Universidade Federal da Bahia
Instituto de Matemática
Departamento de Ciência da Computação

Programa de Mestrado em Mecatrônica

Sistemas Computacionais

THE RELIABILITY OF BROADCASTING PROTOCOLS FOR MOBILE AD-HOC NETWORKS

Talmai Brandão de Oliveira

Dissertação de Mestrado

Salvador
23 de Novembro de 2007

Universidade Federal da Bahia
Instituto de Matemática
Departamento de Ciência da Computação

Talmai Brandão de Oliveira

THE RELIABILITY OF BROADCASTING PROTOCOLS FOR MOBILE AD-HOC NETWORKS

Trabalho apresentado ao Programa de Mestrado em Mecatrônica do Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Mestre em Mecatrônica.

Orientadora: *Profa. Dra. Fabíola Gonçalves Pereira Greve*

Salvador
23 de Novembro de 2007

Ficha catalográfica elaborada pela Biblioteca Bernadete Sinay Neves,
Escola Politécnica da UFBA

Oliveira, Talmai Brandão de
O48r The Reliability of Broadcasting Protocols for Mobile Ad-hoc Networks /
Talmai Brandão de Oliveira. – Salvador, 2007.

— f. : il.

Orientadora: Profr^{a}. Dr^{a}. Fabíola Gonçalves Pereira Greve.
Dissertação (mestrado) – Universidade Federal da Bahia, Escola Politécnica, 2007.

1. Computação Móvel – Simulação (Computador). 2. Sistemas de comunicação móvel. 3. Sistemas de comunicação sem fio. I. Greve, Fabíola Gonçalves Pereira. II. Universidade Federal da Bahia. Escola Politécnica. III. Título.

CDD 20.ed. 621.3845

TERMO DE APROVAÇÃO

Título da Dissertação: *THE RELIABILITY OF BROADCASTING PROTOCOLS FOR MOBILE AD-HOC NETWORKS*

Estudante: Talmai Brandão de Oliveira

Esta Dissertação de Mestrado foi aprovada como requisito parcial para a obtenção do grau de Mestre em Mecatrônica, Universidade Federal da Bahia, pela seguinte banca examinadora:

Profa. Dra. Fabíola Gonçalves Pereira Greve (Orientadora)
Docteur en Informatique, Université de Rennes I, França
Professora Universidade Federal da Bahia

Prof. Dr. Flávio Morais de Assis Silva (Examinador PPGM)
Dr.-Ing, Technische Universität Berlin, Alemanha
Professor Universidade Federal da Bahia

Prof. Dr. Elias Procópio Duarte Jr. (Examinador Externo)
Ph.D. in Computer Science, Tokyo Institute of Technology, Japan
Professor Universidade Federal do Paraná

23 de Novembro de 2007

À Raquel. Você é um presente de Deus na minha vida.

AGRADECIMENTOS

Durante o período deste mestrado tenho contado com o apoio e suporte de uma quantidade tão grande de pessoas, que a aparente simples tarefa de enumerá-las é quase impossível. Inicio essa tentativa agradecendo ao meu Deus por tudo. A Ele a gratidão pela oportunidade de finalizar mais uma etapa da minha vida. Obrigado Jesus. Agradeço aos meus pais, e principalmente à minha mãe, Iara, pelo incentivo e apoio durante toda a minha vida. À Raquel, minha amada esposa, pelo carinho e compreensão durante todas as noites que eu precisava virar a madrugada escrevendo, e que consequentemente não pude estar ao seu lado. Saiba que sem você eu não teria chegado até aqui. Agradeço também aos amigos e irmãos da Igreja Batista Metropolitana. Essas pessoas foram tão importantes nessa conquista que um mero obrigado parece muito pouco.

À professora Fabíola eu agradeço pela orientação, paciência e amizade. Não houve um momento sequer durante esses anos que ela não estivesse atolada de coisas para fazer, mas isso nunca a impediu de dedicar atenção aos seus alunos. Fabíola tem uma visão e sensibilidade invejosa, e não consigo me imaginar com outra orientadora. Perdi conta das vezes que as suas sugestões mudavam radicalmente minha linha de raciocínio; pelas observações de sutilezas que eu jamais enxergaria; pela paciência em rever uma a uma as críticas dos *referees* e no incentivo a melhorar a escrita dos artigos. Ao professor Leizer meus sinceros agradecimentos por me motivar a escrever o que seria meu primeiro artigo internacional.

Agradeço aos outros inúmeros professores, funcionários e alunos do Programa de Pós-Graduação em Mecatrônica pelo apoio prestado. À querida Lúcia Lago agradeço por sempre estar disposta a ajudar os alunos do PPGM em qualquer assunto que estivesse ao seu alcance. Ao professor Iuri Pepe por ter ministrado as duas matérias que mais influenciaram no direcionamento da minha vida profissional. A Victor Costa, amigo e pesquisador, agradeço pelo enorme esforço que teve durante nossa implementação dos protocolos e, principalmente, na montagem do ambiente de simulação. Agradeço também ao Amadeu Júnior que, sem absolutamente nenhuma obrigação, se dispôs a ajudar na montagem de toda a infra-estrutura das simulações nos computadores do *cluster* para mim.

Meus agradecimentos ao CNPQ e à FAPESB pelo apoio financeiro concedido durante quase a totalidade desse mestrado. E principalmente à grande parcela da população brasileira que, sem saber e muitas vezes até sem poder, financiou minha formação.

Louvado seja o nome de Deus para todo o sempre; a sabedoria e o poder a Ele pertencem. Ele muda as épocas e as estações; destrona reis e os estabelece. Dá sabedoria aos sábios e conhecimento aos que sabem discernir. Revela coisas profundas e ocultas; conhece o que jaz nas trevas, e a luz habita com Ele. Eu Te agradeço e Te louvo,ó Deus dos meus antepassados; Tu me deste sabedoria e poder, e me revelaste o que te pedimos...

—DANIEL (Bíblia Sagrada, Livro de Daniel, Cap. 2:20–23)

RESUMO

Titulo: A Confiabilidade de Protocolos de Difusão em Redes Móveis Auto-Organizáveis

Uma rede móvel ad-hoc (*MANET*) é formada por um grupo de dispositivos móveis (também conhecidos como nós) que podem se comunicar diretamente apenas com os nós restritos à área delimitada pelos seus rádio transmissores. Este tipo de rede está impulsionando aplicações inovadoras que combinam a computação móvel, a comunicação sem fio, além de sensores e atuadores especializados. O processo pelo qual um nó envia uma mensagem para todos os outros nós da rede é conhecida como difusão (*broadcasting*). Esta é uma primitiva de comunicação fundamental devido à sua utilização na coleta de informações da rede, no suporte aos algoritmos de endereçamento e no apoio aos protocolos de roteamento. Não obstante a sua importância em redes *MANET*s, pouca atenção tem sido dedicada à satisfação de requisitos de confiabilidade. Tais requisitos buscam garantir a entrega segura e correta de mensagens enviadas através desta primitiva.

Este trabalho estuda o problema de difusão em redes *MANET*s. Diversos protocolos foram propostos para esta primitiva e muitos deles suportam bem a mobilidade dos nós e os problemas de colisão e congestionamento da rede. De fato, pode-se considerar que todos eles são tolerantes a falhas do tipo *fail-stop*. Entretanto, quando cenários de execução mais realistas são considerados, outra classe de falhas – as de omissão – podem ocorrer. Estas, modelam melhor falhas transientes que incorrem durante a comunicação. Desta forma, neste trabalho, avaliamos o desempenho dos protocolos através de experimentos de simulação num cenário de falhas mais realista, caracterizado por omissão. Em conclusão, mostramos que boa parte dos protocolos existentes exibem uma queda significativa nas suas taxas de entrega quando colocados nesse cenário. Como resultado direto dos estudos conduzidos, um novo mecanismo é proposto capaz de aumentar a confiabilidade de protocolos de difusão através da identificação dos melhores vizinhos para comunicação. Este mecanismo, além de suportar crescimento em escala da rede, é capaz de garantir boas taxas de entrega com tempos relativamente baixos, mesmo em ambientes com falhas por omissão. Resultados de simulações demonstram sua eficácia.

Palavras-chave: Redes Móveis Ad-Hoc, Difusão Confiável, Tolerância a Falhas, Comunicação Sem Fio Confiável

ABSTRACT

A mobile ad-hoc network (*MANET*) consists of a group of mobile devices (also called nodes) that are capable of communication restricted to their wireless transmission range. Ad-hoc networks are making a new set mobile applications possible by combining computing power, wireless communication capabilities and specialized actuators and sensors. Broadcasting refers to the process by which one node sends messages to all other nodes in the network. It is an essential operation, since it may be used to collect global information, to support addressing algorithms and to help routing protocols. Ensuring reliable communication between nodes, however, is still a major challenge in *MANET*s due to a large number of problems such as node failures and transient network partitions. But in spite of the importance of the broadcasting primitive, not so much attention has been devoted to reliability requirements beyond best-effort.

This work studies the broadcasting problem in *MANET*s. Existing solutions are able to deal with mobility, congestion and even collisions, but only when under a fail-stop failure model. Unfortunately this model does not adequately represent real scenarios of faults such as link failures, temporary network partitions, topology changes and momentary node failures. Therefore in this work we evaluate – through the aid of simulation experiments – how well broadcasting protocols behave under a more realistic failure model characterized by omission faults. The study conducted here shows that most protocols are highly impacted by failures and are not capable of maintaining high delivery rates. Some even exhibit coverage levels that are unreasonable to expect from broadcasting protocols when placed in such a scenario. As a result of the study conducted, a new mechanism that helps to enhance the capability of broadcasting algorithms to deal with these kinds of faults is proposed. By relying on the mechanism, nodes are able to identify neighboring links that are more reliable prior to transmission. The proposed solution is capable of ensuring good delivery rates, in spite of failures, while maintaining relatively low end-to-end delays. Simulation results demonstrate its efficacy.

Keywords: Mobile Ad-Hoc Networks, Fault Tolerance, Reliable Broadcasting, Reliable Wireless Communication

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Always listen to experts. They'll tell you what can't be done and why.
Then do it.
—ROBERT HEINLEIN

Mechatronic systems – that is, systems that include mechanical, electronic and software components – are commonly seen in modern society, ranging from simple ATM machines to complex fuzzy logic controllers for real-time industrial applications. Although closed loop control over wireless networks has become a popular research topic in recent years [FA02, KA05, WYY06, LMK07], solutions that truly exploit networking capabilities to optimize and enhance the functionality of mechatronic systems are still limited. It is then natural to expect from advanced mechatronic systems that autonomous intelligent units cooperate with one another through wireless networks while performing their tasks, and adapt their local behavior in order to improve overall performance [Rze03, BGH$^+$07]. Whenever this happens, these systems need to rely on a particular kind of wireless network known as ad-hoc.

Mobile ad-hoc networks (*MANET*s) are a special kind of wireless network where the mobile devices (also called nodes) are capable of communication restricted to their wireless transmission range without depending on existing infrastructures, such as base stations. Whenever two nodes are incapable of direct communication with each other they must rely on intermediate nodes to forward messages back and forth between them. Ad-hoc networks are making a new set of mobile applications possible by combining computing power, wireless communication capabilities and specialized actuators and sensors. Well documented examples include military applications, emergency situations and rescue missions [ZL00, GW02, YHE04, LW02]; traffic jam detection via inter-vehicle communication [Bri01]; intelligent environments capable of monitoring temperature, lighting conditions and other variables [LMAH04], as well as in educational circumstances such as when used in classrooms where students and teachers are able to establish a communication network using hand held computers and other mobile devices [ZL00, LW02].

All of these applications benefit from the fast deployment, lower costs and unobtrusive characteristics of this kind of network [BKP03, BCB99]. But while *MANET*s provide clear advantages in cost and flexibility, it brings with it a host of new technical challenges. The mobile devices move and thus the network topology is dynamic, frequently changing. Wireless signal propagation is significantly affected by terrain, obstacles, unanticipated interference and unpredictable fading, causing constant link failures and fluctuating communication channels. In such dynamic networks, the number of participating devices is not previously known, nor is it limited to a maximum size, so solutions must be scalable. Mobile devices may fail to function correctly, but nevertheless the service provided by the network must be able to tolerate such failures. Furthermore, the lack of any previously established hierarchical infrastructure or of a central coordinator forces the devices to organize the network by themselves.

