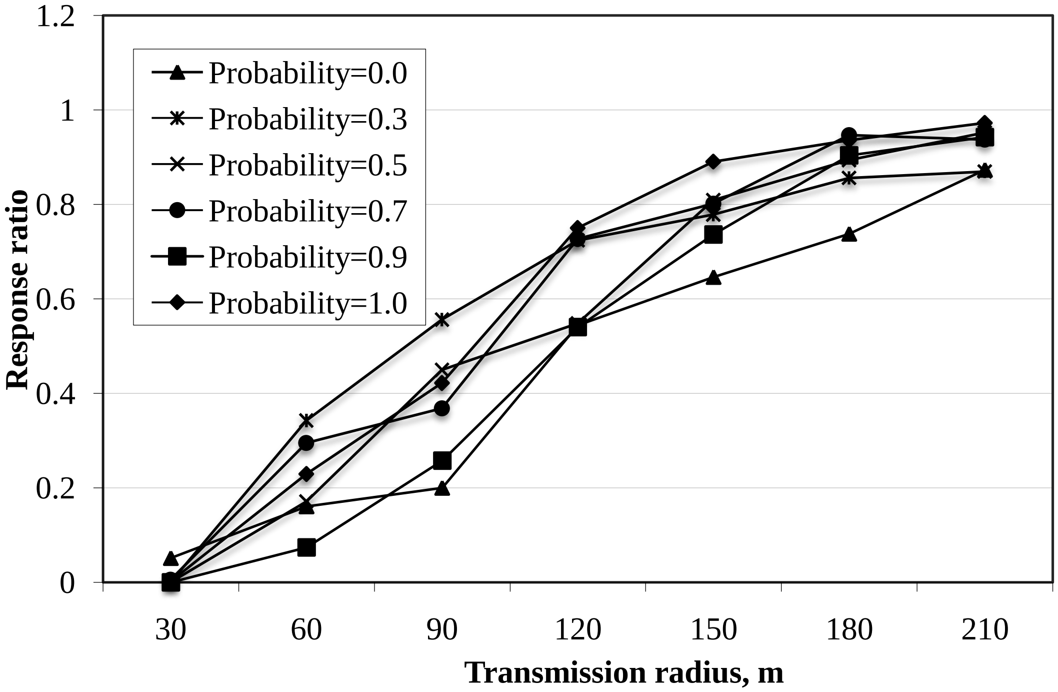

Figure 14: Response ratio, n_s, versus transmission radius with link availability *l*=0.5 and maximal node speed *V*= 5 km/h.

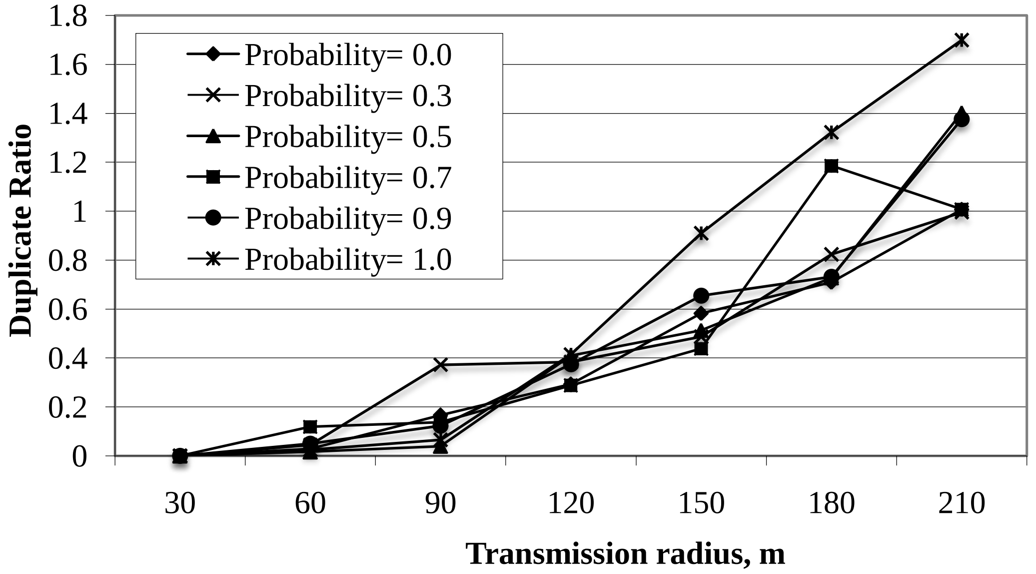

Figure 15: Duplicate ratio, N_r, versus transmission radius with link availability *l*=0.5 and maximal node speed *V*= 5 km/h.

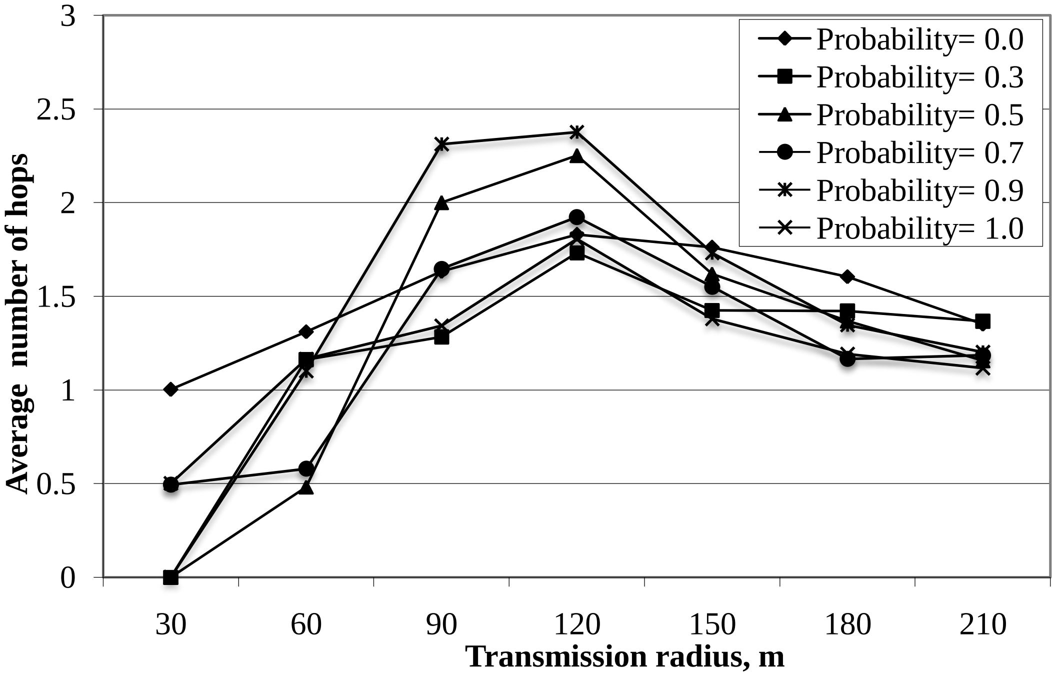

Figure 16: Average number of hops, *h*, versus transmission radius with link availability *l*=0.5 and maximal node speed *V*= 5 km/h.

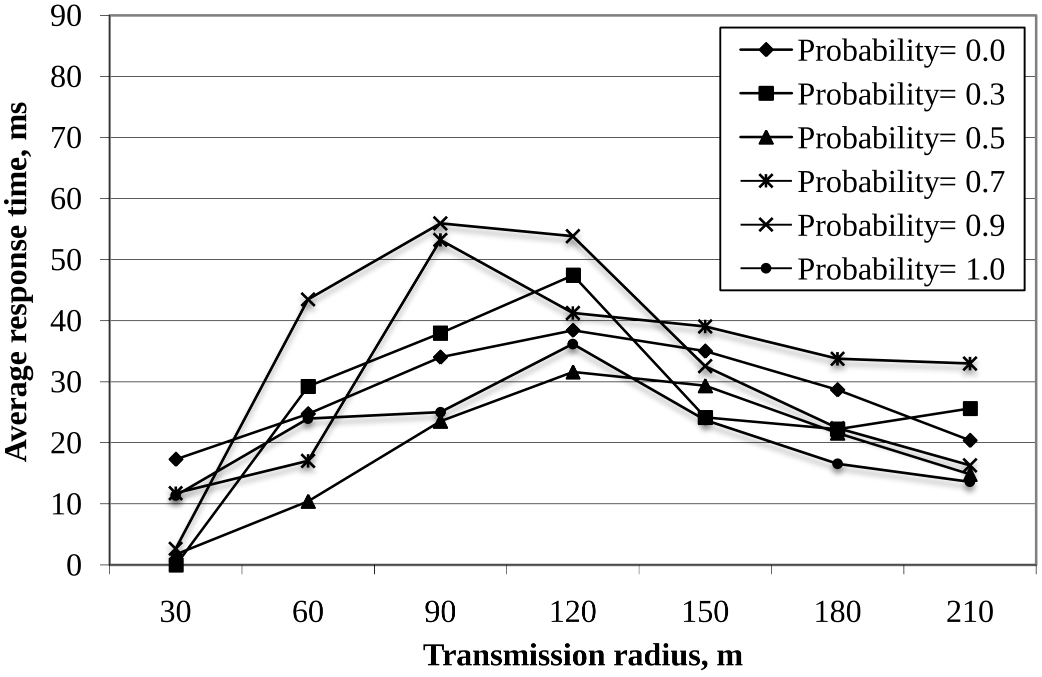

Figure 17: Average response time, *R*, versus transmission radius with link availability *l*=0.5 and maximal node speed *V*= 5 km/h.

Table 9: Simulation results for link availability *l*=0.7, probability of changing direction *p*= 0.0 and maximal node speed *V*= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.060	1.596	17.199	0.000	1.066
	2	0.000	1.612	0.000	0.000	0.000
	3	0.035	1.471	17.986	0.000	1.043
	4	0.051	1.357	15.943	0.000	1.020
60	1	0.183	3.539	25.623	0.084	1.360
	2	0.328	4.471	32.171	0.309	1.563
	3	0.351	4.921	25.619	0.172	1.343
	4	0.022	3.591	80.247	0.000	3.136
90	1	0.662	10.949	29.103	0.354	1.489
	2	0.349	11.991	60.507	0.100	2.521
	3	0.348	10.733	41.895	0.073	1.907
	4	0.689	20.022	41.226	0.187	1.914
120	1	0.820	29.117	28.166	0.647	1.506
	2	0.801	28.102	29.309	0.621	1.530
	3	0.856	35.362	29.610	0.831	1.621
	4	0.809	33.644	31.583	0.808	1.686
150	1	0.913	50.634	20.594	1.188	1.337
	2	0.978	50.180	15.302	1.659	1.157
	3	0.937	52.669	22.168	1.523	1.417
	4	0.861	51.333	26.156	1.321	1.574
180	1	0.958	53.313	17.950	1.595	1.270
	2	0.958	54.183	17.356	1.371	1.193
	3	0.961	53.561	20.060	1.269	1.334
	4	0.948	53.967	20.291	1.525	1.368
210	1	0.991	53.063	13.120	1.890	1.104
	2	0.890	53.783	23.690	1.250	1.582
	3	0.907	53.739	24.610	1.284	1.593
	4	0.876	54.452	28.954	1.079	1.807

Table 10: Simulation results for link availability *l*=0.7, probability of changing direction *p*= 0.0 and maximal node speed *V*= 30 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.043	1.465	14.792	0.000	1.011
	2	0.058	1.487	19.663	0.000	1.112
	3	0.018	1.484	22.031	0.000	1.194
	4	0.037	1.485	23.719	0.000	1.230
60	1	0.220	3.904	27.471	0.041	1.391
	2	0.144	3.615	28.215	0.132	1.413
	3	0.213	3.037	25.304	0.148	1.321
	4	0.119	3.266	32.007	0.017	1.577
90	1	0.568	13.835	34.088	0.265	1.665
	2	0.582	14.976	36.476	0.260	1.742
	3	0.568	13.716	34.109	0.256	1.649
	4	0.408	10.596	40.428	0.234	1.882
120	1	0.689	35.037	39.577	0.535	1.917
	2	0.730	34.140	36.519	0.439	1.785
	3	0.780	38.666	34.829	0.713	1.774
	4	0.687	32.957	41.699	0.404	1.987
150	1	0.919	49.843	27.820	0.955	1.579
	2	0.828	50.172	28.548	0.862	1.618
	3	0.850	48.525	29.968	0.939	1.635
	4	0.879	51.812	28.165	0.886	1.583
180	1	0.952	54.286	24.151	1.246	1.431
	2	0.911	53.911	22.573	1.125	1.468
	3	0.885	53.672	25.516	1.092	1.533
	4	0.854	53.559	27.649	1.119	1.648
210	1	0.944	53.754	19.472	1.565	1.390
	2	0.939	53.809	19.848	1.389	1.360
	3	0.914	54.033	22.440	1.210	1.468
	4	0.940	53.734	20.514	1.291	1.364

Table 11: Simulation results for link availability *l*=0.7, probability of changing direction *p*= 0.0 and maximal node speed *V*= 50 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.017	1.452	19.317	0.029	1.114
	2	0.031	1.357	18.624	0.000	1.127
	3	0.030	1.399	22.489	0.000	1.148
	4	0.025	1.364	18.580	0.040	1.140
60	1	0.168	3.240	33.344	0.104	1.582
	2	0.183	4.112	32.525	0.095	1.553
	3	0.225	3.827	24.808	0.082	1.298
	4	0.223	2.873	24.527	0.029	1.309
90	1	0.597	14.370	33.469	0.321	1.635
	2	0.507	12.666	39.442	0.141	1.805
	3	0.405	12.509	37.926	0.248	1.789
	4	0.369	11.040	42.519	0.195	1.943
120	1	0.749	33.158	32.726	0.678	1.684
	2	0.693	34.227	45.596	0.477	2.149
	3	0.712	34.176	40.406	0.508	1.950
	4	0.800	33.713	30.177	0.591	1.560
150	1	0.869	50.381	28.447	0.835	1.593
	2	0.913	50.607	26.482	1.068	1.567
	3	0.822	49.250	31.122	0.890	1.709
	4	0.855	49.620	30.020	0.805	1.643
180	1	0.916	53.417	22.378	1.164	1.443
	2	0.922	53.333	22.641	1.202	1.430
	3	0.931	53.656	23.270	1.230	1.472
	4	0.927	53.735	20.904	1.292	1.369
210	1	0.958	53.814	17.078	1.528	1.264
	2	0.940	53.828	18.875	1.428	1.344
	3	0.957	53.703	17.619	1.532	1.301
	4	0.928	53.616	18.918	1.496	1.356

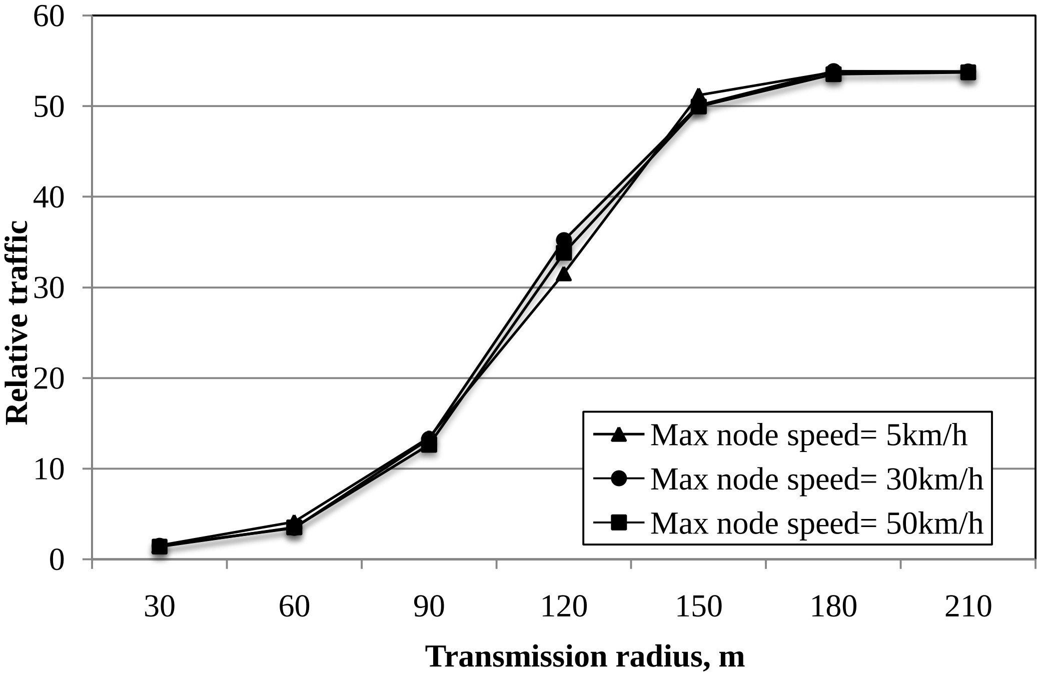

Figure 18: Relative traffic, n_f, versus transmission radius with link availability l=0.7 and different maximal node speed.

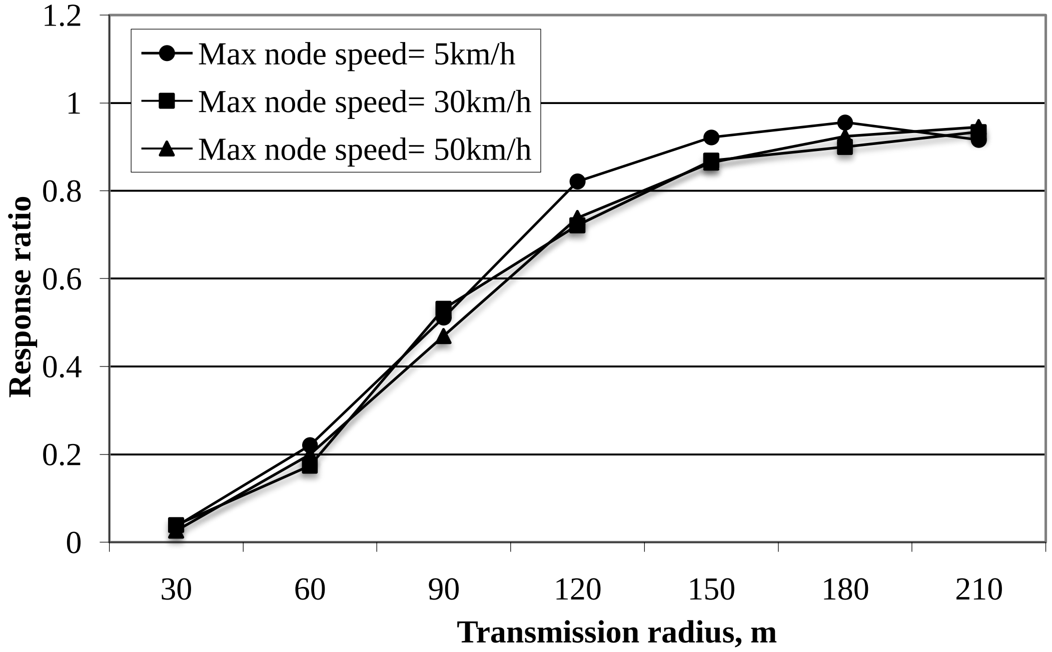

Figure 19: Response ratio, n_s, versus transmission radius with link availability l=0.7 and different maximal node speed.

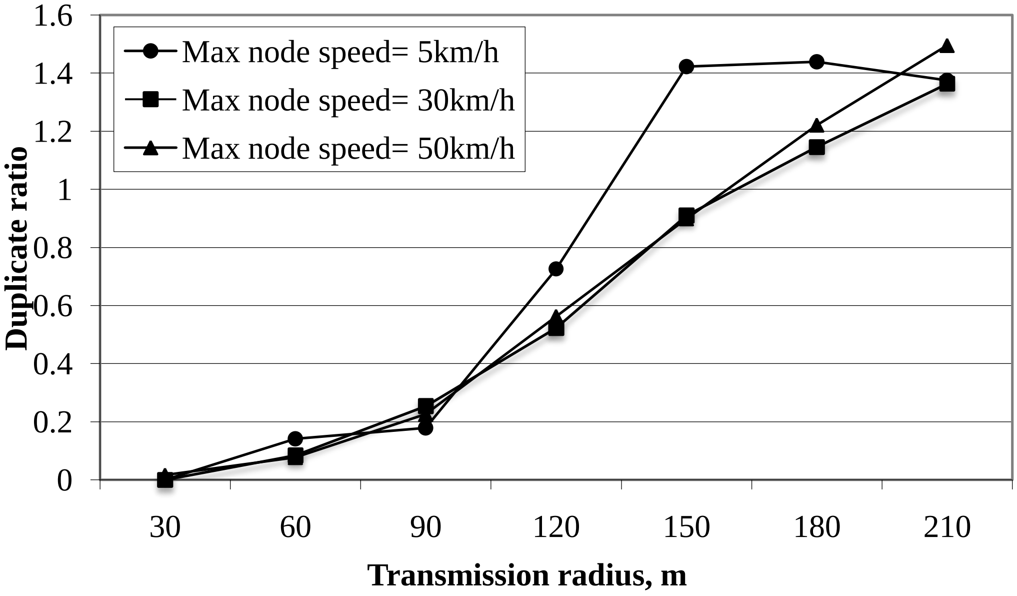

Figure 20: Duplicate ratio, N_r, versus transmission radius with link availability l=0.7 and different maximal node speed.

