Ministry of Higher Education and Scientific Research
University of Saad Dahleb - BLIDA 1
Faculty of Sciences
Department of Informatics

A Simulation Study of Probabilistic Broadcast in MANETs

Report submitted for the fulfillment of the Master degree
Domain: MI
Affiliation: Informatics
Specialization: Computer systems and networks

ILYES OUSSAID AND **ABDESSAMED BOUAZZA**

Supervisor: Mr. Ould-Khaoua Mohamed

Jury:
President: Mr. Kameche Abdellah.
Examiner: Mrs. Nasri Ahlem

Academic year: 2021/2022

Acknowledgment

We would like to greatly thank our supervisor: Prof. Mohamed Ould-Khaoua, for his invaluable advice and guidance during the course of this project.

Our sincere thanks also go to the members of the jury for their interest in our project by agreeing to examine our work and enrich it with their suggestions and recommendations.

Our gratitude is due to our instructors who taught us during our License and Master degrees.

Finally, we would like to thank everyone who participated either directly or indirectly in the achievement of this work.

Abstract

Probabilistic flooding schemes use probabilities to help a node decide whether it rebroadcasts its transiting packet or not. Even though probabilistic schemes have been around for a while, only a few research attempts have analyzed the performance behavior in a Mobile Ad Hoc Network (MANET) environment. Our present study investigates the effects of mobility, injected traffic load and network density on the performance of probabilistic flooding in MANETs. In order to reach our goal, we have implemented probabilistic flooding in the widely used ns-3 discrete-event simulator. After extensive testing through our ns-3 simulator the collected statistics have revealed that the performance metrics including reachability and saved rebroadcast are heavily affected by the injected traffic load and node mobility. Moreover, our performance results have revealed that increasing network mobility results in a decreasing reachability and a high saved rebroadcast.

Résumé

Probabilistic flooding schemes utilisent des probabilités pour aider les nœuds à déterminer s'il faut ou non retransmettre un paquet en transit. Même si **Probabilistic flooding schemes** existent depuis un certain temps, seules quelques études ont examiné leur rendement dans un environnement de réseau mobile ad hoc (MANET). La présente étude vise à déterminer comment la mobilité, la charge de trafic injecté et la densité du réseau influent sur le rendement de **Probabilistic flooding** dans les MANET. Pour atteindre notre objectif, nous avons utilisé le simulateur à événements discrets ns-3, largement répandu, pour implémenter la méthode **Probabilistic flooding** Après des essais approfondis avec notre simulateur ns-3, les statistiques recueillies ont révélé que la charge de trafic injecté et la mobilité des nœuds ont une incidence importante sur les mesures de rendement comme **Reachability** et **Saved Rebroadcast**. De plus, nos résultats de performance montrent que l'augmentation de la mobilité du réseau entraîne une diminution de **Reachability** et une augmentation de **Saved rebroadcast**.

ملخص

تستخدم Probabilistic flooding schemes الاحتمالات لمساعدة العقد في تحديد ما إذا كان سيتم إعادة بث حزمة متنقلة أم لا. على الرغم من أن probabilistic schemes كانت موجودة منذ فترة ، إلا أن عددا قليلا فقط من الدراسات نظرت في كيفية أدائها في بيئة شبكة مخصصة متنقلة (Manet). تبحث الدراسة الحالية في كيفية تأثير حركية العقد وحمولة حركة البيانات وكثافة الشبكة على أداء probabilistic schemes في Manet. لتحقيق هدفنا ، استخدمنا محاكاة الحدث المنفصل NS-3 المستخدمة على نطاق واسع لتنفيذ probabilistic schemes. بعد اختبار مكثف مع جهاز محاكاة NS-3 لدينا ، كشفت الإحصاءات التي تم جمعها أن حمولة حركة البيانات و حركية العقد لها تأثير كبير على مقاييس الأداء مثل إمكانية الوصول ونسبة إعادة البث. علاوة على ذلك ، تظهر نتائج أدائنا أن زيادة حركية العقد في الشبكة تؤدي إلى انخفاض إمكانية الوصول و انخفاض إعادة البث.

Table of Contents

List of figures

List of tables

Chapter 1: General Introduction

Due to recent rapid advances in mobile computing devices and wireless technology, wireless communication is now one of the world's fastest growing technologies. Laptops, personal digital assistants (PDAs), and mobile phones have become lightweight and portable enough for mobile users to carry with them.

Wireless networks are divided into two types. The first, and most common category is a wireless network built on top of a wired network, resulting in a reliable infrastructure wireless network. The wireless nodes are also linked to a wired network and can function as bridges in this type of network. They are commonly referred to as base stations or access points. A cellular phone network is an example of this as a phone connects to a base-station. When the phone moves out of range of a base station, it performs a hand-off and switches to a nearby base station. The hand-off should be quick enough for network users to notice. [1]

The second category of wireless networks is Mobile Ad hoc Networks (or MANETs for short), which are formed of wireless devices that communicate without the need for a pre-existing network infrastructure such as that provided by access points. In MANETs, each mobile node serves not only as a host for applications, but also as a router, sending and receiving packets and forwarding packets for other nodes in the network. [1]

In contrast to one-hop station-based cellular networks, MANETs are also known as multi-hop packet radio networks. Self-configuring nature of MANETs makes them suitable for a wide range of civilian and military applications. Furthermore, MANETs could be useful in disaster relief, battlefields, temporary conference meetings, and uninhabited field searching. In such environments, where there is frequently little or no communication infrastructure or where the existing infrastructure is inconvenient to use, wireless mobile users could communicate by forming a MANET quickly. [2]

Broadcasting is a fundamental operation in MANETs in which a source node sends the same packet to all network nodes. In multi-hop MANETs, where all nodes may be beyond the source's transmission range, intermediate nodes may be required to assist in the broadcast operation by retransmitting the packet to other remote nodes in the network. In traditional broadcast settings, packet distribution frequently consumes valuable network resources such as node power and communication bandwidth. As a result, it is critical to carefully select the intermediate nodes in order to avoid transmission redundancy in the dissemination process.

The typical method for broadcasting a message to mobile network nodes is simple flooding. Flooding basically occurs when a source node broadcasts a packet to all network nodes. Once the packet is received for the first time by each of those receivers, it is eventually rebroadcast. A duplicate packet is simply dropped. This process continues until the packet is received by every node that can be reached. Even while this method has a high guarantee of reachability and it's easy to execute, it has a large transmission overhead and can cause harmful bandwidth congestion in what is referred to as a broadcast storm.

The goal of our project is to reduce the degrading effects of the broadcast storm problem. In order to achieve that, we have implemented a probabilistic mechanism to reduce the number of occurring rebroadcasts in the wireless medium.

In probabilistic broadcasting, when a node receives a packet, it retransmits it with a fixed probability p. Or alternatively, it may decide to drop the packet with probability 1-p. The net effect of this probabilistic broadcast scheme is the reduction of packet transmissions. This often results in the reduction in packet collisions and in turn reduces the deleterious impact of the broadcast storm problem.

In order to conduct our study, we have implemented probabilistic broadcasting in the well-known ns-3 discrete event simulator. We have then conducted extensive experiments to assess its performance under different traffic loads, network sizes and mobility patterns.

The rest of the report is organized as follows.

Chapter 2 provides a general background on MANETs, routing protocols.

Chapter 3 discuss researches in probabilistic broadcast scheme

Chapter 4 includes simulation advantages and implementation of the proposed work.

Chapter 5 includes result evaluation of proposed work.

Chapter 6 concludes this report and provides some possible direction for the continuation of our study.

Chapter 2: MANETs background

A mobile ad hoc network (MANET) is a network of wireless nodes joined together and communicating via wireless links with limited bandwidth. Each wireless node has three different modes of operation: sender, receiver, and router. When a node is a sender, it can deliver messages along a specific route to any defined destination node. It can take messages from other nodes as a receiver. The node can relay the packet to the destination or the following router along the route when it acts as a router. Each node can buffer packets for transmission if necessary. [3]

Because the nodes move randomly, an ad hoc network forms between them at any given time, leading to an arbitrary network structure. Any group of wireless users can dynamically construct MANETs without any preexisting infrastructure or configuration.

2.1. MANET characteristics

A MANET has a number of distinct features. It lacks a centralized infrastructure, to start. It differs from conventional mobile wireless networks in that it is not necessary to build base stations, access points, or servers before using the networks. Figure 2.1 shows the operation of a wireless infrastructure network.

The ad hoc network is instead decentralized, with all mobile nodes acting as routers and all wireless devices being connected to one another, as shown in Figure 2.2. This implies that the MANET is also a self-configuring network in which network operations like message delivery and topology discovery are carried out by the nodes themselves.

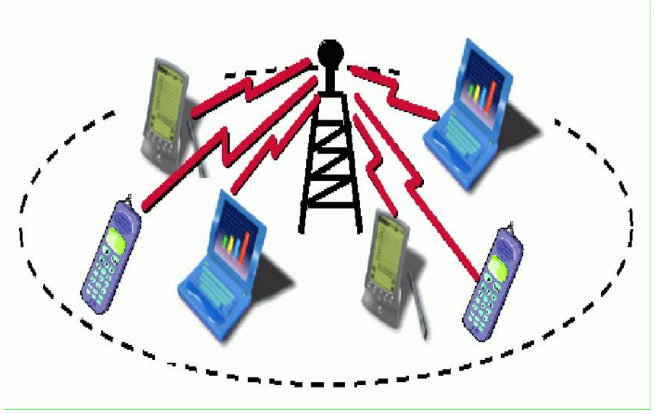


Figure 2-1 Infrastructure-based wireless network.

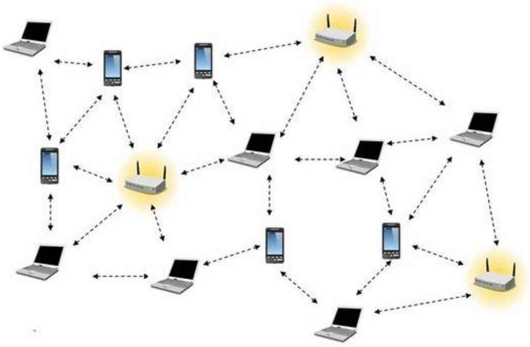
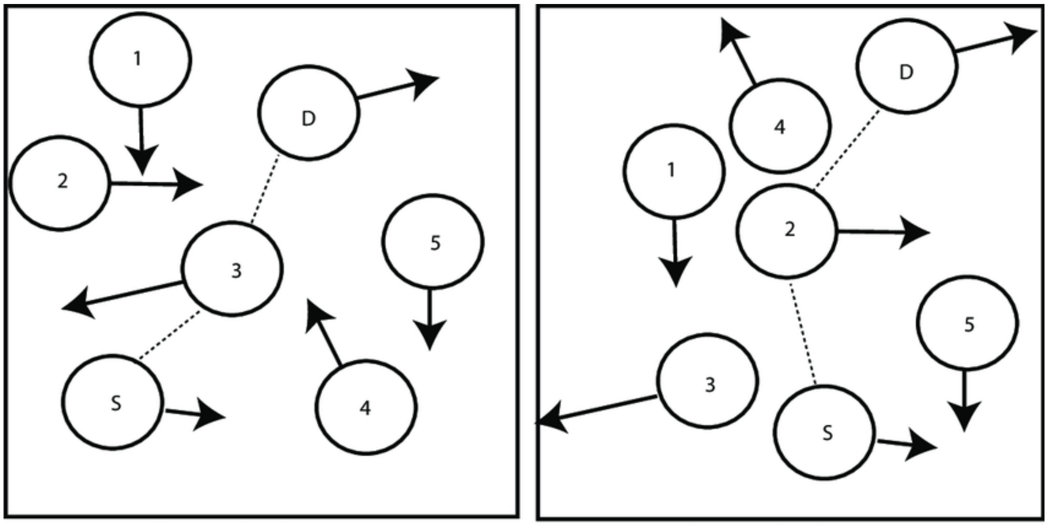


Figure 2-2 Ad hoc wireless network.

A dynamic topology is MANET's second feature. Nodes can move around at will, which leads to frequent and unpredictable changes in the network structure over time. Data packets are then automatically sent over the network's multi-hop paths using alternate routes that were discovered. MANETs achieve this through a variety of routing techniques. This is further elaborated in Annex A.



A. Communication at time t_0. B. Communication at $t_0+\Delta t$.

Figure 2-3 Dynamic Topological Changes in the MANET.

Thirdly, a MANET relies on variable-capacity lines that are bandwidth restricted. Compared to hard-wired links, wireless links have a much lower capacity. A MANET, as a result, has links with high bit error rates, unstable and asymmetric links, and relatively little bandwidth. In contrast, wired networks are characterized by links with high bandwidth, low bit error rates, and links that are stable and symmetric. Congestion is frequently the norm rather than the exception due to a low link capacity. [4]

Fourthly, energy-constrained procedures frequently constrain a [4]. This is due to the fact that its nodes are frequently battery-powered handheld devices. Power conservation is crucial in the design of a MANET system because mobile nodes depend on these finite energy sources.

There is also a lack of adequate physical security. In comparison to fixed-cable networks, mobile wireless networks are more vulnerable to physical security risks such as eavesdropping, interception, denial-of-service, and routing assaults [4]. Therefore, security measures must be used to lessen these dangers. Nodes prefer to transmit as seldom as feasible and with the least amount of power possible. As a result, there will be less chance of detection and interception. In contrast to centralized networks, the decentralized nature of network control will increase robustness against failure.

2.2. Limitations of MANETs

The requirement for dynamic ad hoc networking technologies is both present and future. Although very adaptable, this technology nonetheless has a number of limitations.

2.2.1. Throughput Drops with More Hops

The data packets can be sent directly between nodes that are within their transmission range. The data packets must be routed through a series of several hops, with the intermediate nodes serving as routers, when the node has to deliver data to a non-neighboring node. This suggests that more hops have been taken. The throughput will drop sharply as the number of hops rises. With reference to the four-hop network shown in Figure 2, this can be explained.

Figure 2-4 Multi-hop network.

A node cannot transmit and receive at the same time when connection 1-2 is active, hence link 2-3 cannot be active at the same time. Due to the possibility of node 3's communication interfering with node 2's, Link 3-4 is likewise dormant [5]. As a result, the throughput decreases as the number of hops increases due to an increase in idle nodes.

2.2.2. Throughput Drops with Increasing Mobility

Due to frequent topology changes, highly mobile nodes will incur additional overheads. This is due to an increase in the number of routing packet transmissions as a result of the need to determine new routes following route failures. When using the routing table, each node maintains a list of all available destinations as well as the number of hops required to reach each destination. Topology changes will be reflected in the routing table. Any changes to the routing table are broadcast to all other nodes. This increases the overall network overhead. When the overhead is high, a lower percentage of the packet is dedicated to data transmission, resulting in a lower throughput.

2.2.3. Delay

Delay is the average time it takes a packet from the time it leaves a source to the time it arrives at its destination. As previously stated, there is a need to keep the nodes busy with packet transmission and reception in order to increase the network's throughput. This means that the queue of each node is never empty, resulting in a longer delay.