In a nutshell, while mobile computing is not considered a new computing paradigm, dependable mobile computing can be considered as such due to the large number of issues still to be solved. Developing applications capable of dealing with these situations can be a very difficult task, and allowing each application developer to build their own specific solution is a complex and error prone process. This task, nevertheless, can be considerably simplified if fundamental primitives in fault-tolerant group communication are implemented by the underlying communication protocol [VE05]. One of these primitives is known as *reliable broadcast*. Broadcasting in mobile ad-hoc networks refers to the process by which one node sends messages to all other nodes in the network. But the usefulness of a broadcast service is determined by its reliability guarantees [CDG$^+$05]. Therefore the reliable broadcast primitive was created in order to ensure that all nodes receive the same set of messages. The practical implementation of this primitive in *MANET*s, however, has been thought to be unobtainable. Practical evaluations suggest that even with complex collision avoidance mechanisms under low traffic conditions, communication losses can be as high as 20%–50% [KNG$^+$04, CDG$^+$05]. Weaker primitives have then been proposed as alternatives, including protocols that provide at most probabilistic or best-effort guarantees.

In this work, the broadcasting problem in *MANET*s is studied. This is a problem well studied by such a large number of researchers that it might seem at first that no new substantial contribution can be made. However purely theoretical work has many times dismissed and ignored difficulties that are found when one attempts to *build* and *implement* real systems. The maturity of this field depends on the integration of theory and practice, and this work ventures on this new trail.

Initially, existing approaches to broadcasting are presented and explained. The practical implementation of each one of these approaches could only be correctly judged through simulations, or better yet, through real world experimentations. Since the latter was inaccessible to us, a selected number of protocols are then simulated under a scenario where nodes do not fail, but where they are mobile, thus inducing a network topology which is highly dynamic and frequently changing. Then strategies that attempt to raise the delivery ratio of protocols in order to ensure sufficient message coverage while reliably broadcasting are detailed. Protocols that implement such strategies are also simulated under similar conditions.

All of the studied protocols are then introduced to a new omission fault model where temporary node failures occur. This model helps extend simulation scenarios to represent not only transient link availability with node mobility, but also external environmental interferences that are evidenced to affect message propagation, network topology changes and temporary partitions. Performance results from all of these simulations show that although a significant number of broadcasting protocols do exist for *MANET*s, some are unable to cope well with failures under a realistic scenario of momentary failures, and consequently are not able to guarantee message delivery beyond best-effort.

A novel reliability metric mechanism is then proposed and implemented. This mechanism is applied to some of the protocols and simulated for performance comparison. Results indicate an expressive increase in message delivery, even in presence of omission failures, indicating that broadcasting with delivery rates beyond best-effort can be implemented and used in a real world scenario after all.

1.1 THESIS STRUCTURE

Chapter 2 starts off describing with additional details the characteristics of mobile ad-hoc networks, as well as problems commonly experienced in this kind of network. The system model and assumptions used will be explicitly defined. Observations on the simulation environment are also made and the default scenario is described. Chapter 3 focuses on the broadcasting problem in *MANET*s. Approaches to broadcasting are described and selected protocols are simulated. Simulation results are presented and discussed.

Chapter 4 then introduces the reliable broadcasting problem in *MANET*s. A comprehensive study is made on existing strategies that attempt to solve this problem. Protocols that rely on such strategies are then simulated and their performance results are shown. It is only in Chapter 5 that the broadcasting protocols are introduced to a more realistic fault model. The motivations behind this model are obvious, but other less evident reasons are also described. Once again protocols are submitted to simulation runs, this time under this new model, and analyzed. All simulations results helped produce a list of lessons learned from this analysis.

The poor performance of broadcasting protocols indicate a need for additional mechanisms to ensure better delivery rates, and in Chapter 6 an experiment with a hybrid approach led to the development of a new reliability metric mechanism called *R.S.V.P.* The algorithm behind this mechanism is thoroughly described, and formal correctness arguments are produced. Practical simulation results are also presented. Finally, Chapter 7 concludes and lists future work.

1.2 PUBLICATIONS

Until now, this work has produced the following publications:

1) Oliveira, Talmai Brandão de; Costa, Victor Franco; Greve, Fabíola. On the Behavior of Broadcasting Protocols for MANETs Under Omission Faults Scenarios. In *Proceedings of Third Latin-American Symposium on Dependable Computing (LADC 2007)* volume 4746 of *Lecture Notes in Computer Science*, p. 142-159, Springer, 2007.

2) Oliveira, Talmai Brandão de; Greve, Fabíola. Boosting the Reliability of Deterministic Broadcasting Protocols for MANETs. In *Workshop on Dependable Application Support for Self-Organizing Networks (DASSON 2007)* in conjunction with the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007, Edinburgh, UK. *Supplemental Volume of the Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2007. v. 1. p. 7–12.

3) Oliveira, Talmai Brandão de; Greve, Fabíola. The Node Reliability Approach to Broadcasting in Manets: Raising Reliability With Low End-to-End Delay. In *Proceedings of the Fourth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2007)*, 2007, Pisa, Italy.

4) Oliveira, Talmai Brandão de; Costa, Victor Franco; Greve, Fabíola; Schnitman, Leizer. Evaluating the Impact of Faults on Broadcasting Protocols for MANETs. In *VII Workshop on Fault Tolerant Computing (WTF 2006)* in conjunction with the 24th Brazilian Symposium on Computer Networks, 2006, Curitiba, Paraná. *Proceedings of the 24th Brazilian Symposium on Computer Networks*, Porto Alegre : SBC - Sociedade Brasileira de Computação, 2006. v. 1. p. 49–60.

5) Oliveira, Talmai Brandão de; Greve, Fabíola. On Augmenting the Reliability of Broadcasting Protocols for Scalable MANETs. In *Proceedings of the 5th International Information and Telecommunication Technologies Symposium (I2TS 2006)*, Cuiabá, Mato Grosso, 2006. v. 1. p. 22–29.

CHAPTER 2

CHALLENGES IN MOBILE AD-HOC WIRELESS NETWORKS AND SYSTEM MODEL

> *You see, wire telegraph is a kind of a very, very long cat. You pull his tail in New York and his head is meowing in Los Angeles. Do you understand this? And radio operates exactly the same way: you send signals here, they receive them there. The only difference is that there is no cat.*
>
> —ALBERT EINSTEIN

2.1 INTRODUCTION

Ever since Nikola Tesla demonstrated the possibility of wireless communication in 1893, it has been successfully applied in many every-day objects and its use is constantly increasing. Possibly one of the best known examples of wireless communication is that of cellular phones. To be able to talk to anyone while mobile has drastically altered society and the speed with which it changes [Rit03]. More recently, the proliferation of wireless local area networks [Ass99a, Ass99b, Ass03a] has given people access to the Internet in diverse scenarios including homes, offices and even public places such as airports, hotels and coffee shops. In all of these examples, each local geographical area is covered by a fixed transceiver commonly known as base station or access point (AP), that normally has two communication interfaces - one wireless and the other wired - and is responsible to provide radio coverage to mobile devices (also called nodes) within its transmission range, while at the same time acting as a gateway to the wired network.

The architecture of these wireless networks relies on existing infrastructure and an example of such an architecture can be seen in Figure 2.1. Within this architecture, nodes must always first communicate with the AP, even if the communication is meant to be with a nearby wireless node. Notice how node 1 must first send its message to the AP who forwards it to node 2. Since radio energy dissipates over distance, nodes must stay near the AP in order to sustain acceptable communication. Whenever a receiving node is not connected to the same AP as the sending node, it is left to the AP the responsibility of discovering the AP to which the receiver is connected, and of forwarding the message through its wired interface. Wireless communication in this kind of wireless network is most of the time limited to the first (or last) hop.

A mobile ad-hoc network ($MANET$), on the other hand, is a special kind of wireless network where the nodes are capable of communication restricted to their wireless transmission range, but do not depend on existing infrastructure. Thus they are only able to communicate directly with neighboring nodes, as can be seen in Figure 2.2. The lack of fixed and wired AP forces cooperation between the nodes every time a message

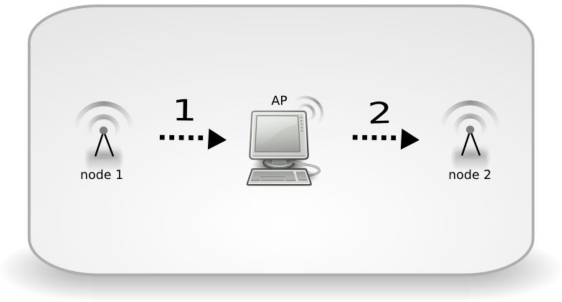


Figure 2.1. Architecture of a mobile network that relies on existing infrastructure

has to be forwarded [1]. Whenever two nodes are incapable of direct communication with each other they must rely on intermediate nodes to forward messages back and forth between them [Eph02]. A simple example of message routing in *MANET*s can be seen when node 3 assumes the role of forwarding messages between nodes 1 and 4.

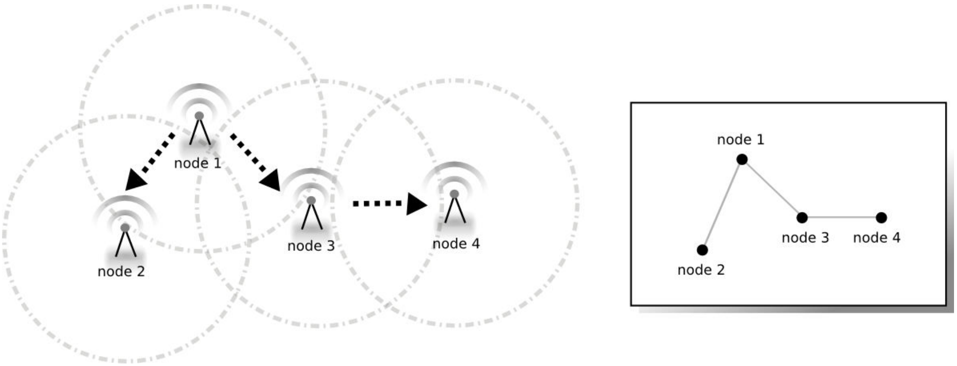


Figure 2.2. Architecture of an ad-hoc wireless network

Additionally, a circle has been placed around the nodes and corresponds to the simplest radio model, known as free-space [Fri46], that is based only on distance across flat terrain, and assumes the ideal propagation condition in which there is a clear line-of-sight path between the transmitter and receiver. All radio communication is perfectly received within this circular range and not at all outside it. Nodes that are located inside this area are considered neighbors. A *MANET* can then be represented as a graph. This has been highlighted in the right portion of Figure 2.2.

It has been shown however that real radios most often exhibit a non-circular and non-uniform communication pattern and thus distance can not be used as the only parameter to determine node connectivity [CJWK02]. Signals tend to decay slowly and there is no exact point where communications cutoff. Real radios are much more elaborate than this simple model and complexities such as variations due to antenna differences, elevation and obstacles can have an expressive impact on the communication between nodes [KNE03].

[1] Although hybrid *MANET*s exist, where the capacity, reliability and performance of the network is somewhat improved [GW02] through the use of strategically placed *AP*.

One can then easily notice that network message capacity varies depending on the chosen architecture. In the former, since every message necessarily gets first forwarded to an *AP*, these can act as centralized coordinators, managing everything from medium access control to collision and contention repression. On the other hand, *MANET*s have no fixed pre-defined central coordinator and thus many problems arise. A project of a *MANET* is then influenced by many technical challenges including fault tolerance, scalability, cost and complexity of network installation and expansion, dynamic network topology, application-specific QoS, real-time and reliability requirements and, last but not least, hardware restrictions such as energy consumption and limited computational power. Thus *MANET*s present a large number of problems that cannot be analyzed nor solved using traditional approaches, demanding revisions when applied to this kind of environment [Sat01, BKP03, EE01]. In the next sections, challenges faced by *MANET*s will be further detailed.

2.2 WIRELESS COMMUNICATION WOES

Wireless communication allows location-independent access to services and data, permitting a highly dynamic network topology. But ensuring effective communication between nodes is a major challenge regarding this kind of communication. Wireless signal propagation is inherently unreliable and significantly affected by terrain, obstacles, material surface reflection and absorption, unanticipated interference, unpredictable fading and link asymmetries, causing constant link failures and fluctuating communication channels - a link that is strong one instant may be weak in the next [BKP03, CDGN05, PM04, BV05, KNG$^+$04].