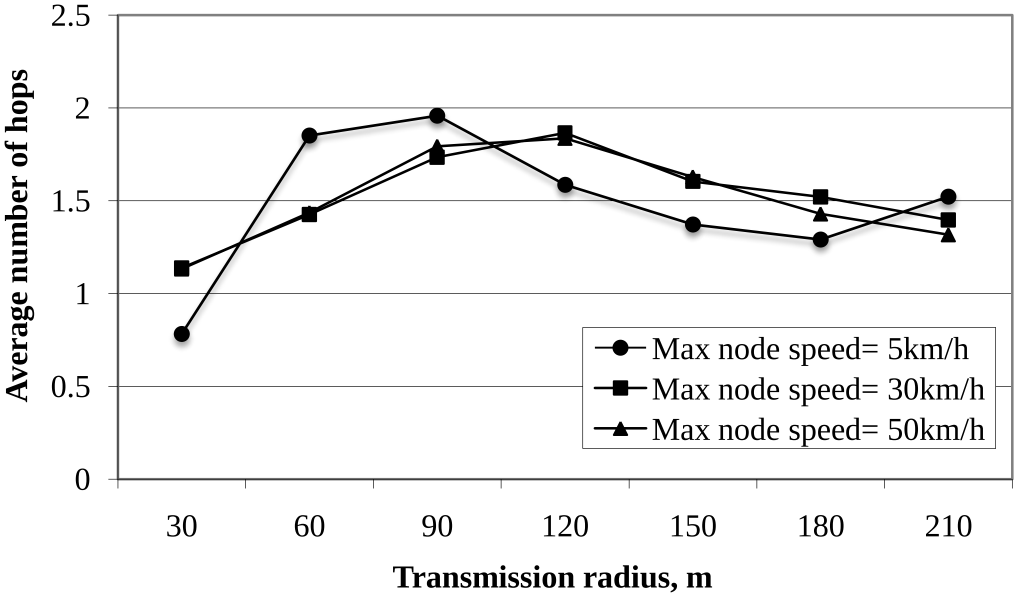

Figure 21: Average number of hops, h, versus transmission radius with link availability l=0.7 and different maximal node speed.

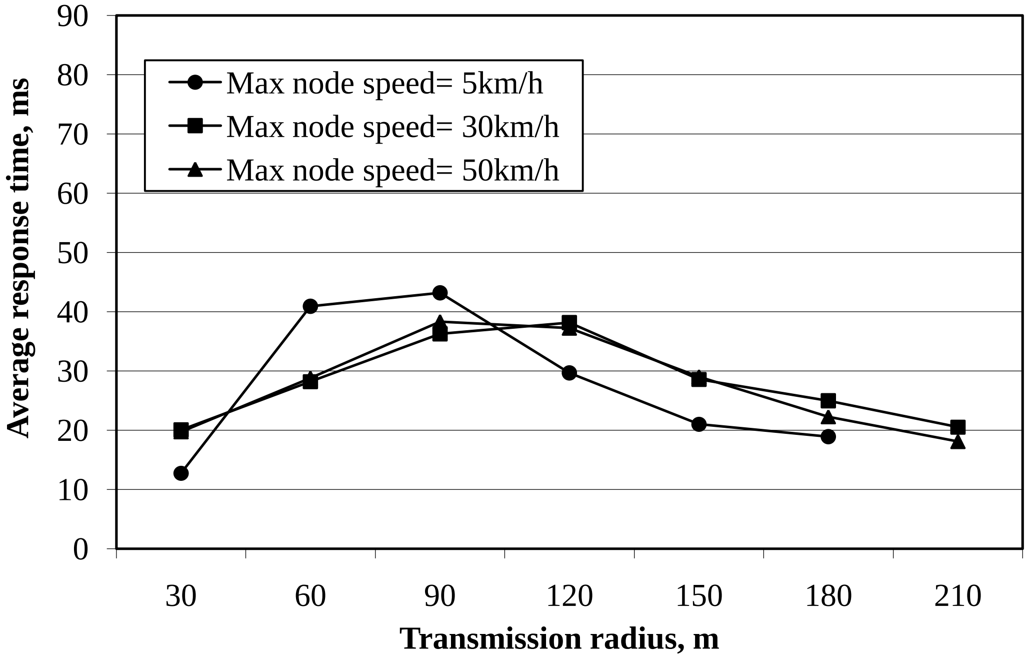

Figure 22: Average response time, R, versus transmission radius with link availability *l*=0.7 and different maximal node speed.

Table 12: Simulation results for link availability *l*=0.05, TTL=4 and maximal node speed *V*= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.047	1.081	14.709	0.000	1.000
	2	0.003	1.041	13.168	0.000	1.000
	3	0.004	1.036	17.203	0.000	1.000
	4	0.009	1.052	16.307	0.000	1.000
60	1	0.012	1.215	16.967	0.000	1.000
	2	0.009	1.125	16.019	0.889	1.000
	3	0.010	1.164	16.048	0.000	1.000
	4	0.031	1.168	15.596	0.000	1.000
90	1	0.043	1.482	21.278	0.070	1.140
	2	0.016	1.272	16.385	0.000	1.000
	3	0.060	1.346	18.774	0.000	1.124
	4	0.045	1.418	15.494	0.067	1.011
120	1	0.048	1.613	16.314	0.031	1.000
	2	0.058	1.952	14.370	0.000	1.017
	3	0.080	1.835	16.426	0.149	1.006
	4	0.000	1.431	9.969	0.000	1.000
150	1	0.058	2.063	13.523	0.308	1.000
	2	0.082	1.780	18.210	0.171	1.134
	3	0.053	1.990	15.860	0.000	1.000
	4	0.094	2.067	16.791	0.000	1.026
180	1	0.116	2.590	13.081	0.155	1.000
	2	0.129	2.854	18.405	0.043	1.089
	3	0.131	2.651	16.470	0.080	1.023
	4	0.125	2.343	17.276	0.068	1.040
210	1	0.115	2.720	14.970	0.229	1.004
	2	0.096	3.546	13.812	0.212	1.026
	3	0.136	3.419	14.955	0.260	1.018
	4	0.115	3.468	18.780	0.000	1.087

Table 13: Simulation results for link availability *l*=0.1, TTL=4 and maximal node speed *V*= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.000	1.129	17.791	0.000	1.000
	2	0.010	1.148	15.795	0.000	1.000
	3	0.003	1.132	14.508	0.000	1.000
	4	0.021	1.124	15.135	0.000	1.000
60	1	0.021	1.258	18.103	0.000	1.071
	2	0.013	1.190	17.233	0.000	1.000
	3	0.029	1.411	27.155	0.000	1.431
	4	0.038	1.182	23.406	0.000	1.211
90	1	0.070	1.500	19.090	0.114	1.143
	2	0.017	1.732	17.218	0.000	1.000
	3	0.026	2.009	19.323	0.000	1.096
	4	0.026	1.741	21.714	0.000	1.154
120	1	0.113	2.576	15.020	0.062	1.009
	2	0.161	3.186	16.893	0.000	1.047
	3	0.080	2.699	18.940	0.050	1.149
	4	0.064	2.135	18.098	0.008	1.094
150	1	0.182	4.126	16.190	0.223	1.069
	2	0.095	4.329	21.450	0.016	1.189
	3	0.124	3.783	19.914	0.000	1.165
	4	0.086	3.023	19.466	0.012	1.122
180	1	0.133	5.170	20.190	0.401	1.217
	2	0.067	5.173	28.578	0.000	1.444
	3	0.159	6.185	18.884	0.006	1.113
	4	0.157	4.098	19.606	0.102	1.169
210	1	0.299	6.529	16.159	0.191	1.050
	2	0.241	8.756	25.744	0.077	1.369
	3	0.205	9.104	16.074	0.119	1.054
	4	0.298	8.456	16.635	0.180	1.070

Table 14: Simulation results for link availability *l*=0.3, TTL=4 and maximal node speed *V*= 5 km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.017	1.223	15.072	0.000	1.000
	2	0.037	1.244	17.451	0.000	1.054
	3	0.008	1.151	13.693	0.000	1.000
	4	0.074	1.185	17.395	0.000	1.060
60	1	0.110	2.434	25.385	0.000	1.364
	2	0.092	2.474	19.561	0.000	1.157
	3	0.007	2.646	15.951	0.000	1.000
	4	0.029	2.540	22.803	0.000	1.138
90	1	0.240	4.133	25.106	0.044	1.316
	2	0.017	3.714	25.152	0.000	1.294
	3	0.126	3.780	25.035	0.095	1.313
	4	0.156	5.218	24.617	0.102	1.288
120	1	0.378	6.302	21.879	0.046	1.242
	2	0.194	9.721	36.564	0.123	1.710
	3	0.027	7.234	66.471	0.000	2.556
	4	0.250	6.017	28.577	0.062	1.460
150	1	0.537	11.160	18.925	0.330	1.159
	2	0.392	13.457	30.571	0.322	1.573
	3	0.639	17.865	23.272	0.435	1.322
	4	0.437	15.273	23.038	0.391	1.285
180	1	0.576	19.150	22.317	0.443	1.304
	2	0.314	14.323	35.883	0.290	1.739
	3	0.479	20.977	24.421	0.356	1.346
	4	0.505	21.781	29.284	0.256	1.488
210	1	0.729	30.673	20.419	0.694	1.268
	2	0.416	26.180	32.088	0.433	1.671
	3	0.496	25.948	32.276	0.239	1.633
	4	0.466	24.472	26.338	0.334	1.425

Table 15: Simulation results for link availability l=0.5, TTL=4 and maximal node speed V= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.049	1.353	22.590	0.000	1.202
	2	0.015	1.557	14.470	0.000	1.000
	3	0.083	1.425	15.504	0.204	1.072
	4	0.026	1.273	18.896	0.000	1.132
60	1	0.082	2.598	23.679	0.018	1.236
	2	0.083	2.288	19.130	0.000	1.060
	3	0.131	3.292	42.325	0.000	1.886
	4	0.095	2.648	16.602	0.042	1.021
90	1	0.306	7.222	30.438	0.168	1.524
	2	0.321	6.367	25.756	0.369	1.371
	3	0.223	3.291	22.449	0.068	1.213
	4	0.364	4.537	24.132	0.303	1.303
120	1	0.490	13.471	28.721	0.333	1.460
	2	0.459	16.255	42.970	0.335	2.017
	3	0.412	13.300	37.889	0.150	1.814
	4	0.490	14.008	34.231	0.155	1.655
150	1	0.662	24.006	29.016	0.768	1.588
	2	0.553	24.648	39.083	0.583	1.886
	3	0.835	27.968	21.681	0.678	1.290
	4	0.617	15.610	29.520	0.263	1.515
180	1	0.748	38.422	26.427	0.772	1.527
	2	0.706	39.944	30.524	0.626	1.736
	3	0.908	37.908	20.064	0.911	1.294
	4	0.692	34.087	29.376	0.605	1.574
210	1	0.792	43.223	21.188	1.055	1.389
	2	0.920	40.482	16.641	1.079	1.174
	3	0.923	37.386	17.668	0.982	1.211
	4	0.924	42.273	16.876	1.221	1.221

Table 16: Simulation results for link availability *l*=0.7, TTL=4 and maximal node speed *V*= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.116	1.542	17.419	0.017	1.082
	2	0.025	1.309	14.175	0.000	1.000
	3	0.019	1.196	15.443	0.000	1.000
	4	0.121	1.360	16.497	0.066	1.021
60	1	0.211	3.884	27.298	0.052	1.404
	2	0.082	3.288	46.300	0.000	1.988
	3	0.105	3.339	41.068	0.000	1.791
	4	0.102	3.359	36.135	0.000	1.688
90	1	0.510	9.604	34.346	0.353	1.691
	2	0.632	9.142	28.154	0.192	1.445
	3	0.733	7.445	22.277	0.222	1.265
	4	0.462	8.755	38.421	0.156	1.775
120	1	0.725	17.370	29.212	0.519	1.542
	2	0.702	23.978	32.186	0.758	1.663
	3	0.710	15.676	36.434	0.786	1.814
	4	0.548	18.238	36.569	0.418	1.801
150	1	0.857	30.554	26.494	1.109	1.575
	2	0.852	36.439	29.780	1.016	1.676
	3	0.895	29.588	26.685	0.844	1.522
	4	0.825	20.934	19.172	1.091	1.258
180	1	0.957	33.988	16.834	1.392	1.176
	2	0.961	35.557	14.527	1.781	1.115
	3	0.887	43.244	26.484	1.225	1.598
	4	0.967	42.327	17.899	1.704	1.290
210	1	0.962	44.952	17.567	1.482	1.266
	2	0.895	45.909	20.468	1.363	1.414
	3	0.968	44.338	14.922	1.665	1.190
	4	0.795	46.061	31.359	0.849	1.910

Table 17: Simulation results for link availability l=0.05, TTL=7 and maximal node speed V= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.047	1.081	14.641	0.000	1.000
	2	0.004	1.036	13.168	0.000	1.000
	3	0.009	1.052	16.307	0.000	1.000
	4	0.031	1.059	17.926	0.000	1.000
60	1	0.001	1.130	16.914	0.000	1.000
	2	0.009	1.125	16.019	0.000	1.000
	3	0.010	1.164	16.048	0.089	1.000
	4	0.031	1.168	15.596	0.000	1.000
90	1	0.048	1.475	15.605	0.000	1.000
	2	0.069	1.323	17.360	0.000	1.065
	3	0.042	1.431	16.323	0.094	1.000
	4	0.026	1.479	14.429	0.000	1.000
120	1	0.063	1.711	17.022	0.024	1.032
	2	0.051	1.553	14.570	0.000	1.000
	3	0.063	1.731	17.751	0.000	1.063
	4	0.038	1.533	15.801	0.000	1.038
150	1	0.057	2.067	15.238	0.032	1.000
	2	0.051	2.740	18.363	0.020	1.068
	3	0.007	2.127	19.544	0.000	1.200
	4	0.095	2.162	19.128	0.019	1.131
180	1	0.084	3.034	18.039	0.243	1.118
	2	0.039	3.265	22.561	0.038	1.102
	3	0.063	2.955	17.833	0.000	1.196
	4	0.082	2.562	18.336	0.098	1.109
210	1	0.120	4.056	17.273	0.166	1.083
	2	0.113	3.878	20.254	0.031	1.154
	3	0.154	3.917	16.271	0.026	1.035
	4	0.093	3.789	17.685	0.022	1.081

Table 18: Simulation results for link availability *l*=0.1, TTL=7 and maximal node speed *V*= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.002	1.092	17.791	0.000	1.000
	2	0.010	1.148	18.505	0.000	1.000
	3	0.002	1.067	14.899	0.000	1.000
	4	0.003	1.062	15.937	0.000	1.000
60	1	0.021	1.258	18.103	0.000	1.071
	2	0.013	1.190	17.233	0.094	1.000
	3	0.095	1.423	15.226	0.000	1.000
	4	0.042	1.202	14.699	0.000	1.000
90	1	0.073	1.976	19.994	0.027	1.137
	2	0.036	1.862	16.312	0.000	1.069
	3	0.016	1.529	19.525	0.000	1.125
	4	0.081	2.300	22.182	0.025	1.235
120	1	0.070	2.527	17.334	0.170	1.064
	2	0.116	3.056	17.441	0.039	1.082
	3	0.102	2.161	21.225	0.000	1.190
	4	0.102	2.161	17.660	0.039	1.190
150	1	0.204	4.531	15.441	0.166	1.029
	2	0.173	4.812	17.173	0.000	1.075
	3	0.054	2.979	17.192	0.009	1.083
	4	0.027	4.048	25.522	0.000	1.222
180	1	0.133	6.377	15.782	0.271	1.079
	2	0.081	6.057	21.401	0.009	1.173
	3	0.160	9.529	24.654	0.000	1.312
	4	0.039	5.162	19.000	0.253	1.139
210	1	0.275	10.116	19.077	0.167	1.176
	2	0.172	9.868	22.019	0.340	1.227
	3	0.406	10.791	14.909	0.160	1.034
	4	0.325	11.741	16.685	0.061	1.048

Table 19: Simulation results for link availability l=0.3, TTL=7 and maximal node speed V= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.017	1.223	15.072	0.000	1.000
	2	0.037	1.244	17.451	0.000	1.054
	3	0.008	1.151	13.693	0.000	1.000
	4	0.074	1.185	17.395	0.000	1.060
60	1	0.076	2.141	19.863	0.000	1.118
	2	0.064	2.145	25.423	0.094	1.351
	3	0.025	1.534	18.328	0.000	1.040
	4	0.040	1.630	17.879	0.000	1.075
90	1	0.260	4.200	20.845	0.137	1.194
	2	0.112	5.796	28.220	0.000	1.424
	3	0.230	3.322	20.686	0.089	1.163
	4	0.253	4.246	22.607	0.215	1.214
120	1	0.353	11.141	23.734	0.123	1.289
	2	0.215	14.093	35.991	0.079	1.658
	3	0.340	14.313	25.807	0.048	1.368
	4	0.342	16.963	25.243	0.094	1.349
150	1	0.378	24.579	27.692	0.421	1.441
	2	0.442	24.890	27.597	0.150	1.406
	3	0.235	20.905	31.948	0.051	1.594
	4	0.291	25.103	39.998	0.098	1.926
180	1	0.557	37.179	23.369	0.380	1.321
	2	0.712	38.285	20.562	0.599	1.262
	3	0.557	37.258	19.640	0.632	1.194
	4	0.746	38.363	24.519	0.371	1.351
210	1	0.779	45.229	18.943	0.687	1.215
	2	0.640	45.672	25.816	0.411	1.440
	3	0.619	42.564	23.969	0.514	1.382
	4	0.629	44.763	23.622	0.469	1.356

Table 20: Simulation results for link availability l=0.5, TTL=7 and maximal node speed V= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.049	1.353	22.589	0.000	1.202
	2	0.015	1.544	14.469	0.000	1.000
	3	0.045	1.247	15.966	0.000	1.067
	4	0.011	1.243	16.961	0.000	1.000
60	1	0.108	3.035	25.325	0.069	1.290
	2	0.110	2.194	21.627	0.005	1.200
	3	0.249	4.315	18.055	0.297	1.134
	4	0.152	2.449	16.763	0.039	1.036
90	1	0.299	10.449	33.064	0.184	1.614
	2	0.239	8.848	35.621	0.161	1.678
	3	0.390	9.453	28.020	0.008	1.425
	4	0.153	8.878	51.947	0.046	2.293
120	1	0.684	21.598	25.959	0.560	1.404
	2	0.563	21.463	40.102	0.170	1.869
	3	0.428	25.701	40.975	0.079	1.875
	4	0.583	22.309	34.820	0.310	1.716
150	1	0.870	41.396	21.791	0.772	1.301
	2	0.768	40.830	27.770	0.561	1.483
	3	0.493	31.181	30.176	0.661	1.570
	4	0.680	41.839	31.840	0.545	1.641
180	1	0.850	50.830	24.664	0.616	1.423
	2	0.578	51.525	39.752	0.357	2.040
	3	0.800	49.571	22.720	0.668	1.378
	4	0.818	48.631	24.743	0.597	1.417
210	1	0.824	52.556	21.837	0.828	1.410
	2	0.957	51.916	15.077	1.579	1.168
	3	0.884	52.036	19.345	1.064	1.313
	4	0.860	51.480	20.100	0.752	1.299

Table 21: Simulation results for link availability *l*=0.7, TTL=7 and maximal node speed *V*= 5km/h.

Maximum distance of transmission of a node, m	Simulation run	n_s	n_f	R	N_r	h
30	1	0.060	1.596	17.199	0.000	1.066
	2	0.000	1.612	0.000	0.000	0.000
	3	0.035	1.471	17.986	0.000	1.043
	4	0.051	1.357	15.943	0.000	1.020
60	1	0.183	3.539	25.623	0.084	1.360
	2	0.328	4.471	32.171	0.309	1.563
	3	0.351	4.921	25.619	0.172	1.343
	4	0.022	3.591	80.247	0.000	3.136
90	1	0.662	10.949	29.103	0.354	1.489
	2	0.349	11.991	60.507	0.100	2.521
	3	0.348	10.733	41.895	0.073	1.907
	4	0.689	20.022	41.226	0.187	1.914
120	1	0.820	29.117	28.166	0.647	1.506
	2	0.801	28.102	29.309	0.621	1.530
	3	0.856	35.362	29.610	0.831	1.621
	4	0.809	33.644	31.583	0.808	1.686
150	1	0.913	50.634	20.594	1.188	1.337
	2	0.978	50.180	15.302	1.659	1.157
	3	0.937	52.669	22.168	1.523	1.417
	4	0.861	51.333	26.156	1.321	1.574
180	1	0.958	53.313	17.950	1.595	1.270
	2	0.958	54.183	17.356	1.371	1.193
	3	0.961	53.561	20.060	1.269	1.334
	4	0.948	53.967	20.291	1.525	1.368
210	1	0.991	53.063	13.120	1.890	1.104
	2	0.890	53.783	23.690	1.250	1.582
	3	0.907	53.739	24.610	1.284	1.593
	4	0.876	54.452	28.954	1.079	1.807

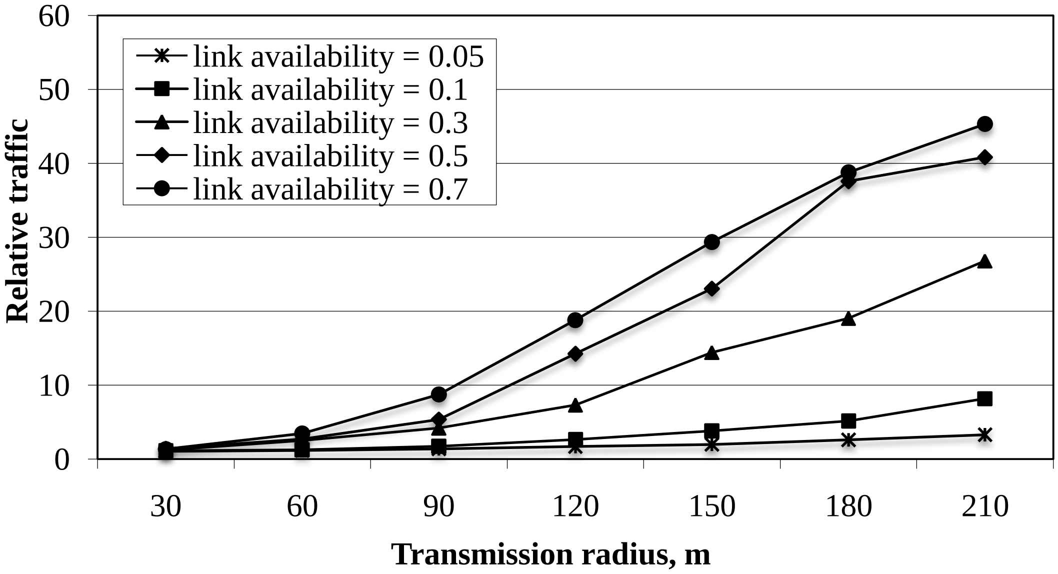

Figure 23: Relative traffic, n_f, versus transmission radius with different link availability for TTL=4.