2.3. Manet applications

Although initially mobile ad hoc networks were conceived as a general-purpose network, in terms of real-world deployment and industrial adoption, MANET applications are envisaged as specialized networks that are managed by a single authority and tailored to solve specific problems. For example, in military networks, vehicular networks or sensor networks. Additionally, MANETs are expected to become a key component in the 4G architecture. Similarly, ad hoc networking technique is expected to become an important functionality of overall next-generation wireless network technologies.

In this context, typical applications and some of the most illustrative use cases nowadays are described below:

2.3.1. Tactical networks

MANETs are suitable for facilitating communication and coordination between soldiers, military vehicles, and information headquarters. MANETs have numerous advantages in military applications. For starters, their distributed architecture reduces the risk of communication failure because there is no fixed infrastructure that can be attacked. Second, tactical networks based on MANETs can be quickly deployed even in areas with no infrastructure.

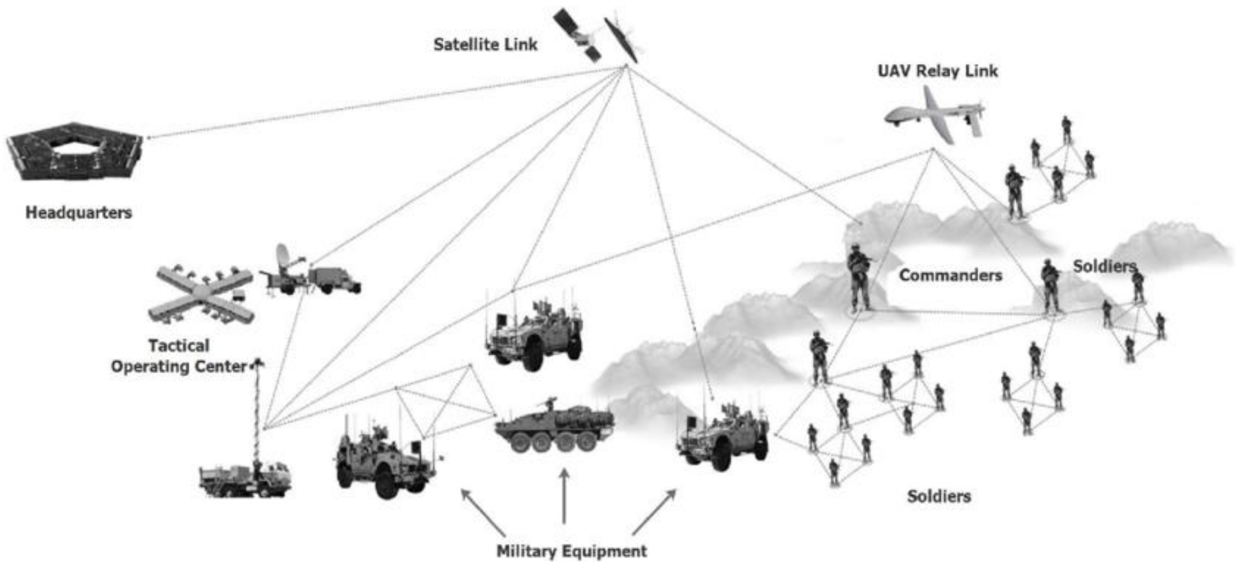


Figure 2-5 Military communication using MANET.

2.3.2. Extended network connectivity

MANETs can also be used to connect devices or the Internet in areas with limited infrastructure or intermittent access. The Interplanetary Internet and the Rural Internet system are two examples. The first is a multi-hop communication system that connects devices such as satellites, shuttles, and surface stations to provide a solution to the extreme distances that single-hop communication can cover. The Rural Internet is a networking system that extends Internet connectivity to remote areas via personal (e.g., mobile phones) and community-owned electronic devices.

2.3.3. Emergency Services

Ad hoc networks have traditionally been used after disasters or catastrophes such as floods, earthquakes, or fires. Ad hoc networks not only allow emergency teams to communicate, but they can also be used to rapidly deploy self-organized devices such as unmanned aerial vehicles (UAVs) or autonomous robots, reducing response time and maximizing the efficiency of available resources.

2.3.4. Vehicular Networks (VANETs)

The novel Intelligent Transportation Systems make use of Vehicular Networks, a type of ad hoc network (VANETs). Vehicles in these networks are outfitted with wireless interfaces that allow them to communicate with one another (V2V) or with road-side fixed infrastructure (V2I). On the one hand, V2V communications enable vehicles to participate in vehicle coordination platforms as well as the routing of other communications; on the other hand, V2I connectivity enables vehicles to obtain information about road conditions, traffic congestion, or accident warnings; and control platforms can update traffic lights in real time to optimize traffic flow in the event of congestion. Actually, IEEE is working on defining an architecture and a complementary, standardized set of services and interfaces that will enable secure V2V and V2I wireless communications in the IEEE 1609 family of standards.

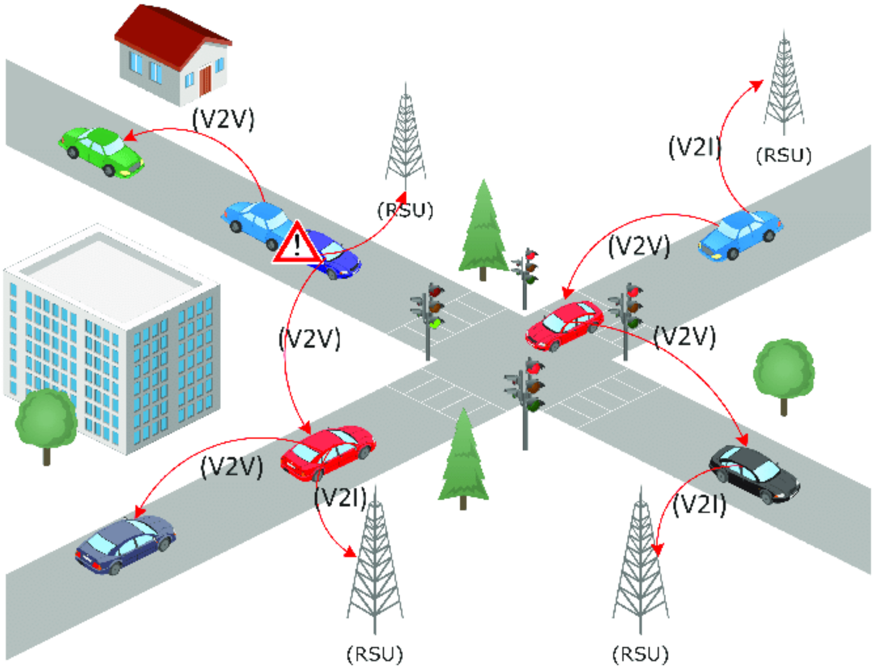


Figure 2-6 Example of the communication between VANETs components.

2.3.5. Wireless Personal Area Networks (PANs)

Ad hoc networks enable proximal electronic devices with specific functions, such as cameras, storage devices, televisions, mobile phones, or laptops, to dynamically share information via an autonomous home network.

2.3.6. Body Area Networks (BANs)

Carrying and holding a computer or laptop is impractical in some situations, such as on an assembly line or while working at a height. A wearable computer can provide a solution for ubiquitous computing in these cases by distributing computer components (e.g., head-mounted, processors, mass storage, displays, earphones, and microphones) on the body. Similarly, a system of wireless medical sensors can be placed in or around a human body to serve as a health monitoring system, which can be used for a variety of purposes such as remote health/fitness monitoring, rehabilitation, disease prevention, or sports training.

2.3.7. Wireless Sensor Networks (WSNs)

MANETS, in addition to BANs, PANs, VANETs, and military networks, can be used to interconnect a collection of low-cost and low-power sensor devices deployed in the environment or carried by animals. These devices are typically embedded in buildings, bridges, streets, animals, or mountains and are used for environmental or industrial monitoring, as well as monitoring events and phenomena more broadly.

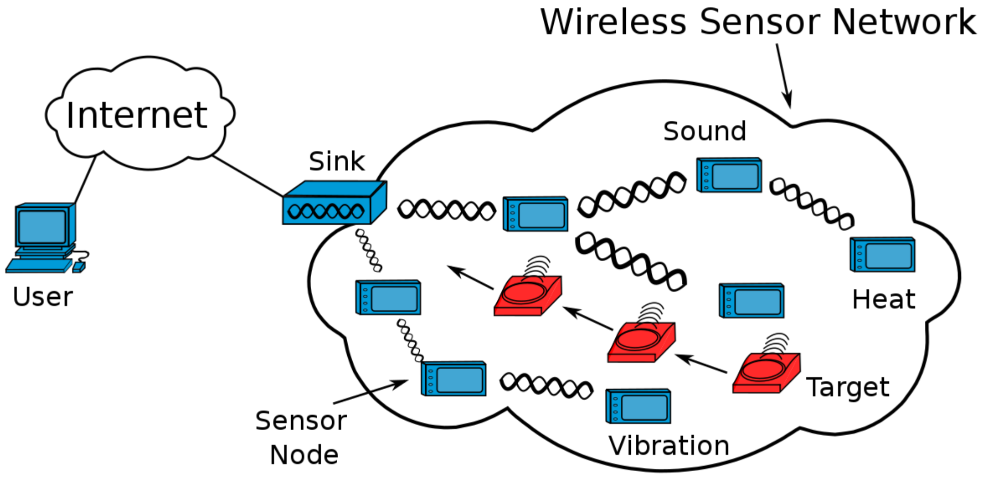


Figure 2-7 Example of Wireless Sensor Networks topology

2.3.8. Smart Cities

The recently proposed model of intelligent resource management in cities, known as "Smart Cities," emerges as another application of wireless ad hoc networking. Smart Cities make use of a wireless communication system to connect the smart devices deployed in the city and to facilitate content dissemination between users. Wireless sensors and actuators can provide new services and applications for controlling and coordinating physical services (such as adaptive traffic lighting, waste management, and smart signaling), as well as learning citizens' behaviors and needs and adapting city services.

2.3.9. Internet of Things (IoT)

The Internet of Things (IoT) is a novel communication paradigm that outlines a new scenario of modern wireless communication in which all devices communicate via a common platform and can thus be easily controlled remotely. The main goal is to build a virtual platform where objects or things (such as a sensor network, vehicle, appliance, mobile phones, and so on) can interact with one another and freely participate and offer/receive any type of service.

2.4. MANET Routing Protocols

Nodes in a Mobile Ad hoc Network (MANET) do not know the topology of their network; instead, they must discover it on their own because the topology in an ad hoc network is dynamic. The basic rules are that when a new node joins an ad-hoc network, it must announce its arrival and presence, as well as listen for similar announcement broadcasts from other mobile nodes.

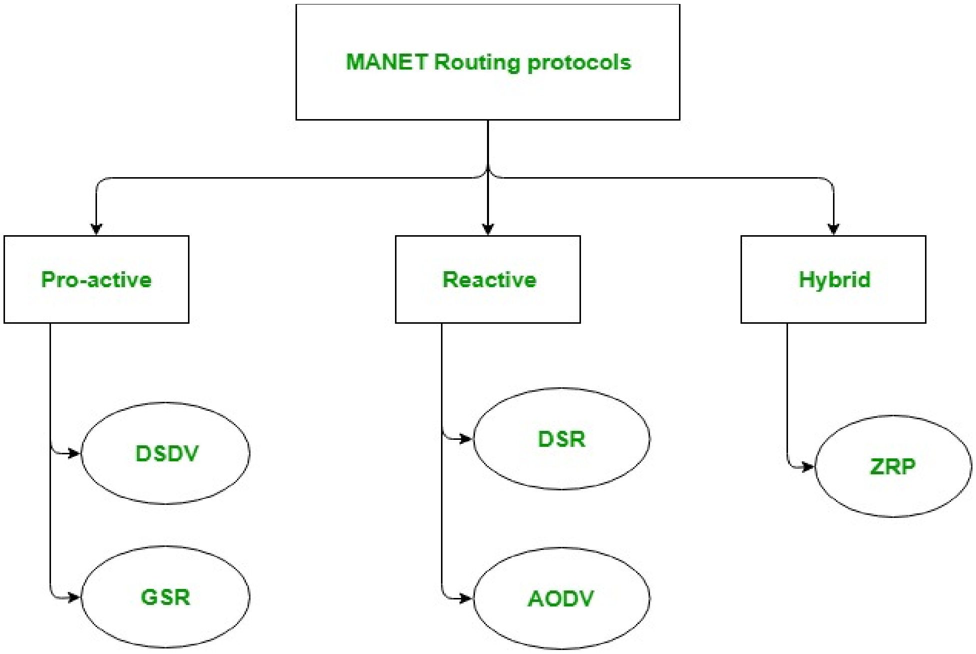


Figure 2-8 Classification of routing protocol in manet.

2.4.1. Proactive routing protocols

Table-driven routing protocols are another name for these. Each mobile node has its own routing table that contains information about the routes to all possible destination mobile nodes. Because the topology of a mobile ad-hoc network is dynamic, these routing tables are updated on a regular basis as the network topology changes. It has the limitation that it does not work well for large networks because the routing table entries become too large as they need to maintain route information to all possible nodes.

2.4.2. Destination Sequenced Distance Vector Routing Protocol (DSDV)

Destination Sequenced Distance Vector (DSDV) is a hop-by-hop vector routing protocol that requires each node to broadcast routing updates on a regular basis. Based on modifications to the Bellman-Ford routing mechanism, this is a table-driven algorithm. Each node in the network keeps a routing table with entries for all of the network's destinations and the number of hops required to reach each of them. Each entry is assigned a sequence number, which aids in the identification of stale entries. This mechanism enables the protocol to avoid routing loop formation. To advertise its location, each node sends updates that are tagged throughout the network with a monotonically increasing even sequence number. New route broadcasts include the destination's address, the number of hops required to reach the destination, the sequence number of the information received about the destination, and a new sequence number unique to the broadcast. The most recent sequence number labeled route is always used. When the neighbors of the transmitting node receive this update, they recognize that they are one hop away from the source node and include this information in their distance vectors. In their routing table, each node stores the "next routing hop" for every reachable destination. The route with the highest sequence number, i.e. the most recent, is used. When a neighbor B of A discovers that A is no longer reachable, it advertises the route to A with an infinite metric and a sequence number one greater than the route's most recent sequence number, forcing any nodes with B on the path to A to reset their routing tables. [6]

2.4.3. Reactive routing protocols

These are also known as on-demand routing protocols. In this type of routing, the route is discovered only when it is required. The route discovery process is carried out by flooding route request packets across the mobile network. It has two major phases: route discovery and route maintenance.