Testbed experiments have shown that transient links occur frequently, permitting situations where one message is correctly sent and received but the next consecutive message is not [CJWK02]. The actual transmission range of mobile hosts has also been known to vary when in different environments [DW04]. Many other factors also impede correct message transmission and reception as well, including hardware failure, battery exhaustion and node mobility [HJR04, KT03, SAL$^+$03]. Thus wireless communications, and specially ad-hoc wireless communications, have a significantly lower quality than that of a conventional wired network resulting in lower bandwidth, higher error rates, frequent network partitioning, lower network throughput and overall higher communication delays due to retransmissions [GC04, BKP03, BT02].

2.2.1 Shared Transmission Channels

It is assumed that nodes communicate using wireless radios that are half-duplex (nodes can either send or receive, but not both at the same time), use a shared transmission channel where only a simple collision-avoidance scheme can be applied. Each node can sense if other nearby neighbors are using the channel, but while transmitting they have no access to the status of the channel nor are capable of detecting collisions due to the noise it produces. Collision detection featured in wired networks are extremely difficult to achieve in a wireless network since most full duplex radios are still exceedingly expensive.

Shared transmission channels are commonly assumed in other works related to *MANET*s as well, including - but not limited to - [WD05, LW04, VE03, ZA05, WL99, PL00]. The upmost reason for this is that there exists a wide range of physical link technologies, each of which have different design goals. In some cases these goals even have conflicting criterion, like for example high bandwidth, energy efficiency and reliability. So in order to keep the research general purpose, the simplest one is chosen.

Because of the shared transmission channels, nodes are not able to selectively transmit: whenever it sends a message, all of its neighbors receive it. The downside to this is that whenever message transmission from nearby nodes overlap, or when the nodes try to simultaneously transmit messages, collisions and contention may occur preventing correct reception [MGL04, GW02, LW02, RCS05, CDGN05, BV05, PM04]. Common techniques to overcome this includes utilizing an exponential backoff delay whenever a node overhears neighbors' transmission, or applying a small jitter (typically $< 1ms$) before forwarding a message [DW04, CDGN05]. Both of these can easily be used when nodes are equipped with only one wireless interface and the medium access control (MAC) layers are limited to best effort techniques, such as CSMA/CA (*carrier sense multiple access with collision avoidance*) protocols.

More advanced (and expensive) techniques require the use of multiple wireless interfaces, directional antennas or multiple channels allied with a channel hopping mechanism in order to overcome temporary failures and to reduce collisions and contention [MCS$^+$06, WD05, KT03]. The past several years have seen a rapid growth of these kind of solutions, mainly due to high-quality low-cost devices. Most of the focus has been either on relatively long-ranged high-data-rate communication devices and protocols such as WiFi (IEEE Standards 802.11a, b and g [CWKS97]) and Ultra Wideband (IEEE Standard 802.15.3 [Ass03b]) or, on short-ranged low-data-rate communication devices and protocols like Bluetooth [MW05, Ass05].

But while the delay-based solutions are not really adequate for *MANET*s due to long backoff periods when in highly mobile scenarios and inefficient channel utilization [KSD06, CZI03], the complexity, scalability problems and the lack of flexibility on defining topology of the latter makes it expensive and improper for some simple applications requiring low cost, memory, processing and power consumption [MW05]. Some of these issues have been better handled with the advent of new standards, like ZigBee and the IEEE Standard 802.14.4 [Kin03, Ass03c]. Although recent studies have shown that they are only really adequate for very low-data-rate usages (around 1 $message/s$) [ZL04], these have recently been adopted by most vendors/operators to ensure that nodes use compliant radios permitting, even in a heterogeneous wireless network, the use of a common frequency band for communication [ASD$^+$06].

2.2.2 Hidden Node Problem

A common problem in wireless communication is known as the hidden node problem [TK75]. This problem occurs when two or more nodes, which are not within each others' communication range, try to simultaneously communicate with another node known to both. In this case, the two incoming transmissions will collide with each other. Thus, even if the medium is free near the transmitter, it may not be near the intended receiver. This can be seen in Figure 2.3. Node 2 can communicate with both of the other two nodes 1 and 3, but neither node 1 nor node 3 are aware of each others' existence. Since none of them are able to detect an ongoing transmission from the other node, both can wrongly conclude that they may transmit. Nothing can be assumed about nodes' 2 reception, since it could have had either received noise - due to transmission collision - or most likely the transmission which had the strongest signal.

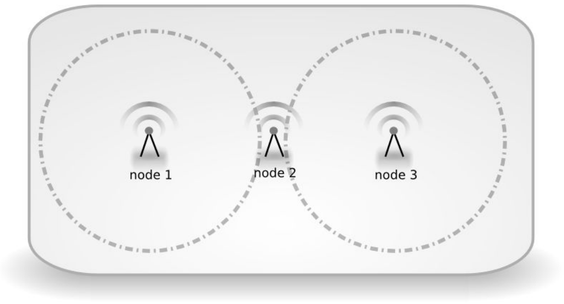


Figure 2.3. The hidden node problem

As the load on the network increases, interference goes up as well, and consequently the efficiency of the network decreases, eventually driving network throughput to zero, since all nodes will try to transmit at the same time. This lack of coordination between transmitting nodes cannot easily be handled since nodes may not be able to directly communicate with each other.

2.2.3 Exposed Node Problem

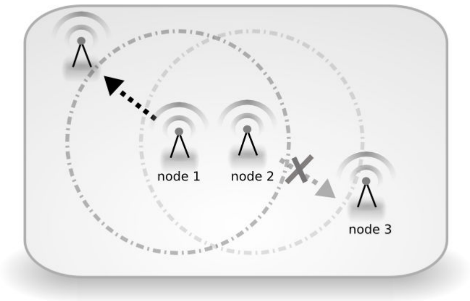


Figure 2.4. The exposed node problem

The exposed node problem takes place whenever a node detects an outgoing transmission and is then unnecessarily restrained from transmitting. This situation decreases the communication rate of the network and is depicted in Figure 2.4. During the timespan that node 1 transmits, node 2, in an attempt to reduce interference, will be prevented from transmitting as well. Although the channel is busy near the transmitter, it may be free near the intended received.

2.2.4 Overall Unpredictability

*MANET*s face a multitude of additional problems, besides the communication issues previously listed, making it impossible to foretell whether or not a node is up or not [FLP85, HJR04]. The mobile nodes move, thus inducing a network topology which is highly dynamic and frequently changing. The number of nodes that will participate in the network is not previously known, nor is it limited to a fixed quantity. At any moment new nodes may be added or current nodes removed. The lack of any previously established hierarchical infrastructure or of a central coordinator forces nodes to organize the network by themselves. Furthermore, nodes are unreliable and can simply fail, be it due to battery exhaustion or to hardware/software malfunction [CDGN05, KT03, SAL$^+$03].

2.3 SYSTEM MODEL

In this section a set of assumptions are made about the environment. These are a realistic approximation of the physical system in which nodes will operate and, therefore, are not too restrictive. It is the system model that provides the basis upon which any future argument must hold. It is also used to evaluate broadcasting protocols described in the next chapters. It also helps influence and guide the design of the reliability mechanism that will be proposed.

In the remainder of this work, a *MANET* is defined as:

Definition 2.1 *(Mobile Ad-Hoc Network) A (MANET) is a graph $G = (V, E)$ in which V represents a set of asynchronous mobile nodes and E represents a set of edges. An edge (x, y) is defined whenever two nodes x and y are able to communicate with each other; so that, x and y are considered as neighbors*[2].

Each one of the nodes has identical software and hardware, and is possibly cooperating towards a common goal without the help of any infrastructure for supporting communication. That is, there are no fixed *AP* and nodes communicate entirely through an ad-hoc network. Furthermore, each node has an unique predetermined identification. There is no common clock or shared memory, and the relative speed and processing capacity of an individual node is undetermined. Nodes are presumed to have no access to position related information (GPS), nor can they determine the relative distance between nodes (through signal strength analysis, for example). Regarding node mobility, no previous knowledge of mobility patterns is assumed to be known, except that movement has some upper bound V_{max} on speed. Additionally, it is assumed that nodes are battery operated, having a limited power supply.

Each node is equipped with a CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) transceiver which can access the air medium. The wireless channels are shared by all nodes and can be accessed by any node at any time although collisions cannot be detected. Since every node can transmit several messages, it is important to be able to correctly identify the sender and to be able to distinguish between other messages sent by a particular sender. In order to make every message unique, it is

[2] The definition of a neighbor is deliberately broad, as this concept will be formally defined shortly.

then assumed that every message transmitted includes at least the identification of the sender and a sequence number generated by the sender that does not repeat itself.

Nodes are uniform (omni-directional antennas and same transmission range), but nevertheless asymmetric links may occasionally occur. Furthermore, a real world scenario characterized by interference introduced by the environment, link instability and transmission failure due to node movement is assumed. Failures related to the communication channels are known as *link failures*, and the following type will be considered:

- A communication channel between two nodes commits an *omission failure* on a message if the message is transmitted by the sender, but any possible receivers (whose wireless transmission range and antenna orientation allows them to receive the message) either never captures the message or the message received is unusable.

Associated with each node is an automaton. Briefly put, an automaton is a machine evolving from one state to another under the action of transitions [BBF$^+$01]. The set of possible transitions defines legal sequences of operations (or steps) that a node can execute. A node is assumed to execute an infinite sequence of such steps, but failures may occur. Informally, node failures are defined as deviations from correct behavior, and are known to be of the following types [HT94]:

- A node commits a *crash failure* when it stops executing further operations.
- A node commits a *send-omission failure* on a message if it completes executing the *send step*, but the message is never transmitted by its transceiver.
- A node commits a *receive-omission failure* on a message if it captures the message through its transceiver, but does not receive the message after completing the *receive step*.
- A node commits a *byzantine failure* (also known as *arbitrary* or *malicious* failures) if the sequence of steps that it executes does not follow the legal sequence of steps defined by its associated automaton, thus exhibiting any kind of behavior.

Throughout this work, however, only omission type failures (including both node and link failures) will be considered, that is, nodes are assumed to fail following an omission fault failure semantic where they may crash, omit to respond, or delay for an unknown amount of time to a request [CDK94]. Formally, a faulty node can now be defined as:

Definition 2.2 *(Faulty Node) A faulty node may react within an unpredictable amount of time or even crash. Nevertheless, a faulty node will not act maliciously.*

Since all of these difficulties may impede correct communication between nodes, what constitutes an acceptable and error-free message exchange is now properly defined.

Definition 2.3 *(Stable Communication Channel) Consider two nodes n_1 and n_2 whose wireless transmission range allows them to send messages to one another. A stable communication channel is said to exist from n_1 to n_2 if at least one of an unlimited number of consecutive transmission attempts made by n_1 to send a message to n_2 is successful.*

Note, however, that for this definition to be true, the stable communication channel is not required to be symmetric, that is no stable communication channel is required from n_2 to n_1. Although this may seem strange at first, asymmetric channels may occur [GKW$^+$02, LW04, LW07, KNG$^+$04, KNE03]. This is due to, among other reasons, local congestion and external interference. Based on the previous definition, the notion of neighbors can now be formally established.

Definition 2.4 *(Local Neighborhood) A local neighborhood (N) of n_1 is defined as the set of nodes that can establish a stable communication channel to n_1.*

In other words, suppose that n_1, n_2 and $n_3 \in V$. $N(n_1) = (n_2, n_3)$ is said to be true if both n_2 and n_3 have a stable communication channel to n_1. Once again we must note that, for this definition to be true, n_1 need not be considered a neighbor of any other node, as there may not exist a stable communication channel between n_1 and another node.

Intuitively it is argued that unless nodes are able to correctly exchange messages with neighboring nodes, they might as well not be considered neighbors. The network is dynamic and nodes may come and go at any time. Since *MANET*s are dynamic in nature, global information exchange is no longer reasonable to expect and support. Nodes must then somehow limit themselves to local information on topology in order to determine neighboring nodes. For this reason, the number of nodes participating in the group at any given time can vary, but might not be globally known. The concept of group membership is not strict, as it is assumed the existence of only a single group in each run to omit group identifiers. In fact, it is the union of all individual local neighborhoods that composes the group.