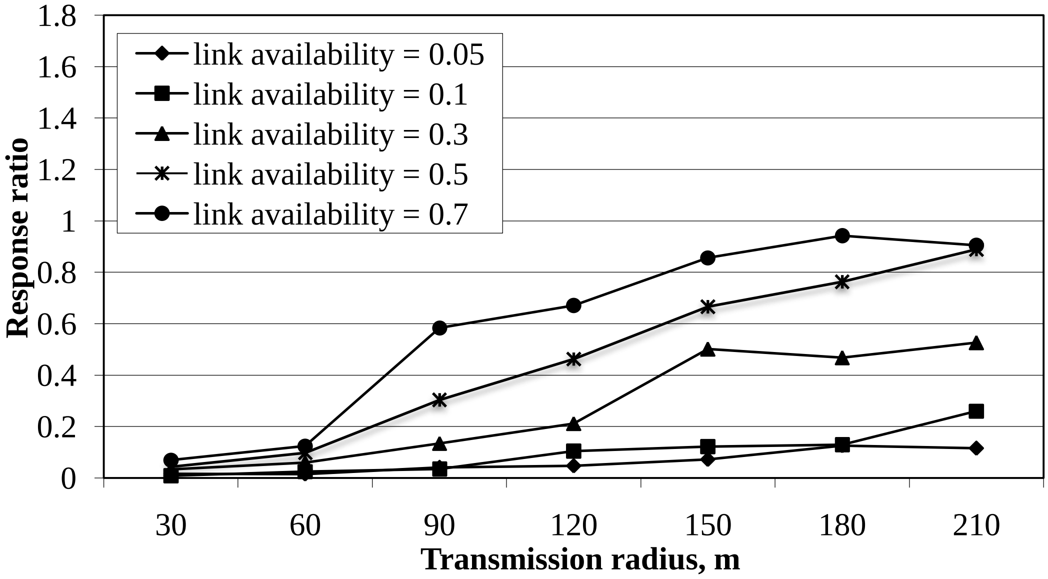

Figure 24: Response ratio, n_s, versus transmission radius with different link availability for TTL=4.

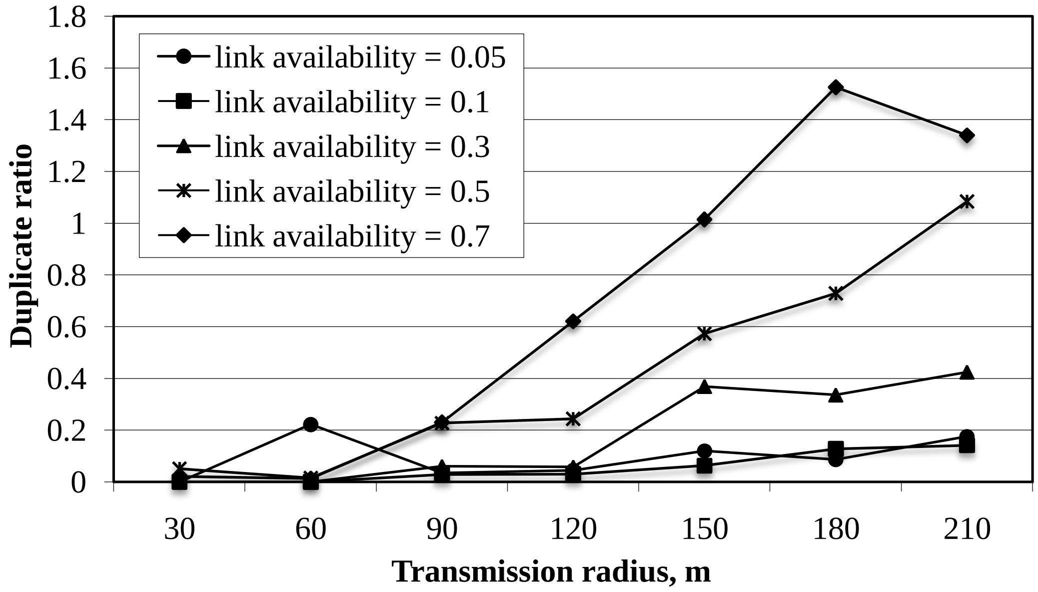

Figure 25: Duplicate ratio, N_r, versus transmission radius with different link availability for TTL=4.

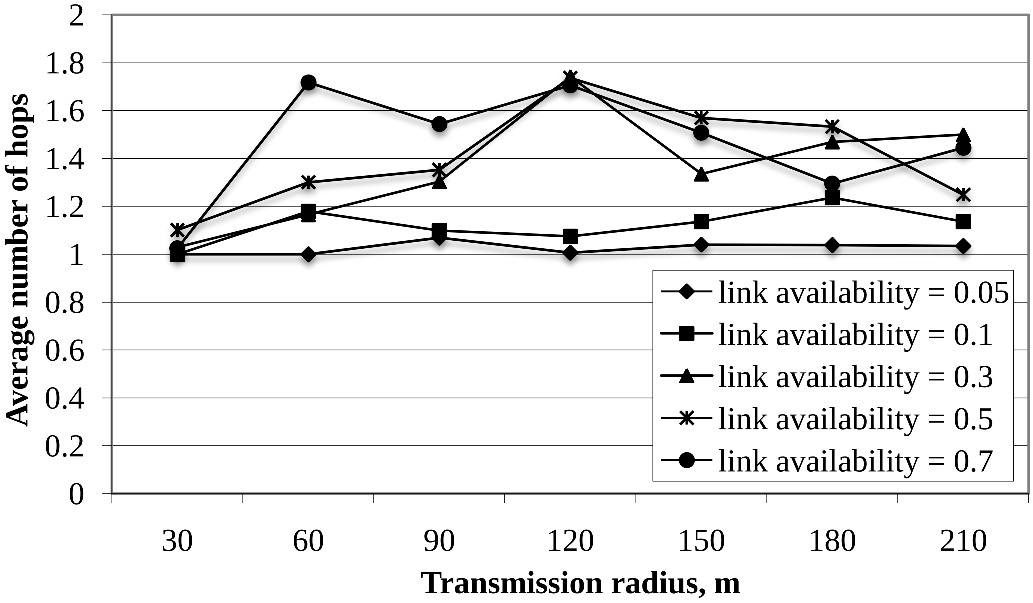

Figure 26: Average number of hops, h, versus transmission radius with different link availability for TTL=4.

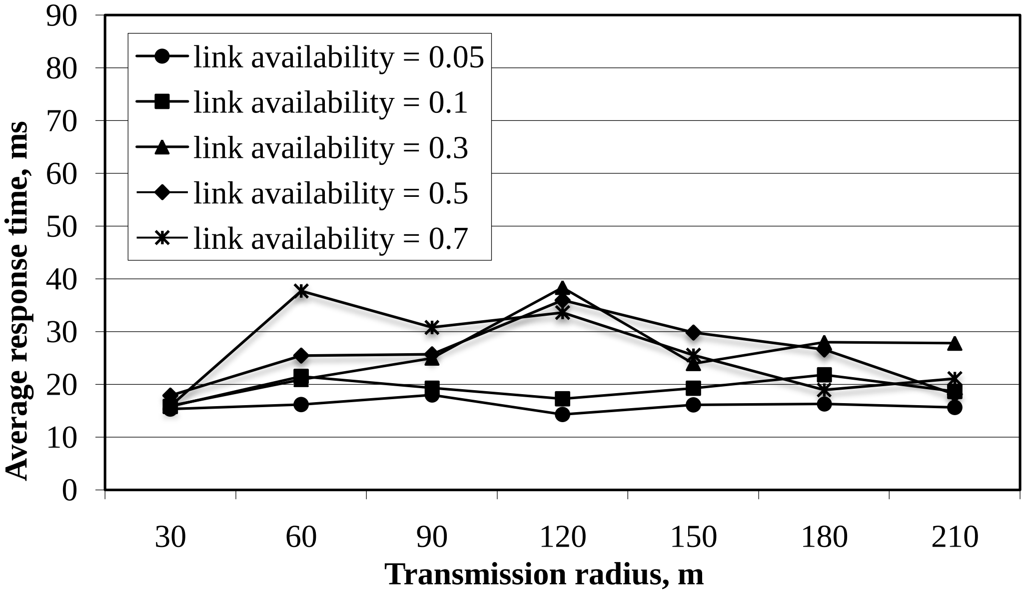

Figure 27: Average response time, R, versus transmission radius with different link availability for TTL=4.

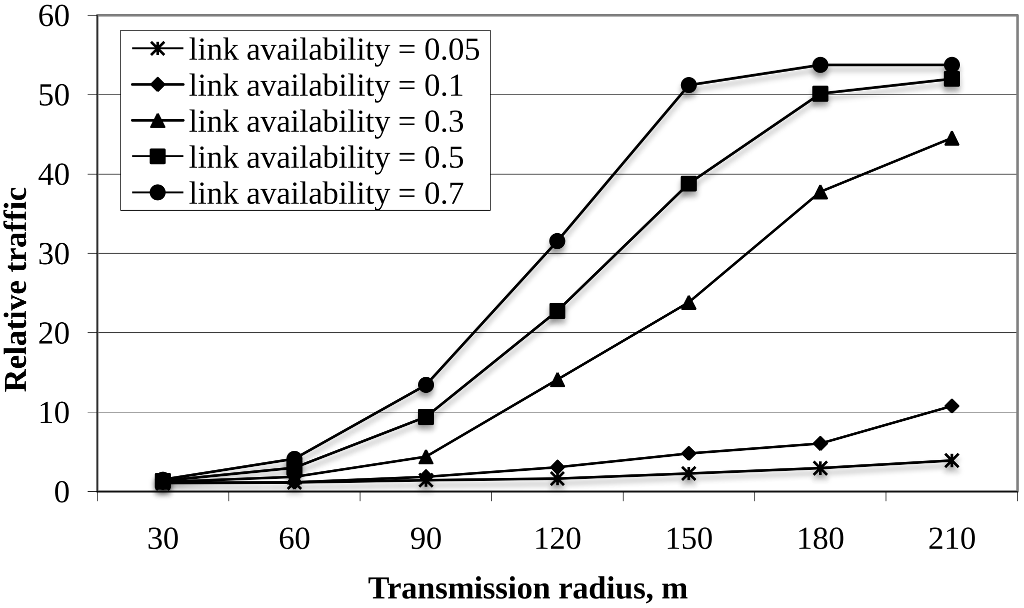

Figure 28: Relative traffic, n_f, versus transmission radius with different link availability for TTL=7.

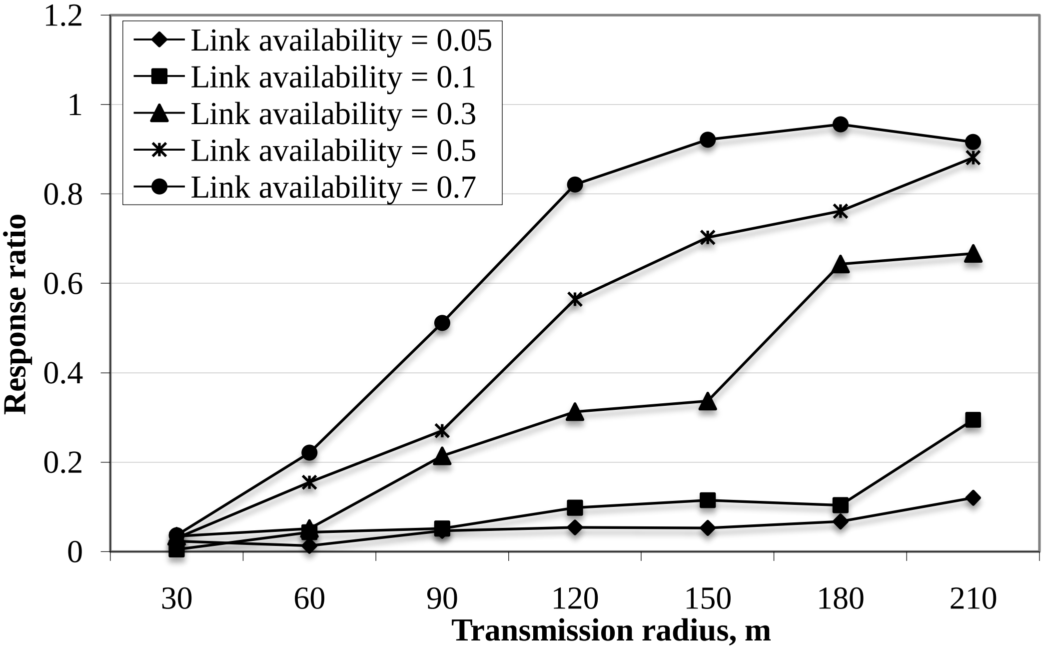

Figure 29: Response ratio, n_s, versus transmission radius with different link availability for TTL=7.

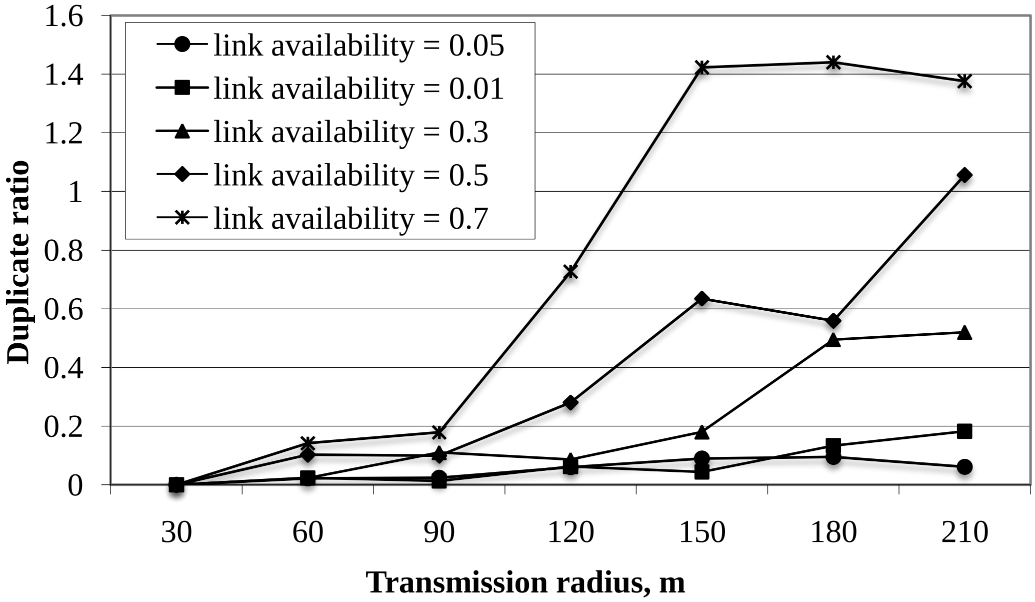

Figure 30: Duplicate ratio, N_r, versus transmission radius with different link availability for TTL=7.

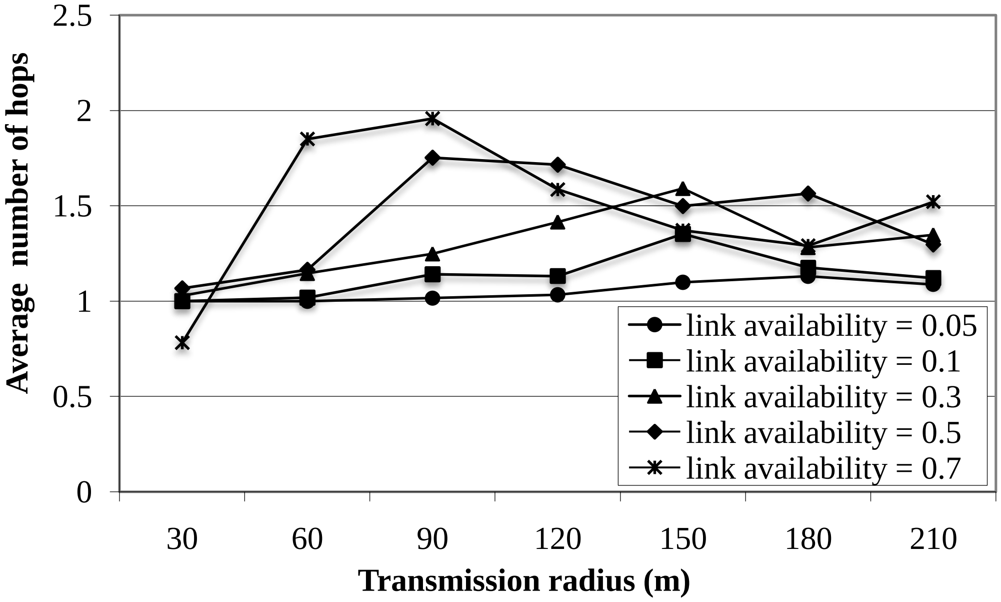

Figure 31: Average number of hops, *h*, versus transmission radius with different link availability for TTL=7.

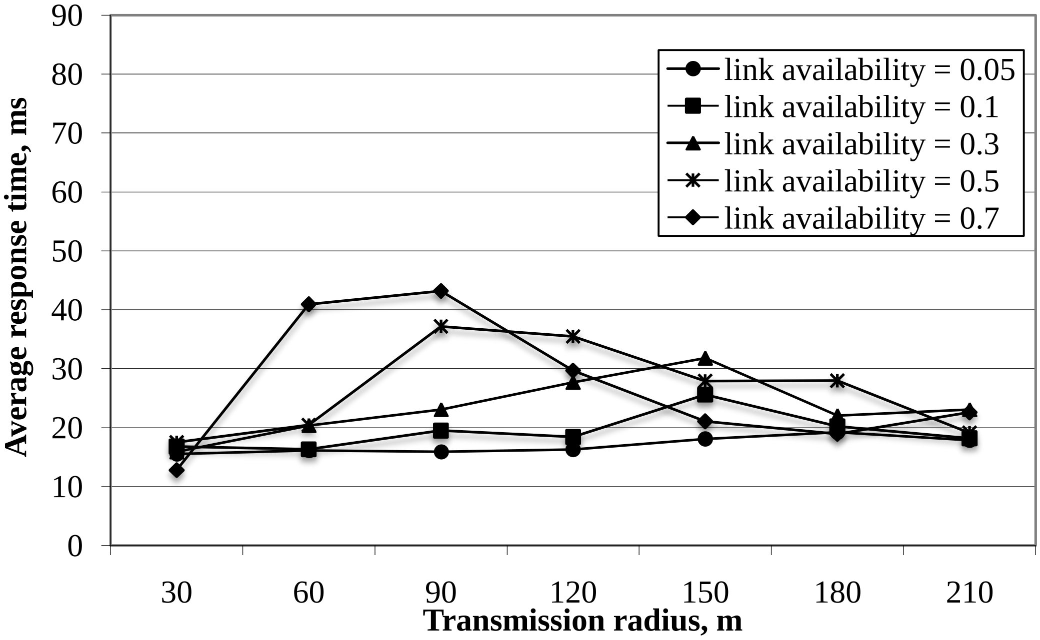

Figure 32: Average response time, *R*, versus transmission radius with different link availability for TTL=7.