2.4.4. Ad-Hoc On Demand Vector Routing protocol (AODV)

The AODV protocol constructs routes between nodes only when source nodes request them. As a result, AODV is classified as an on-demand algorithm because it generates no extra traffic for communication along links. The routes are kept open for as long as the sources require them. They also form trees to connect members of multicast groups. To ensure route freshness, AODV employs sequence numbers. They are self-starting and loop-free, as well as scalable to a large number of mobile nodes. Networks in AODV are silent until connections are established. Network nodes that require connections broadcast a connection request. The remaining AODV nodes forward the message and keep track of which node requested a connection. As a result, they establish a series of temporary routes back to the requesting node. A node that receives such messages and holds a route to a desired node sends a backward message to the requesting node via temporary routes. The node that initiated the request takes the path with the fewest hops through other nodes. After a certain period of time, entries that are not used in routing tables are recycled. If a link fails, the routing error is returned to the sending node, and the process is repeated. [7]

2.4.5. Hybrid Routing protocol

Hybrid protocols, such as Zone Routing Protocol, combine the benefits of reactive protocol scalability and proactive protocol stability (ZRP). A hybrid protocol-enabled vehicle/node divides its neighbors into two zones: "intra-zone" and "inter-zone." Intra-zone uses reactive protocol to establish a route to destination networks, whereas interzone uses proactive protocol to update routing table information only when the network topology changes.

2.5. Broadcasting in MANETs

Broadcasting is a fundamental operation in MANETs in which a source node sends the same packet to all network nodes. A transmission by a node in the one-to-all model can reach all nodes within its transmission radius, whereas in the one-to-one model, each transmission is directed toward only one neighbor [8]. The one-to-many model can be used to reach multiple neighbors at once, and can also be considered. [9]

2.5.1. Applications of Broadcasting

Broadcasting has a wide range of applications, and several MANET protocols rely on the availability of an underlying broadcast service [10, 11]. Broadcasting is used in applications such as paging a specific node or disseminating information to the entire network (alarm signal for example). It can also be used in reactive protocols for route discovery. A route request is broadcasted in the network in Ad Hoc On-demand Distance Vector Routing (AODV) [7] and Dynamic Source Routing (DSR) [12] to discover a path to a specific destination. Each node remembers the broadcast ID as well as the name of the node from which the packet was received. When the destination is reached, it responds with a unicast (point-to-point) packet, and each intermediate node can then establish return routes [7].

2.5.2. Characteristics of Broadcasting

Because broadcasting is spontaneous, any mobile node can initiate a broadcast operation at any time. This broadcast packet is propagated in the network to reach all the nodes with a minimal number of re-transmission. All other nodes have a responsibility to help in propagating the packet by re-broadcasting it. An attempt should be made to successfully distribute the packet to as many nodes as possible without incurring substantial computational and communication overhead.

2.5.3. Broadcast storm caused by flooding

Flooding is a method of rapidly distributing routing protocol updates to every node in a large network. When a broadcast message is received for the first time, the host is required to rebroadcast it. This clearly costs n transmissions in a network of n hosts [9]. The disadvantages of flooding in a CSMA/CA network include:

Redundant rebroadcasts. When a mobile host decides to rebroadcast a broadcast message to its neighbors, the message is already in the hands of all of the neighbors.

Contention. After a mobile host broadcasts a message, if many of its neighbors decide to rebroadcast the message, these transmissions (which are all from nearby hosts) may severely contend with each other.

Collision. Collisions are more likely to occur and cause more damage due to the lack of RTS/CTS dialogue.

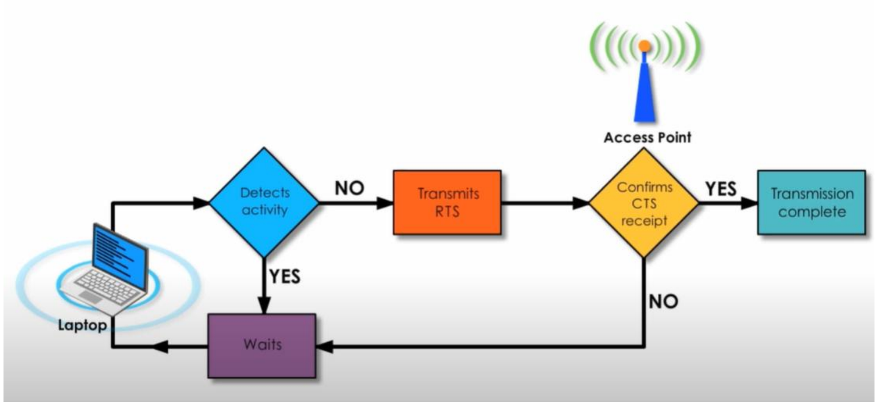


Figure 2-9 CSMA/CA — Wireless Medium Access Control Protocol.

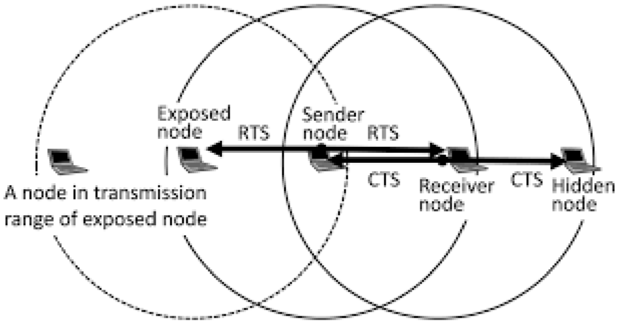


Figure 2-10 RTS/CTS for Exposed Node Reduction in IEEE 802.11 Ad Hoc Networks.

2.5.4. Mechanisms to reduce redundancy, contention, and collision

One solution to the broadcast storm problem is to prevent some hosts from rebroadcasting in order to reduce redundancy and thus contention and collision [9]. Here are some schemes to do so:

Probabilistic Scheme: Probabilistic rebroadcasting is a simple way to reduce rebroadcasts. When a broadcast message is received for the first time, a host will rebroadcast it with probability P. When P = 1, this scheme is clearly equivalent to flooding.

Counter-based scheme: Do not rebroadcast if a host has received a broadcast packet more than C times.

Distance-Based Scheme: Calculate the distance to the sending host d(min) = Minimum distance between sending hosts. If d(min) is less than D (a threshold), then do not rebroadcast.

2.6. Conclusion

This chapter introduced the mobile ad hoc network (manet). These include MANET CHARACTERISTICS, MANET LIMITATIONS, and MANET APPLICATIONS. We also discussed several routing protocols, such as proactive, reactive, and hybrid routing protocols. Following that, we discussed broadcasting in manet, including the characteristics and applications, as well as describing the damage caused by the broadcast. Finally, we discussed some potential solutions to the broadcast storm problem.

The following chapter will present research studies on the probabilistic broadcast scheme.

Chapter 3: Probabilistic broadcast schemes

In order to alleviate the broadcast storm problem, many broadcast protocols have been proposed in the last decade. A basic classification of broadcast schemes divides them into two categories, deterministic schemes and probabilistic schemes. In deterministic techniques, only a subset of nodes are allowed to take part in the broadcasting process. Multi Point Relay (MPR) and Connected Dominating Set (CDS) are some examples of deterministic broadcast algorithms. However, this could lead to repeated use of the same nodes. In addition, under mobility conditions this set of nodes should change very frequently because of the topological changes. [13]

Probabilistic broadcast schemes however balance the power consumption among all the nodes in the network by selecting well balanced routes over the network lifetime. In probabilistic broadcast, nodes forward the incoming broadcast packets according to certain probability values, so all nodes are allowed to participate in the broadcast process. Moreover, probabilistic schemes are more robust against failures, attacks, and are unaffected by the mobility of nodes like the deterministic schemes. [13]

Probabilistic broadcast schemes are classified into two main categories, fixed probability schemes and adaptive schemes.

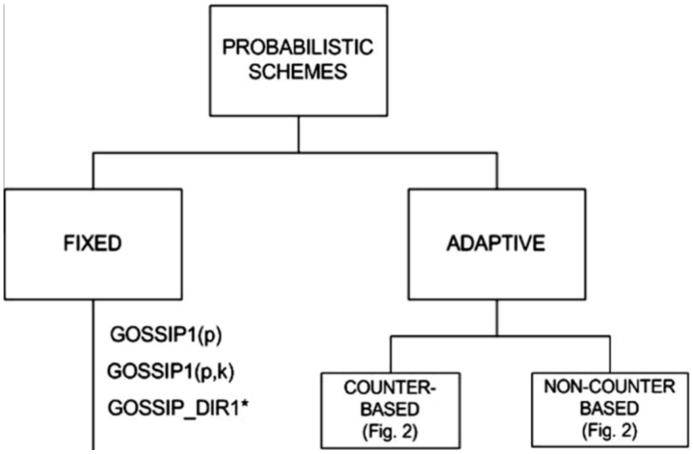


Figure 3-1 Classification of probabilistic schemes.

3.1. Fixed probability schemes

These schemes use a constant forwarding probability value so every node has the same forwarding probability in the network. Fixed probabilistic schemes have been studied using percolation theory and the phase transition phenomenon used in random networks. However, there are important differences between random networks and ad hoc networks so the results observed in random networks cannot be said to be true for ad hoc networks. Consequently, there is not an optimal forwarding probability for all possible scenarios. This optimal forwarding probability may depend on many parameters such as density, distance among nodes, and speed. [13]

The simplest probabilistic scheme is known as GOSSIP [14,15]. In this scheme, nodes forward an incoming packet with a fixed probability p, and the probability of not forwarding the incoming packet is 1 p. This scheme is called GOSSIP1(p).

A further expansion is GOSSIP1(p,k) [15] in which nodes forward an incoming packet with a probability equal to 1 for the first k hops. This ensures certain connectivity among nodes up to a distance of k hops from the source node.

A variant of GOSSIP1 is GOSSIP_DIR1 [16], which uses directional antennas instead of omnidirectional antennas.

3.2. Adaptive schemes

In adaptive schemes local or global parameters, such as density metrics, speed, energy is used to determine the forwarding probability. The adaptive schemes can also be classified into two categories, non-counter-based schemes and counter-based schemes. The main difference between these two categories is that the counter-based schemes use the number of copies of a given broadcast packet as a feedback from the broadcast process in the node’s neighborhood. The objective of the counter-based mechanism is to avoid the die out problem of broadcasting. The die out problem happens when the broadcast process is stopped due to forwarding decisions. Both counter-based and non counter-based adaptive schemes can be further classified according to the parameters used to adjust the forwarding probability such as density, distance, speed, number of neighbors covered (self-pruning), energy and artificial intelligence. [13]

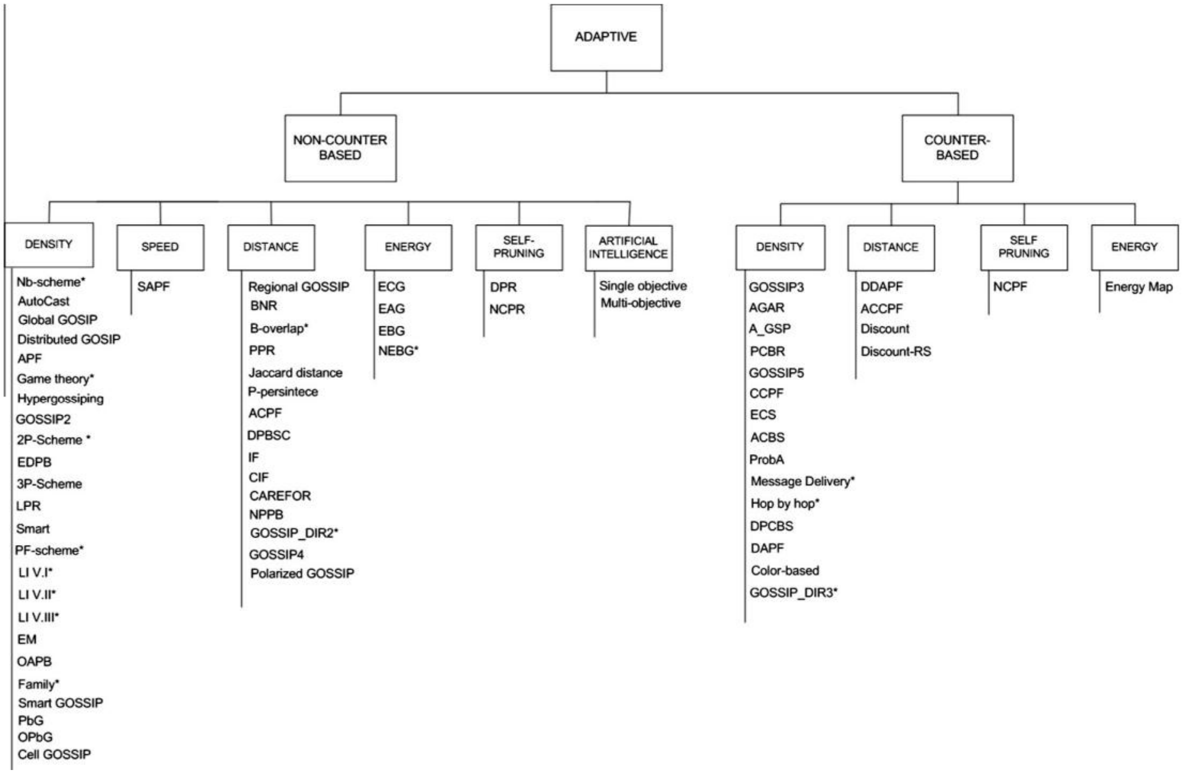


Figure 3-2 Classification of adaptive schemes. [13]

An example of adaptive non-counter-based scheme is nb-scheme [17]. The forwarding probability in nb-scheme is adjusted to the inverse proportional to the number of neighbors of a node so:

$$p = \frac{k}{n_b}$$

Where k is the propagation factor and by adjusting its value, the maximum and minimum probability can be adjusted. The value of n_b can be easily obtained via hello packets. Although this scheme is very simple, the optimum value of k can depend on many topological parameters.

An example of an adaptive counter-based scheme is the AGAR scheme [18]. This scheme combines the counter-based mechanism with the node's number of neighbors in order to adjust the forwarding probability. a node forwards the incoming packet with a probability:

$$p = \frac{pi}{n_b + 1}$$

whenever it receives fewer than h messages. As a result, the probability will depend on the node's neighbors resulting in a reduction in the number of broadcast messages.

3.3. Conclusion

In this section we have reviewed some research studies that have been done in probabilistic broadcast schemes. This review includes comparison of the two classes of this scheme: fixed probabilistic scheme and adaptive probabilistic scheme. As well as some examples of these two classes.

The next chapter talks about simulation and explains the implementation of our work.