The only guarantee made regarding overall network connectivity is that no permanent partition occur, but all other scenarios are considered, including partition-less scenarios, eventually disconnected scenarios (where partitions occur rarely but reconnect quickly), and eventually connected scenarios (where partitions occur most of the time, eventually reconnect, but quickly partition once again). Formally,

Definition 2.5 *(Network Connectivity Requirement) A network satisfies the network connectivity requirement if no permanent partition exists between any two nodes of the network.*

Unfortunately, a network that meets the connectivity requirement still cannot guarantee that messages can be correctly sent from one node to the other unless they are local neighbors. Messages sent from non-neighboring nodes must be forwarded by intermediate nodes, and unless there exists a multi-hop stable communication path between the nodes, no further guarantees can be made. Formally,

Definition 2.6 *(Multi-hop Stable Communication Path) A multi-hop stable communication path is said to exist from a node n_1 to a node n_2 when there exists stable communication channels between all intermediate nodes.*

Only if such a path exists it is assumed that information dissemination between any two nodes can then occur within a maximum delay.

2.4 OVERVIEW OF THE SIMULATION ENVIRONMENT

The most adequate way to test a wireless protocol would be to run it over a real ad-hoc network. Unfortunately, state-of-the-art work is still insufficient for deploying sizable networks [WCH06], so most often a smaller scale or a simplified scenario is used. But, these are hardly accessible to researchers, so therefore tests must rely on network simulators.

Such simulators are widely used throughout *MANET* related research, however, its use must necessarily model radio propagation using simplifying assumptions which have raised some issues within the *MANET* research community [KNE03]. A further complication arises from the lack of consensus on the choice of mobility model to use during simulations. Most studies rely on the Random Waypoint Model [BMJ+98] in which nodes move randomly in a pre-determined two-dimensional space with no obstacles. Because of its simplicity, however, it does not adequately capture certain mobility characteristics of more realistic scenarios. Furthermore, as stated on previous sections, *MANET*s face an array of problems that would need to be implemented in the simulator in order to hold in a “real” wireless network, and that most often as ignored or simplified by protocol designers [KNG+04, CJWK02]. All these factors have been taken into consideration throughout this work. Motivated by these findings, and in order to attest that our simulation results were valid, we chose more realistic parameters with which to simulate. The simulation parameters used throughout the simulations in this work, unless otherwise noted, can be seen in Table 2.1.

Simulations were executed with the NS-2 network simulator (version 2.30) [NS-07], where nodes were confined within an area of $1300m$ x $1300m$, and had a constant transmission range of 250 m. While the simple free-space model [Fri46] is simpler to implement, a single direct line of sight path between two mobile nodes is seldom the only means of propagation. The *Two-Ray Ground Reflection* model [Rap01] considers both the direct path and a ground reflection path between the sender and the receiver. The chosen MAC layer follows the IEEE 802.11 specification with no RTS/CTS/ACK for all message transmissions, and the movement pattern of each node follows the Gauss-Markov [LH03] mobility model. This mobility model implements random mobility, but takes additional precautions to eliminate sudden stops and sharp turns. BonnMotion v1.3a [dW07] was used as the mobility scenario generator and up to an 18 *second* pause time was allowed between consecutive node movements.

To allow proper initialization and settling, 3000 seconds of node movement was discarded. Each simulation then ran for a total of 500 seconds. In the first 100 seconds only *“Hello”* type messages were sent to allow for updated local topology information to be exchanged throughout the network. It was during the next 100 seconds that nodes were configured to broadcast data messages as well. For the last 300 seconds no *new* data messages were broadcasted, but nodes still exchanged *"Hello"* messages, retransmitted buffered messages as needed and moved. This was to allow for proper message delivery termination, such as unsent queued messages, as well as possible re-transmission attempts. Each simulation was repeated 20 times to achieve at least a 95% confidence interval for the results. To obtain the value of the broadcast rate, we first simulated a *MANET* where mobility was fixed at $1m/s$, but the broadcast rate varied from 1 $message/s$ to 111 $message/s$, the number of nodes varied from 10 to 160 and node failure varied from 0% to 50%. As expected, too high or too low of a number of retransmissions affect communication. 10 $messages/s$ was the value chosen since, on average, even when taking node failure into consideration, had the best overall effect on every metric measured and permitted the most stable and reliable communication.

We made similar simulations to obtain the node speed, in this case fixing broadcast rate at 10 $messages/s$ but varying node speed between 1 m/s and 20 m/s. Previous studies on the impact of mobility varied the speed of the nodes between 1 m/s and 160 m/s and proved that mobility was a major cause of delivery failure as noted by [WD05, PR99, LW02]. In our case, since we kept the speeds relatively low, the negative effects of mobility were less visible, being node failure much more significant. We ended up choosing the final value of 1 m/s - which represented the scenario where the protocols would be least impacted by mobility. We refer the reader to [dOCGS06] for more details on these simulations.

Simulation Parameters	
Simulator	NS-2 (2.30)
Network Area	1300 m x 1300 m
Transmission Range	250 m
Simulation Time	500 s
# of Trials	20
Mobility Model	Gauss-Markov
Max Pause Time	18 s
Radio Propagation Model	Two-Ray Ground
Broadcast Rate	10 msg/s
Node Speed	1 m/s
Node Message Queue Size	50 msg
Confidence Interval	95 %

Table 2.1. Simulation Parameters

2.4.1 Simulation Metrics

In order to evaluate the performance and behavior of the broadcast protocols when in a fault injected environment we have defined five metrics with which we have divided the evaluation studies. The metrics are *reliability, forwarding ratio, number of gateways, end-to-end delay* and *number of dropped messages*. Since our main priority is analyzing the reliability of the protocols, both energy concerns and protocol overhead-related metrics (such as "hello" message exchange) were not taken into consideration.

- Reliability: A high delivery ratio is the primary goal of any broadcast protocol. It shows not only if the broadcast protocol in question does what it is supposed to do, but will help to show how each protocol deals with failure. Since the number of nodes participating in the simulation is known by the simulator, we are able to extract and analyze the percentage of nodes that received any given message.

- Number of Gateways: Protocol efficiency is given by the number of gateway nodes that retransmit and take an active role in the broadcast. Therefore, an efficient broadcast protocol is one that uses the lowest number of gateways to reach the highest number of nodes, which in turn will lead to a lower number of messages and consequently to less congestion and collision.

- Forwarding Ratio: Number of gateways reveal how many nodes got involved in the transmission of a message, but in order to better understand protocol behavior, this is not enough. After all, the protocol might not be capable of achieving full network coverage, and nodes can fail. Efficiency, therefore, is better measured as a ratio of the number of nodes that received a message (*#receptions*) to the number of nodes that acted as gateways (*#gateways*). That is, *#gateways/#receptions*. We denominate this the forwarding ratio. The higher the forwarding ratio, the greater the number of nodes that had to forward the message. A low forwarding ratio then means that the protocol is efficient, since it uses the lowest number of gateways to reach the highest number of nodes. However, we remind the reader that this does not mean that an efficient protocol is also reliable.

- End-to-End Delay: This metric measures how long it takes any given message to reach every node. It can also be used in conjunction with the others to help understand how congestion has affected the protocols.

- Number of Dropped Messages: Obviously, a congested network causes a rise in the number of collisions and, in most cases, this is the result of an increase in the broadcast rate or in the size of the broadcast messages. Naturally, we chose to measure the number of dropped messages to represent congestion and collision.

2.5 RESEARCH DIRECTIONS

In this chapter, some of the main characteristics of *MANET*s have briefly been outlined, as the fundamental problems that surround wireless communications were surveyed. Any of those issues can easily bring down a wireless communication link between two neighboring nodes. However, as new applications for *MANET*s are suggested, other criterion will also gain importance. Quality of service requirements such as fairness, latency and reliability may be vital to some of these applications, and will need to be ensured by the underlying *MANET* before they can be safely used. Thus, the network architecture must be planned to be able to protect against, or at least reduce the effects of communication failures.

This chapter has also defined the system model that is assumed in the rest of this work. Some observations on the simulation environment have also been made. In particular, the default values have been exposed. The metrics that will be evaluated during all simulation runs were described as well. In the next chapter the broadcast problem in *MANET*s is defined. Broadcasting is a fundamental building block for dealing with routing and reaching consensus [SCS03, LK00, WD05, WC02, VE03, MGL04]. A number of broadcasting protocols were simulated and analyzed as well.

CHAPTER 3

BROADCASTING IN MOBILE AD-HOC NETWORKS

BROAD·CAST
1. To transmit or signal for public or general use.
2. To send out or communicate.
3. To make known over a wide area.
—AMERICAN HERITAGE DICTIONARY

3.1 INTRODUCTION

Broadcasting in mobile ad-hoc networks refers to the process by which one node sends messages to all other nodes in the network. Even in traditional networks, it is an essential operation since it may be used to collect global information, to support addressing algorithms and to implement multicasting. But, particularly in *MANET*s, it is a fundamental building block that helps, for instance, routing protocols propagate routing-related information as well as reaching consensus [SCS03, LK00, WD05, WC02, VE03, MGL04]. Broadcasting is an active research topic and the most significant challenge in its development is compensating between the number of messages broadcast and the number of nodes reached; in finding the balance between redundancy and reliability [ZA05]. On one hand, a high number of retransmission attempts can result in more nodes being reached, but the extra attempts can lead to a rise in the number of collisions, to a reduction in overall network throughput and in increasing transmission delay times. On the other hand, risking too small of a number of retransmissions can lead to only partial network message delivery.

In this chapter, existing approaches to broadcasting in *MANET*s are described. Five different protocols are simulated. Simulation results are discussed based on the previously selected metrics. The protocols chosen considered many of the dynamic aspects expected in an ad-hoc network, and were Blind Flooding, Dynamic Probabilistic Approach [ZA05], Wu and Li's Protocol [WL99], Scalable Broadcast Algorithm [PL00] and Dominant Pruning [LK01]. Initially, each protocol is individually analyzed. Afterwards, they are compared with one another.

All of the simulation values produced are available through logs generated by the simulator. Note that all specific values referred to in the analysis of each protocol are mean values, but the results in the graphs have confidence intervals of 95% plotted as well, although these are extremely small and when plotted on a full scale graph, can hardly be seen. For this reason, the exact plot values for the reliability metric results are also shown.

Simulation runs assume no *node* or *link* failures of any kind, but link breakages may occur due to node mobility. As stated in Chapter 2, a real world scenario is characterized by interference introduced by the environment, link instability and transmission failure due to node movement. These, however, would unnecessarily complicate the simulations as they could possibly have a huge impact of overall protocol results. Failures are then only assumed in Chapter 5.

3.2 THE BROADCAST PROBLEM IN MANETS

The broadcast problem considered throughout this work is assumed to be spontaneous, that is, any node can broadcast at any time. It is also assumed that global network knowledge - although possible to collect - will be inaccurate due to node mobility and failures. In fact, it is assumed that local connectivity information is constantly updated and collected during network operation. Furthermore, since a broadcast is transmitted on top of a CSMA/CA transceiver, the broadcasting is considered unreliable.

A large number of algorithms for broadcasting in *MANET*s exist. Each one of these approaches aim to select the smallest number of neighboring nodes that will become involved in message retransmission, while attempting to reach the highest number of nodes in the network. Nodes that actively take part in the forwarding process are commonly called *gateway* nodes. A node that has received the broadcast message is called a *covered* node and a node that has not yet received the broadcast message is called an *uncovered* node.

If the topology of the network is known and static, the problem of finding the most effective set of gateway nodes has been proven to be similar to the problem of finding the minimum connected dominating set (or *MCDS*) [LK00]. An *MCDS* is the smallest set of gateway nodes such that every node in the set is connected, and all nodes which are not in the set are within transmission range of at least one node in the *MCDS*. Once found, the process of forwarding messages can be handled by the nodes within the *MCDS* set. For example, in Figure 3.1, the *MCDS* is a graph of size equal to 4, formed by the set of nodes $\{3, 6, 7, 8\}$. This has also been highlighted in the figure. All other nodes are within transmission range of at least one of those nodes, and all of them are connected to one another.