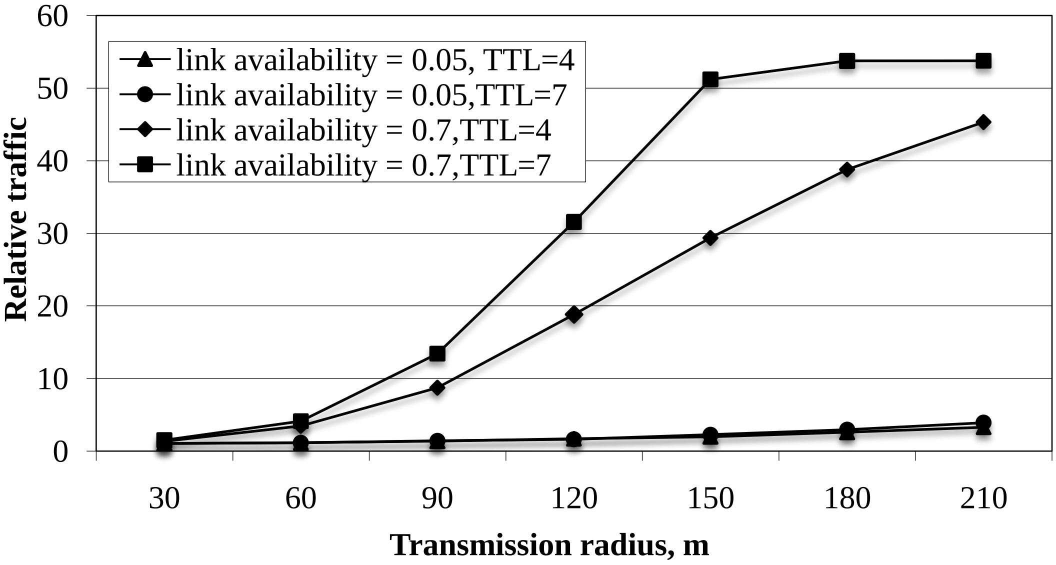

Figure 33: Relative traffic, n_f, versus transmission radius with different link availability for TTL=7, and 4.

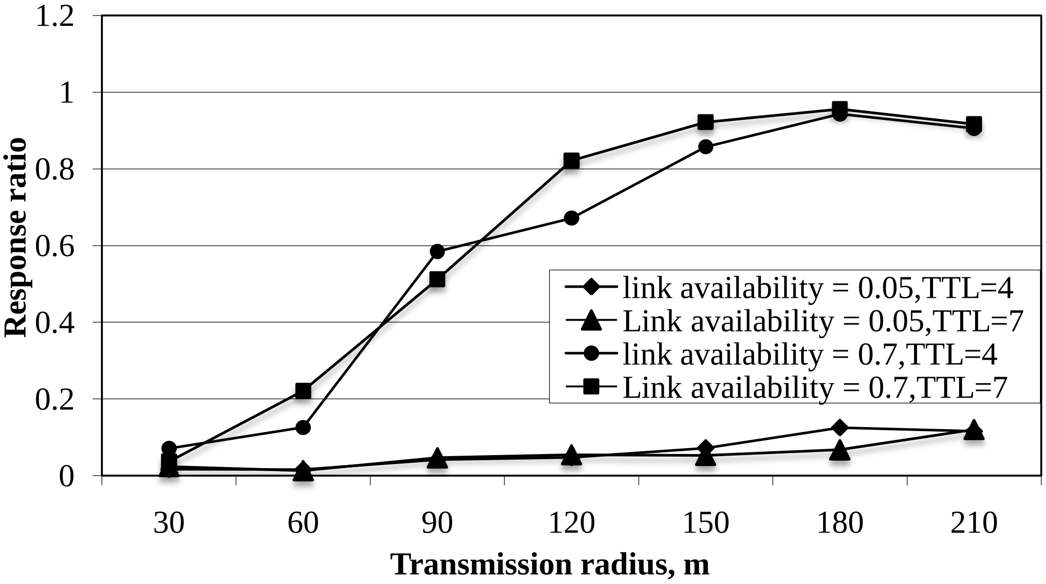

Figure 34: Response ratio, n_s, versus transmission radius with different link availability for TTL=7 and 4.

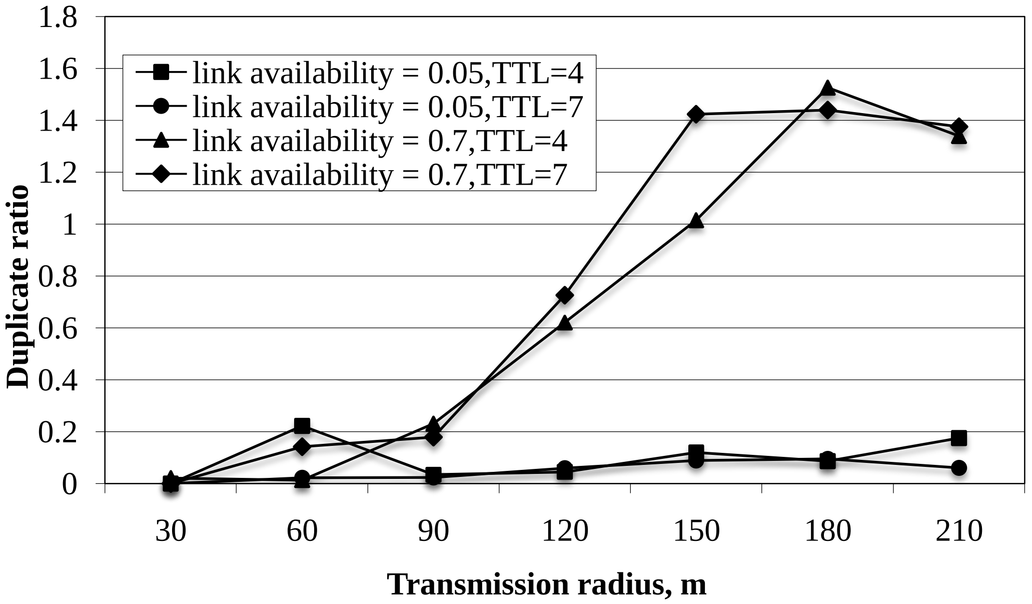

Figure 35: Duplicate ratio, N_r,, versus transmission radius with different link availability for TTL=7 and 4.

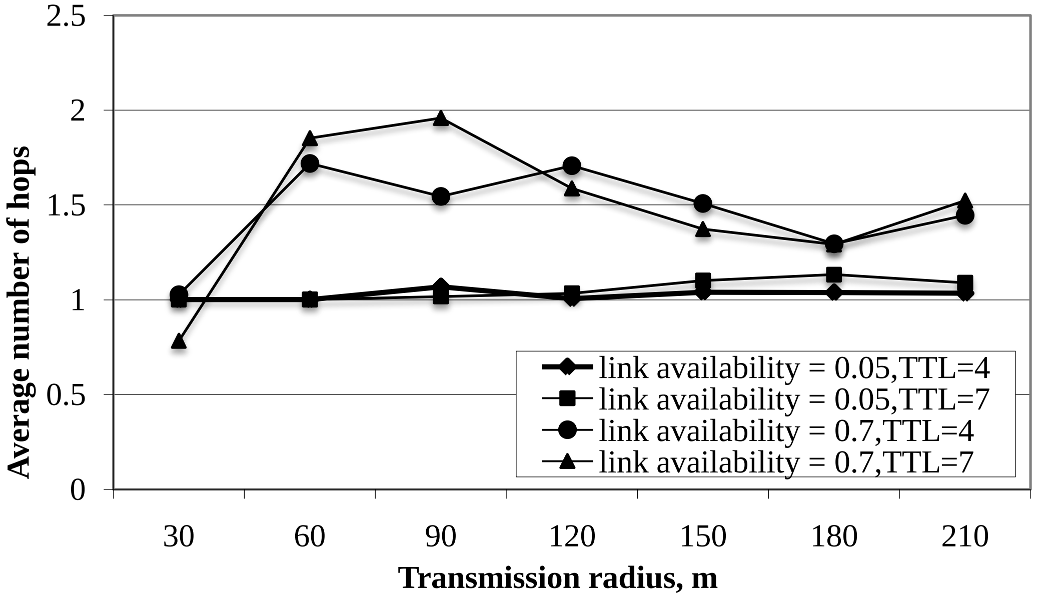

Figure 36: Average number of hops, h, versus transmission radius with different link availability for TTL=7 and 4.

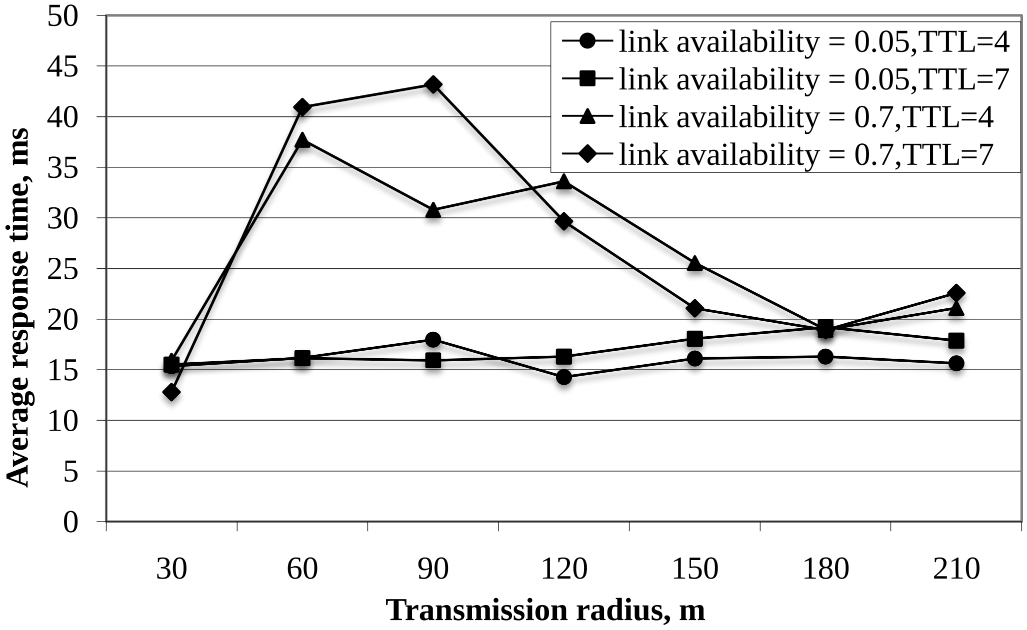

Figure 37: Average response time, *R*, versus transmission radius with different link availability for TTL=7 and 4.

3.9. Discussion of the Simulation Results

The following comments and observations can be made using the simulation results:

1. All the performance metrics are dependent on the transmission radius, but the character of this dependence is different for different performance metrics.
2. As Figures 14, 19, 24 and 29 demonstrate, the response ratio is quite low for small values of transmission radius, but it approaches the highest value of 1 at the transmission radius of 210m. However, for small link availability *l*=0.05, the response ratio remains quite low even at transmission radius of 210m, since a large number of packets are lost on the path from the source node to server and back.
3. At a small transmission radius of 30 m, the response ratio is low even for high value of link availability l = 0.7. The reason is that, with N = 50 nodes in the network, there is a high probability that each transmitting or forwarding node has

no neighbors within this transmission radius. This means that a packet transmitted by a node in network area has a very low chance to be received by at least one other node in this area.

4. Response ratio has direct relation with link availability. Increasing the link availability results in increment of response ratio, and it becomes more obvious in higher transmission radiuses (Figures 24 and 29). Also, as you can understand from Figures 14, 19, and 34, different probabilities of changing direction, node speeds, and TTL have no significant effect on this performance metric.
5. As Figures 16, 21, 26 and 31 show, the average number of hops is quite low at a small transmission radius. It initially increases with the increase of the transmission radius, reaching some maximum and then decreases. Such a behavior of this metric can be explained in the following way. When the transmission radius is small, then, as was explained earlier, many transmitted or forwarded packets will be received mainly by a close neighbor. It means, the packet can reach the destinations if only the destination is a close neighbor of source, with a low number of hops. On the other hand, with a very large transmission radius, many nodes will find their destination node in the coverage area, so packet can be transmitted with only one transmission. This reduces the average number of hops again.
6. The average number of hops metric was usually varying in the range (1-2) for different link availabilities and node speeds, except for different probability of changing direction which exceeds this range (Figure 16). Figures 26 and 31 indicate that for small link availabilities (l=0.05 and 0.1) it shows small changes and always stays close to 1 even with increasing the transmission radius. This performance metric is the same for different TTL with small link availability as

shown in Figure 36, but it reaches higher average number of hops for TTL=7. As packets with smaller TTL will be discarded in their way to destination, it results in a minor increase in this performance metric.

7. The third metric, the relative traffic, can be quite high for a large value of link availability (Figures 23, and 28), especially at large transmission radius, when more and more nodes are involved in the retransmission of packets (Figures 13 and 18). With variable values of TTL, the number of nodes involved in packet transmission is reduced. As shown in Figure 33, a value of TTL=4 has a small impact on the performance of the pure flooding scheme.
8. As Figures 17, 22, 27 and 32 show, the average response time is quite low at a small transmission radius. It initially increases with the increase of the transmission radius, reaches some maximum and then decreases. As explained before, when the transmission radius is small, less numbers of nodes are involved in the transmission. On the other hand, with a large transmission radius, many transmitted packets will find their destination node in the area with only one transmission. This reduces response time of the packets. Plus, maximum average response time is larger for packets with bigger TTL as you can see in Figure 37.
9. Figures 15, 20, 25, and 30 show that the duplicate ratio, the last metric, is quite low for a small transmission radius, but can be high for a large value of link availability (Figures 25, and 30). At the transmission radius of 210m as it approaches the highest value, since more than one server can be in the range of the transmitted packets and contribute to duplicate replies.
10. As indicated in Figure 35, in a network with all its nodes having the same link availability, changing the packet’s TTL doesn’t have a visible effect on the duplicate ratio.

11. As graphs in Figures 18-22 demonstrate, change of the maximum possible node speed in the range from 5 Km/h to the medium speed of a car in a city of 50 Km/h does not result in considerable change of all performance metrics.
12. As Figures 13-18 indicate, performance metrics does not have a noticeable dependence on the pattern of motion.

Chapter 4

EXPERIMENTAL STUDIES IN WIRELESS AD HOC NETWORKS

4.1. Application-layer Program

This section describes the structure of the application-layer program used in the experimental investigation of wireless ad hoc networks. The program was developed based on the prototype program [27] and presented in [29]. In the program [44], one of the most basic and fundamental routing algorithms - so called the pure flooding - is used for data dissemination in the wireless network [40]. This scheme uses hop-by-hop broadcasting and delivers each transmitted packet to possibly every node in the network many times. Pure flooding is a topology-independent and stateless mechanism which offers high reliability and minimal state information maintenance.

In the implementation of the pure flooding mechanism, area-restricted multicast mode of transmission, which represents a limited broadcast form, is used to send each packet to the destination node. The socket mechanism with the UDP transport protocol was used to multicast packets. IP and CSMA/CA protocols were considered at the network layer and MAC layer, respectively. The MAC layer performs the collusion detection by expecting the reception of an acknowledgment to any transmitted frame except multicast frames [32].

4.2. Organization of Experiments

The laptop computers used in the experiments as a network node have similar specifications with Intel core 2 duo 2.2 GHz processor and they are joining to the network by 802.11 b/g Wi-Fi wireless interface adapter. During the experiments laptops are powered by 9-cell batteries. All the experiments were performed during daytime with temperature varying from 20°C to 30°C. In each experiment, the number of requests, which were sent from the source node to the destination node, was 2000.

This section focuses on analyzing the underlying network behavior for various scenarios and presenting the results. Due to real-world environmental factor such as fading, attenuation, scattering and presence of other interfering factor [45], after a number of experiments with one of the network configuration, it was observed that the results were not same. So in order to interpret the results, each experiment was done three times and the average of each value was taken to provide realistic and accurate results. All the experiments were performed in outdoor environment and places of the source node and destinations node were fixed during experiments.

The first set of experiments was done to investigate the effect of inter-packet transmission time to the network performance. In the second set of experiments, the effect of the position of laptop from the ground level was investigated. Figure 38. shows the setting of the network configuration for the fixed set of experiments.

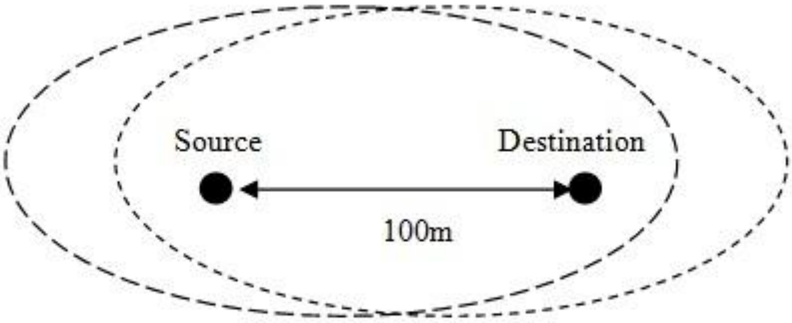

Figure 38: The network configuration in an open field.

In this group of experiments, we had two different scenarios, where the distance between the source node and the destination node was fixed at 100cm. For the first scenario, laptops computers were placed at 50cm height from the ground. The inter-packet transmission time (delay between transmissions of each packet) was varied in the range from 10ms to 3000ms. At the source node in addition to the inter-packet transmission time, the application data size was varied from 100 bytes to 8000 bytes for each selected delay and used as a parameter. In the second scenario, the inter-packet transmission time was fixed as 100ms, and laptop computers were placed at three different heights (0cm, 50cm, and 100cm) from the ground. The application data size was varied from 100 bytes to 8000 bytes and used as a parameter for each experiment.

4.3.Experimental Study Results and Analysis

There are so many performance metrics to describe the efficiency of wireless ad hoc networks. Most popular performance metrics used in simulations are the delivery ratio (or response ratio) and the the end-to-end delay (or average round trip time) per delivered packet.

The results of the experiments with fixed configurations are presented in the form of tables, Tables 22 – 31, and graphs, in Figures 39-42.

Table 22: Experiment results for different heights with application data size=100bytes.

Height from the ground, cm	Experiment	Response ratio	Average response time, ms
0	1	0.867	3.711
	2	0.873	6.193
	3	0.886	6.020
50	1	0.901	3.377
	2	0.901	4.038
	3	0.887	4.384
100	1	0.937	3.686
	2	0.902	3.411
	3	0.884	2.380

Table 23: Experiment results for different heights with application data size=1000bytes.

Height from the ground, cm	Experiment	Response ratio	Average response time, ms
0	1	0.846	20.902
	2	0.791	24.649
	3	0.771	24.826
50	1	0.785	18.773
	2	0.847	20.763
	3	0.796	20.553
100	1	0.939	20.716
	2	0.902	19.120
	3	0.885	19.190

Table 24: Experiment results for different heights with application data size=2000bytes.

Height from the ground, cm	Experiment	Response ratio	Average response time, ms
0	1	0.577	16.698
	2	0.711	15.980
	3	0.492	15.652
50	1	0.488	17.129
	2	0.434	16.181
	3	0.483	16.360
100	1	0.791	16.112
	2	0.798	15.976
	3	0.854	16.195

Table 25: Experiment results for different heights with application data size=4000bytes.

Height from the ground, cm	Experiment	Response ratio	Average response time, ms
0	1	0.519	47.524
	2	0.275	61.177
	3	0.219	57.112
50	1	0.369	34.544
	2	0.747	36.387
	3	0.729	39.694
100	1	0.790	37.692
	2	0.765	31.582
	3	0.761	31.614

Table 26: Experiment results for different heights with application data size=8000bytes.

Height from the ground, cm	Experiment	Response ratio	Average response time, ms
0	1	0.029	129.148
	2	0.349	78.696
	3	0.413	78.021
50	1	0.250	78.064
	2	0.445	78.596
	3	0.520	78.452
100	1	0.586	78.000
	2	0.610	78.013
	3	0.654	78.025

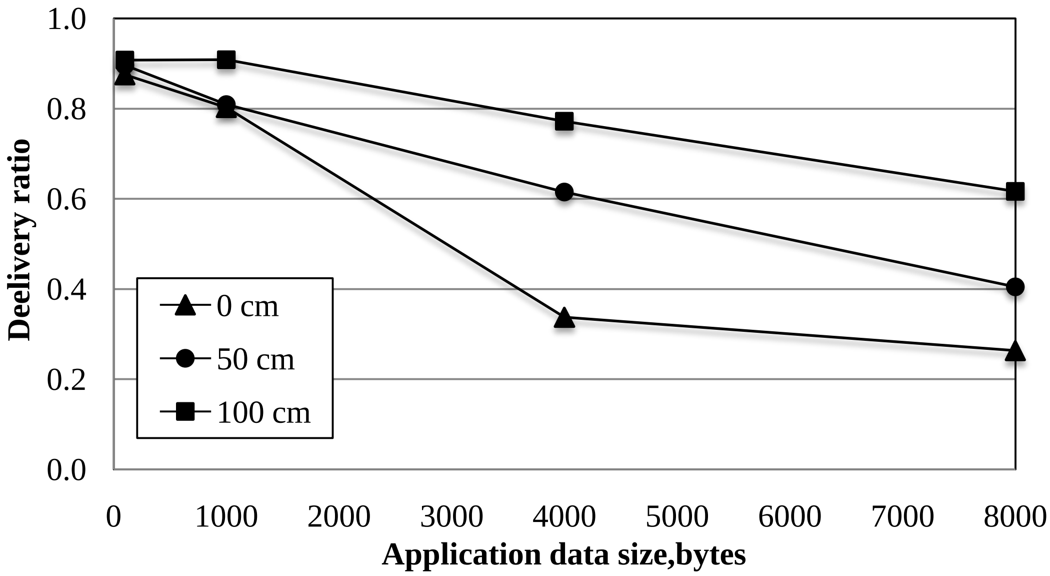

Figure 39: The delivery ratio versus application data size, for different heights.