Chapter 4: Implementation

4.1. Simulation

Computers are one of the primary sources for research. They are crucial for the analysis of vast volumes of data as well as for the simulation-based analysis of models, which provides more possibilities for confirming the interactions between model components. Simulation is utilized in research because it enables testing of most theoretical model capabilities and constraints and because it helps in the creation of artificial conditions that are challenging to replicate in a real experiment. [19]

For instance, any laboratory sample can be re-created with a model in a computing environment; the physical device would be the computer program or software, and the measurements would be the computer tasks. [19]

Researchers can predict how a system will behave by using simulation, which is application or computer processes that mimic physical processes by generating a similar reaction. It can therefore be used as an experimental setup or as a tool to help with operational decisions. Additionally, it is used to research challenging and complex systems before investing resources in a real experiment. [19]

There are two types of systems: discrete and continuous. In a discrete system, the state variables change instantly at different points in time. In a continuous system, the state variable changes continuously over time. [19]

In computer networks, many systems function as discrete systems (LAN, cellular infrastructure, wireless networks); in them, specific events or interactions change the state and the behavior of the entire system. [19]

4.2. Network Simulators

A network simulator is a software program that predicts the performance of a computer's network. Network simulators are used when communication networks become too complex for fixed analytical tools to provide a precise insight of system performance. In a simulator, the computer network can be shaped with the help of links, devices and applications, allowing researchers to study the performance of a network. [20]

The different types of network simulators/ network simulation tools are open source and commercial:

- Network Simulator version 2 (NS-2)
- Ns3

- Netkit
- Marionnet
- JSIM (Java-based Simulation)
- OPNET
- QualNet
- Glomosim

The open-source simulators are Marrionet, Netkit, NS-2, NS-3, JSIM, Glomosim
The commercial simulators are OPNET and QualNet

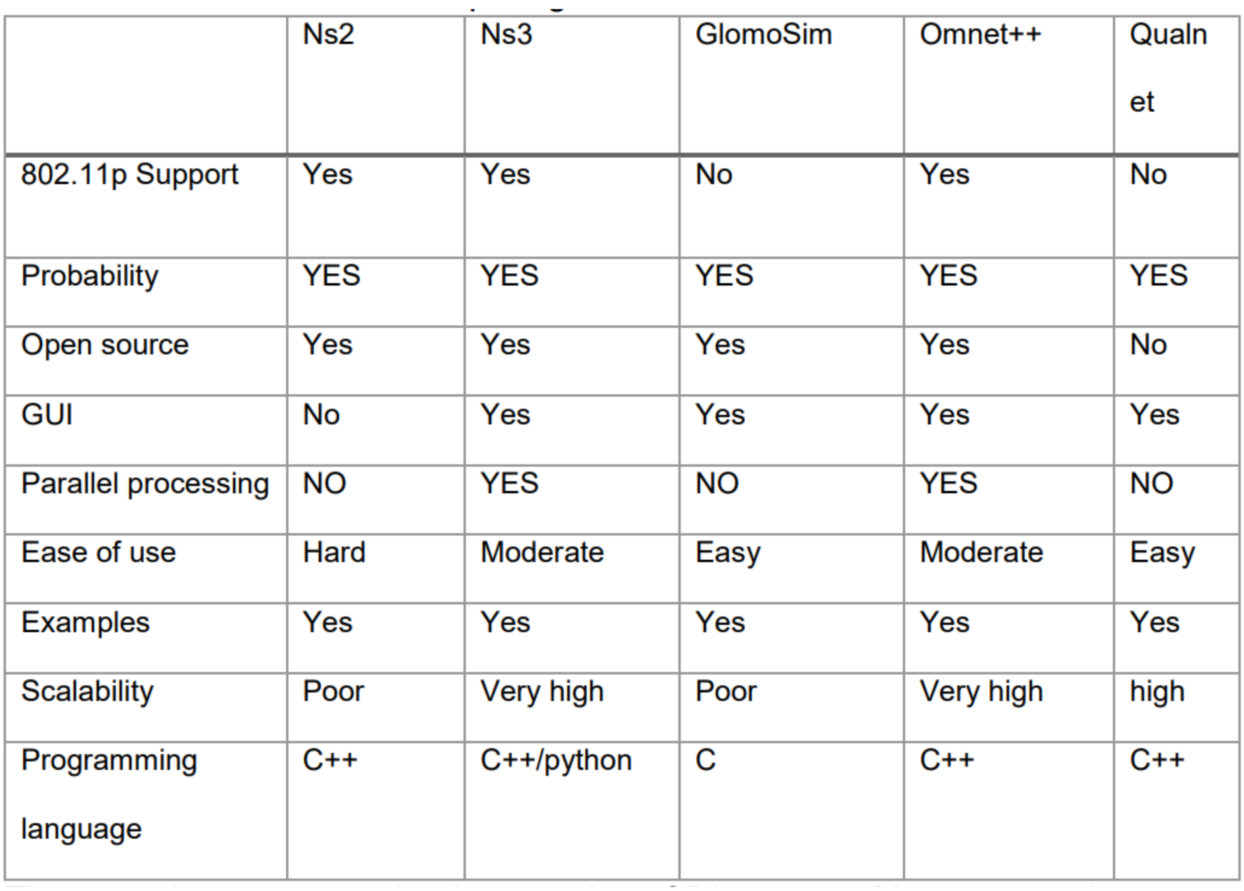
4.3. The ns-3 network simulator

The Ns-3 is a discrete event driven network simulator created with C++ language for research and educational purposes. The simulator is developed as an open source project, and it promotes community involvement by offering a public mailing list for user and developer discussions, a bug tracker, and a wiki containing user-contributed content. GPL2v is the license used for all software. [21]

The goal of the ns-3 simulator is to provide realistic simulation of all parts of the network down to the physical level, so that the gap from simulation to real implementation is decreased, while providing an architecture that is easily configurable and extended. [21]

Table 1 compares different network simulators:

Table 4.1: comparison of different network simulators [22].



	Ns2	Ns3	GlomoSim	Omnet++	Qualnet
802.11p Support	Yes	Yes	No	Yes	No
Probability	YES	YES	YES	YES	YES
Open source	Yes	Yes	Yes	Yes	No
GUI	No	Yes	Yes	Yes	Yes
Parallel processing	NO	YES	NO	YES	NO
Ease of use	Hard	Moderate	Easy	Moderate	Easy
Examples	Yes	Yes	Yes	Yes	Yes
Scalability	Poor	Very high	Poor	Very high	high
Programming language	C++	C++/python	C	C++	C++

4.4. Protocols for the NS-3 simulator

By modeling the nodes more like actual computers with applications, protocol stacks, and network interfaces, ns-3 is in line with real systems and supports crucial interfaces like Berkeley sockets and IP/device driver interface for Linux to allow for the reuse of application and kernel code. Additionally, PCAP1 and ns-2 mobility traces are supported, enabling software integration with other tools. [21]

The robust attribute system in ns-3 makes all simulation settings configurable, and the callback-based tracing system separates the generation of trace events, for example, from data logging and analysis. [21]

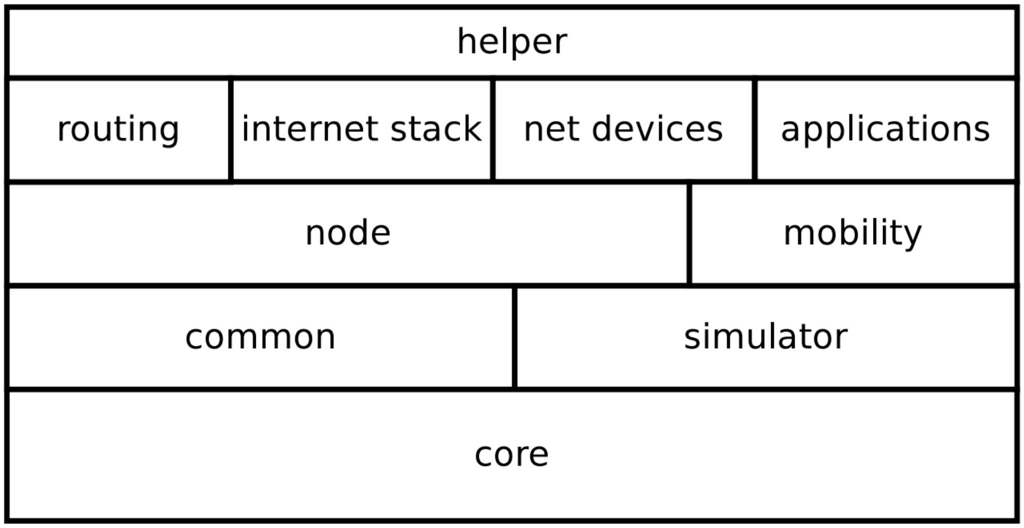


Figure 4-1 ns-3 software architecture [21].

The implementation of ns-3 uses a modular design. The different models implements:

Core: Contains generic core functions of the smart pointers, callbacks, logging.
Simulator: Contains the implementation of events, schedulers and time objects that forms the simulator.
Common: Library of objects that are common to all networks packet and packet headers.
Node: Contains the fundamental simulator models nodes, channels, base class for net devices and sockets.
Mobility: Contains different mobility models static, ns-2 traces and random walk.
Routing: Contains different router implementations AODV2 and OLSR3.
Internet Stack: Contains Internet related models, models TCP, UDP and IPv4.
Net Devices: Different net device implementations, Ethernet, Wifi, Wimax etc.
Application: Base classes for implementing applications.
Helper: Provides a simpler interface to lower-layer APIs. [21]

4.5. C++ language

Appeared in the early 1990s, the C++ language is currently one of the most widely used languages worldwide for scientific applications and software development. As the successor to the C language, C++ is efficient. But it also has powerful features, such as "Class" concepts, which allows the application of object-oriented programming techniques. [5]

There are two types of source files:

- Files that actually contain instructions; their names are EX.cpp
- Files that only contain declarations; their names are EX.h

4.6. Simulation environment

For the Operation System, NS-3 is well optimized to work with Linux. We chose Ubuntu 20.04 which is a Linux distribution.

The Ubuntu OS is installed on a virtual machine using VMware workstation. VMware Workstation is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems; it enables users to set up virtual machines on a single physical machine and use them simultaneously along with the host machine. [24]

Virtual machine settings:
Memory size: 2 GB
Number of processors: 2
Hard disk size: 30 GB

4.7. NS-3 installation

NS-3 is installed by executing the following commands:

Update the Ubuntu Repo and Existing Applications

```
sudo apt update
sudo apt -y upgrade
```

Install Core Dependencies for Build and Compilation

```
sudo apt install build-essential libsqlite3-dev libboost-all-dev libssl-dev
git python3-setuptools castxml
```

Install Dependencies for NS-3 Python bindings

```
sudo apt install gir1.2-goocanvas-2.0 gir1.2-gtk-3.0 libgirepository1.0-dev
python3-dev python3-gi python3-gi-cairo python3-pip python3-pygraphviz
python3-pygccxml
```

Install Dependencies for NS-3

```
sudo apt install g++ pkg-config sqlite3 qt5-default mercurial ipython3
openmpi-bin openmpi-common openmpi-doc libopenmpi-dev autoconf cvs bzr
```

```
unrar gdb valgrind uncrustify doxygen graphviz imagemagick python3-sphinx
dia tcpdump libxml2 libxml2-dev cmake libc6-dev libc6-dev-i386 libclang-
6.0-dev llvm-6.0-dev automake
```

Install Additional Python Packages

```
sudo su
cd
pip3 install kiwi
exit
Cd
```

Download and Install NS-3

```
# download from ns-3 server
wget -c https://www.nsnam.org/releases/ns-allinone-3.33.tar.bz2
# extract tar.bz2
tar -xvjf ns-allinone-3.33.tar.bz2
# go back to home folder
cd

# navigate to ns-3 directory (not the NS-3 all in one)
cd ns-allinone-3.33/ns-3.33/
# Configure the installation
./waf configure --enable-examples
# Build ns-3 installation
./waf
# to check whether installation was a success
./waf --run hello-simulator
```

NS-3 files can be found in bake/source/ns-3 directory

4.8. Implementation

Flooding is implemented for route discovery in various routing protocols such as aodv and dsr. We choose to work on route discovery functions from ns-3 AODV source code to set up flooding broadcast implementation. *Manet-routing-compare.cc* program allows one to run ns-3 AODV under a typical random waypoint mobility model. The program is located in the example/routing directory and it is a free software.

The random waypoint model has been used to simulate mobility patterns. Nodes follow a motion-pause recurring mobility state, where each node at the beginning of the simulation remains stationary for pause time seconds, then chooses a random destination and starts moving towards it with speed selected from a uniform distribution (0, max_speed]. After the node reaches that destination, it again stands still for a pause time interval (pause_time) and picks up a new destination and speed. This cycle repeats until the simulation terminates.

The simulation runs for 300 simulated seconds, of which the first 50 are used for start-up time. The number of nodes is 50. Nodes move according to RandomWaypointMobilityModel with a speed of 2 m/s and no pause time within a 1000x1000 m region. The WiFi is in ad hoc mode with a 2 Mb/s rate (802.11b). Number of nodes and moving speed can be changed to other values as a workload for the simulation.

The following code represent the configuration of packet size and data rate:

```
int nWifis = 50;
NodeContainer adhocNodes;
adhocNodes.Create (nWifis);
```

The following code represent the configuration of node moving speed:

```
int nodeSpeed = 2;
std::stringstream ssSpeed;
  ssSpeed << "ns3::UniformRandomVariable[Min=0.0|Max=" << nodeSpeed << "]";
  std::stringstream ssPause;
  ssPause << "ns3::ConstantRandomVariable[Constant=" << nodePause << "]";
  mobilityAdhoc.SetMobilityModel ("ns3::RandomWaypointMobilityModel",
                                  "Speed", StringValue (ssSpeed.str ()),
                                  "Pause", StringValue (ssPause.str ()),
                                  "PositionAllocator", PointerValue
(taPositionAlloc));
```

There is one source/sink data pair sending UDP data at an application rate of 4.096 Kb/s each. This is typically done at a rate of one 512-byte packet per second. Application data is started at a random time between 50 and 51 seconds and continues to the end of the simulation. Packet number can be changed to other values as a workload for the simulation.

The following code represent the configuration of packet size and data rate:

```
std::string rate ("4096bps");
Config::SetDefault  ("ns3::OnOffApplication::PacketSize",StringValue ("512"));
Config::SetDefault ("ns3::OnOffApplication::DataRate",  StringValue (rate));
```

The following code set AODV as a routing protocol for this topology:

```
AodvHelper aodv;
```

```
list.Add (aodv, 100);
internet.SetRoutingHelper (list);
internet.Install (adhocNodes);
```

Aodv-routing-protocol.cc is the AODV implementation file in ns-3, it can be found in src/aodv/model directory.