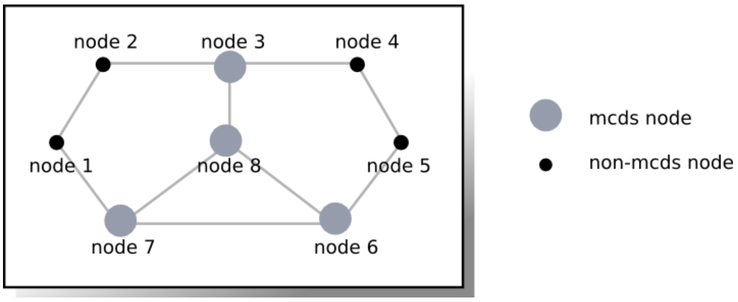


Figure 3.1. Example of a Minimum Connected Dominating Set

By relying on the nodes belonging to the *MCDS*, a message can be forwarded to all nodes in the network with the smallest number of forwarding messages. If the sending node belongs to the *MCDS*, then - assuming network connectivity remains constant and messages are properly transmitted - the number of retransmissions necessary to reach all nodes would be equal to the size of the *MCDS*. If the sending node does not belong to the *MCDS*, then this value would be raised by one. This can be further exemplified by Figure 3.2 where message retransmission is visualized in three different snapshots of network transmission and reception. Notice how after all nodes in the *MCDS* forward the message, the whole network has received the message.

Since the problem of finding a *MCDS* has been proven to be NP-complete [ZA05, GK96, LK00], the use of efficient approximation algorithms is necessary [GK96]. Unfortunately, many of these solutions rely on complete or partial global network topology information exchange such as link/node states, network membership and routing tables. In an ad hoc environment, where the nodes are free to move, it may be impossible to gather such information in a correct and timely manner, and therefore finding a *MCDS* becomes even harder.

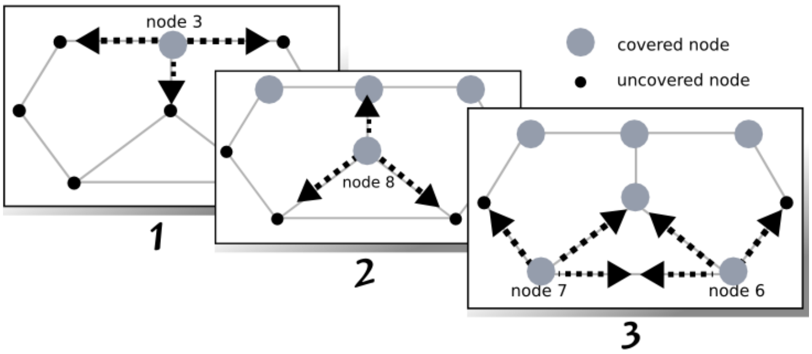


Figure 3.2. Broadcasting with the help of a MCDS

3.3 EXISTING APPROACHES FOR BROADCASTING IN MANETS

The solutions for broadcasting in *MANET*s are commonly classified based on their delivery guarantees, and they can either be *probabilistic* or *deterministic*. Probabilistic protocols are those that guarantee delivery with a certain probability [SCS03, ZA05, LEH04, NTCS99, OVT01, LEH03, CRB01]. Each node determines whether or not it is a gateway based on a probability P. The value of P is determined individually by each node, and when well chosen, a high ratio of delivery can be obtained. Protocols which embrace the probabilistic approach have less constraints and assumptions when compared to deterministic protocols; are usually simpler to implement; and normally have small or little memory requirements.

Deterministic protocols, on the other hand, are those which assume non-probabilistic delivery guarantees [LW02, LK00, WD05, WL99, PR99, HJR04, JM04, PL00, LW04, VE04, OGA06, LK01]. These use knowledge of local topology to determine the gateways. By periodically sending *"Hello"* messages, nodes are able to construct a local view of their neighbors. Different algorithms are used in order to define the most effective set of gateway nodes. Unfortunately, local neighborhood information can (and probably will) be imprecise and inconsistent, since between any two *"Hello"* messages, a node may move, its neighbors may crash, a link may become unstable or many other situations may rise. Therefore, in spite of the name, the deterministic approach when applied to "real world" conditions (with mobility, contention and collision of messages), cannot guarantee exact (deterministic) coverage. Instead, it is capable of a better delivery rate with the need of less redundant message transmission. Deterministic broadcasting protocols can be further classified as either *self-pruning* or *neighborhood designating*. In self-pruning algorithms a node that receives a message decides by itself whether it is a gateway. While in neighborhood designating algorithms it is the sending node who selects the neighboring nodes that should become gateways by piggy-backing this list in the broadcast message.

Regarding existing approaches for broadcasting in *MANET*s, the choice of protocols that will now be presented is deliberately partial. Not only for brevity, but the following algorithms have either produced results that inspired this work or have been widely cited in the literature.

3.3.1 Blind Flooding

Description. One of the simplest solutions to broadcasting in *MANET*s is known as *blind flooding*. Through this approach every message received by a node is forwarded exactly once, and therefore every node becomes a gateway. So in a network with n nodes, n messages will be forwarded. Flooding is categorized as a probabilistic broadcasting protocol where the probability P of forwarding a message is always equal to 1. Much like all probabilistic protocols, it is seen as an option to tackle the lack of determinism of *MANET*s by applying a non-deterministic solution [DGH+87, VB00].

Known Limitations. While there exists many papers that use this naïve approach, it has been shown in [NTCS99, LW02, TNS03] that it leads to:

- Contention: Suppose a node n_1 sends a message. If all neighboring nodes who received such message try to rebroadcast, contention may occur because nodes surround n_1 are likely to be in close vicinity and, thus, contend with each other for the wireless medium. Clearly, the contention is expected to be higher the more dense the network becomes.

- Collision: The CSMA/CA protocol requires a host that intends to broadcast to first sense the channel for existing communication: if the channel is sensed idle, then the node is permitted to transmit; if not, the station has to enter a backoff period prior to transmission. If during this backoff period no communication is heard, the node is allowed to transmit. If multiples nodes exit their backoff period at the same time, multiple transmissions will be sent causing serious interferences. Furthermore, nodes will keep transmitting since no collisions are detected, leading to wasted bandwidth.

- Redundancy: Since every message is broadcast by every node, regardless of the necessity of not, flooding causes unreasonable message redundancy transmissions. By analyzing Figure 3.3, it can be clearly seen how much redundancy is generated when nodes rely on flooding. Observe how, in an optimal situation, only 2 messages would need to be transmitted in order to reach all nodes, while flooding would impose 5 additional transmissions.

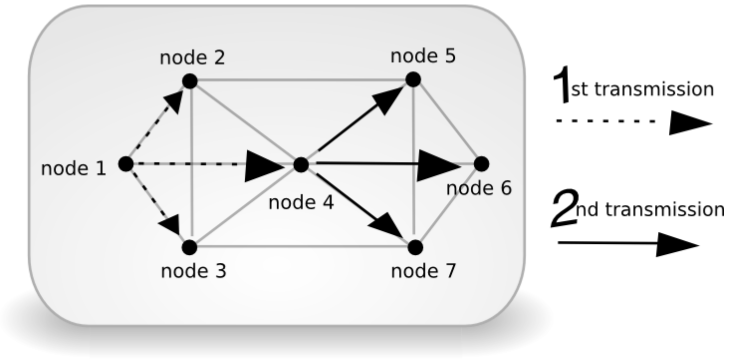


Figure 3.3. Optimal Transmission Situation

These problems have been denominated the *broadcast storm* problem [NTCS99], and are known to interfere in the coverage and increase latency of the broadcast. Obviously, the broadcast storm problem is not limited to flooding and may surface in any broadcasting protocol that does not take additional precautions.

Simulation Results

Efficiency. The efficiency results of flooding are as expected: all nodes that receive a message also retransmit, corroborating the well known fact that efficiency is not one of this protocol's best trait. Invariant on the number of nodes in the network, the forwarding ratio is always 100%[1]. This can be seen in Figure 3.4. The graph on the left plots the number of gateways involved for each simulation run with the specified number of nodes in the network. The graph on the right plots the forwarding ratio results.

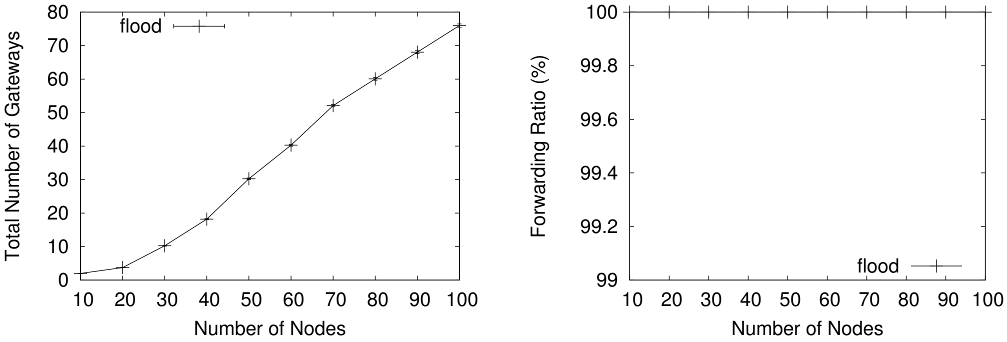


Figure 3.4. Simulation Results: Efficiency of Flooding Protocol

End-to-End Delay and Dropped Messages. End-to-end transmission delays and drop rates, as shown in Figure 3.5, appears to indicate the effects of broadcasting storm problems, with dropped messages reaching values as high as 500 when in a high-density

[1] The reader is reminded that it is a low forwarding ratio determines that a protocol is efficient, since it uses the lowest number of gateways to reach the highest number of nodes.

scenario. While a $0.2s$ end-to-end delay seems acceptable, when latter compared with the other protocols, it will in fact validate that a broadcast storm is occurring.

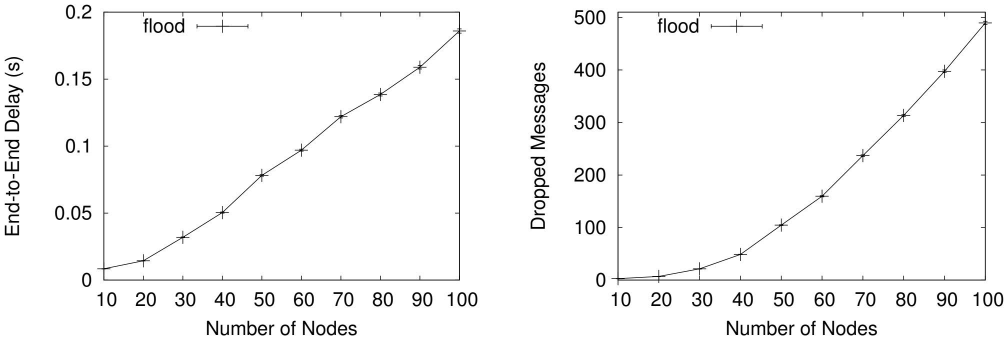
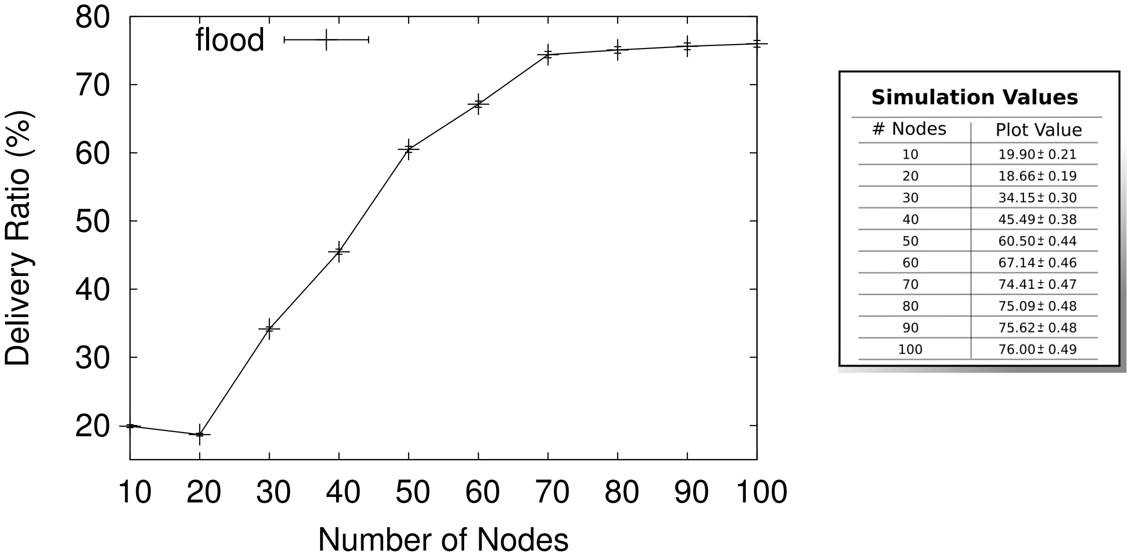


Figure 3.5. Simulation Results: Delay and Drop Rates of Flooding Protocol

Reliability. Reliability results can be seen in Figure 3.6. Discouragingly the delivery ratio nears the 80% mark only in high-density networks, but for the most part does not produce adequate network coverage. Previous research has indicated that reliability and fault-tolerance is only assumed because of the high redundancy [KMG03], but in fact results show that flooding is unable to guarantee message delivery to all nodes. Unfortunately, although for a more static scenario it is not recommended, many extremely mobile and dynamic scenarios can only rely on this approach to broadcast. Actually, flooding is used by many existing broadcasting protocols as a last resort when "all else fails" [OVT01].