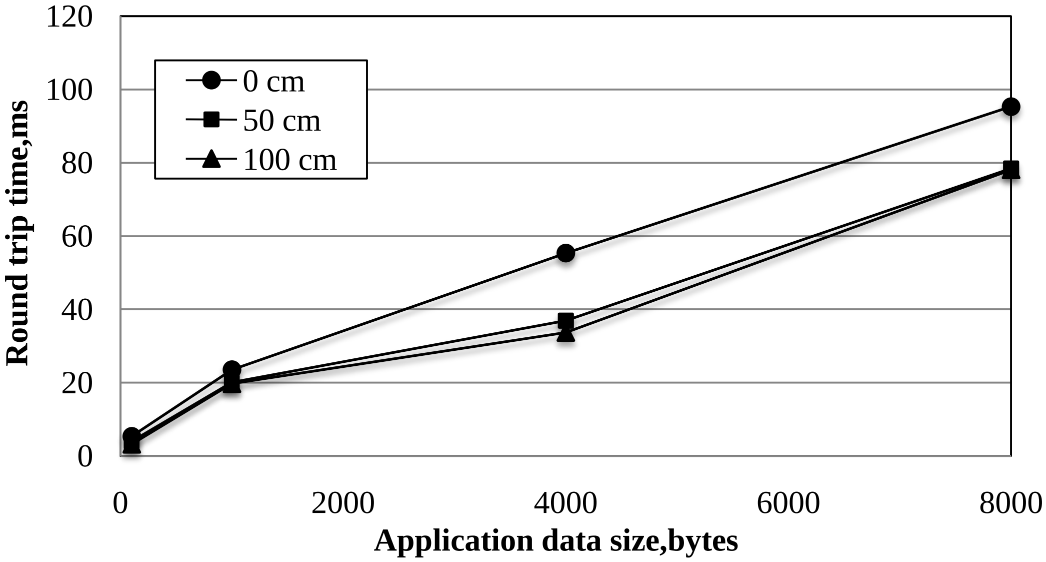

Figure 40: The average round trip time versus application data size, for different heights.

Table 27: Experiment results for different application data sizes with delay=10ms.

Application data size, bytes	Experiment	Response ratio	Average response time, ms
100	1	0.881	10.688
	2	0.824	4.638
	3	0.777	0.294
1000	1	0.686	111.356
	2	0.683	96.971
	3	0.699	72.258
2000	1	0.311	34.098
	2	0.299	33.372
	3	0.289	33.567
4000	1	0.463	85.505
	2	0.425	83.940
	3	0.231	78.654
8000	1	0.098	148.933
	2	0.107	147.276
	3	0.083	143.458

Table 28: Experiment results for different application data sizes with delay=50ms.

Application data size, bytes	Experiment	Response ratio	Average response time, ms
100	1	0.870	3.900
	2	0.930	4.077
	3	0.912	3.984
1000	1	0.776	22.619
	2	0.798	20.843
	3	0.747	20.672
2000	1	0.459	16.062
	2	0.534	16.225
	3	0.577	16.676
4000	1	0.516	31.911
	2	0.679	31.414
	3	0.292	31.937
8000	1	0.212	104.703
	2	0.260	106.002
	3	0.244	103.150

Table 29: Experiment results for different application data sizes with delay=100ms.

Application data size, bytes	Experiment	Response ratio	Average response time, ms
100	1	0.901	3.377
	2	0.901	4.038
	3	0.887	4.384
1000	1	0.785	18.773
	2	0.847	20.763
	3	0.796	20.553
2000	1	0.488	17.129
	2	0.434	16.181
	3	0.483	16.360
4000	1	0.369	34.544
	2	0.747	36.387
	3	0.729	39.694
8000	1	0.250	78.064
	2	0.445	78.596
	3	0.520	78.452

Table 30: Experiment results for different application data sizes with delay=500ms

Application data size, bytes	Experiment	Response ratio	Average response time, ms
100	1	0.927	4.053
	2	0.935	4.276
	3	0.944	3.899
1000	1	0.905	19.772
	2	0.909	20.194
	3	0.912	19.882
2000	1	0.424	18.559
	2	0.645	18.353
	3	0.599	17.609
4000	1	0.839	31.922
	2	0.877	32.729
	3	0.776	32.853
8000	1	0.784	78.645
	2	0.843	78.380
	3	0.749	78.782

Table 31: Experiment results for different application data sizes with delay=1000ms

Application data size, bytes	Experiment	Response ratio	Average response time, ms
100	1	0.877	6.202
	2	0.871	6.379
	3	0.872	6.234
1000	1	0.823	18.707
	2	0.903	20.683
	3	0.558	25.446
2000	1	0.780	17.639
	2	0.730	17.776
	3	0.720	15.601
4000	1	0.584	32.372
	2	0.753	32.769
	3	0.219	39.856
8000	1	0.312	80.446
	2	0.117	78.209
	3	0.141	74.330

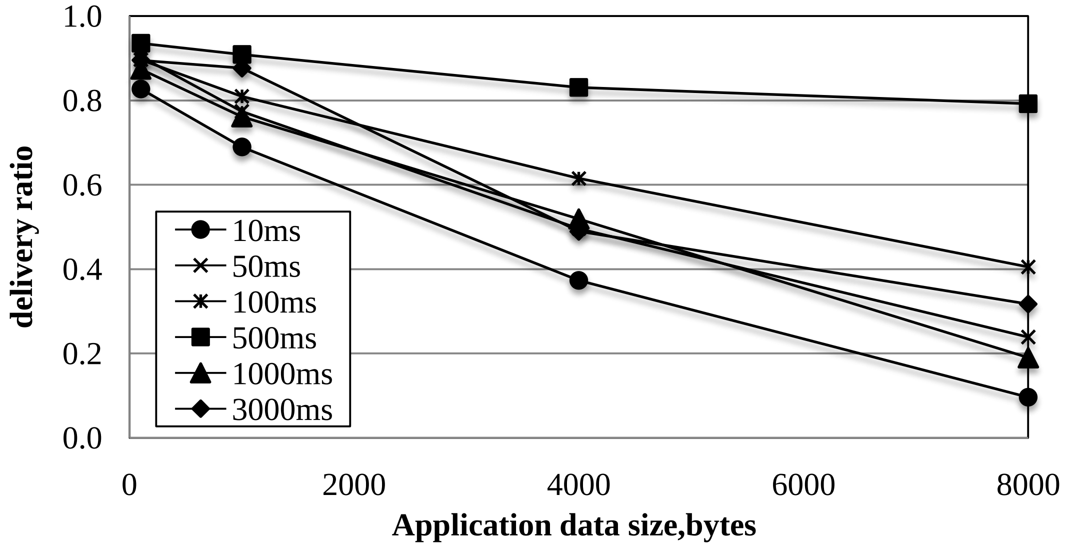

Figure 41: The delivery ratio versus application data size, for different inter-packet transmission times.

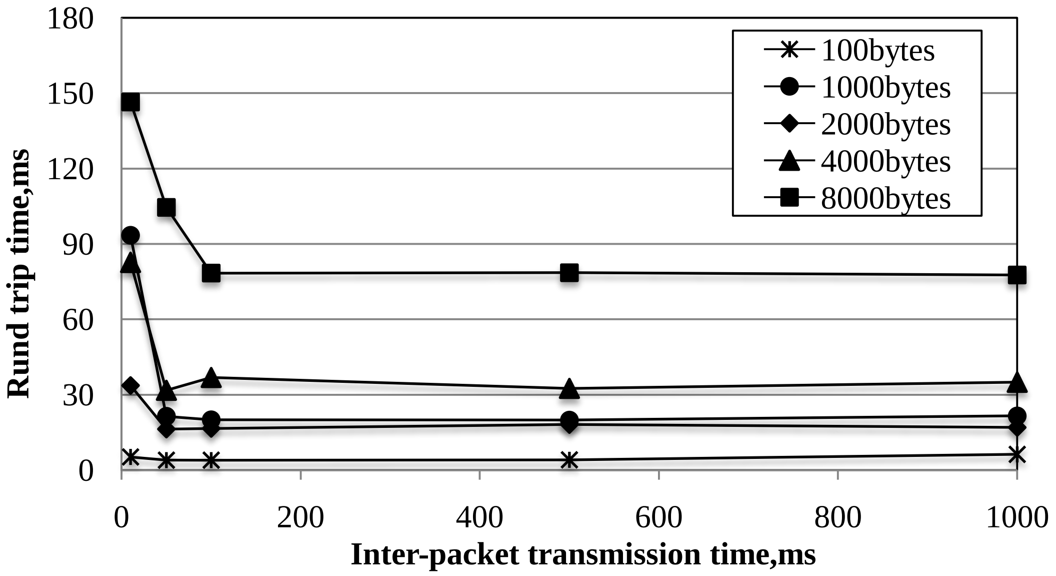

Figure 42: The average round trip time versus inter-packet transmission time, for different application data sizes

4.4.Discussion of the Experimental Results

The following comments and observations can be driven from the obtained results of experiments.

1. In the outdoor fixed experiments (Figures 39-42), the performance metrics used depend on the application data sizes.
2. Figures 39 and 41 show that the delivery ratio considerably decreases with the increase in the application data size. Figure 41 shows that delivery ratio varies with different inter-packet transmission times and it reaches the highest value with 500ms delay for all application data sizes. Moreover, as Figure 39 presents, for low application data sizes, delivery ratio is high for all heights (0cm, 50cm and 100cm) from the ground. The same figure, also shows that the position of the laptops affect delivery ratio considerably since there is less reflection and fading at high positions from the ground. For large application data sizes, more than one packet is transmitted since there is a limitation on the frame size in IEEE 802.11 MAC layer [46].

3. As graphs in Figures 40 and 42 demonstrate, the average round trip time is low for small application data sizes and it increases accordingly. From Figure 42 one can see that, for large application data sizes, the average round trip time varies for small delays but it does not change considerably for small data sizes. Also, when there is an increment in the inter-packet transmission time, the average round trip time decreases for large application data sizes (especially for 4000 bytes and 8000 bytes) up to a certain interval. Beyond that, the average round trip time does not show any improvement. As Figure 40 shows, the average round trip time does not considerably change with different heights of laptops from the ground.

Chapter 5

CONCLUSION

In this work, an anycast flooding simulation model of mobile ad hoc networks was developed, and practically important performance metrics are investigated. In order to have a clear representation of parallelism of events and processes in the distributed system of wireless LANs, a class of extended Petri nets was used to implement the model. The dependence of practically important performance metrics on the transmission radius, link availability, maximal possible node speed and probability of changing direction was investigated by conducting a large number of simulation studies. These metrics are the delivery ratio, the average number of hops, the relative traffic, the response time and the duplicate ratio.

Simulation results show that different probability of changing direction has a small affect the performance of the network. Also, changing the maximum node speed has small effect on performance of the network. For the small link availabilities performance metrics remain quite low even at transmission radius of 210m. On the other hand, decreasing the TTL can result in less traffic when other performance metrics are nearly same.

A series of experiments have been carried out in an outdoor real-world network environment using a program running on Microsoft Windows Vista. The performance of the wireless ad hoc network was investigated by conducting a large number of experimental runs with some important performance metrics obtained by the transmission of different sized packets, inter-packet transmission time and varying height of the source node and the destination node from the ground. As a final conclusion, we can say that increasing the height from the ground can improve the performance metrics. Beside, increment in the inter-packet transmission time is helpful for large application data sizes to decreases average round trip time up to a certain point.

The developed model of wireless ad hoc networks can be extended in a number of ways such as allowing joining and leaving of some nodes to/from the network, modeling of failure of nodes and changing the number of source nodes or anycast server inside the network. It is also possible to conduct real-world experiments with restricted-flooding and its result can be compared with our simulation results. These and some other extensions can be a subject of a further study.

In summary, the present simulation results and the results of the experiments together with the implemented scheme and the simulation model can be used for exploring different aspects of routing and data transmission in wireless ad hoc networks, implementation of a more efficient routing scheme, and a better and closer understanding toward anycasting in ad hoc wireless networks.

REFERENCES

[1] Wu S.L., Tseng Y.C., (2007) Wireless Ad Hoc Networking, *Auerbach Publications*, ISBN 978-0-8493-9254-2.

[2] C K Toh (2002) Ad Hoc Mobile Wireless Networks: Protocols & Systems. *Prentice Hall Publishers*, ISBN 0130078174.

[3] C. Ho, Obraczka K., Tsudik G., and Viswanath K., (1999) Flooding for reliable multicast in multi-hop ad hoc networks, *in Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM).*

[4] Royer E., Toh C. (1999) A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks, *IEEE Personal Communications 6*, pp.46–55.

[5] Johnson D. B., & Maltz D. A. (1996) Dynamic Source routing in ad hoc wireless networks in Mobile Computing. Kluwer Academic Publishers, vol. 353, Chapter 5, pp. 153-181.

[6] C. E. Perkins and P. Bhagwat. (1994) Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. *In Proceedings of the conference on Communications architectures, protocols and applications (SIGCOMM '94)*, pp. 234-244.

[7] Perkins C. E., & Royer E. M. (1999) Ad Hoc On Demand Distance Vector routing. *Proceedings of 2nd IEEE Workshop, Mobile Computer System and Applications*, pp. 90-100.

[8] Jacquet P., Muhlethaler P., Clausen T., Laouiti A., Qayyum A., & Viennot L. (2001) Optimized link state routing protocol for ad hoc networks. *In Proceedings of the 5th IEEE Multi Topic Conference* (INMIC 2001).

[9] S. Kannan, S. Karthik and ZV.P. Arunachalam. (2010) Performance analysis and comparison of mobile ad-hoc network routing protocols. *Research Journal of Applied Sciences*,Vol. 5, pp. 424-428.

[10] Khatri P., Rajput M., Shastri A. and Solanki K. (2010) Performance study of ad-hoc reactive routing protocols. *Journal of Computer Science*, Vol. 6, Issue: 10, pp.1159-1163.

[11] Samyak S., Amit K., Mahesh S. and Girish B. (2008) Performance Evaluation of Ad Hoc Routing Protocols Using NS2 Simulation. *Mobile and Pervasive Computing, CoMPC–2008*, Chennai, India.

[12] Arun Kumar B. R., Lokanatha C. Reddy, Prakash S. Hiremath. (2008) Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols. *International Journal of Computer Science and Information Security,* Vol. 8 No. 6 pp. 337-343.

[13] Mbarushimana C., Shahrabi A., (2007) Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks," *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on* , vol.2, no., pp.679-684.

[14] Peng G., Yang J., & Gao C (2004) ARDSR: An Anycast Routing Protocol for Mobile Ad hoc Network. *Symp. On Emerging Technologies of IEEE on Mobile and Wireless Communication*.

[15] Wang J., Zheng Y., Jia W. (2003) An AODV-Based Anycast Protocol in Mobile Ad hoc Network. *International Symp. of IEEE on Personal,Indoor, and Mobile Radio Communication Proceedings*.

[16] Saeed A., Khan L., Shah N., & Ali H. (2009) Performance comparison of two Anycast based reactive routing protocols for mobile Ad hoc networks. *Computer, Control and Communication, 2009. IC4 2009*. 2nd International Conference on , vol., no., pp.1-6.

[17] Jianxin W., Yuan Z., Cheng L., Weijia Jia., (2003) A-DSR: A DSR-based anycast protocol for IPv6 flow in mobile ad hoc networks. *Vehicular Technology Conference, 2003.* VTC 2003-Fall. 2003 IEEE 58th , vol.5, no., pp. 3094- 3098 Vol.5.

[18] Royer E. M., & Perkins C. E. (1999) Multicast Operation of the Ad hoc On-Demand Distance Vector Routing Protocol. *In Proc. Of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, pp. 207-218.

[19] Lei C.; Mnaouer, A.B. (2004) Performance evaluation of new hybrid multicast routing protocols for ad-hoc networks. *Communications Systems, 2004. ICCS 2004. The Ninth International Conference on* , vol., no., pp.70-74.

[20]] Martin M., & Takuro S. (2009) Route-Count Based Anycast Routing in Wireless Ad Hoc Networks. *Vehicular Technology Conference Fall (VTC 2009-Fall),* 2009 IEEE 70th.

[21] Kashif S., Lijuan C., Tu W., & Teresa D. (2008) A Hybrid Anycast Routing Protocol for Load Balancing in Heterogeneous Access Networks. *Computer Communications and Networks, 2008. ICCCN '08*. Proceedings of 17th International Conference.1-6.

[22] S. Jadhav T., Brown S., Doshi D. H., and Thekkekunnel R. G. (2005) Lessons Learned Constructing a Wireless Ad Hoc Network Test Bed. *In Proc. of the 1st Workshop on Wireless Network Measurements (WINMee).*

[23] Kotz D., Newport C., Gray R.S, Liu J., Yuan Y., & Elliott C. (2004) Experimental Evaluation of Wireless Simulation Assumptions. *In Proceedings of MSWiM 2004.*

[24] Gray R. S., Kotz D., Newport C., Dubrovsky N., Fiske A., Liu J., Masone C., McGrath S., and Yuan Y. (2004). Outdoor experimental comparison of four ad hoc routing algorithms. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems* (MSWiM '04). ACM, USA, pp. 220-229.

[25] Lin Y., Mohsenian Rad A. H., W. S. Wong V., and Song J-H. (2005). Experimental comparisons between SAODV and AODV routing protocols. In *Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling* (WMuNeP '05). ACM, USA, pp. 113-122.

[26] Oz G., Ozen Y. (2009) Experimental investigation of data transmission in wireless ad hoc networks. *Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, ICSCCW 2009. Fifth International Conference on*, pp.1-4.

[27] Oz G., Kostin A., Oyeniyi A. M., Sharghi Z. and Seifzadeh S. (2008) Prototype Application-Layer Test Bed for Implementation and Investigation of Routing and Data Dissemination in Wireless Mobile Ad Hoc Networks. *in 10th International Workshop on Computer Science and Information Technologies CSIT2008*, pp.27-31.

[28] Oz G., Abed A.K., Ozen, Y. (2009) Application-layer testbed for real-world experimentation in wireless ad hoc networks. *10th International Workshop on Computer Science and Information Technologies CSIT2009*, pp. 27-31.

[29] Abed A.K., Komili M., Oz, G. (2010) Experimental study of data dissemination in wireless ad hoc networks. *12th International Workshop on Computer Science and Information Technologies CSIT2010*.

[30] Kostin, A. & Ilushechkina, L. (2010) Modeling and Simulation of Distributed Systems”, *World Scientific Publication*.

[31] Kostin A. & Ilushechkina L. (2003) WinSim:Software and Manual, http://www.informatik.uni-hamburg.de/TGI/PetriNets/tods/quick.html.

[32] Brenner P. (1997) A Technical Tutorial on the IEEE 802.11 Protocol. BreezeCom.

[33] Kostin, A.E. (1989). Models and Algorithms for Organization of Distributed Data Processing in Information Systems, Diss. DSc , Moscow Institute of Electronic Technology (Technical University),(in Russian).

[34] Kostin A. and Ilushechkina, L. (2004) Winsim: A tool for performance evaluation of parallel and distributed systems. *LNCS*, vol. 3261. Springer, Berlin.

[35] Camp T. and Boleng J. and Davies V. (2002) A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications and Mobile Computing Special Issue*: Mobile Ad Hoc Networking – Research, Trends and Applications Volume 2, Issue 5, pp. 483–502.

[36] Royer E., Melliar-Smith P.M., and Moser L.(2001) An analysis of the optimum node density for ad hoc mobile networks. *In Proceedings of the IEEE International Conference on Communications (ICC).*

[37] Lin G., Noubir G, Rajaraman R. (2004) Mobility models for ad hoc network simulation. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* , vol.1, no., pp. 4.

[38] Jardosh A., Belding-Royer E. M, Almeroth K.C., and Suri S.(2003) Towards realistic mobility models for mobile ad hoc networks. *In Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom '03)*. ACM, USA, pp. 217-229.

[39] Gowrishankar S., Basavaraju T. G., Sarkar S. K. (2007) Effect of Random Mobility Models Pattern in Mobile Ad hoc Networks. *International Journal of Computer and Network Security (IJCSNS),*VOL.7 No.6.

[40] Obrachzka, K., & Viswanath, K. (2001). Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks 7*, pp. 627-634.

[41] Gelenbe E. , Pujolle G. (1998) Introduction to queueing networks. *WILEY Publication.*

[42] Haci, H. (2010). MS Thesis: Performance investigation of simulation models of wireless mobile ad hoc networks. EMU.

[43] IEEE 802.11-2007, Standard for LAN/MAN - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[44] Azizi R., Oz G.(2011) Performance evaluation of data dissemination in real-world wireless ad hoc networks, *Communications and Information Technology (ICCIT)*, 2011 International Conference on , pp.176-179.

[45] Jardosh A.P., Belding-Royer E.M., Almeroth K.C., Suri S. (2005) Real-World Environment Models for Mobile Network Evaluation. *Selected Areas in Communications, IEEE Journal on*, Vol: 23, Issue: 3, pp: 622 – 632.

[46] Crow B. P., Widjaja I., Kim J. G. and Sakaim P. T. (1997) IEEE 802.11 Wireless Local Area Networks, IEEE communication Magazines, pp. 116-126.