In AODV, nodes discover routes in request-response cycles. A node requests a route to a destination by broadcasting an RREQ message to all its neighbors. When a node receives an RREQ message but does not have a route to the requested destination, it in turn broadcasts the RREQ message. [25]

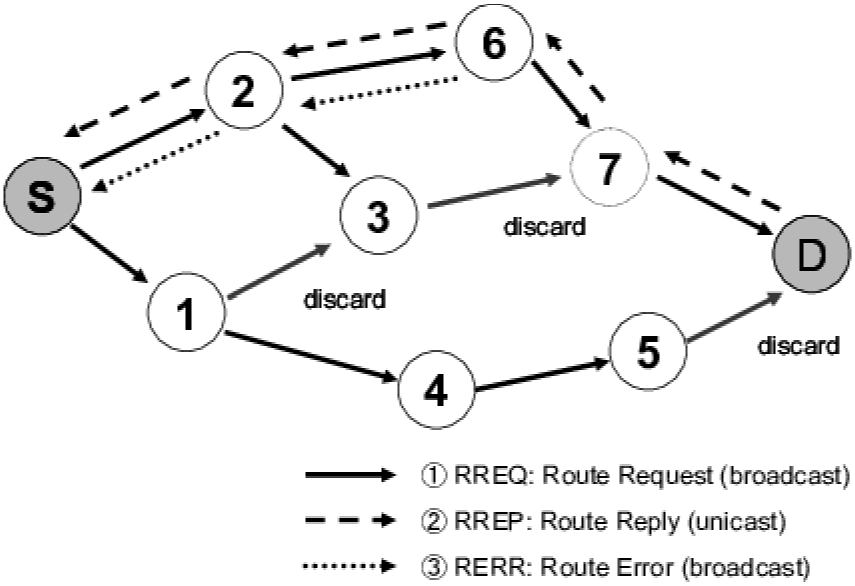


Figure 4-2 AODV route discovery.

In *Aodv-routing-protocol.cc* program, 2 functions are involved in the request broadcast:
SendRequest: Starts the broadcast by sending the RREQ packet.
RecvRequest: Receive the RREQ packet, then do one of 3 following decisions:

- Drop the packet if node received it before, we refer to it as a duplicated RREQ packet.
- Send a RREP packet (route reply) if the node is the destination or it has a route to the destination.
- Rebroadcast the RREQ message.

The the default broadcast algorithm is as follows:

```
Node receive RREQ;
If ( RREQ is duplicated) then drop RREQ;
Else if (this node is the destination or it have a route for destination) then send RREP;
Else rebroadcast RREQ;
```

The broadcast is done by flooding,which means that the rebroadcast probability is 1 by default. We modify the program to add probabilistic broadcast implementation.

Our proposed algorithm is as follows:

```
Node receive RREQ;
If ( RREQ is duplicated) then drop RREQ;
Else if (this node is the destination or it have a route for destination) then send RREP;
Generate random number x between 0 and 1;
if (x < p) then rebroadcast RREQ packet;
Else drop RREQ packet;
```

Where **p** is the probability of rebroadcast.

The rebroadcast probability condition is added in the *SendRequest* function before rebroadcasting the RREQ message:

```
double r = ((double) rand() / (RAND_MAX));
  if(r < 0.8){ //rebroadcast probability
  std::cout << "Positve " << id << " " << origin << " " << receiver << " " << src << " " <<
dst << "\n";
  //std::cout << m_ipv4->GetAddress (1,0).GetLocal () << " sent RREQ\n";

  for (std::map<Ptr<Socket>, Ipv4InterfaceAddress>::const_iterator j =
         m_socketAddresses.begin (); j != m_socketAddresses.end (); ++j)
    {
      Ptr<Socket> socket = j->first;
      Ipv4InterfaceAddress iface = j->second;
      Ptr<Packet> packet = Create<Packet> ();
      SocketIpTtlTag ttl;
      //ttl.SetTtl (tag.GetTtl () - 1);
```

```
      packet->AddPacketTag (ttl);
      packet->AddHeader (rreqHeader);
      TypeHeader tHeader (AODVTYPE_RREQ);
      packet->AddHeader (tHeader);
      // Send to all-hosts broadcast if on /32 addr, subnet-directed otherwise
      Ipv4Address destination;
      if (iface.GetMask () == Ipv4Mask::GetOnes ())
        {
          destination = Ipv4Address ("255.255.255.255");
        }
      else
        {
          destination = iface.GetBroadcast ();
        }
      m_lastBcastTime = Simulator::Now ();
      Simulator::Schedule (Time (MilliSeconds (m_uniformRandomVariable->GetInteger (0,
10))), &RoutingProtocol::SendTo, this, socket, packet, destination);

   }
    } else { std::cout << "Negative " << id << " " << origin << " " << receiver << " " <<
src << " " << dst << "\n"; }
```

We track all RREQ packets to study the broadcast performance. We categorize all the events for RREQ packets into 4 types:

Received: when a node receives a RREQ packet.
Duplicated: when a node drops a duplicated RREQ packet.
Positive: when a node rebroadcasts a RREQ packet based on a rebroadcast probability.
Negative: when a node drops a RREQ packet based on a rebroadcast probability.

Every RREQ packet has a route request ID, organizer address, source and destination addresses. The ID defines the group of broadcasts for one route request. Each simulation can have one or more broadcast IDs depending on the number of route requests. The organizer is the node requesting a route for the destination. The node checks if the packet has been received before by comparing the ID and organizer to the previous packets.

The following code prints each event, the current node address for that event and RREQ packet informations:

```
std::cout << "Received " << id << " " << origin << " " << receiver << " " << src <<
" " << dst << "\n";
std::cout << "Duplicated " << id << " " << origin << " " << receiver << " " << src
<< " " << dst << "\n";
std::cout << "Positive " << id << " " << origin << " " << receiver << " " << src <<
" " << dst << "\n";
std::cout << "Negative " << id << " " << origin << " " << receiver << " " << src <<
" " << dst << "\n";
```

We run the following command to start the simulation:

```
~/bake/source/ns-3.32$ ./waf --run aodv >> results.txt
```

The argument “>> results.txt” is used to print *std::cout* output in a text file.

We then import the extracted file into a Microsoft office excel file where we filter each event by its route request ID.

We calculated the reachability for each route request ID using the following formula:

Reachability = r - d

where **r** is the number of received RREQ packets for all nodes and **d** is the number of duplicated RREQ packets.

Then we calculated the reachability average (RE) for all route request IDs:

RE = Reachability / Number of route request IDs

Saved rebroadcast (SRB) is calculated using the following formula:

SRB = (RE - b) / b

Where **b** is the average number of nodes that rebroadcasted the RREQ packet. We get **b** by calculating the average of “positive” events for all route request IDs.

After repeating the same process for all the different simulations, we collect and organize each simulation result into groups based on the workload. reachability average and saved rebroadcast for each group is collected in an excel file in order to create graphs:

For each group of simulations, we have a graph for saved rebroadcasts (SRB) and a graph for reachability (SE). The axis for both graphs is rebroadcast probability with 10 different values from 0 to 1.

4.9. Conclusion:

In this chapter we have talked about simulation and its importance in the research space. We’ve introduced the NS-3 simulator and its benefits. We’ve also identified the environment used for our simulation. Next, we explained the implementation of our simulation.

Chapter 5: Performance Analysis of Probabilistic Flooding

A probabilistic approach to flooding has been suggested in [9,26]. In the purpose of reducing redundant rebroadcast packets and decreasing the damages caused by the broadcast storm problem. In the probabilistic scheme, when a node receives a packet the probability of the same node re-broadcasting that packet is a set probability p, when p=1 we are in the case of blind flooding.

It has been revealed in previous studies that probabilistic broadcasts demand a lower overhead in comparison to blind flooding while most of the nodes receive the broadcast packet using this scheme. In our study we will take in consideration a number of factors that could better the performance of a typical MANET, such as node mobility, network density and traffic load.

In this chapter we will describe the simulation setup and present performance results showing the effect of node mobility, network density and traffic load on the performance of probabilistic flooding.

5.1. Simulation Setup

In our simulation scenario we have 50 mobile nodes roaming in an area of 1000X1000m. The density of the nodes is sufficient to maintain a good connectivity in the network, each node has a 250 meters radius that can engage communication from within and the bandwidth of 2mbps. The probability of a rebroadcast varied between 0.1 and 1 percent increment by 0.1 percent per simulation trial.

We selected one node as the data source where a CBR traffic is attached to the selected node, we used the random waypoint model to simulate a mobility pattern, the maximum speed of the node varied from 2m/s to 20 with a pause time of 0 sec.

Table 5.1: Summary of the parameters used in the simulation experiments.

Parameter	**Value**
Transmitter range	250 meters
Bandwidth	2 Mbps
Simulation time	300 seconds
Pause time	0 seconds
packet size	512 bytes
Topology size	1000×1000 m2
Number of node	25,50,100
Maximum speed	2, 10 and 20 m/s
Traffic Load	1,5,10 packet/s

Broadcast protocols performance can be measured by a number of metrics [9,26] a commonly used one is the number of packets retransmitted with respect of the number of nodes in the network, in our research we used a saved rebroadcast and reachability.

Blind flooding is known to have the worst SRB with the reachability level being at its best. Though it's only achievable at the expense of excessive redundant re-broadcasting packets. Therefore, our goal in this study was to improve SRB while maintaining the same level of reachability. We observe from figure 5.1 the variations of the broadcast probabilities vary from 0.1 to 1.0% with 0.1% increment when the traffic load is at its lowest, 1 packet/s is injected into the network. While the SRB is explored in the figure 5.1 at low mobility conditions of maximum speeds of 2 m/s and 0 pause time. When the SRB is at 90% the value of p=0.1, meanwhile when p is increased to 0.6 the SRB decreased to 40%, observing the results we conclude that as much as the rebroadcast probability increases the SRB decreases. When p=1 (blind flooding) SRB is 0%. This is because as the probability of the transmission increases for every node, this implies that there are more candidates for broadcast re-transmissions in a given area, and as a result the number of nodes that transmit the packet increases which increases the number of redundant rebroadcast packets and that leads to a higher chance of collision and contention due to the increases in redundant rebroadcast packets.

As the probability of the transmission increases for every node it'll allow for more candidates for broadcast re-transmissions (in a given area) for example when p=1 the SRB is at 0%, in contrast there will be a higher chance of collision and construction due to the fact of the increase in redundant rebroadcast packets.

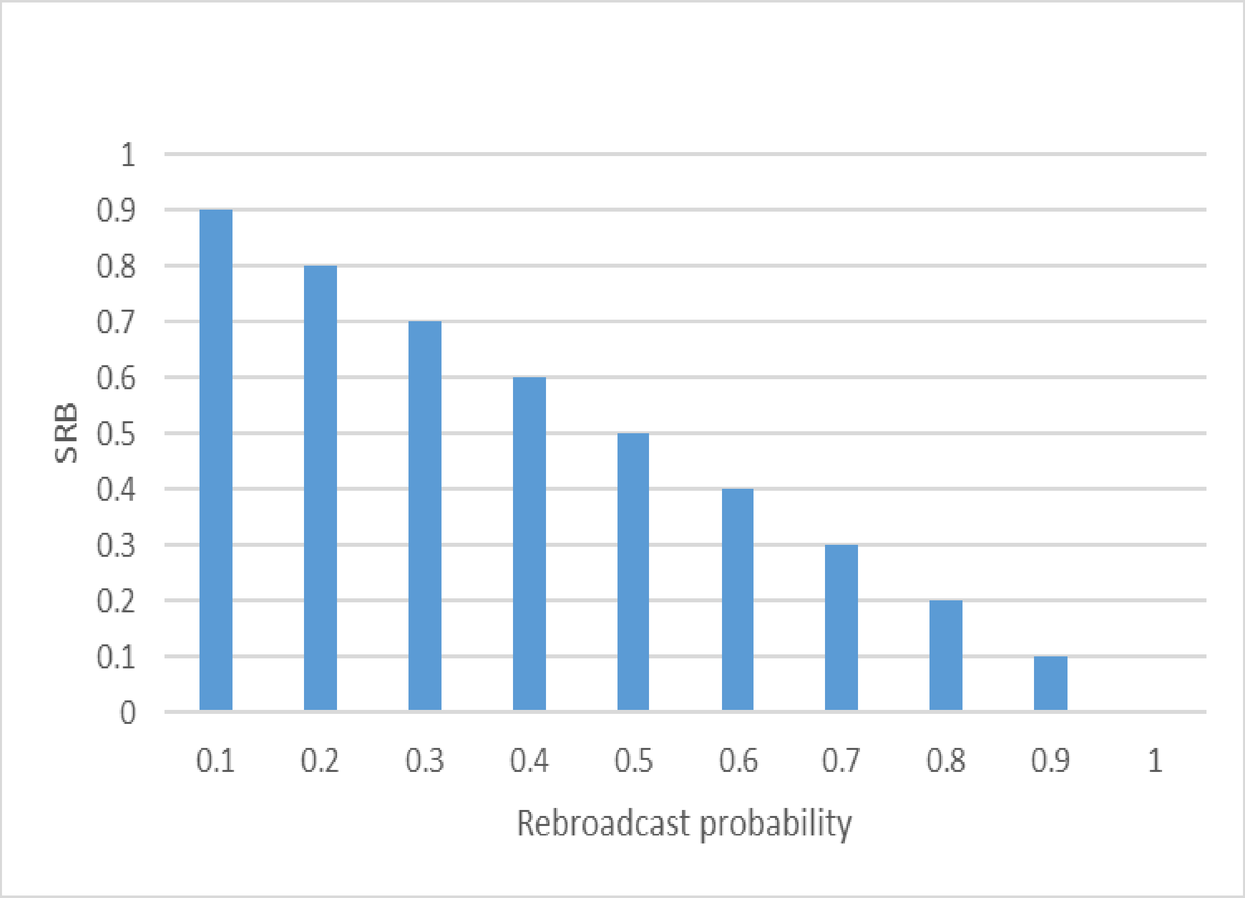


Figure 5-1 SRB vs. rebroadcast probability for a network size of 50 nodes and a node speed 2 m/s.

We observe from figure 5.2 the variations of the broadcast probabilities vary from 0.1 to 1.0% with 0.1% increment when the traffic load is at its lowest, 1 packet/s is injected into the network. While reachability (RE) is explored in the figure2020 at low mobility conditions of maximum speeds of 2 m/s and 0 pause time.

observing the results, we observe that as much as the rebroadcast probability increases the RE increases. For example, when p=0.1 RE is around 15% and when p is increased to 1.0 RE is around 70%.

As the probability of the transmission increases for every node it'll allow for more candidates for broadcast re-transmissions (in a given area). Hence the number of nodes receiving the broadcast packet increases over the total number of mobile nodes that are reachable, as a result of the increase of the number of nodes that transmit the packet.