Simulation Values

# Nodes	Plot Value
10	19.90 ± 0.21
20	18.66 ± 0.19
30	34.15 ± 0.30
40	45.49 ± 0.38
50	60.50 ± 0.44
60	67.14 ± 0.46
70	74.41 ± 0.47
80	75.09 ± 0.48
90	75.62 ± 0.48
100	76.00 ± 0.49

Figure 3.6. Simulation Results: Reliability of Flooding Protocol

3.3.2 Dynamic Probabilistic Approach

Description. In order to reduce the number of nodes involved with the flooding approach, alternative probabilistic solutions apply various threshold mechanisms to help a node determine whether or not a retransmission attempt is redundant [WC02, NTCS99, TNS03]. Depending on network conditions, a well chosen value for P, will help establish a high ratio of delivery without causing a broadcasting storm.

In this line of reasoning, the authors of [NTCS99] proposed the use of a counter to keep track on the number of times a message has been received. When a node receives a broadcast message for the first time, it will set a counter to 1. Then it will schedule the message for rebroadcasting. However, before the message is sent, every time the node hears the same message being transmitted, it will increase the counter by one. To avoid the broadcast problem, before retransmission it checks the internal counter value. If the counter equals an internal counter threshold, a node assumes – due to the number of excessive broadcasts overheard – that the message has been received by all neighbors and inhibits transmission. Thus, in a dense area of the network, some nodes will not rebroadcast, while in sparse areas of the network, most likely all nodes rebroadcast.

Unlike most approaches to probabilistic broadcasting that assume a fixed probability of P [NTCS99, WC02], Zhang and Agrawal proposed the *dynamic probabilistic approach* [ZA05] which extends the counter-based approach and adjusts the value of P according to the density of the network. The rationale is that in dense areas, a single transmission will already reach a large number of neighbors, therefore reducing the need of retransmissions. High-density areas lowers retransmission probability P, while sparse areas do the exact opposite. Network density is estimated by using an internal counter that increases whenever a node detects a neighbor and decreases periodically.

Known Limitations. The Dynamic Probabilistic Approach algorithm assumes that the network topology does not change drastically so that the probability calculated can be a reasonable approximation of the optional probability for the next message transmission. This, unfortunately, is only the case for networks where speed is low. Furthermore, while the probability of broadcasting is dynamically adjusted, it becomes dependent upon other fixed parameters that need also be carefully selected (like for example, the exact value of timeouts).

Simulation Results

Efficiency. By inhibiting redundant retransmissions, selectively choosing the gateway nodes, using neighborhood topology and density information and even by randomizing the timing of the retransmissions, the dynamic probabilistic approach expected to reduce interference problems. And this it did, as can be concluded by analyzing the efficiency results in Figure 3.7. The total number of gateways was kept at a minimum, as was the forwarding ratio. Meaning that this is a more efficient manner to broadcasting in *MANET*s.

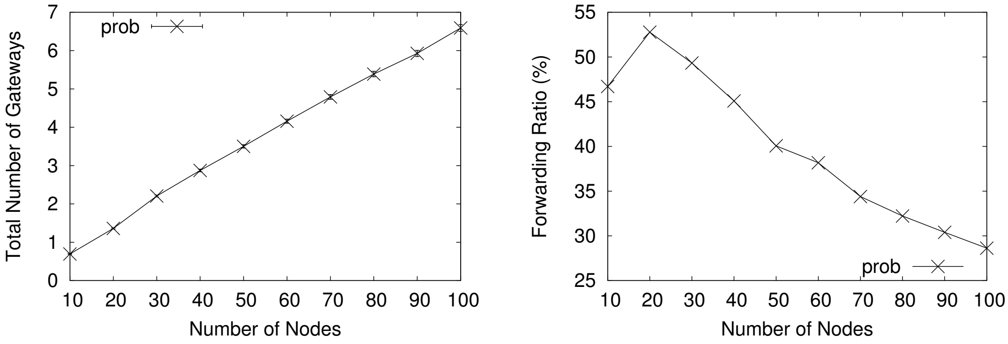


Figure 3.7. Simulation Results: Efficiency of Dynamic Probabilistic Approach

End-to-End Delay and Dropped Messages. In Figure 3.8 the low end-to-end delay can be verified as the highest value barely reaches $0.025s$. This approach also has a low number of dropped messages, even when in a very high density (100 node) network.

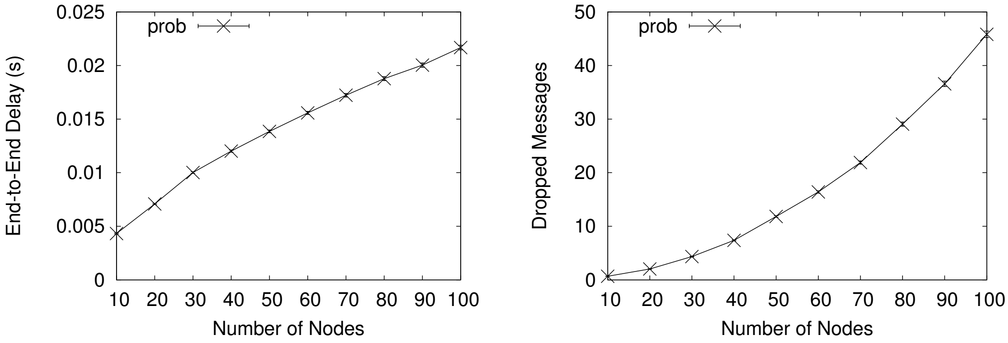


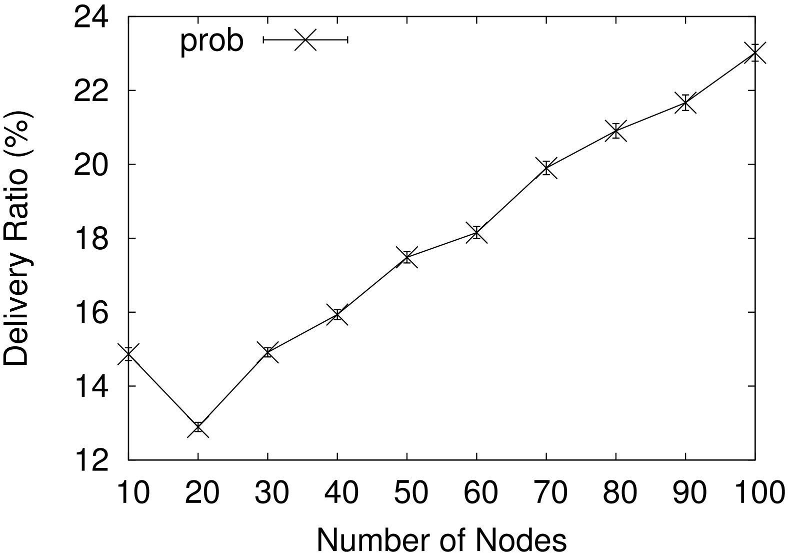
Figure 3.8. Simulation Results: Delay and Drop Rates of Dynamic Probabilistic Approach

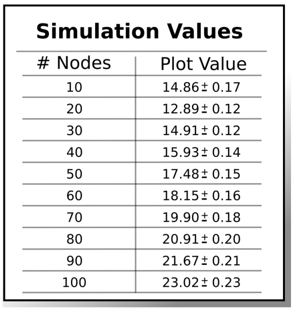
Reliability. While capable of producing favorable results for the previous metrics with a high efficiency and low end-to-end delay, the dynamic probabilistic approach can not do the same for reliability. Although reasonable, results indicate that node density is not the most adequate metric to adjust the probability of retransmission. As seen in Figure 3.9, the delivery ratio barely reaches 24% of the network, even dipping below the 15% mark in low density scenarios.

3.3.3 Wu and Li's Protocol

Description. Wu and Li [WL99] proposed a deterministic self-pruning algorithm to calculate a set of nodes that form a connected dominating set (CDS). According to the authors, their solution reduces the number of forwarding nodes while maintaining a high delivery ratio. They also state that it is scalable, adapting to many diverse network scenarios.

Their gateway selection process is simple and relies on constant neighborhood set exchange between nodes: a node is marked as a gateway if it has two neighbors that are not directly connected. Each node determines locally if it belongs or not to the CDS by analyzing the neighborhood topology information collected. This kind of protocol




Simulation Values

# Nodes	Plot Value
10	14.86 ± 0.17
20	12.89 ± 0.12
30	14.91 ± 0.12
40	15.93 ± 0.14
50	17.48 ± 0.15
60	18.15 ± 0.16
70	19.90 ± 0.18
80	20.91 ± 0.20
90	21.67 ± 0.21
100	23.02 ± 0.23

Figure 3.9. Simulation Results: Reliability of Dynamic Probabilistic Approach

is classified as self-pruning based. The protocol uses a constant number of rounds to calculate the CDS which is directly related to the number of neighbors each node has. Clearly, after neighborhood set exchange, each node knows its 2-hop neighbors. According to the authors, the resultant dominating set includes nodes of the shortest path.

Additionally, two pruning rules are introduced in order to reduce the number of participating nodes:

- Rule 1: States that a gateway looses its gateway status whenever all of its neighbors are also neighbors of another gateway with a higher priority. Priorities are determined based on individual node id and degree (number of 1-hop neighboring nodes). The priority values are used in order to establish a total order among all nodes of the *MANET*.

- Rule 2: States that whenever the neighbors of a gateway node are covered by 2 other nodes that are connected and with higher priorities, than it will become a non-gateway node.

Wu and Li's protocol is well known and has been used and extended by many others [DW03, DW04, WD04]. But these works where all inspired on reducing the number of gateways nodes and increasing broadcasting efficiency, and not on achieving high message delivery ratio. Simulation results clearly show that although older, the original protocol still ensures higher message coverage [DW04]. This is why it was chosen over the newer protocols.

Known Limitations. As previously stated, the resultant dominating set defined by the protocol includes nodes of the shortest path. But, in an high-density ad-hoc environment where the nodes are free to move, the shortest path tends to be the most unstable and prone to link failure due to the *edge effect* [LSL$^+$02].

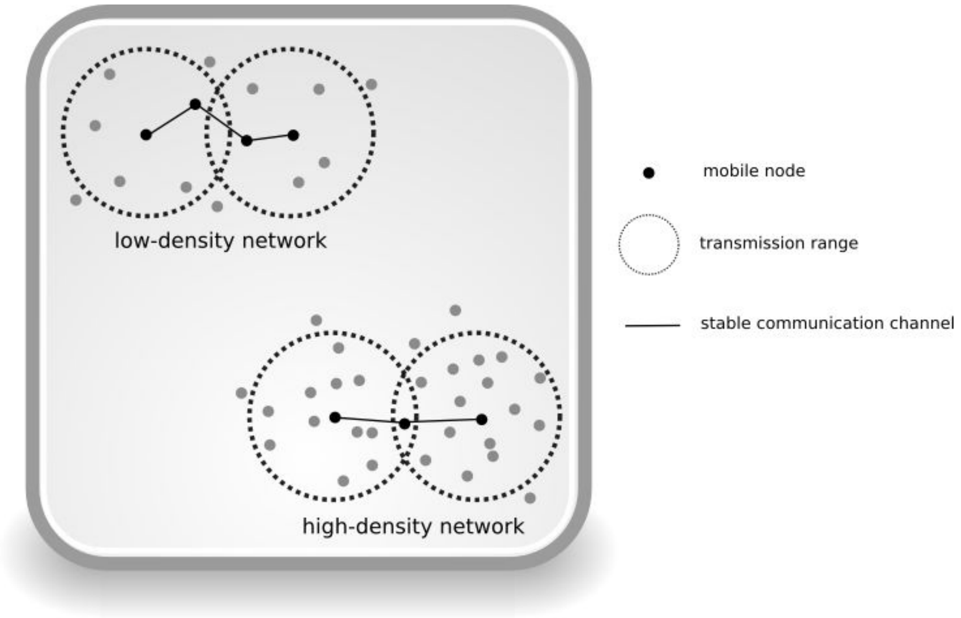


Figure 3.10. The Edge Effect

The edge effect occurs in high-density networks. The authors of [LSL+02] found that shortest paths are often composed of communication channels between the farthest neighbors located on the edge of their transmission range. This can be better observed in Figure 3.10. Notice how in a high-density network the average distance of a stable communication channel is longer and almost the same size of the transmission range, while in a low-density network messages are forwarded by more nodes. This means that even a small movement to the left or to the right of the only forwarding node can break the communication link in the high-density network, but in the low-density network, the forwarding nodes may move the same amount and still not interfere with the communication. This effect is not taken into consideration by Wu and Li.