APPENDICES

Appendix A: The source text of the model of the switching module.

```
(****************************************************************************)
(*                             Switching module                             *)
(*                       for the model of anycasting                        *)
(*                   in an ad-hoc mobile wireless network                   *)
(*                                                                          *)
(*                                                                          *)
(*            File  SWITCH.JOM          Date 10 January 2011                *)
(****************************************************************************)
SEGMENT SWITCH, TICK = MSEC;
ATTRIBUTES
    MTYP: INTEGER;    (* Message type: 1 - initializing data from switcher
                                       2 - multicast request,
                                       3 - unicast reply,
                                       4 - reserved                         *)
    SRC : INTEGER;                          (* Source node of mcast request *)
    DEST: INTEGER;                            (* Destination node, not used *)
    SNDR: INTEGER;        (* If MTYP > 1, last sending node of this message *)
    NEXT: INTEGER;         (* Next node to pass message by switching module *)
    MID : INTEGER;         (* Message Id: 1, 2, ... ; incremented by source *)
    WRK : INTEGER;                                     (* Working attribute *)
    XCD : REAL;                                   (* X-coordinate of sender *)
    YCD : REAL;                                   (* Y-coordinate of sender *)
    RX  : REAL;                                 (* X-coordinate of receiver *)
    RY  : REAL;                                 (* Y-coordinate of receiver *)
    TTL : INTEGER;                (* The number of hops for message to pass *)
    HOPS: INTEGER;          (* The number of hops already passed by message *)
    TIM : REAL;                    (* Moment of generation of mcast request *)
    ND1 : INTEGER;             (* Id of the first passed node (source node) *)
    ND2 : INTEGER;                          (* Id of the second passed node *)
    ND3 : INTEGER;                           (* Same for other passed nodes *)
    ND4 : INTEGER;
    ND5 : INTEGER;
    ND6 : INTEGER;
    ND7 : INTEGER;
    ND8 : INTEGER;
    ND9 : INTEGER;
    ND10: INTEGER;

DATA
    PDIR /0.0/: REAL;   (* Probability to change direction of node movement *)
    NODS /0/: INTEGER;                 (* Total number of nodes in the area *)
    NSRV /0/: INTEGER;           (* Number of anycast servers, NSRV << NODS *)
    XMIN /0.0/: REAL;                   (* Minimal X-coordinate of the area *)
    XMAX /0.0/: REAL;                   (* Maximal X-coordinate of the area *)
    YMIN /0.0/: REAL;                   (* Minimal Y-coordinate of the area *)
    YMAX /0.0/: REAL;                   (* Maximal Y-coordinate of the area *)
    DX /0.0/:   REAL;                      (*  Max step size in X-direction *)
    DY /0.0/:   REAL;                      (*  Max step size in Y-direction *)
    XNEW /0.0/: REAL;                 (* New node coordinate in X-direction *)
    YNEW /0.0/: REAL;                 (* New node coordinate in Y-direction *)
    DT /0.0/: REAL;                     (* Time  to recompute node position *)
    IDNU /0/: INTEGER;                       (* Counter of node identifiers *)
    DMAX /0.0/: REAL;                                (* Transmission radius *)
    CNT /0/: INTEGER;        (* Counter of nodes within transmission radius *)
    FLG /0/: INTEGER;                        (* Flag used in T1001 and X101 *)
    XPOS: ARRAY [50] OF REAL;             (* Current X-coordonates of nodes *)
    YPOS: ARRAY [50] OF REAL;             (* Current Y-coordonates of nodes *)
    XNXT: ARRAY [50] OF REAL;                          (* Next X-coordinate *)
    YNXT: ARRAY [50] OF REAL;                          (* Next Y-coordinate *)
    DIST: ARRAY [49] OF INTEGER;                        (* IDs of neighbors *)

(* Pascal section with  working variables *)
INTERFACE
IMPLEMENTATION
var
   inod: integer;                                             (* Node index *)
   OK,j,i: integer;                                     (* Flag and indexes *)
   xi, yi: real;                                       (* Working variables *)
   dxx, dyy: real;               (* Coordinate difference between two nodes *)
   dis: real;                                 (* Distance between two nodes *)
```

```
   prob: real;                                 (* Value of function FRANDOM *)
END.

(* Preparation of common initializing data  for all nodes and
   generation of initial positions of  nodes in given area *)
NET   T4: S4/S3, S5;
TRANS T4: %S5.MTYP := 1;                        (* Initializing messge type *)
          %S5.WRK := %NODS;                            (* The number of nodes *)
          %S5.MID := %NSRV;                          (* The number of servers *)

          (* Specification of initial and next positions of nodes *)
          for inod := 1 to %NODS do
          BEGIN
          %XPOS[inod] := UNIFRM (1, %XMIN, %XMAX);
          %YPOS[inod] := UNIFRM (1, %YMIN, %YMAX);
          %XNXT[inod] := UNIFRM (1, %XPOS[inod] - %DX, %XPOS[inod] + %DX);
          %YNXT[inod] := UNIFRM (1, %YPOS[inod] - %DY, %YPOS[inod] + %DY);
          if (%XNXT[inod] < %XMIN) then %XNXT[inod] := %XMIN;
          if (%XNXT[inod] > %XMAX) then %XNXT[inod] := %XMAX;
          if (%YNXT[inod] < %YMIN) then %YNXT[inod] := %YMIN;
          if (%YNXT[inod] > %YMAX) then %YNXT[inod] := %YMAX;
          (* Writeln('Ndpos ', inod, ':', %XPOS[inod], ' ', %YPOS[inod]); *)
          (* Writeln('Ndnxt ', inod, ':', %XNXT[inod], ' ', %YNXT[inod]); *)
          END;


(* Periodical re-computing of positions of all nodes in the area *)
NET   Y1: S3, S2/S1;
NET   T1: S1/S2;
TIME  T1: %DELAY := %DT;
TRANS T1:                          (* Re-computing after elapsing step time *)
          for inod := 1 to %NODS do
          BEGIN                                            (* External loop *)
          prob := FRANDOM(1);
          xi := %XNXT[inod] + (%XNXT[inod] - %XPOS[inod]);
          yi := %YNXT[inod] + (%YNXT[inod] - %YPOS[inod]);
          if (prob < %PDIR) OR (xi < %XMIN) OR       (* Change of direction *)
          (xi > %XMAX) OR (yi < %YMIN) OR (yi > %YMAX)  (* Boundary reached *)
          then                                     (* Compute new direction *)
            BEGIN
            OK := 0;                          (* Init flag for X-coordinate *)
            WHILE (OK = 0) do          (* Determine new X-coodinate of node *)
              BEGIN
              xi := UNIFRM (1, %XNXT[inod] - %DX, %XNXT[inod] + %DX);
              if (xi > %XMIN) AND (xi < %XMAX)
              then BEGIN %XPOS[inod] := %XNXT[inod];
              %XNXT[inod]:= xi; OK := 1; END;
              END;
            OK := 0;                          (* Init flag for Y-coordinate *)
            WHILE (OK = 0) do          (* Determine new Y-coodinate of node *)
              BEGIN
              yi := UNIFRM (1, %YNXT[inod] - %DY, %YNXT[inod] + %DY);
              if (yi > %YMIN) AND (yi < %YMAX)
              then BEGIN %YPOS[inod] := %YNXT[inod];
              %YNXT[inod] := yi; OK := 1; END;
              END;

              (* Writeln('Nd ', inod, ':', %XPOS[inod],' ', %YPOS[inod]); *)
            END
            else                              (* Move in the same direction *)
             BEGIN
             %XPOS[inod] := %XNXT[inod]; %XNXT[inod] := xi;       (* Along X *)
             %YPOS[inod] := %YNXT[inod]; %YNXT[inod] := yi;       (* Along Y *)
             END;

            (* Writeln('Nd ', inod, ':', %XPOS[inod],' ', %YPOS[inod]); *)
            (* Writeln('Nd ', inod, ':', %XNXT[inod],' ', %YNXT[inod]); *)
          END; (* External loop *)

(* Initial distribution of identifiers to nodes *)
NET   Y2: S5, S6/S7;
TRANS Y2: %IDNU := %IDNU + 1;        (* Id for the next node, can be > NODS *)
          %S7.NEXT := %IDNU;          (* Store Id of next node in attribute *)

NET   X2: S7/S8, S9;
CONTR X2: IF %IDNU <= %NODS                          (* All IDs distributed? *)
          THEN %OUT := 1                        (* Not yet, next Id message *)
```

```
          ELSE Begin
               %IDNU := %NODS;                        (* Restore maximal Id *)
               %OUT := 2;                                (* Finish the loop *)
               End;
NET   T8: S8/S10, S6;                                            (* Looping *)
NET   T9: S9;                                               (* Absorb token *)
NET   Y3: S10, S11, S12/S2000;                        (* Prepare to pass message *)

(* Inputs  from network nodes *)
NET  Y1000: s101,s102,s103,s104,s105,s106,s107,s108,s109,s110,
            s111,s112,s113,s114,s115,s116,s117,s118,s119,s120,
            s121,s122,s123,s124,s125,s126,s127,s128,s129,s130,
            s131,s132,s133,s134,s135,s136,s137,s138,s139,s140,
            s141,s142,s143,s144,s145,s146,s147,s148,s149,s150/S99;

(* Analyzing what was submitted from a node *)
NET   X100: S99/Q1000, Q1001, S50;
CONTR X100: if %S99.MTYP = 2
            then %OUT := 1                             (* Multicast request *)
            else if %S99.MTYP = 3
                 then %OUT := 2                            (* Unicast reply *)
                 else %OUT := 3;                           (* Anything else *)

NET T50: S50;                                           (* Absorb token *)

(* Handling the desire of a node to pass a request to neighbor nodes *)
NET   T1000: Q1000, S1000/S501;
TRANS T1000: i := %Q1000.SNDR;                 (* ID of the requesting node *)
             %CNT := 0;
             %S501.XCD := %XPOS[i];  (* X-coordinate of the requesting node *)
             %S501.YCD := %YPOS[i];  (* Y-coordinate of the requesting node *)
             for j :=1 to %NODS do  (* Determing neighbors of the requester *)
             BEGIN                                         (* External loop *)
               if i<> j
               then BEGIN
                    (*writeln('i= ', i, '  j= ', j);*)
                    dxx := %XPOS[i] - %XPOS[j];
                    dyy := %YPOS[i] - %YPOS[j];
                    dis := SQRT(dxx*dxx + dyy*dyy);  (* Distan to next node *)
                    if dis < %DMAX        (* Node j is in the coverage area *)
                    then BEGIN
                    (* Writeln('Dis    i    j', dis, '    ',i,'    ', j); *)
                         %CNT := %CNT + 1;                (* Count the node *)
                         %DIST[%CNT] := j;                  (* Store its ID *)
                         END;
                    END;
             END;  (* External loop *)
             (* Writeln('T1000:DIST array ', %DIST[%CNT]); *)

(* Passing the requested message to reachable neighbor nodes, if any *)
NET   Y500: S500, S501/S502;
NET   X502: S502/S503, S1000;
CONTR X502: IF %CNT = 0
            THEN %OUT := 2                        (* No more neighbor nodes *)
            ELSE %OUT := 1;                   (* Consider the next neighbor *)
NET T503: S503/S500, S11;   (* Prepare to pass the message to next neighbor *)
TRANS T503: i := %DIST[%CNT];                         (* ID of the neighbor *)
            %S11.NEXT := i;
            %CNT := %CNT - 1;
            %S11.RX := %XPOS[i];            (* Coordinate of receiving node *)
            %S11.RY := %YPOS[i];
            (* Writeln('T503:DIST array ', %DIST[%CNT]); *)
            (* WRITELN('SWITCH: Nd ', %S11.SNDR, ' --> ', %S11.NEXT); *)

(* Handling the desire of a node to pass a ucast reply to only one node *)
NET   T1001: Q1001/S1001;
TRANS T1001: i := %Q1001.SNDR;                 (* ID of the requesting node *)
             J := %Q1001.NEXT;                  (* ID of addressed receiver *)
             %FLG := 0;
             %S1001.XCD := %XPOS[i];      (* X-coord of the requesting node *)
             %S1001.YCD := %YPOS[i];      (* Y-coord of the requesting node *)
             dxx := %XPOS[i] - %XPOS[j];
             dyy := %YPOS[i] - %YPOS[j];
             dis := SQRT(dxx*dxx + dyy*dyy);  (* Distance to addressed node *)
             if dis < %DMAX            (* Addressed node j in coverage area *)
             then %FLG := 1;                                    (* Set flag *)
```

```
NET   X101: S1001/S12, S13;
CONTR X101: if %FLG = 1
            then %OUT := 1                   (* Addressed node is reachable *)
            else %OUT := 2;                    (* The node is not reachable *)
NET   T13: S13;                                                 (* Absorber *)

(* Passing a message to a node with ID = S2000.NEXT *)
NET   X2000: s2000/s201,s202,s203,s204,s205,s206,s207,s208,s209,s210,
s211,s212,s213,s214,s215,s216,s217,s218,s219,s220,
s221,s222,s223,s224,s225,s226,s227,s228,s229,s230,
s231,s232,s233,s234,s235,s236,s237,s238,s239,s240,
s241,s242,s243,s244,s245,s246,s247,s248,s249,s250;

CONTR X2000: %OUT := %S2000.NEXT;

(* Attaching copies of the node segment MONOD to the switching segment *)
ATTACH MONOD/NOD1,NOD2,NOD3,NOD4,NOD5, NOD6, NOD7, NOD8, NOD9, NOD10,
NOD11, NOD12, NOD13, NOD14, NOD15, NOD16, NOD17, NOD18, NOD19, NOD20,
NOD21, NOD22, NOD23, NOD24, NOD25, NOD26, NOD27, NOD28, NOD29, NOD30,
NOD31, NOD32, NOD33, NOD34, NOD35, NOD36, NOD37, NOD38, NOD39, NOD40,
NOD41, NOD42, NOD43, NOD44, NOD45, NOD46, NOD47, NOD48, NOD49, NOD50/;

(* Linking switching segment  with node segments (copies of MONOD) *)
LINK SWITCH,NOD1:  S101,S100/S201,S200;
LINK SWITCH,NOD2:  S102,S100/S202,S200;
LINK SWITCH,NOD3:  S103,S100/S203,S200;
LINK SWITCH,NOD4:  S104,S100/S204,S200;
LINK SWITCH,NOD5:  S105,S100/S205,S200;
LINK SWITCH,NOD6:  S106,S100/S206,S200;
LINK SWITCH,NOD7:  S107,S100/S207,S200;
LINK SWITCH,NOD8:  S108,S100/S208,S200;
LINK SWITCH,NOD9:  S109,S100/S209,S200;
LINK SWITCH,NOD10: S110,S100/S210,S200;
LINK SWITCH,NOD11: S111,S100/S211,S200;
LINK SWITCH,NOD12: S112,S100/S212,S200;
LINK SWITCH,NOD13: S113,S100/S213,S200;
LINK SWITCH,NOD14: S114,S100/S214,S200;
LINK SWITCH,NOD15: S115,S100/S215,S200;
LINK SWITCH,NOD16: S116,S100/S216,S200;
LINK SWITCH,NOD17: S117,S100/S217,S200;
LINK SWITCH,NOD18: S118,S100/S218,S200;
LINK SWITCH,NOD19: S119,S100/S219,S200;
LINK SWITCH,NOD20: S120,S100/S220,S200;
LINK SWITCH,NOD21: S121,S100/S221,S200;
LINK SWITCH,NOD22: S122,S100/S222,S200;
LINK SWITCH,NOD23: S123,S100/S223,S200;
LINK SWITCH,NOD24: S124,S100/S224,S200;
LINK SWITCH,NOD25: S125,S100/S225,S200;
LINK SWITCH,NOD26: S126,S100/S226,S200;
LINK SWITCH,NOD27: S127,S100/S227,S200;
LINK SWITCH,NOD28: S128,S100/S228,S200;
LINK SWITCH,NOD29: S129,S100/S229,S200;
LINK SWITCH,NOD30: S130,S100/S230,S200;
LINK SWITCH,NOD31: S131,S100/S231,S200;
LINK SWITCH,NOD32: S132,S100/S232,S200;
LINK SWITCH,NOD33: S133,S100/S233,S200;
LINK SWITCH,NOD34: S134,S100/S234,S200;
LINK SWITCH,NOD35: S135,S100/S235,S200;
LINK SWITCH,NOD36: S136,S100/S236,S200;
LINK SWITCH,NOD37: S137,S100/S237,S200;
LINK SWITCH,NOD38: S138,S100/S238,S200;
LINK SWITCH,NOD39: S139,S100/S239,S200;
LINK SWITCH,NOD40: S140,S100/S240,S200;
LINK SWITCH,NOD41: S141,S100/S241,S200;
LINK SWITCH,NOD42: S142,S100/S242,S200;
LINK SWITCH,NOD43: S143,S100/S243,S200;
LINK SWITCH,NOD44: S144,S100/S244,S200;
LINK SWITCH,NOD45: S145,S100/S245,S200;
LINK SWITCH,NOD46: S146,S100/S246,S200;
LINK SWITCH,NOD47: S147,S100/S247,S200;
LINK SWITCH,NOD48: S148,S100/S248,S200;
LINK SWITCH,NOD49: S149,S100/S249,S200;
LINK SWITCH,NOD50: S150,S100/S250,S200;

SEGEND.
```