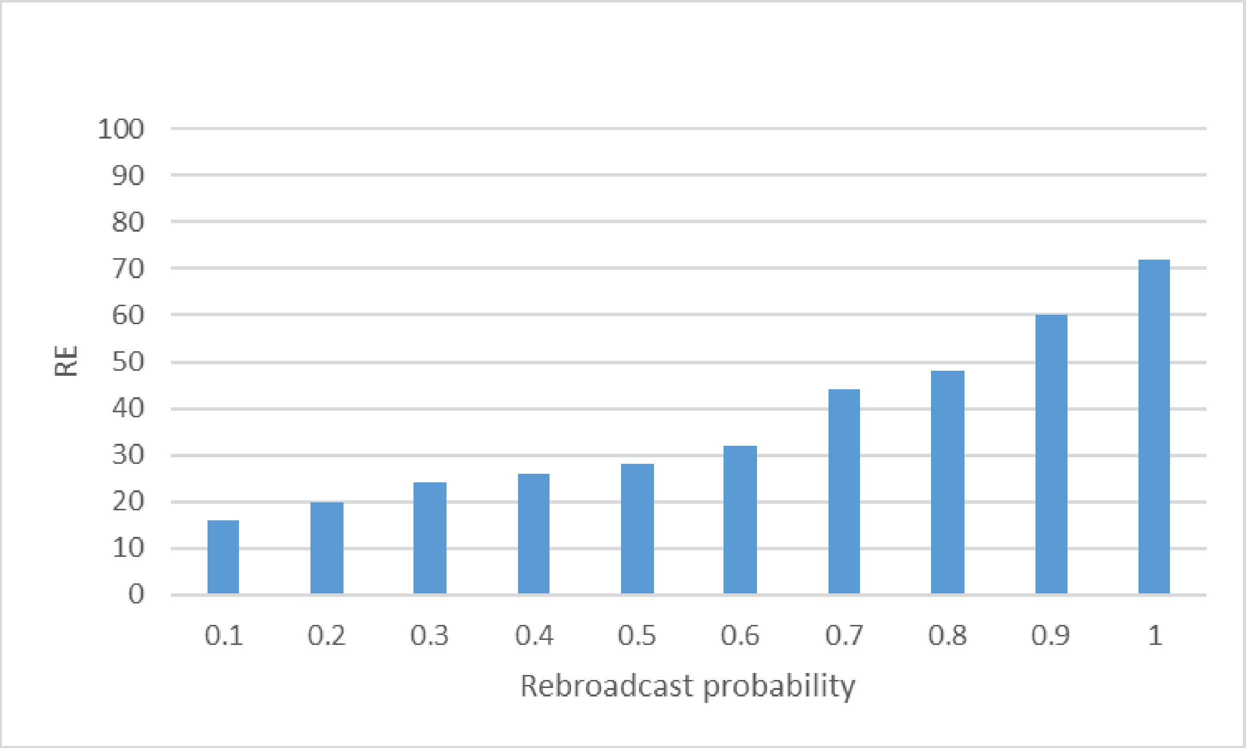


Figure 5-2 RE vs. rebroadcast probability for a network size of 50 nodes and node speed 2 m/s.

5.2. Effects of Network Density

To further the study of the performance effects of varying network density meaning the number of network nodes per unit area for a given transmission range, we examined three levels of network density as stated below:

- Low density: 25 nodes;

- Medium density: 50 nodes;

- High density: 100 nodes;

Considering the three different network densities and two different nod speeds depicted by the results shown in figures 5.3 and 5.4. As suggested by the figures the RE increases with a higher network density. The improvement of the RE can be linked to the following reasons. The increase of the density of the nodes is coherent to the number of the nodes covering a particular area. With the probability of re-broadcast being fixed for every node, for each “coverage“ area it's implied that there are more candidates for transmission. Therefore, there is a greater chance that a broadcast retransmission occurs, which resulted in the increasing of RE. RE increases proportionally to p, as p increases. For example, when N=100 the probability p=1.0 is required to achieve RE of 90%.

Considering a given transmission range, the network density increases coherently with the network connectivity.

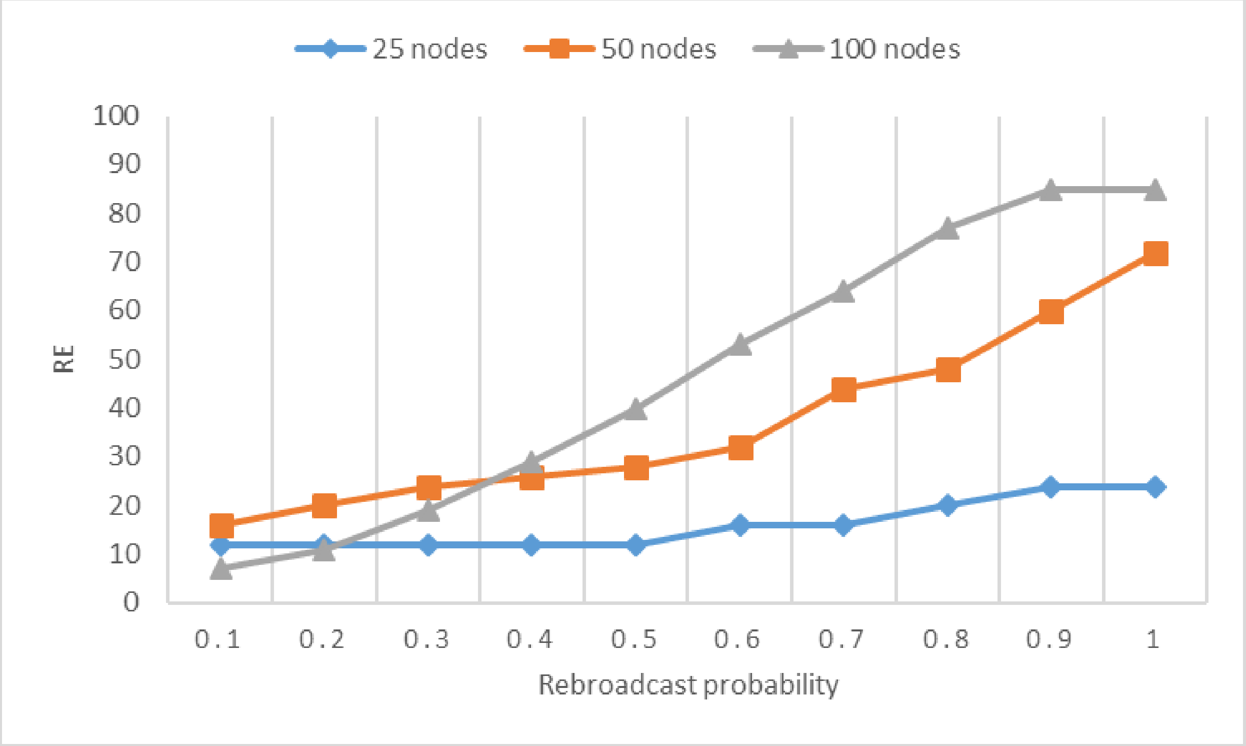


Figure 5-3 RE vs. rebroadcast probability for different network densities 25, 50, and 100 nodes and a node speed 2 m/s.

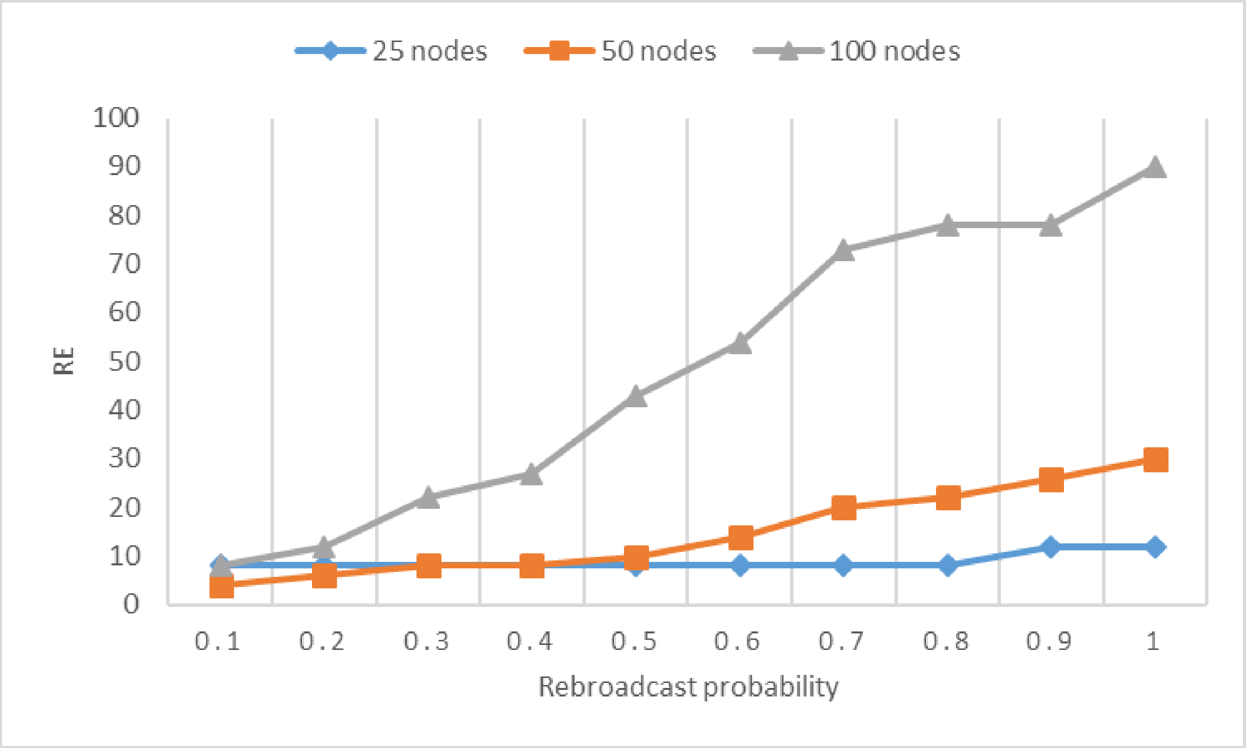


Figure 5-4 RE vs. rebroadcast probability for different network densities 25, 50, and 100 nodes and a node speed 20 m/s.

As can be seen in the figures, SRB decreases with a higher network density. The increase in number of nodes in a given area means that each node will have a higher number of neighbors as a result there will be a higher number of redundant packets, leading to a decline in SRB. However, as the node speed goes up, SRB does the same. Looking at the figures we noticed that when we increased the rebroadcast probability SRB decreases, this is related to the fact that increasing the probability of a rebroadcast lead to an increase in the redundant rebroadcast packet; this happens when a node rebroadcast a packet that a neighbor node already has a copy of the same packet. furthermore, increasing the rebroadcast probability leads to a higher possibility for collisions.

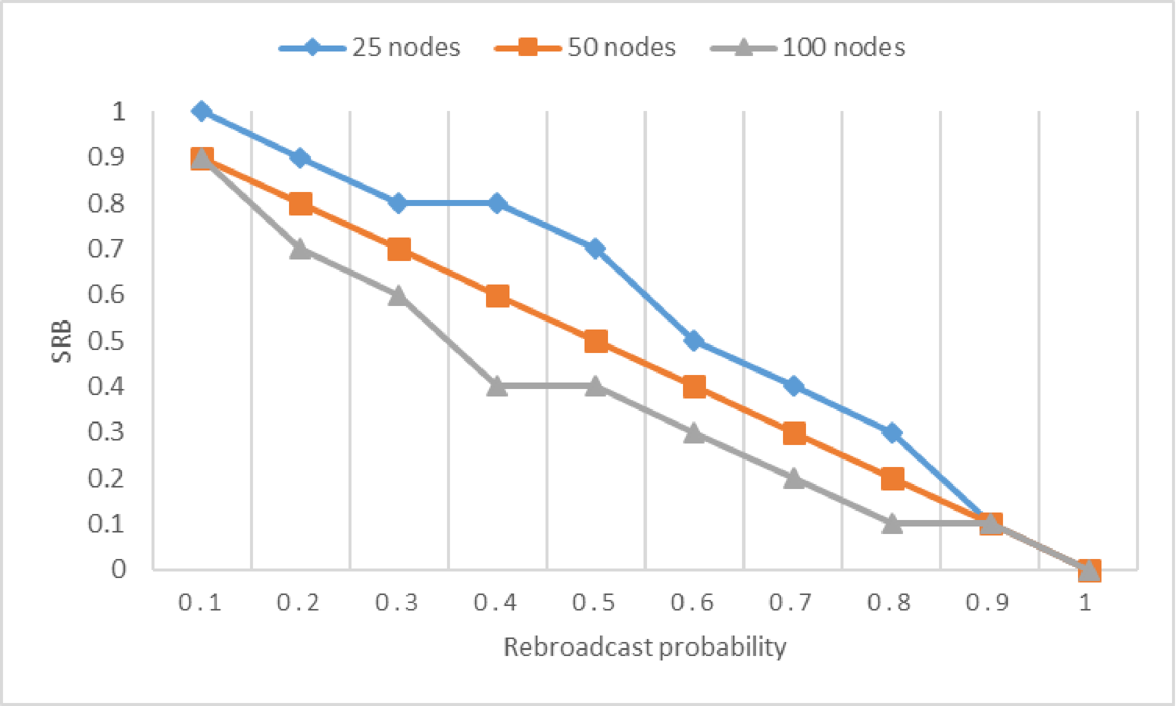


Figure 5-5 SRB vs. rebroadcast probability for different network densities 25, 50, 100 nodes and node speed 2 m/s.

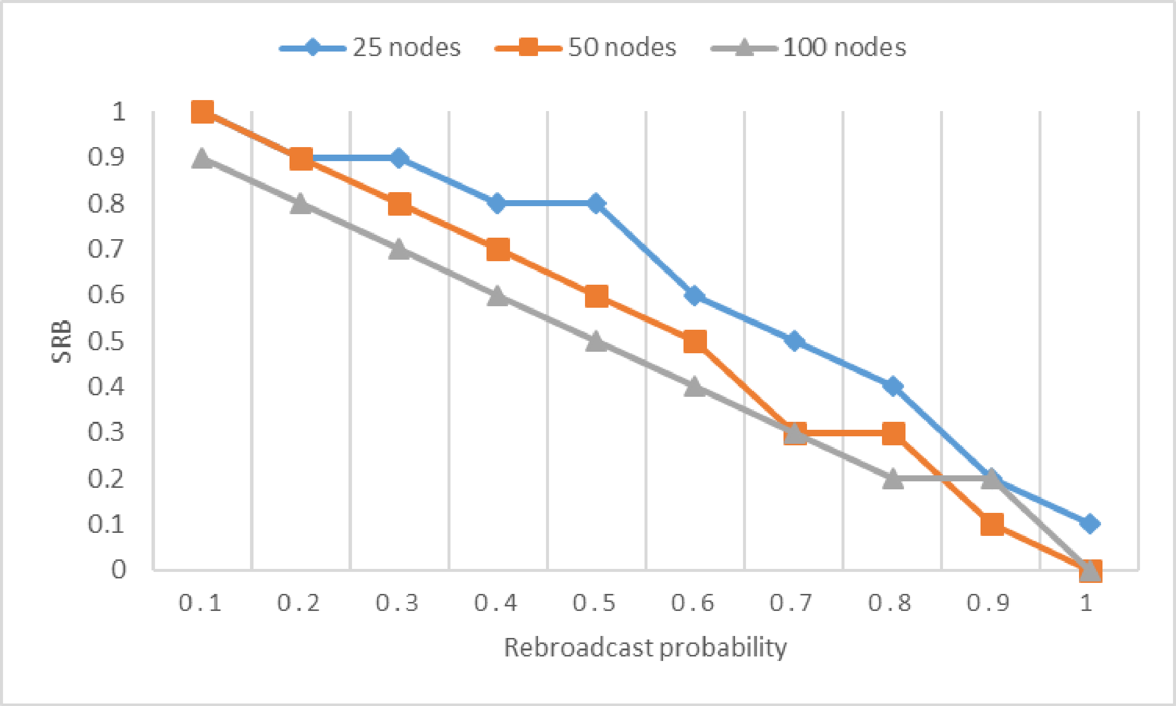


Figure 5-6 SRB vs. rebroadcast probability for different network densities 25, 50, 100 nodes and node speed 20 m/s.