An additional known limitation is the fact that no guarantees are ever made that a gateway is correctly forwarding the messages and, therefore, no message delivery is ever ensured.

Simulation Results

Efficiency. The simulation results of the number of gateways and forwarding ratio of Wu and Li's protocol can be seen in Figure 3.11. These results indicate that the overall efficiency of the protocol is greater as network density increases. This was an expected behavior since, in a network where nodes are closer to one another, the connected dominating set obtained will most likely involve less nodes. In a network between 40 and 50 nodes, the forwarding ratio decreases from a little over 50% to just under 43%.

End-to-End Delay and Dropped Messages. As more nodes transmit, both the end-to-end delay as well as dropped message increase. This can be verified in Figure 3.12.

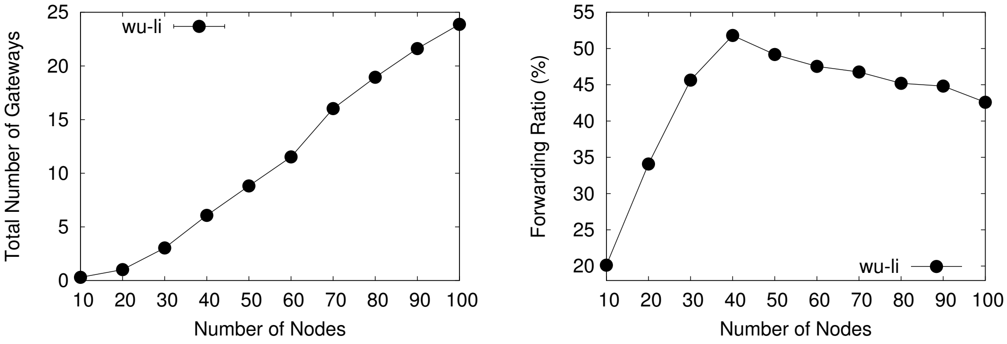


Figure 3.11. Simulation Results: Efficiency of Wu and Li's Protocol

Note how, in a 100 node network, the number of dropped messages reach values as high as 160.

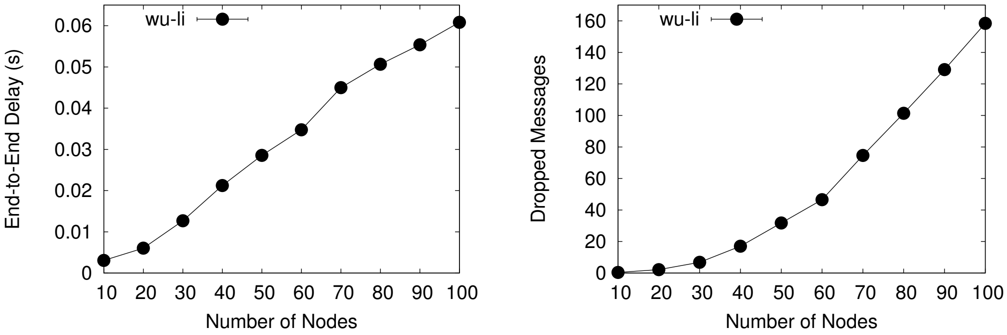


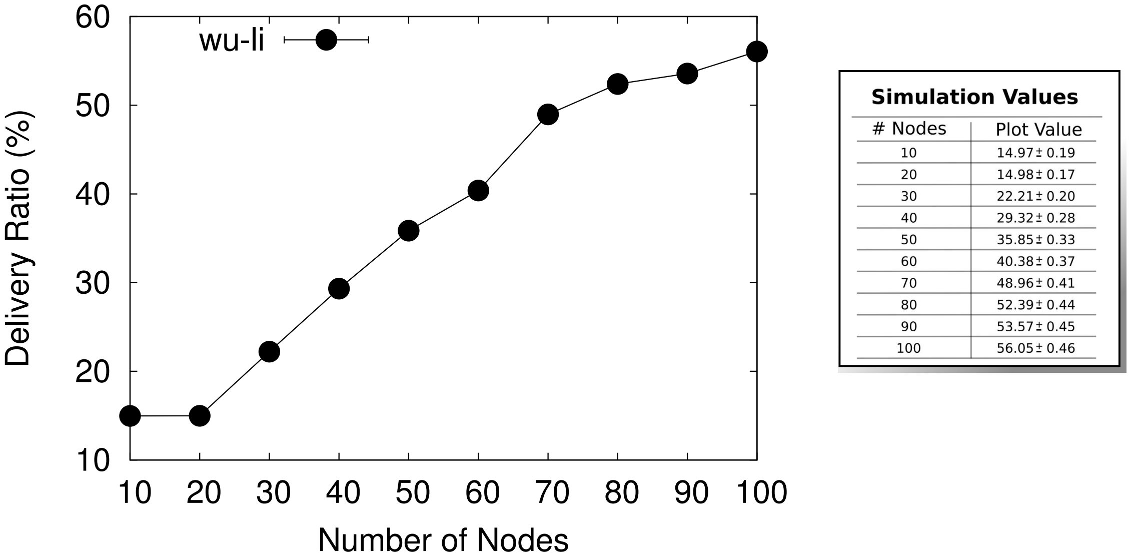
Figure 3.12. Simulation Results: Delay and Drop Rates of Wu and Li's Protocol

Reliability. Apparently the reduction in the number of gateway nodes had its toll in the reliability results of the protocol, as can be seen in Figure 3.13. In most of the simulation runs, less than half of the network received the messages. In fact, not until at least 70 nodes existed in the network was this protocol capable of delivering to over 50% of the network, maxing out at 56%. Surely, this is not an acceptable - nor expected - behavior of a broadcasting protocol.

3.3.4 Scalable Broadcast Algorithm

Description. The main idea of the deterministic self-pruning broadcasting algorithm proposed by Peng and Lu [PL00] is that a node does not need to rebroadcast a message that already has been received by neighboring nodes. In order to determine this, each node needs to have knowledge of local 2-hop topology and of duplicate messages. The Scalable Broadcasting Algorithm (namely SBA) is executed in 2 steps: local neighborhood discovery and data broadcasting.

- Local neighborhood discovery: Consists of exchanging neighborhood sets between local nodes in order to learn 2-hop topology information (exactly like Wu and Li's protocol).



Simulation Values

# Nodes	Plot Value
10	14.97 ± 0.19
20	14.98 ± 0.17
30	22.21 ± 0.20
40	29.32 ± 0.28
50	35.85 ± 0.33
60	40.38 ± 0.37
70	48.96 ± 0.41
80	52.39 ± 0.44
90	53.57 ± 0.45
100	56.05 ± 0.46

Figure 3.13. Simulation Results: Reliability of Wu and Li's Protocol

- Data broadcasting: For this step, whenever a node t receives from its neighbor v a message, before forwarding the message it checks which nodes belong to v's neighborhood. Since v transmitted, node t knows all the nodes that should have received the message. By looking at its own neighborhood set, t can determine if there are still any other neighbors which have not been covered. Only when there exists uncovered neighbors will t schedule a retransmission. But if the initial transmission covered all the neighbors of t, the redundant retransmission is unnecessary and can be canceled.

Instead of immediately retransmitting, the authors proposed a random backoff delay based on the density of the neighborhood. Nodes with more neighbors will have a higher priority and will rebroadcast earlier, thus raising the chances of a single transmission reaching a greater number of nodes. This also helps reduce the chances of occurring a *broadcast storm*.

Known Limitations. One drawback of SBA is that it requires up-to-date neighborhood information. Without it, unfortunately, a node that is receiving a message will erroneously calculate its forward status. But even with perfect topology information, due to mobility and transmission errors, a node has absolutely no guarantees that the message correctly arrived at the other nodes. The backoff delay also has the drawback of longer overall delay to transmit messages.

Simulation Results

Efficiency. The authors of SBA proposed an algorithm that inhibited retransmissions whenever the node assumed that all of its neighbors had already received the same message. Figure 3.14 plots two graphs related to the efficiency of SBA and the results seem to indicate that, unfortunately, this is not what happens during simulation runs. What would be expected is that, as the network density increases, a single transmission would have covered many nodes reducing the number of further retransmissions (raising with it the efficiency of the protocol). But results point to the opposite direction.

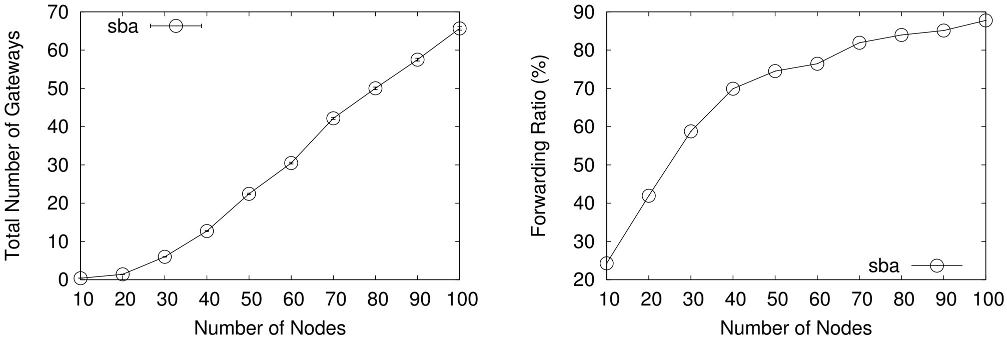


Figure 3.14. Simulation Results: Efficiency of SBA

End-to-End Delay and Dropped Messages. Figure 3.15 seem to clarify what occurs during SBA's simulation runs: contention and collisions. Note the high number of dropped messages and the rise in the end-to-end delay. With 50 nodes, delays range within 0.07*s* and dropped messages at about 100. But with 100 nodes these values jump to 0.18*s* and 450.

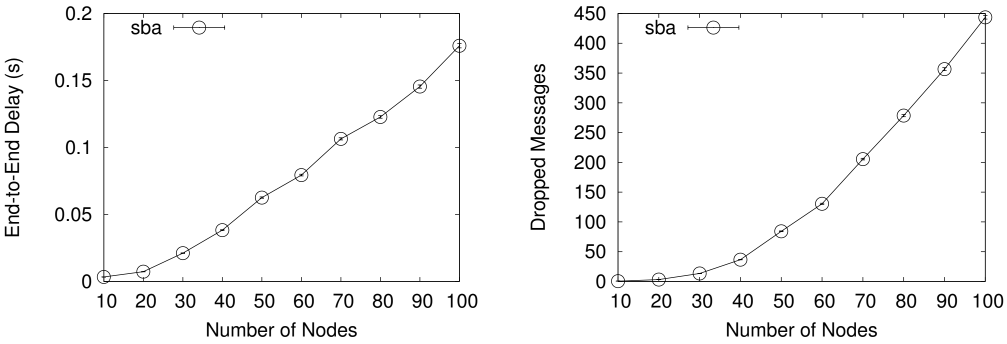
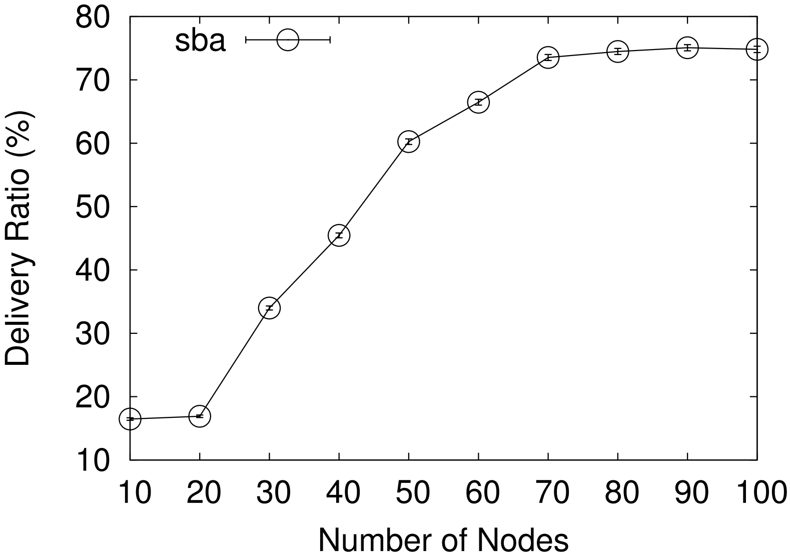


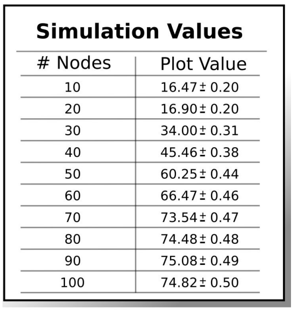
Figure 3.15. Simulation Results: Delay and Drop Rates of SBA

Reliability. The reliability results of SBA are shown in Figure 3.16. It shows that delivery ratios only reach satisfactory levels when at least 50 nodes are present in the network; which is when at least 60% of the network is covered. But examining this information from another angle, it can be noted that it takes the participation of at least 75% of the network to allow coverage at such rates. Worse still, with less than 70% of network participation, delivery rates are unable to reach half of the network.