Appendix B: The source text of the model of a node module.

```
(****************************************************************************)
(*                               Node module                                *)
(*                       for the model of anycasting                        *)
(*                            in an ad hoc WLAN                             *)
(*                                                                          *)
(*                                                                          *)
(*            File  MONOD.JOM            Date  10 January   2011            *)
(****************************************************************************)
SEGMENT MONOD, TICK = MSEC;
ATTRIBUTES
    MTYP: INTEGER;    (* Message type: 1 - initializing data from switcher
                                       2 - multicast request,
                                       3 - unicast reply,
                                       4 - reserved                         *)
    SRC : INTEGER;                          (* Source node of mcast request *)
    DEST: INTEGER;                            (* Destination node, not used *)
    SNDR: INTEGER;        (* If MTYP > 1, last sending node of this message *)
    NEXT: INTEGER;         (* Next node to pass message by switching module *)
    MID : INTEGER;         (* Message Id: 1, 2, ... ; incremented by source *)
    WRK : INTEGER;                                     (* Working attribute *)
    XCD : REAL;                                   (* X-coordinate of sender *)
    YCD : REAL;                                   (* Y-coordinate of sender *)
    RX  : REAL;                                 (* X-coordinate of receiver *)
    RY  : REAL;                                 (* Y-coordinate of receiver *)
    TTL : INTEGER;                (* The number of hops for message to pass *)
    HOPS: INTEGER;          (* The number of hops already passed by message *)
    TIM : REAL;                    (* Moment of generation of mcast request *)
    ND1 : INTEGER;             (* Id of the first passed node (source node) *)
    ND2 : INTEGER;                          (* Id of the second passed node *)
    ND3 : INTEGER;                           (* Same for other passed nodes *)
    ND4 : INTEGER;
    ND5 : INTEGER;
    ND6 : INTEGER;
    ND7 : INTEGER;
    ND8 : INTEGER;
    ND9 : INTEGER;
    ND10: INTEGER;


DATA
    SELF /0/: INTEGER;                            (* Unique Id of this node *)
    MNUM /0/: INTEGER;                    (* Counter of message identifiers *)
    T0 /0.0/: REAL;          (* Period of request generation by source node *)
    NODS /0/: INTEGER;         (* Number of nodes in the area, with servers *)
    MLST /0/: INTEGER;           (* ID of previously received mcast request *)
    ULST /0/: INTEGER;             (* ID of previously received ucast reply *)
    MNEW /0/: INTEGER;                       (* Flag of a new mcast request *)
    UNEW /0/: INTEGER;                         (* Flag of a new ucast reply *)
    RMIN /0.0/: REAL;        (* Min distance to a close node, reliable link *)
    RMAX /0.0/: REAL;        (* Max distance to a close node, reliable link *)
    TON /0.0/: REAL;                       (* Mean time of link in ON state *)
    TOFF /0.0/: REAL;                     (* Mean time of link in OFF state *)
    TREC /0.0/: REAL;                  (* Period of checking of link states *)
    DNUM /8/: INTEGER;      (* The number of directional links of each node *)
    TTLM /7/: INTEGER;             (* Maximal TTL, must be not more than 10 *)
    DILS: ARRAY [8] OF INTEGER;         (* States of links: 1 - ON, 0 - OFF *)
    TLS: ARRAY [8] OF REAL;        (* Moments of state termination of links *)
    NSRV /0/: INTEGER;                         (* Number of anycast servers *)
    STRT /0.0/: REAL;                     (* Starting moment of time-out T0 *)
    REMT /0.0/: REAL;                         (* Remaining time of time-out *)
    INR  /0/: INTEGER; (* Interruprt flag: 0 - no interrupt, 1 - interrupt  *)
    TTIM /0.0/: REAL;            (*calculating total response time for all packets*)
    TCNT /0/: INTEGER;

(* Pascal section with  working variables *)
INTERFACE
IMPLEMENTATION
var
   inod,i: integer;                                           (* Node index *)
   ind: integer;                                           (* Working index *)
   OK:   integer;                                                 (* A flag *)
   xi, yi: real;                                       (* Working variables *)
   prob : real;                                          (* Value of FRANDOM *)
```

```
   curtime: real;                                 (* Current simulation time *)
   dist :real;                   (* Distance from receiving to sending node  *)
   dxx, dyy : real;                               (* Coordinate differencies *)
   tang : real;           (* Tangent of angle from receiving to sending node *)
END.

(* Input from the switching segment *)
NET   X1: s200/s1, s2;
CONTR X1: IF %S200.MTYP = 1
          THEN %OUT := 1          (* Initializing data from switching module *)
          ELSE %OUT := 2;            (* A request or reply from another node *)

NET   T1: S1/S40,S4;          (* Copying init data from the switching module *)
TRANS T1: %NODS := %S1.WRK;               (* The number of nodes in the area *)
          %NSRV := %S1.MID;                         (* The number of servers *)
          %SELF := %S1.NEXT;                     (* Identifier for this node *)
          (* WRITELN ('My Id = ', %SELF); *)

          (* Initializing of random link states and state durations *)
          for i := 1 to %DNUM do
          BEGIN                                             (* Next link *)
          prob := FRANDOM (2);
          if prob < 0.5          (* ON and OFF states with equal probability *)
          then begin                         (* ON link state in direction i *)
               %DILS[i] := 1;
               %TLS[i] := EXPON (2, %TON)
               end
          else begin                        (* OFF link state in direction i *)
               %DILS[i] := 0;
               %TLS[i] := EXPON (2, %TOFF)
               end
          END;

(* Periodic checking and changing of link states *)
NET   Y40: S40, S42/S41;
NET   T41: S41/S42;
TIME  T41: %DELAY := %TREC;                               (* Constant period *)
TRANS T41: curtime := CLOCK(1);
           for i := 1 to %DNUM do
           BEGIN
           if curtime >= %TLS[i] (* If link state durati elapsed, change it *)
           then BEGIN
                if %DILS[i] = 1                           (* It was ON state *)
                then begin
                     %DILS[i] := 0;
                     %TLS[i] := curtime + EXPON(3, %TOFF)      (* Set to OFF *)
                     end
                else begin                               (* It was OFF state *)
                     %DILS[i] := 1;
                     %TLS[i] := curtime + EXPON(4, %TON)  (* Set to ON state *)
                     end
                END
           else continue;                            (* No change for link i *)
           END;

NET   X4: S4/S5, S6;
CONTR X4: if %SELF = %NODS (* Only node with ID = NODS may be a source node *)
          then %OUT := 1
          else %OUT := 2;

(* Generation of an mcast request  by source node, with period T0 *)
NET   Y5: S5, S95, S94/S96;
TRANS Y5: %S96.MTYP := 2;                               (* A request message *)
          %S96.SRC := %SELF;                               (* Source node Id *)
          %S96.DEST := 0;                                        (* Not used *)
          %S96.SNDR := %SELF;                    (* ID of immediate sender *)
          %S96.TTL  := %TTLM;                        (* Max number of hops *)
          %S96.HOPS := 0;                                 (* Counter of hops *)
          %S96.TIM  := CLOCK(1);                     (* Moment of generation *)
          %MNUM := %MNUM + 1;                               (* ID of request *)
          %S96.MID := %MNUM;
          %STRT := CLOCK(1);                         (* Start of time-out T0 *)
          %S96.ND1 := 0;            (* Initializing attributes ND1 .. ND10 *)
          %S96.ND2 := 0;
          %S96.ND3 := 0;
          %S96.ND4 := 0;
          %S96.ND5 := 0;
```

```
          %S96.ND6 := 0;
          %S96.ND7 := 0;
          %S96.ND8 := 0;
          %S96.ND9 := 0;
          %S96.ND10 := 0;


NET   T5: S96/S97, S98;
TRANS T5: %INR := 0;                (* Init: no replies to this request yet *)

(* Period of request generation *)
NET   T98: S98/S81;
TIME  T98: %DELAY := %T0;

(* Any reply received during T0? *)
NET   X81: S81/S94, S82;
CONTR X81: if %S99 > 0
           then %OUT := 2                 (* First reply received during T0 *)
           else %OUT := 1;                            (* No reply during T0 *)
NET   T82: S82/S95;
(* TRANS T82: WRITELN ('Source: reply MID and HOPS are ',%S177.MID:5,%S177.HOPS:5); *)

(* Mcast request or ucast reply passed to this node by switching module *)

NET   X2: S2/S15, Q1, Q2, S31, S3;
CONTR X2: %S2.HOPS := %S2.HOPS + 1; %S2.TTL := %S2.TTL - 1;
          if (%SELF = %NODS)
          then BEGIN
               if (%S2.MTYP = 3)
               then %OUT := 1         (* Source node received a ucast reply *)
               else %OUT := 5     (* Source got its own request, discard it *)
               END
          else BEGIN                           (* This is not a source node *)
               if (%S2.MTYP = 2)
               then %OUT := 2                     (* Received mcast request *)
               else %OUT := 3  (* Ucast reply for forwarding back to source *)
               END;

(* Now check if the  message came from very close node or via ON link *)
          dxx := %S2.XCD - %S2.RX;
          dyy := %S2.YCD - %S2.RY;
          dist := sqrt (dxx * dxx + dyy * dyy);     (* Distance from sender *)
          if dist > UNIFRM (1, %RMIN, %RMAX)       (* Not very close sender *)
          then
            BEGIN           (* Determine a link and its state to the sender *)
            tang := dyy/(dxx + 0.1);                     (* Direction angle *)
            if (tang >= 0.0) AND (tang <= 1.0) AND (dxx > 0) AND (dyy >= 0)
            then i := 1                         (* First directional sector *)
            else
            if (tang > 1.0) AND (dxx > 0) AND (dyy >= 0)
            then i := 2                           (* 2nd directional sector *)
            else
            if (tang <= -1.0) AND (dxx < 0) AND (dyy >= 0)
            then i := 3                           (* 3rd directional sector *)
            else
            if (tang > -1.0) AND (tang <= 0) AND (dxx < 0) AND (dyy >= 0)
            then i := 4                           (* 4th directional sector *)
            else
            if (tang > 0) AND (tang <= 1.0) AND (dxx < 0) AND (dyy < 0)
            then i := 5                           (* 5th directional sector *)
            else
            if (tang > 1.0) AND (dxx < 0) AND (dyy < 0)
            then i := 6                           (* 6th directional sector *)
            else
            if (tang <= -1.0) AND (dxx > 0) AND (dyy < 0)
            then i := 7                           (* 7th directional sector *)
            else i := 8;                          (* 8th directional sector *)
            if %DILS[i] = 0                                  (* Link is OFF *)
            then %OUT := 4;            (* Link is OFF, discard this message *)
            END;                                    (* End determine a link *)

(* Handling of the received ucast reply by source node *)
NET   X15: S15/S99, S16, S17;
CONTR X15: if (%S98 = 0)       (* Current or next period T0 not yet running *)
           then %OUT := 0                (* To wait a little when T0 starts *)
           else if (%INR = 0) AND (%S15.MID = %S98.MID)
                then BEGIN
```

```
                    %TCNT := %TCNT + 1;
                     WRITELN('Current response time for the last packet: ', (CLOCK (1)
- %S15.TIM));
                    WRITELN('Current counter value', %TCNT);
                    %TTIM := %TTIM + (CLOCK (1) - %S15.TIM);
                    WRITELN('Aggregated response time for all packets: ', %TTIM);
                    WRITELN('The average response time', (%TTIM / %TCNT));
                    %OUT := 1; %INR := 1        (* First reply  during T0 *)

                    END
                else BEGIN
                     if %INR = 1        (* First reply was received already *)
                     then %OUT := 2              (* Another reply during T0 *)
                     else %OUT := 3         (* All replies after elapsed T0 *)
                     END;


(*   *************** modified by HH ********** *)

NET T177: S99;


(*   ************ EOM *************** *)

(* Handling  of the received ucast reply by intermediate simple node *)
NET   T20: Q2/S20;
TRANS T20: if %S20.MID = %ULST (* Compare received ID with ID stored before *)
          then %UNEW := 0          (* Duplicated reply from another server *)
          else begin                      (* New reply passed to this node *)
                %UNEW := 1;                        (* Set flag of new reply *)
               %ULST := %S20.MID;                          (* Store its ID *)
               end;
          (* WRITELN('Nd ', %SELF, ' from ', %S9.SNDR, ' MID = ', %S9.MID); *)

(* Decision on the received ucast reply *)
NET   X30: S20/S70, S80, S90;
CONTR X30: %OUT := 1;                        (* Forward the reply on default *)
          if %UNEW = 0
           then %OUT := 3                    (* Dublicated reply, discard it *)
           else if (%S20.TTL = 0)
                then %OUT := 2;            (* Discard the reply with TTL = 0 *)

(* Forwarding the reply on the route to the source node, *)
(* with the use of the last non-zero attribute NDi by switching module *)
NET   T2000: S70/S2000;
TIME  T2000: %DELAY := UNIFRM(1, 1.00, 30.00);    (* Transm and propag time *)
(* Address of next node on route to source node is last nonzero attribute *)
TRANS T2000:      if %S2000.ND10 > 0
             then begin i := %S2000.ND10; %S2000.ND10 := 0 end
             else if %S2000.ND9 > 0
             then begin i := %S2000.ND9; %S2000.ND9 := 0 end
             else if %S2000.ND8 > 0
             then begin i := %S2000.ND8; %S2000.ND8 := 0 end
             else if %S2000.ND7 > 0
             then begin i := %S2000.ND7; %S2000.ND7 := 0 end
             else if %S2000.ND6 > 0
             then begin i := %S2000.ND6; %S2000.ND6 := 0 end
             else if %S2000.ND5 > 0
             then begin i := %S2000.ND5; %S2000.ND5 := 0 end
             else if %S2000.ND4 > 0
             then begin i := %S2000.ND4; %S2000.ND4 := 0 end
             else if %S2000.ND3 > 0
             then begin i := %S2000.ND3; %S2000.ND3 := 0 end
             else if %S2000.ND2 > 0
             then begin i := %S2000.ND2; %S2000.ND2 := 0 end
             else if %S2000.ND1 > 0
             then begin i := %S2000.ND1; %S2000.ND1 := 0 end;
             %S2000.NEXT := i;
WRITELN('Nod:',%SELF:2, %S70.ND1:5,%S70.ND2:5,%S70.ND3:5,%S70.ND4:5,%S70.ND5:5);

(* Handling  of the received mcast request by simple, not source, node *)
NET   T2: Q1/S9;
TRANS T2: if %S9.MID = %MLST (* Coompare received ID with ID stored earlier *)
         then %MNEW := 0                            (* Duplicated request *)
         else begin                    (* New request passed to this node *)
               %MNEW := 1;
              %MLST := %S9.MID;
```

```
              end;
          (* WRITELN('Nd ', %SELF, ' from ', %S9.SNDR, ' MID = ', %S9.MID); *)

(* Decision on the received request *)
NET   X3: S9/S7, S8, S10, S11;
CONTR X3: %OUT := 1;                   (* Retransmit the request on default *)
          if %MNEW = 0
          then %OUT := 3                  (* Dublicated request, discard it *)
          else BEGIN                               (* Delivered new request *)
               if (%SELF <= %NSRV)                 (* This is a server node *)
               then %OUT := 4
               else if (%S9.TTL = 0)
                    then %OUT := 2      (* Discard the message with TTL = 0 *)
               END;

(* Generation of a ucast reply by the server *)
NET   T11: S11/S19;
TRANS T11: WRITE('SRV',%SELF:5,', MID and HOPS:',%S11.MID:5, %S11.HOPS:5);
      WRITELN('Path:',%S11.ND1:3,%S11.ND2:3,%S11.ND3:3,%S11.ND4:3,%S11.ND5:3);
           %S19.TTL  := %TTLM;            (* Max number of hops from server *)
           %S19.HOPS := 0;                   (* Counter of hops from server *)
           %S19.MTYP := 3;                        (* Message is ucast reply *)
            (* Find the last nonzero attribute in list ND1, ..., ND10 *)
             if %S19.ND10 > 0
             then begin i := %S19.ND10; %S19.ND10 := 0 end
             else if %S19.ND9 > 0
             then begin i := %S19.ND9; %S19.ND9 := 0 end
             else if %S19.ND8 > 0
             then begin i := %S19.ND8; %S19.ND8 := 0 end
             else if %S19.ND7 > 0
             then begin i := %S19.ND7; %S19.ND7 := 0 end
             else if %S19.ND6 > 0
             then begin i := %S19.ND6; %S19.ND6 := 0 end
             else if %S19.ND5 > 0
             then begin i := %S19.ND5; %S19.ND5 := 0 end
             else if %S19.ND4 > 0
             then begin i := %S19.ND4; %S19.ND4 := 0 end
             else if %S19.ND3 > 0
             then begin i := %S19.ND3; %S19.ND3 := 0 end
             else if %S19.ND2 > 0
             then begin i := %S19.ND2; %S19.ND2 := 0 end
             else if %S19.ND1 > 0
             then begin i := %S19.ND1; %S19.ND1 := 0 end;
             %S19.NEXT := i;    (* ... and use it as first address to reply *)

TIME  T11: %DELAY := UNIFRM(1, 1.00, 30.00);    (* Transmit and propag time *)

(* Retransmitting  the received  request by a simple node *)
NET   T1000: S7/S1000;
TIME  T1000: %DELAY := UNIFRM(1, 1.00, 30.00);    (* Transm and propag time *)

(* Node output *)
(* Store ID of this node in first free attribute ND1 or ND2 or ... ND10 *)
(* This is done only for mcast requests *)
NET   Y1000: S97, S1000, S19, S2000/S100;
TRANS Y1000: %S100.SNDR := %SELF;                    (* Sender is this node *)
             if (%IN = 1) OR (%IN = 2)
             then
             BEGIN
             if %S100.ND1 = 0
             then %S100.ND1 := %SELF
             else if %S100.ND2 = 0
             then %S100.ND2 := %SELF
             else if %S100.ND3 = 0
             then %S100.ND3 := %SELF
             else if %S100.ND4 = 0
             then %S100.ND4 := %SELF
             else if %S100.ND5 = 0
             then %S100.ND5 := %SELF
             else if %S100.ND6 = 0
             then %S100.ND6 := %SELF
             else if %S100.ND7 = 0
             then %S100.ND7 := %SELF
             else if %S100.ND8 = 0
             then %S100.ND8 := %SELF
             else if %S100.ND9 = 0
             then %S100.ND9 := %SELF
```

```
             else if %S100.ND10 = 0
             then %S100.ND10 := %SELF
             END;

(* Common absorber of unnecessary tokens *)
NET   Y2000: S3, S6, S8, S10, S16, S17, S31, S80, S90;
SEGEND.
```

Appendix C: The file of parameters for a specific combination of model parameters.

```
(**********************************************************************)
(*                              MCL statements                         *)
(*                      for the  model of anycasting                   *)
(*                       in  mobile ad-hoc network                     *)
(*                                                                     *)
(*                File  SWITCH.JZP        Date 10 January  2011         *)
(**********************************************************************)

FOR SEGMENT SWITCH;

SET NODS /50/;                (* The number of all nodes in the given area *)
SET NSRV /5/;                   (* The number of servers in the given area *)
SET XMIN /0.0/;                        (* Minimal X-coordinate of the area *)
SET XMAX /500.0/;                      (* Maximal X-coordinate of the area *)
SET YMIN /0.0/;                        (* Minimal Y-coordinate of the area *)
SET YMAX /500.0/;                      (* Maximal Y-coordinate of the area *)
(* DX: 0.1 (3.6 km/h), 0.2 (7.2), 0.4 (14.4), 0.8 (28.8) and 1.6 (57.6)    *)
SET DX /1.389/;                      (* Max step along X during step time, m *)
SET DY /1.389/;                      (* Max step along Y during step time, m *)
SET DT /100.0/;                (* Step time to recompute node position, ms *)
SET DMAX /30.0/;                                  (* Transmission radius, m *)
SET PDIR /0.0/;        (* Probability to change direction of node movement *)
MARK S4;
MARK S1000;
STATISTICS Y1000, X2000, T8, T1000, T1001,S13,S50;

FOR SEGMENTS NOD1,NOD2,NOD3,NOD4,NOD5,NOD6,NOD7,NOD8,NOD9,
NOD10,NOD11,NOD12,NOD13,NOD14,NOD15,NOD16,NOD17,NOD18,NOD19,NOD20,
NOD21,NOD22,NOD23,NOD24,NOD25;

SET TON /10000.0/;              (* Mean time a link in ON state, ms, fixed *)

(* TOFF values for different values of link availability = TON/(TON+TOFF): *)
(* 190000 (l = 0.05), 90000 (0.1),56667 (0.15), 40000 (0.2), 30000 (0.25)  *)
(* 23333 (0.3), 18570 (0.35), 15000 (0.4), 12222 (0.45), 10000 (0.5),      *)
(* 6667 (0.6), 4286 (0.7), 2500 (0.8), 1111 (0.9)                          *)

SET TOFF /4286.0/;    (* Mean time a link in OFF state, ms, for l = 0.05 *)
SET TREC /2000.0/;                 (* Interval of checking of link states *)
(* Parameters of uniform distribution of distance to very near nodes, m *)
SET RMIN /5.0/;
SET RMAX /10.0/;
STATISTICS T1000,T2,T11,T20, T2000;
STATISTICS S100,S2,S8,S10,S31,S80,S90;


FOR SEGMENTS NOD26,NOD27,NOD28,NOD29,NOD30,NOD31,NOD32,NOD33,NOD34,
NOD35,NOD36,NOD37,NOD38,NOD39,NOD40,NOD41,NOD42,NOD43,NOD44,NOD45,
NOD46,NOD47,NOD48,NOD49,NOD50;

SET TON /10000.0/;              (* Mean time a link in ON state, ms, fixed *)
SET TOFF /4286.0/;    (* Mean time a link in OFF state, ms, for l = 0.05 *)
SET TREC /2000.0/;                 (* Interval of checking of link states *)
(* Parameters of uniform distribution of distance to very near nodes, m *)
SET RMIN /5.0/;
SET RMAX /10.0/;
STATISTICS T1000,T2,T11,T20, T2000;
STATISTICS S100,S2,S8,S10,S31,S80,S90;
```

```
FOR SEGMENT NOD50;                     (* This is only for the source node *)
SET T0 /500.0/;     (* Time interval (ms)  to send requests by source node *)
STATISTICS X2, T5;
STATISTICS S3,S16,S17,S31,S94,S99;
HISTO 1 (HOPS,0,1,7) S99;   (* Hops of replies from servers to source node *)
HISTO 2 (TIM,0,1,7) S15;   (* Avg. response time from server to source node*)
```

Appendix D: Average Values and Confidence Interval of the Investigated Performance Metrics

Average values and 95% confidence interval of investigated performance metrics are provided here.