5.3. Effects of Mobility

The results for reachability in comparison to different rebroadcast probabilities are shown in the following figure, the nodes move continuously with no pause time where the maximum speed is varied from 2 10 and 20 m/s. RE decreases as the node speed increases. For example, when a node moves with a low speed of 2m/s the reachability is 30% with a rebroadcast probability of p=0.5, but to get the same reachability with a node moving with a higher speed of 20m/s the rebroadcast probability need to be up to p=1.In order to maintain a good reachability for a node moving with a higher speed we need to set a high probability for a rebroadcast (e.g. p=1).

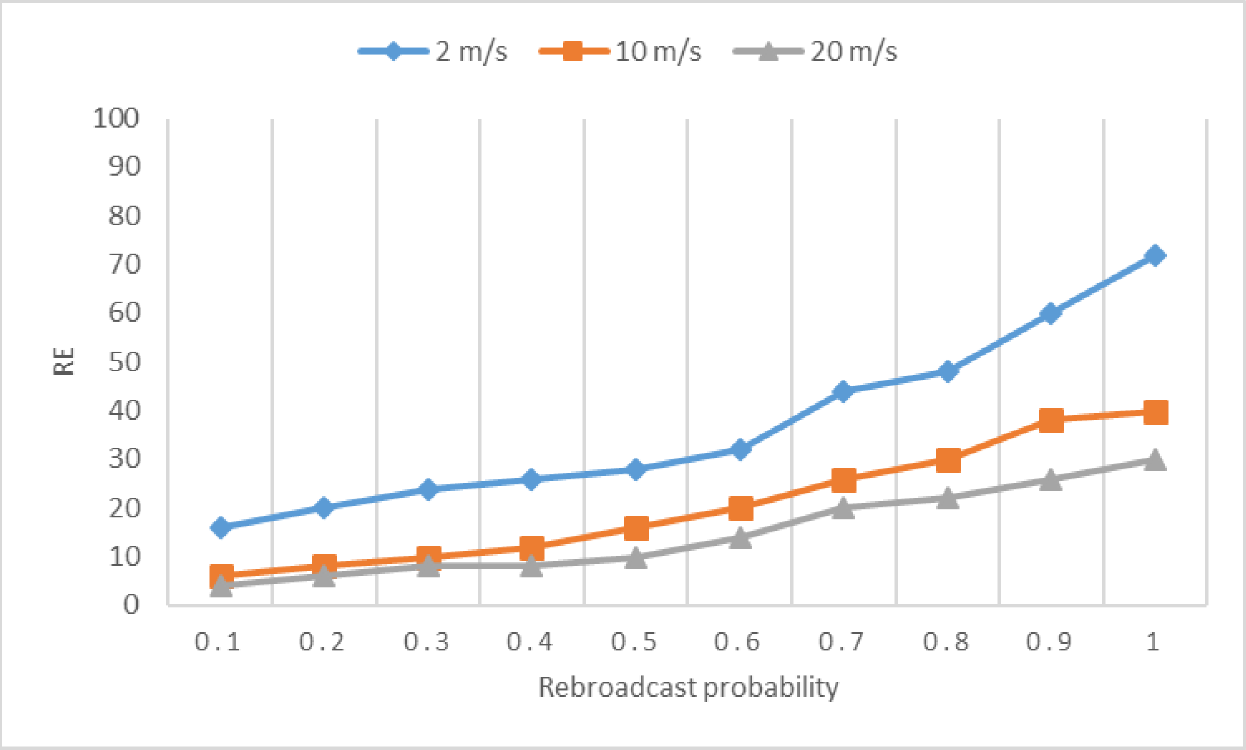


Figure 5-7 RE vs. rebroadcast probability for different node speeds 2, 8, and 20 m/s.

In the next figure we observe the SRB against the rebroadcast probability for three different node speeds and continuous mobility. The node speed has an impact on the observed saved rebroadcast value as seen in the results, hence for a given rebroadcast probability as the node speed increases SRB increases. As an example We notice a 5% decline in SRB when node speed increases from 2 to 20 m/s at the rebroadcast probabilities p=0.55 and to 0% when p=1.The spike of SRB is attributed to the fact that the movement of the nodes may cause a decrease in the retransmission rebroadcast packets. This in succession makes the number of nodes that really transmit the rebroadcast packet decline causing the SRB to be higher.

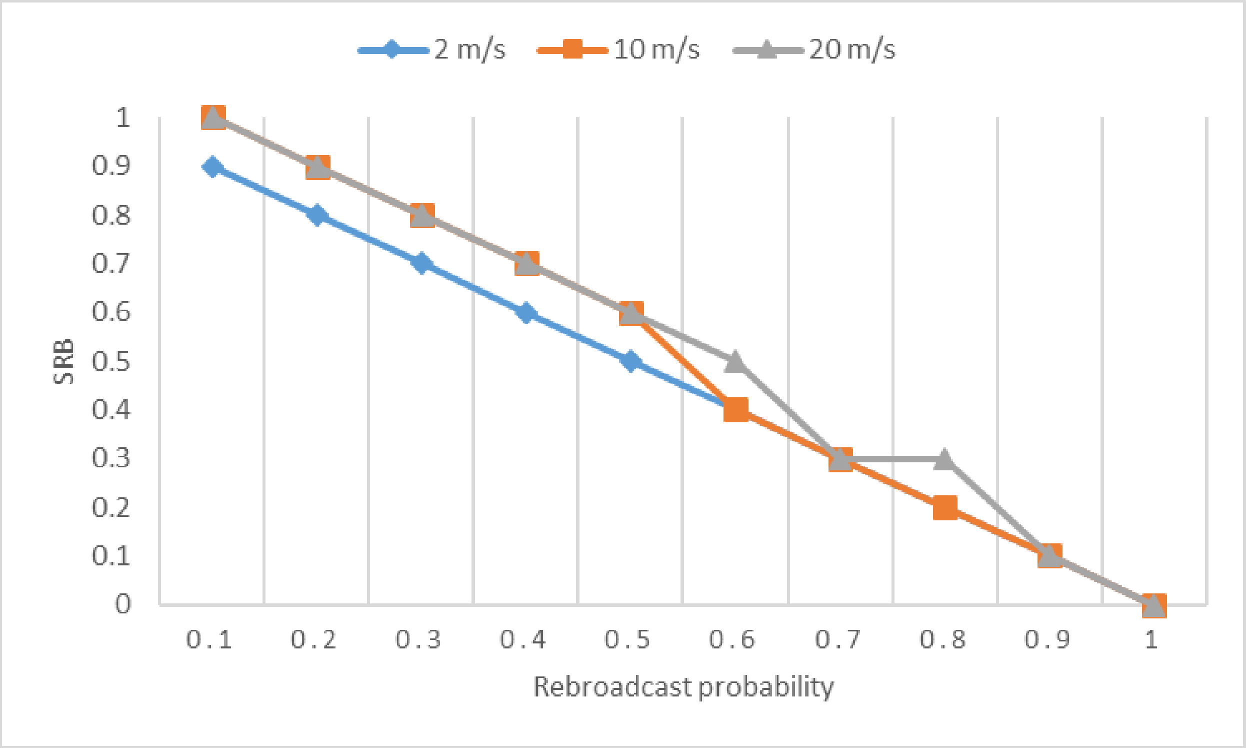


Figure 5-8 SRB vs. rebroadcast probability for different node speeds 2, 8, and 20 m/s.

5.4. Effects of Traffic Load

To be able to have the traffic load in the network varied from light traffic through moderate to heavy traffic. The following rates of broadcast packets generated at the source node are put in mind:

-Light traffic load: 1 packet/s;

- Medium traffic load: 5 packets/s;

- Heavy traffic load: 10 packets/

Figure 5.9 reveals the results of the RE for a varying rebroadcast probability when the traffic load is changing under continuous and consecutive node mobility and a speed of 2 m/s. The figure 2077 also shows that the RE we achieved is going up as the rebroadcast probability is rising, while the traffic load is fluctuating in a cross form manner. We also see the examination of the RE at a high node speed. Figure 5.10 implies that overall RE is heavily and dramatically affected when the node speed increases, as we can see the RE is too low with only 30% when the rebroadcast probability is 1.

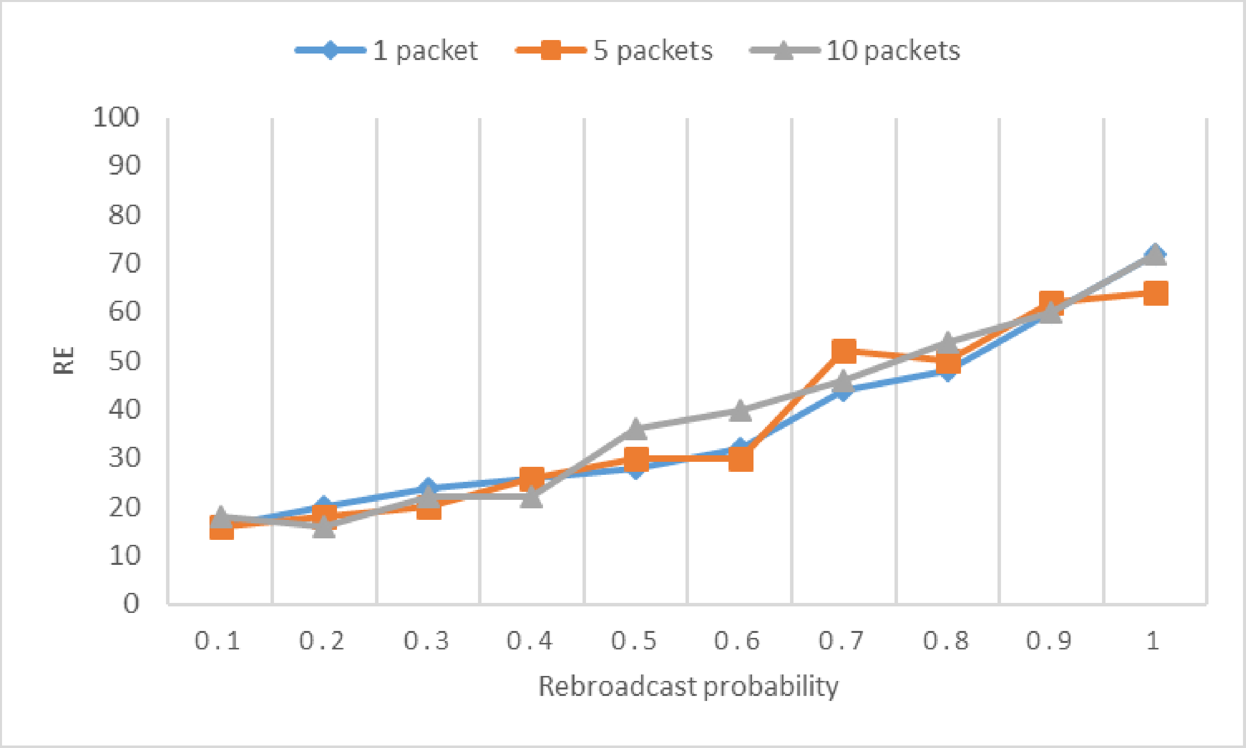


Figure 5-9 RE vs. rebroadcast probability for different traffic loads 1, 5 and 10 packets/s with a node speed of 2 m/s.

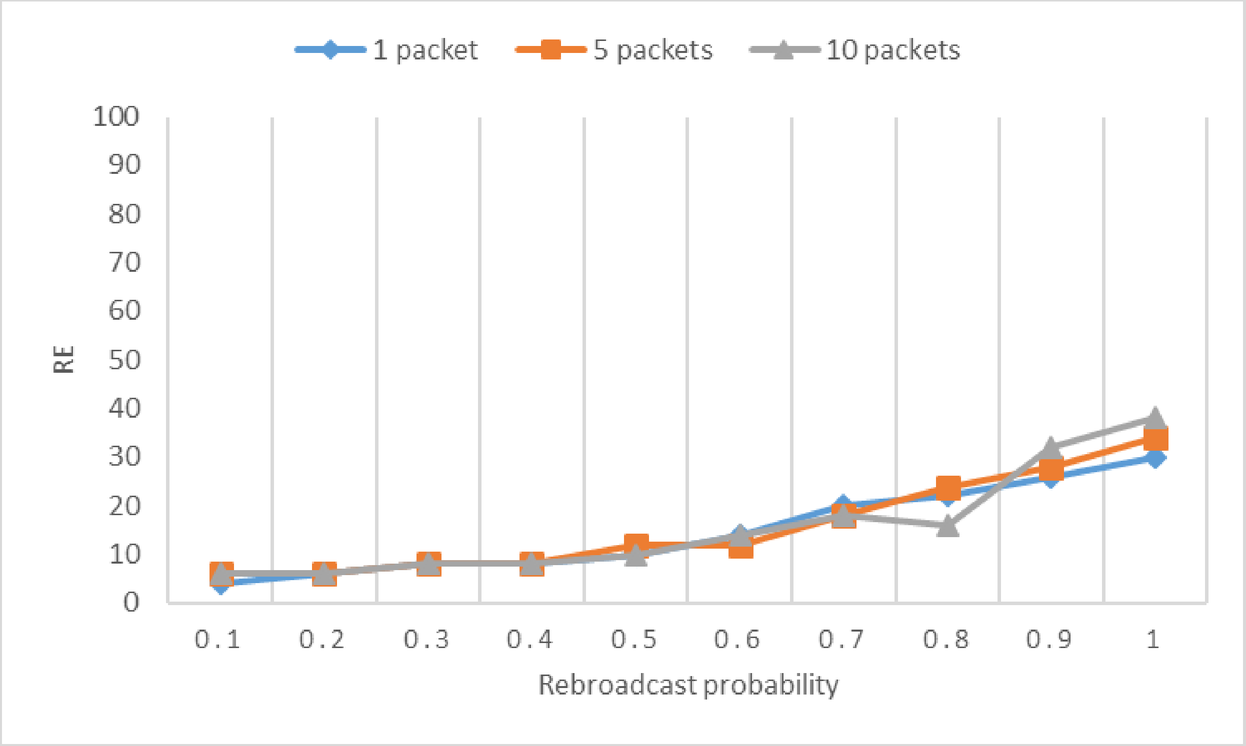


Figure 5-10 RE vs. rebroadcast probability for different traffic loads 1, 5 and 10 packets/s with a node speed of 20 m/s.