Table D.1: Average values and 95% confidence intervals of the performance metrics for maximum node speed *V*=5 km/h, *l*=0.5 and *p*=0.0

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.051 ± 0.006	0.161 ± 0.162	0.200 ± 0.109	0.545 ± 0.230	0.647 ± 0.166	0.738 ± 0.049	0.874 ± 0.100
Average number of hops	1.003 ± 0.008	1.311 ± 0.300	1.633 ± 0.041	1.830 ± 0.322	1.761 ± 0.224	1.605 ± 0.148	1.351 ± 0.181
Relative traffic	1.340 ± 0.152	3.313 ± 0.438	6.144 ± 2.073	24.626 ± 5.480	43.930 ± 3.336	51.008 ± 0.726	52.541 ± 1.016
Average response time, ms	17.272 ± 5.691	24.704 ± 9.515	33.996 ± 2.049	38.424 ± 10.112	34.999 ± 5.984	28.668 ± 2.297	20.363 ± 4.807
Duplicate ratio	0.000 ± 0.000	0.029 ± 0.055	0.167 ± 0.065	0.293 ± 0.080	0.582 ± 0.233	0.709 ± 0.171	1.006 ± 0.328

Table D.2: Average values and 95% confidence intervals of the performance metrics for maximum node speed V=5 km/h, l=0.5 and p=0.3

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.000 ± 0.000	0.342 ± 0.377	0.556 ± 0.310	0.724 ± 0.365	0.778 ± 0.367	0.856 ± 0.377	0.870 ± 0.165
Average number of hops	0.000 ± 0.000	1.161 ± 1.325	1.283 ± 0.269	1.733 ± 0.837	1.425 ± 0.805	1.421 ± 1.030	1.367 ± 0.357
Relative traffic	1.316 ± 0.320	5.116 ± 3.594	16.542 ± 2.710	31.813 ± 7.746	45.296 ± 3.255	51.629 ± 1.523	53.039 ± 0.891
Average response time, ms	0.000 ± 0.000	29.249 ± 16.526	37.941 ± 9.289	47.425 ± 24.004	24.166 ± 23.915	22.218 ± 28.392	25.641 ± 6.051
Duplicate ratio	0.000 ± 0.000	0.045 ± 0.090	0.373 ± 0.246	0.384 ± 0.772	0.487 ± 1.039	0.823 ± 1.238	0.996 ± 1.054

Table D.3: Average values and 95% confidence intervals of the performance metrics for maximum node speed V=5 km/h, l=0.5 and p=0.5

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.000 ± 0.000	0.171 ± 0.314	0.449 ± 0.541	0.548 ± 0.632	0.809 ± 0.232	0.894 ± 0.133	0.952 ± 0.035
Average number of hops	0.000 ± 0.000	0.482 ± 1.532	2.000 ± 1.505	2.252 ± 1.782	1.620 ± 0.769	1.369 ± 0.495	1.156 ± 0.167
Relative traffic	1.108 ± 0.316	4.899 ± 3.939	11.931 ± 8.364	28.344 ± 12.538	48.511 ± 3.601	52.010 ± 1.943	51.799 ± 0.501
Average response time, ms	1.705 ± 5.420	10.408 ± 33.099	23.526 ± 30.290	31.573 ± 39.391	29.342 ± 21.090	21.568 ± 14.063	14.807 ± 4.366
Duplicate ratio	0.000 ± 0.000	0.016 ± 0.052	0.040 ± 0.073	0.410 ± 0.626	0.513 ± 0.438	0.728 ± 0.368	1.402 ± 0.529

Table D.4: Average values and 95% confidence intervals of the performance metrics for maximum node speed V=5 km/h, l=0.5 and p=0.7

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.005 ± 0.017	0.295 ± 0.606	0.368 ± 0.486	0.727 ± 0.379	0.801 ± 0.385	0.947 ± 0.053	0.937 ± 0.059
Average number of hops	0.493 ± 1.567	0.578 ± 1.073	1.646 ± 2.033	1.922 ± 1.233	1.550 ± 0.985	1.165 ± 0.125	1.184 ± 0.277
Relative traffic	1.229 ± 0.443	5.043 ± 4.554	13.310 ± 9.247	31.135 ± 4.034	46.687 6.235	51.201 ± 1.523	52.849 ± 0.902
Average response time, ms	11.744 ± 37.347	17.032 ± 34.199	53.211 ± 33.018	41.253 ± 36.617	39.042 ± 31.166	33.759 ± 50.368	32.975 ± 22.183
Duplicate ratio	0.000 ± 0.000	0.119 ± 0.306	0.137 ± 0.311	0.288 ± 0.412	0.438 ± 0.575	1.185 ± 0.729	1.007 ± 1.060

Table D.5: Average values and 95% confidence intervals of the performance metrics for maximum node speed V=5 km/h, l=0.5 and p=0.9

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.000 ± 0.000	0.074 ± 0.225	0.257 ± 0.604	0.540 ± 0.625	0.737 ± 0.315	0.904 ± 0.156	0.942 ± 0.072
Average number of hops	0.000 ± 0.000	1.100 ± 2.026	2.311 ± 1.210	2.376 ± 1.959	1.730 ± 1.127	1.345 ± 0.323	1.201 ± 0.181
Relative traffic	1.373 ± 0.432	3.998 ± 3.521	7.158 ± 10.290	29.686 ± 15.568	46.563 ± 4.892	51.433 ± 1.073	52.521 ± 0.899
Average response time, ms	2.630 ± 8.363	43.469 ± 50.644	55.934 ± 38.091	53.827 ± 57.915	32.523 ± 30.928	22.393 ± 11.541	16.297 ± 6.316
Duplicate ratio	0.000 ± 0.000	0.050 ± 0.114	0.124 ± 0.394	0.375 ± 0.625	0.655 ± 0.687	0.731 ± 0.812	1.376 ± 0.802

Table D.6: Average values and 95% confidence intervals of the performance metrics for maximum node speed *V*=5 km/h, *l*=0.5 and *p*=1.0

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.002 ± 0.006	0.229 ± 0.353	0.421 ± 0.445	0.750 ± 0.311	0.891 ± 0.145	0.935 ± 0.086	0.972 ± 0.022
Average number of hops	0.500 ± 1.590	1.164 ± 1.356	1.342 ± 0.194	1.805 ± 0.977	1.378 ± 0.251	1.190 ± 0.168	1.116 ± 0.115
Relative traffic	1.132 ± 0.421	3.637 ± 1.818	14.811 ± 6.785	35.500 ± 9.111	48.172 ± 3.701	51.579 ± 1.661	51.851 ± 1.542
Average response time, ms	11.336 ± 36.049	23.952 ± 30.047	25.009 ± 7.423	36.180 ± 28.759	23.711 ± 8.910	16.550 ± 6.241	13.591 ± 3.683
Duplicate ratio	0.000 ± 0.000	0.025 ± 0.081	0.066 ± 0.196	0.415 ± 0.551	0.910 ± 0.722	1.322 ± 0.709	1.699 ± 0.364

Table D.7: Average values and 95% confidence intervals of the performance metrics for maximum node speed *V*=5km/h and *l*=0.7

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.037 ± 0.042	0.221 ± 0.242	0.512 ± 0.300	0.821 ± 0.038	0.922 ± 0.078	0.956 ± 0.009	0.916 ± 0.082
Average number of hops	0.782 ± 0.830	1.851 ± 1.372	1.958 ± 0.676	1.586 ± 0.132	1.372 ± 0.276	1.291 ± 0.122	1.522 ± 0.472
Relative traffic	1.509 ± 0.190	4.130 ± 1.079	13.424 ± 7.049	31.556 ± 5.563	51.204 ± 1.726	53.756 ± 0.624	53.759 ± 0.902
Average response time, ms	12.782 ± 13.615	40.915 ± 41.980	43.183 ± 20.606	29.667 ± 2.259	21.055 ± 7.144	18.914 ± 2.352	22.594 ± 10.685
Duplicate ratio	0.000 ± 0.000	0.141 ± 0.210	0.179 ± 0.201	0.726 ± 0.172	1.423 ± 0.333	1.440 ± 0.234	1.376 ± 0.564

Table D.8: Average values and 95% confidence intervals of the performance metrics for maximum node speed *V*=30 km/h and *l*=0.7

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.039 ± 0.026	0.174 ± 0.080	0.531 ± 0.131	0.722 ± 0.069	0.869 ± 0.063	0.900 ± 0.066	0.934 ± 0.021
Average number of hops	1.137 ± 0.154	1.426 ± 0.173	1.735 ± 0.170	1.866 ± 0.165	1.604 ± 0.043	1.520 ± 0.151	1.396 ± 0.079
Relative traffic	1.480 ± 0.016	3.455 ± 0.607	13.280 ± 2.985	35.200 ± 3.916	50.088 ± 2.149	53.857 ± 0.512	53.832 ± 0.219
Average response time, ms	20.051 ± 6.171	28.249 ± 4.441	36.275 ± 4.749	38.156 ± 4.885	28.625 ± 1.500	24.972 ± 3.421	20.569 ± 2.099
Duplicate ratio	0.000 ± 0.000	0.084 ± 0.104	0.254 ± 0.022	0.523 ± 0.220	0.910 ± 0.070	1.146 ± 0.109	1.364 ± 0.243

Table D.9: Average values and 95% confidence intervals of the performance metrics for maximum node speed *V*=50 km/h and *l*=0.7

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.026 ± 0.010	0.200 ± 0.045	0.470 ± 0.164	0.738 ± 0.075	0.865 ± 0.060	0.924 ± 0.011	0.945 ± 0.023
Average number of hops	1.132 ± 0.023	1.435 ± 0.243	1.793 ± 0.200	1.836 ± 0.421	1.628 ± 0.100	1.428 ± 0.068	1.316 ± 0.067
Relative traffic	1.393 ± 0.069	3.513 ± 0.891	12.646 ± 2.166	33.819 ± 0.791	49.965 ± 1.012	53.535 ± 0.303	53.740 ± 0.159
Average response time, ms	19.753 ± 2.950	28.801 ± 7.610	38.339 ± 5.990	37.226 ± 11.247	29.018 ± 3.205	22.298 ± 1.593	18.123 ± 1.464
Duplicate ratio	0.017 ± 0.032	0.078 ± 0.053	0.226 ± 0.122	0.563 ± 0.144	0.900 ± 0.187	1.222 ± 0.086	1.496 ± 0.077

Table D.10: Average values and 95% confidence intervals of the performance metrics for TTL=4, maximum node speed V=5km/h and l=0.05

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.016 ± 0.033	0.015 ± 0.017	0.041 ± 0.029	0.047 ± 0.054	0.072 ± 0.031	0.125 ± 0.011	0.116 ± 0.026
Average number of hops	1.000 ± 0.000	1.000 ± 0.000	1.069 ± 0.117	1.006 ± 0.013	1.040 ± 0.102	1.038 ± 0.060	1.034 ± 0.058
Relative traffic	1.053 ± 0.032	1.168 ± 0.059	1.379 ± 0.144	1.708 ± 0.369	1.975 ± 0.215	2.609 ± 0.335	3.288 ± 0.609
Average response time, ms	15.347 ± 2.832	16.158 ± 0.918	17.983 ± 4.129	14.270 ± 4.800	16.096 ± 3.130	16.308 ± 3.646	15.629 ± 3.449
Duplicate ratio	0.000 ± 0.000	0.222 ± 0.707	0.034 ± 0.063	0.045 ± 0.113	0.120 ± 0.237	0.086 ± 0.077	0.175 ± 0.189

Table D.11: Average values and 95% confidence intervals of the performance metrics for TTL=4, maximum node speed V=5 km/h and l=0.1

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.009 ± 0.014	0.025 ± 0.017	0.035 ± 0.038	0.105 ± 0.068	0.122 ± 0.069	0.129 ± 0.068	0.261 ± 0.073
Average number of hops	1.000 ± 0.000	1.178 ± 0.302	1.098 ± 0.111	1.075 ± 0.096	1.136 ± 0.084	1.236 ± 0.231	1.136 ± 0.248
Relative traffic	1.133 ± 0.016	1.260 ± 0.168	1.745 ± 0.331	2.649 ± 0.687	3.815 ± 0.913	5.156 ± 1.355	8.211 ± 1.833
Average response time, ms	15.807 ± 2.263	21.475 ± 7.421	19.337 ± 2.932	17.238 ± 2.704	19.255 ± 3.519	21.814 ± 7.219	18.653 ± 7.527
Duplicate ratio	0.000 ± 0.000	0.000 ± 0.000	0.029 ± 0.091	0.030 ± 0.048	0.062 ± 0.170	0.127 ± 0.299	0.142 ± 0.085

Table D.12: Average values and 95% confidence intervals of the performance metrics for TTL=4, maximum node speed *V*=5 km/h and *l*=0.3

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.034 ± 0.047	0.060 ± 0.079	0.135 ± 0.147	0.212 ± 0.232	0.501 ± 0.175	0.469 ± 0.177	0.527 ± 0.221
Average number of hops	1.029 ± 0.053	1.165 ± 0.238	1.303 ± 0.022	1.742 ± 0.915	1.335 ± 0.276	1.469 ± 0.312	1.499 ± 0.299
Relative traffic	1.201 ± 0.065	2.523 ± 0.147	4.212 ± 1.107	7.319 ± 2.677	14.439 ± 4.511	19.058 ± 5.315	26.818 ± 4.259
Average response time, ms	15.903 ± 2.932	20.925 ± 6.492	24.977 ± 0.389	38.373 ± 31.276	23.951 ± 7.701	27.976 ± 9.579	27.781 ± 8.949
Duplicate ratio	0.000 ± 0.000	0.000 ± 0.000	0.060 ± 0.076	0.058 ± 0.081	0.370 ± 0.085	0.336 ± 0.131	0.425 ± 0.312

Table D.13: Average values and 95% confidence intervals of the performance metrics for TTL=4, maximum node speed *V*=5 km/h and *l*=0.5.

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.044 ± 0.048	0.098 ± 0.037	0.304 ± 0.094	0.463 ± 0.059	0.667 ± 0.192	0.763 ± 0.158	0.890 ± 0.103
Average number of hops	1.101 ± 0.137	1.301 ± 0.638	1.353 ± 0.208	1.737 ± 0.376	1.570 ± 0.391	1.533 ± 0.290	1.249 ± 0.152
Relative traffic	1.402 ± 0.192	2.707 ± 0.670	5.354 ± 2.820	14.259 ± 2.170	23.058 ± 8.363	37.590 ± 3.959	40.841 ± 4.084
Average response time, ms	17.865 ± 5.841	25.434 ± 18.500	25.694 ± 5.468	35.953 ± 9.551	29.825 ± 11.347	26.598 ± 7.449	18.093 ± 3.353
Duplicate ratio	0.051 ± 0.162	0.015 ± 0.032	0.227 ± 0.215	0.243 ± 0.167	0.573 ± 0.349	0.729 ± 0.226	1.084 ± 0.159

Table D.14: Average values and 95% confidence intervals of the performance metrics for TTL=4, maximum node speed *V*=5 km/h and *l*=0.7

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.070 ± 0.089	0.125 ± 0.093	0.584 ± 0.194	0.671 ± 0.131	0.857 ± 0.045	0.943 ± 0.060	0.905 ± 0.128
Average number of hops	1.026 ± 0.061	1.718 ± 0.387	1.544 ± 0.370	1.705 ± 0.204	1.508 ± 0.283	1.295 ± 0.342	1.445 ± 0.514
Relative traffic	1.352 ± 0.229	3.468 ± 0.444	8.737 ± 1.476	18.815 ± 5.727	29.379 ± 10.164	38.779 ± 7.450	45.315 ± 1.297
Average response time, ms	15.884 ± 2.220	37.700 ± 12.850	30.800 ± 11.254	33.600 ± 5.667	25.533 ± 7.155	18.936 ± 8.308	21.079 ± 11.477
Duplicate ratio	0.021 ± 0.050	0.013 ± 0.041	0.231 ± 0.136	0.621 ± 0.287	1.015 ± 0.193	1.526 ± 0.416	1.340 ± 0.556

Table D.15: Average values and 95% confidence intervals of the performance metrics for TTL=7, maximum node speed *V*=5km/h and *l*=0.05

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.023 ± 0.031	0.012 ± 0.020	0.046 ± 0.028	0.053 ± 0.018	0.052 ± 0.056	0.067 ± 0.033	0.120 ± 0.040
Average number of hops	1 ± 0	1 ± 0	1.016 ± 0.051	1.033 ± 0.041	1.099 ± 0.136	1.131 ± 0.069	1.088 ± 0.078
Relative traffic	1.057 ± 0.029	1.147 ± 0.035	1.426 ± 0.115	1.632 ± 0.164	2.274 ± 0.497	2.954 ± 0.465	3.909 ± 0.176
Average response time, ms	15.510 ± 3.273	16.144 ± 0.879	15.929 ± 1.959	16.285 ± 2.224	18.065 ± 3.099	19.192 ± 3.585	17.870 ± 2.696
Duplicate ratio	0 ± 0	0.022 ± 0.0706	0.023 ± 0.074	0.005 ± 0.018	0.021 ± 0.021	0.094 ± 0.169	0.061 ± 0.111

Table D.16: Average values and 95% confidence intervals of the performance metrics for TTL=7, maximum node speed V=5 km/h and l=0.1

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.004 ± 0.005	0.043 ± 0.058	0.051 ± 0.048	0.097 ± 0.030	0.114 ± 0.138	0.103 ± 0.085	0.294 ± 0.155
Average number of hops	1 ± 0	1.017 ± 0.056	1.141 ± 0.109	1.131 ± 0.108	1.102 ± 0.132	1.175 ± 0.156	1.121 ± 0.151
Relative traffic	1.092 ± 0.062	1.268 ± 0.170	1.916 ± 0.506	2.476 ± 0.673	4.092 ± 1.282	6.781 ± 3.024	10.629 ± 1.332
Average response time, ms	16.783 ± 2.636	16.315 ± 2.570	19.503 ± 3.851	18.414 ± 2.986	18.832 ± 7.210	20.209 ± 5.966	18.172 ± 4.898
Duplicate ratio	0 ± 0	0.023 ± 0.074	0.013 ± 0.023	0.062 ± 0.118	0.043 ± 0.130	0.133 ± 0.236	0.182 ± 0.184

Table D.17: Average values and 95% confidence intervals of the performance metrics for TTL=7, maximum node speed V=5 km/h and l=0.3

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.034 ± 0.046	0.051 ± 0.036	0.213 ± 0.109	0.312 ± 0.103	0.336 ± 0.145	0.643 ± 0.159	0.666 ± 0.119
Average number of hops	1.028 ± 0.052	1.146 ± 0.223	1.248 ± 0.188	1.415 ± 0.262	1.591 ± 0.377	1.282 ± 0.110	1.348 ± 0.151
Relative traffic	1.201 ± 0.065	1.862 ± 0.518	4.391 ± 1.635	14.127 ± 3.783	23.869 ± 3.160	37.771 ± 1.016	44.557 ± 2.193
Average response time, ms	15.902 ± 2.931	20.373 ± 5.520	23.089 ± 5.612	27.693 ± 8.904	31.808 ± 9.260	22.022 ± 3.655	23.087 ± 4.652
Duplicate ratio	0 ± 0	0.023 ± 0.074	0.110 ± 0.143	0.086 ± 0.049	0.180 ± 0.263	0.495 ± 0.221	0.520 ± 0.189

Table D.18: Average values and 95% confidence intervals of the performance metrics for TTL=7, maximum node speed *V*=5 km/h and *l*=0.5

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.030 ± 0.031	0.155 ± 0.105	0.270 ± 0.159	0.565 ± 0.167	0.703 ± 0.254	0.762 ± 0.197	0.881 ± 0.089
Average number of hops	1.067 ± 0.151	1.165 ± 0.171	1.753 ± 0.598	1.716 ± 0.351	1.499 ± 0.234	1.564 ± 0.505	1.297 ± 0.158
Relative traffic	1.347 ± 0.224	2.999 ± 1.504	9.407 ± 1.190	22.768 ± 3.165	38.811 ± 8.114	50.139 ± 2.052	51.997 ± 0.704
Average response time, ms	17.496 ± 5.639	20.443 ± 6.122	37.163 ± 16.456	35.464 ± 10.964	27.894 ± 6.994	27.970 ± 12.577	19.090 ± 4.566
Duplicate ratio	0.000 ± 0.000	0.102 ± 0.210	0.099 ± 0.137	0.280 ± 0.333	0.635 ± 0.167	0.559 ± 0.220	1.056 ± 0.594

Table D.19: Average values and 95% confidence intervals of the performance metrics for TTL=7, maximum node speed *V*=5 km/h and *l*=0.7

Metric	Transmission radius, m						
	30	60	90	120	150	180	210
Response ratio	0.037 ± 0.042	0.221 ± 0.242	0.512 ± 0.300	0.821 ± 0.038	0.922 ± 0.078	0.956 ± 0.009	0.916 ± 0.082
Average number of hops	0.782 ± 0.830	1.851 ± 1.372	1.958 ± 0.676	1.586 ± 0.132	1.372 ± 0.276	1.291 ± 0.122	1.522 ± 0.472
Relative traffic	1.509 ± 0.190	4.130 ± 1.079	13.424 ± 7.049	31.556 ± 5.563	51.204 ± 1.726	53.756 ± 0.624	53.759 ± 0.902
Average response time, ms	12.782 ± 13.615	40.915 ± 41.980	43.183 ± 20.606	29.667 ± 2.259	21.055 ± 7.144	18.914 ± 2.352	22.594 ± 10.685
Duplicate ratio	0.000 ± 0.000	0.141 ± 0.210	0.179 ± 0.201	0.726 ± 0.172	1.423 ± 0.333	1.440 ± 0.234	1.376 ± 0.564

Table D.20: Average values and 95% confidence intervals of the performance metrics for different delays versus application data size

Delay, ms	Performance metric	Application data size, bytes				
		100	1000	2000	4000	8000
10	Response ratio	0.827 ± 0.121	0.689 ± 0.020	0.300 ± 0.026	0.373 ± 0.290	0.096 ± 0.028
	Average response time, ms	5.207 ± 12.146	93.528 ± 46.011	33.679 ± 0.874	82.700 ± 8.353	146.556 ± 6.533
50	Response ratio	0.904 ± 0.072	0.774 ± 0.060	0.523 ± 0.139	0.496 ± 0.452	0.239 ± 0.057
	Average response time, ms	3.987 ± 0.206	21.378 ± 2.509	16.321 ± 0.740	31.754 ± 0.686	104.618 ± 3.322
100	Response ratio	0.896 ± 0.019	0.809 ± 0.077	0.468 ± 0.069	0.615 ± 0.496	0.405 ± 0.324
	Average response time, ms	3.933 ± 1.190	20.030 ± 2.544	16.557 ± 1.172	36.875 ± 6.071	78.371 ± 0.640
500	Response ratio	0.935 ± 0.020	0.909 ± 0.008	0.556 ± 0.271	0.831 ± 0.119	0.792 ± 0.111
	Average response time, ms	4.076 ± 0.441	19.949 ± 0.509	18.174 ± 1.163	32.501 ± 1.176	78.602 ± 0.476
1000	Response ratio	0.873 ± 1.435	0.761 ± 0.420	0.743 ± 0.075	0.519 ± 0.635	0.190 ± 0.247
	Average response time, ms	6.272 ± 0.219	21.612 ± 8.060	17.005 ± 2.834	34.999 ± 9.798	77.662 ± 7.200

Table D.21: Average values and 95% confidence intervals of the performance metrics for different heights versus application data size

Height, cm	Performance metric	Application data size, bytes				
		100	1000	2000	4000	8000
0	Response ratio	0.875 ± 0.023	5.308 ± 3.224	0.593 ± 0.257	16.110 ± 1.245	0.264 ± 0.479
	Average response time, ms	5.308 ± 3.224	0.896 ± 0.019	16.110 ± 1.245	0.468 ± 0.069	95.288 ± 68.232
50	Response ratio	0.896 ± 0.019	3.933 ± 1.190	0.468 ± 0.069	16.557 ± 1.172	0.405 ± 0.324
	Average response time, ms	3.933 ± 1.190	0.908 ± 0.063	16.557 ± 1.172	0.814 ± 0.080	78.371 ± 0.640
100	Response ratio	0.908 ± 0.063	3.159 ± 1.602	0.814 ± 0.080	16.094 ± 0.257	0.617 ± 0.080
	Average response time, ms	3.159 ± 1.602	19.675 ± 2.099	16.094 ± 0.257	33.629 ± 8.186	78.013 ± 0.029