The SRB is shown in Figure 5.10 as a function of the rebroadcast probability as the traffic load is varying by the increase in the rate of the broadcast packets from 1, 5, and 10 packet/s while the system size is kept at 50 nodes under continuous and consecutive mobility conditions with 0 second pause time. as the node speed is at 2 m/s. We notice from the figure the suggestion that for continuous mobility and a speed of 2 m/s it is shown that as the rebroadcast probability increases the SRB also increases. When the network is subjected to high rebroadcast probability the SRB starts to decrease. For example, the SRB is the same for three traffic loads when p=0.9 that's because under higher traffic loads, it's really hard and difficult to maintain a high SRB level at the same time the probability is high. The reason being in parallel to the load of the nodes increasing, the number of packets present within the network also increases, and as a consequence collision will have a higher chance of occurring as well as reduced access to the shared wireless medium. Which results in diminishing the number of nodes receiving the broadcast packet, and hence reduces SRB.

We also have examined the SRB for a high node speed of 20 m/s and different traffic loads.
It's shown in Figure 5.12 that for a given rebroadcast probability SRB is affected slightly coherently to the increase of the node speed. This is related to the increasing number of collisions also as the reduced channel access when the network is subjected to the increased traffic loads.

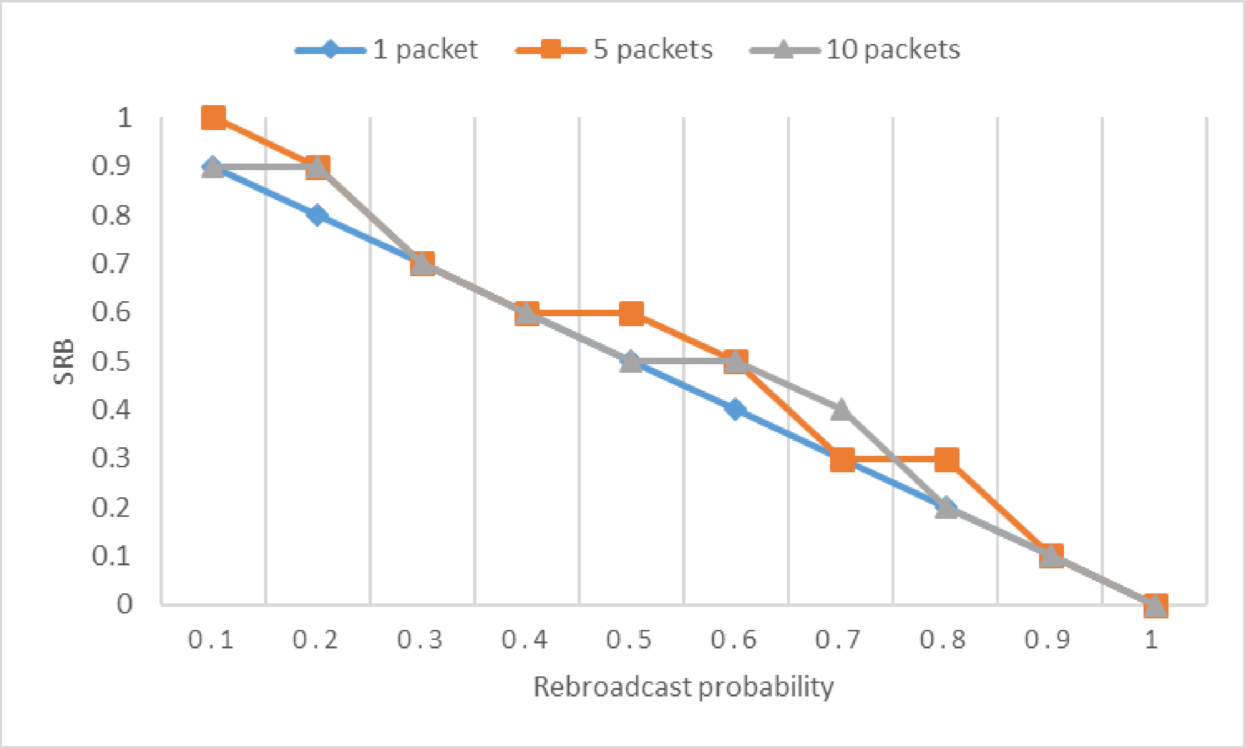


Figure 5-11 SRB vs. rebroadcast probability for different traffic loads 1, 5 and 10 packets/s and node speed of 2 m/s.

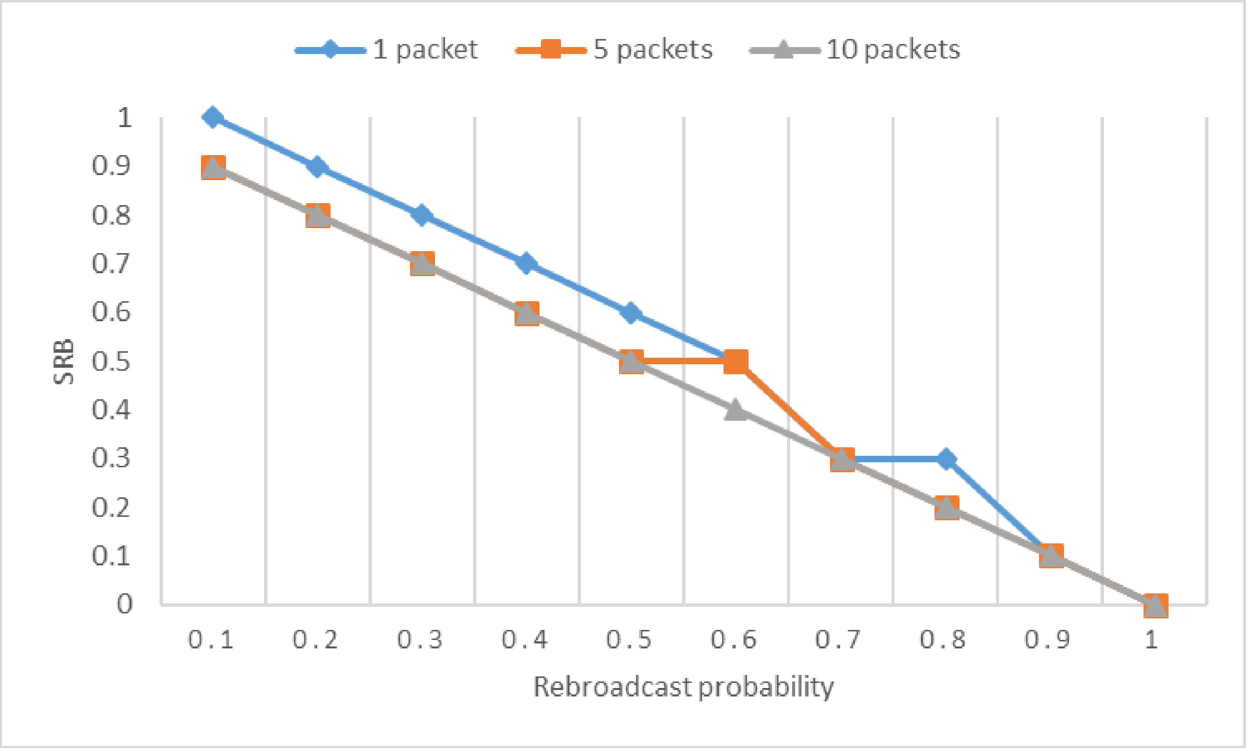


Figure 5-12 SRB vs. rebroadcast probability for different traffic loads 1, 5 and 10 packets/s and node speed of 20 m/s.

5.5. Conclusions

In this chapter we have seen the effects of most of the system parameters in MANETs, including node mobility, traffic load and density on the performance of the probabilistic flooding. Results from ns-3 simulations have revealed that there is a substantial effect on the reachability and saved rebroadcast metrics. As the results have shown for the different rebroadcast probabilities, as the node speed increases, the reachability decreases.

For instance, with a higher node speed of 20 m/s and the rebroadcast probability at 1, the reachability was only at 30%. Frequent and similar performance trends have been noticed and observed when the other system parameters, specifically the network density have been examined in the sense that they have been found to have a great impact on the degree of reachability and the number of saved rebroadcasts achieved by the probabilistic broadcasting scheme. For example, reachability improves by 60% as the node density increases from 25 to 100 nodes. SRB decreases by 30% when the node density increases from 25 to 100 nodes.

Chapter 6: Conclusions and future perspectives

Mobile Ad hoc Networks (MANETs) have received a great deal of attention in the research community in recent years. This has been motivated by recent advancements in mobile computing devices and wireless technology as well as their potential applications that could be realized using such networks, like simple civil and commercial applications, complex high-risk emergency services, battlefield operations, internet of things (IoT), internet of vehicles (IOV), wireless sensor networks(WSN) and smart cities.

Probabilistic broadcast schemes have been shown to be suitable for MANETs due to a variety of benefits such as low overhead, balanced energy consumption, robustness against failures, and node mobility. Many probabilistic broadcast schemes have been proposed by researchers over the last decade. Although probabilistic flooding schemes have been around for a while, only a a few research attempts have been made to analyze their performance behavior in a MANET environment. In our study we have examined some of the most important parameters in a MANET system, such as node mobility, network density, and traffic load, and their impact on the performance of the probabilistic approach to broadcasting in MANETs.

In this probabilistic broadcast, all the nodes use the same fixed probability for rebroadcasting packets in the network. We have successfully implemented the probabilistic broadcast scheme in the well-known ns-3 software simulator. After that, we have carried out extensive simulation experiments in order to evaluate the performance of probabilistic broadcast in MANETs.

After conducting extensive ns-3 simulation experiments, our results have revealed that node mobility has a substantial effect on the saved rebroadcast (SRB) and reachability metrics. The results have shown that for different rebroadcast probabilities, as the node density increases, SRB decreases and reachability increases. For example, reachability improves by 60% as the node density increases from 25 to 100 nodes. SRB decreases by 30% when the node density increases from 25 to100 nodes. Similar performance trends have been observed when the other system parameters, such as network mobility and traffic load, have been examined in that they have been found to have a great impact on the degree of reachability and the number of saved rebroadcasts achieved by the probabilistic broadcasting scheme.

There are several possible directions that can be pursued to further extend our present work, which are listed below.

- **Underwater sensor networks:** These can be used in a variety of commercial and military scenarios. Underwater networks can communicate in a variety of ways, but when nodes are separated by great distances, underwater acoustic communication is the only viable option. The routing protocol must support intermittent connectivity as well as mobile network topologies like autonomous underwater vehicle networks. Without congesting the network, the medium access control protocol must manage medium access with high latency and potentially high packet loss ratios.

- **Smart mobility:** Smart mobility refers to the use of alternative modes of transportation in addition to or instead of owning a gas-powered vehicle. This can take many forms, including ride-sharing, car-sharing, public transportation, walking, biking, and other activities. The demand for smart mobility arose as a result of increasing traffic congestion and its associated side effects, such as pollution, fatalities, and lost time.

- **The Internet of Drones:** The Internet of Drones is an architecture designed to provide unmanned aerial vehicles (UAVs), also known as drones, with coordinated access to controlled airspace. Drone technology has many applications, including on-demand package delivery, traffic and wildlife surveillance, infrastructure inspection, search and rescue, agriculture

References

[1] I. Stojmenovic. Handbook of wireless networks and mobile computing, Wiley, New York, 2002

[2] C-K. Toh. Ad hoc mobile wireless networks, protocols and systems, PrenticeHall, New York, 2002.

[3] He, C. (2003) Throughput and delay in Wireless Ad hoc Networks. Final report of EE359 Class Project, Stanford University.

[4] Corson, S. and Macker, J. (1999). Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. Network Working Group.

[5] Holland, G & Vaidya, N. (2002). Analysis of TCP Performance over Mobile Ad Hoc Networks. Texas A&M University

[6] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distancevector routing (DSDV) for mobile computers. Proceedings of the conference on Communications architectures, protocols and applications, pages 234-244, 1994.

[7] C. Perkins and E. M. Royer, Ad-hoc on-demand distance vector routing. Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), pages 90–100, 1999.

[8] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. Proceedings ACM Symposium Mobile Ad Hoc Networking & Computing (MOBIHOC' 02), pages 194–205, 2002.

[9] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. Proceedings Of ACM/IEEE Mobicom'99, pages 151-162, August 1999.

[10] R. Chandra, V. Bahl, and P. Bahl. Connecting to multiple IEEE 802.11 networks using a single wireless card. Proceedings of Twenty- Third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), volume 2, pages 882 - 893, March 2004.

[11] R. Sivakumar, P. Sinha and V. Bharghavan. CEDAR: A core extension distributed ad hoc routing algorithm. IEEE Journal of Selected Areas in Communications volume 17(8), pages 1454-1465, 1999.

[12] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless Network. Kluwer Academic Publishers, 1996.

[13] D.G. Reina, S.L. Toral, P. Johnson, F. Barrero.A survey on probabilistic broadcast schemes for wireless ad hoc networks. Elsevier. 2014.

[14] Z.J. Haas, J.Y. Halpern, L. Li, Gossip-based ad hoc routing, IEEE/ACM Trans. Networking 14 (3) (2006) 479–491.

[15] Z.J. Haas, M.R. Pearlman, The performance of query control schemes for the zone routing protocol, IEEE/ACM Trans. Networking 9 (4) (2001) 427–438.

[16] C.-C. Shen, Z-Huang, C. Jaikaeo, Directional broadcast for mobile ad hoc networks with percolation theory, IEEE Trans. Mobile Comput. 5 (4) (2006) 317–332.

[17] J. Cartigny, D. Simplot, Border node retransmission based probabilistic broadcast protocols in ad-hoc networks, in: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS '03), 2003

[18] Z. Shi, H. Shen, Adaptive gossip-based routing algorithm, in: IEEE International Conference on Performance, Computing, and Communications, February 2004, pp. 323–324.

[19] Henry Zárate Ceballos, Jorge Ernesto Parra Amaris, Hernan Jiménez Jiménez, Diego Alexis Romero Rincón, Oscar Agudelo Rojas, Jorge Eduardo Ortiz Triviño."Wireless Network Simulation: A Guide using Ad Hoc Networks and the ns-3 Simulator". Apress, 2021.

[20] "What is Network Simulation : Types and Its Advantages". Elprocus. https://www.elprocus.com/what-is-network-simulation-types-and-its-advantages/.

[21] Fredrik Herbertsson."Implementation of a Delay-Tolerant Routing Protocol in the Network Simulator NS-3".22 december 2010.

[22] Ahmed M. Alwakeel."IMPLEMENTATIONS OF THE DTM, DADCQ AND SLAB VANET BROADCAST".Degree of Master of Science, college of engineering, Florida Atlantic University.

[23] Lahcen Touidjine, Bendjeddou Mohammed."Routage avec Qualité de service dans les réseaux Ad hoc".Master degree. Mohamed El Bachir El Ibrahimi de Bordj Bou Arreridj university.

[24] "VMware Workstation". Wikipedia. https://en.wikipedia.org/wiki/VMware_Workstation.29 August 2022.

[25] "Description of the AODV Protocol". Usenix. https://www.usenix.org/legacy/publications/library/proceedings/osdi02/tech/full_papers/musuvathi/musuvathi_html/node12.html

[26] Y. Sasson, D. Cavin, A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks, Swiss Federal Institute of Technology Technical report IC/2002/54, 2002.