3.3.5 Dominant Pruning

Description. The dominant pruning algorithm (namely, DP) [LK01] is a deterministic neighborhood-designating broadcasting protocol that uses 2-hop neighborhood




Simulation Values

# Nodes	Plot Value
10	16.47 ± 0.20
20	16.90 ± 0.20
30	34.00 ± 0.31
40	45.46 ± 0.38
50	60.25 ± 0.44
60	66.47 ± 0.46
70	73.54 ± 0.47
80	74.48 ± 0.48
90	75.08 ± 0.49
100	74.82 ± 0.50

Figure 3.16. Simulation Results: Reliability of SBA

information to reduce redundant transmissions. In DP, whenever a node sends a message, it selects the smallest number of forwarding nodes that can cover all nodes in a 2-hop distance. That is, when node j wants to send a message, it selects from his local neighborhood set $N(j)$ the minimum number of nodes that should act as gateways to reach all nodes in $N(N(j))$. By determining this, it will then loop through $N(j)$ and select the smallest number of nodes that are able to guarantee coverage. These nodes will become forwarding nodes. Since it is a neighborhood-designating protocol, it piggybacks this list in the broadcast message.

Although there exist newer algorithms that extend DP, such as [LW02] and [WD05], where simulation results show that neighborhood information is more effectively used (lower number of gateway nodes) and even more redundant messages are eliminated, they unfortunately seem to produce results which have lower delivery rates. This obviously makes sense since it is the redundant messages that help raise message coverage.

Known Limitations. Strangely the DP protocol assumes that when node j first transmitted the message, all of its 1-hop neighbors (which is the set $N(j)$) correctly received the message (which inherently means it assumes that no errors occur!). It also assumes that when a selected gateway node $k \in N(j)$ forwards the message, all of its 1-hop neighbors ($N(k)$) will correctly receive the message as well. But in a fault-enabled environment one cannot just assume that message transmissions are always correct and this should produce perceptible consequences to message delivery in a real-world scenario.

Simulation Results

Efficiency. Dominant pruning's neighborhood-designating approach seems to produce a pretty efficient protocol, as can be seen in Figure 3.17. Forward ratios are kept low, reaching for most of the simulation values below 40%, and peaks off at 42%. The number of gateways is relatively low as well.

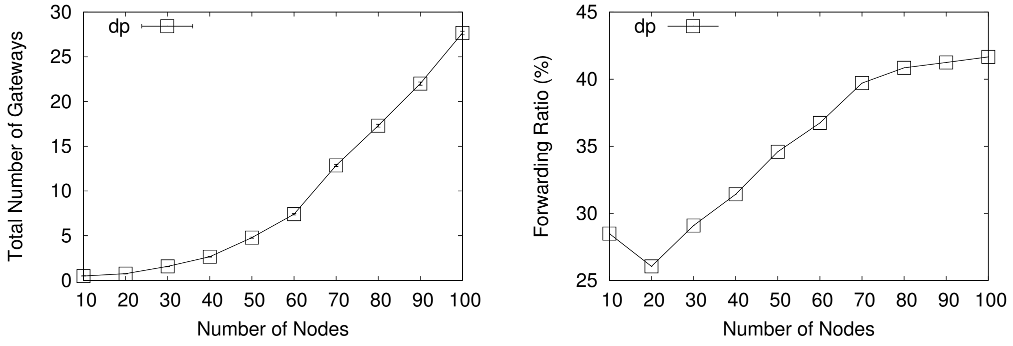


Figure 3.17. Simulation Results: Efficiency of DP

End-to-End Delay and Dropped Messages. Figure 3.18 plots the results of end-to-end delay and of dropped messages. Both of these metrics increase with the number of nodes in the network. With less than 50 nodes, end-to-end delay is lower than $0.02s$, but this values grows to almost $0.08s$ when in a high-density scenario. Dropped messages, when at most 60 nodes exist in the network, are limited to 50, but bolts to 200 messages when 100 nodes are in the network. It is presumed once again that collision and contention related problems start to surge when network density increases.

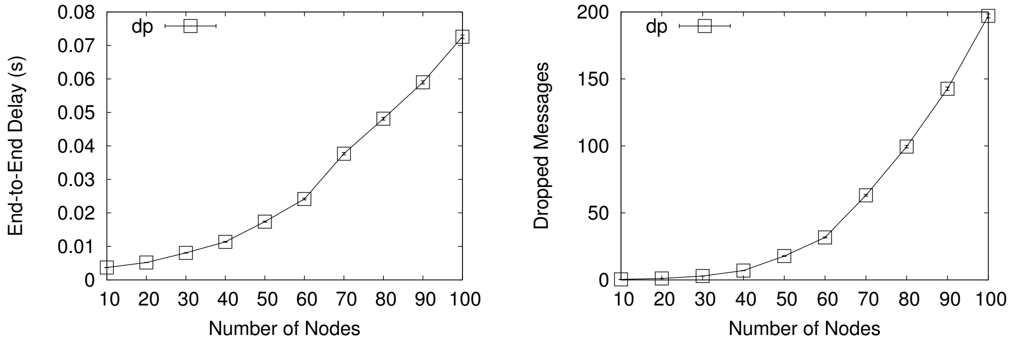
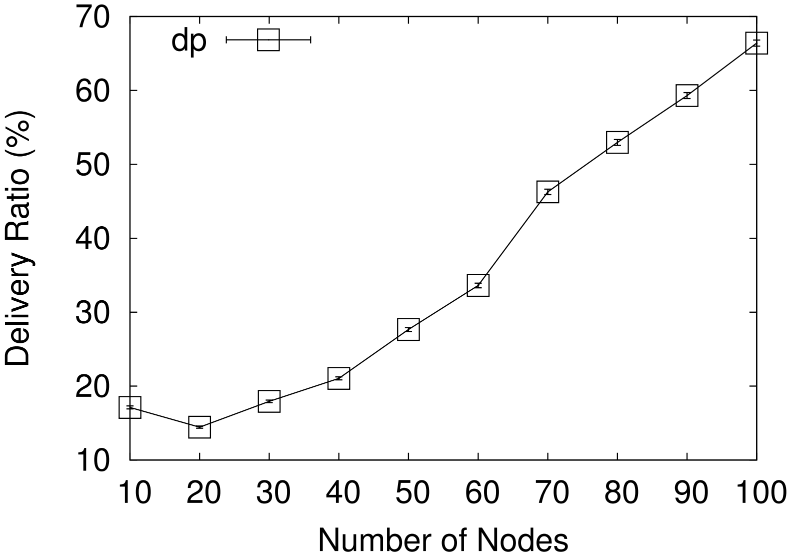
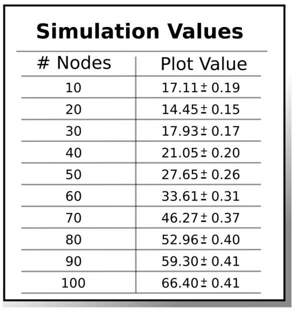


Figure 3.18. Simulation Results: Delay and Drop Rates of DP

Reliability. It seems that the reliability of DP is dependent on the network density. Figure 3.19 reveals that at least 70 to 80 nodes are needed to achieve a 50% delivery ratio. With 60 nodes, network coverage barely reaches 35%. When simultaneously evaluated with the efficiency results, it can be noticed that only when 40% of the network becomes a gateway is that delivery ratios start to increase. Nevertheless, even in a 100 node network, DP is only capable of ensuring delivery to 67% of the nodes.





Simulation Values

# Nodes	Plot Value
10	17.11± 0.19
20	14.45± 0.15
30	17.93± 0.17
40	21.05± 0.20
50	27.65± 0.26
60	33.61± 0.31
70	46.27± 0.37
80	52.96± 0.40
90	59.30± 0.41
100	66.40± 0.41

Figure 3.19. Simulation Results: Reliability of DP

3.4 PERFORMANCE COMPARISON OF PROTOCOLS

Figure 3.20, Figure 3.21 and Figure 3.22 presents the results of the simulations of all protocols in a attempt to directly compare their performance. Comparisons will be limited to three of the metrics: efficiency, end-to-end delay and reliability.

Efficiency. Figure 3.20 presents all the protocols' efficiency results plotted on the same graph. While Flooding has the worst efficiency of all protocols regardless of the number of nodes in the network, Probabilistic Broadcasting is capable of maintaining efficiency values even better than some deterministic protocols (with the exception of very sparse networks). Dominant Pruning also has very efficient results. The biggest surprise would have to be SBA, since one would expect that messages overheard would lead to less message retransmissions. But results show a different reality.

End-to-End Delay. The end-to-end delays of all protocols can be compared in Figure 3.21. Probabilistic broadcasting has the smallest delays and is the most uniform of all results. The deterministic broadcasting protocols all need to exchange local topology information, which by itself already adds an extra delay. So the results from both Dominant Pruning as well as Wu and Li's protocol are as expected. But SBA resembles too much Flooding, and clearly indicates that a broadcasting storm is occurring.

Reliability. The reliability of broadcasting protocols is the most important metric of all, but disturbingly, even in a failure-free scenario, simulation results show that complete network coverage is never obtained. Figure 3.22 presents the results of all protocols where clearly, delivery ratios never even reach 80% of the network. In fact, the most efficient protocols had the worst reliability results, indicating that algorithms that incorrectly inhibit message retransmissions can cause more loss than gains.

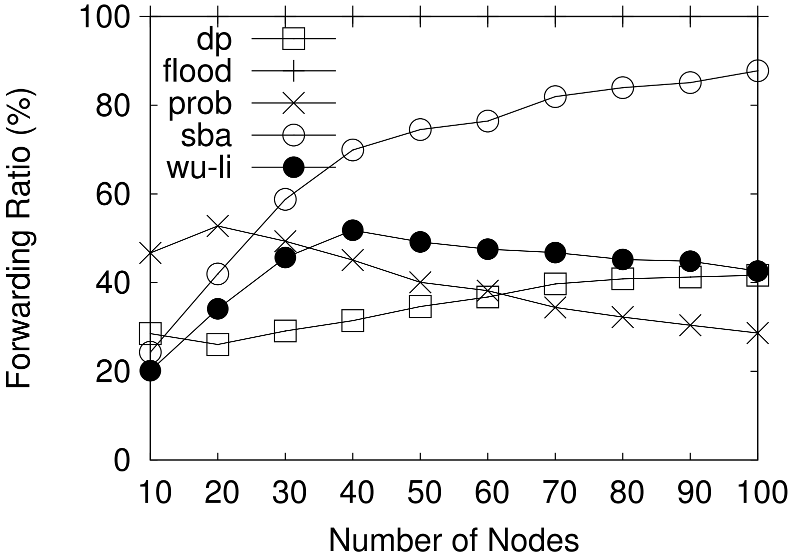


Figure 3.20. Simulation Results: Efficiency of all Protocols

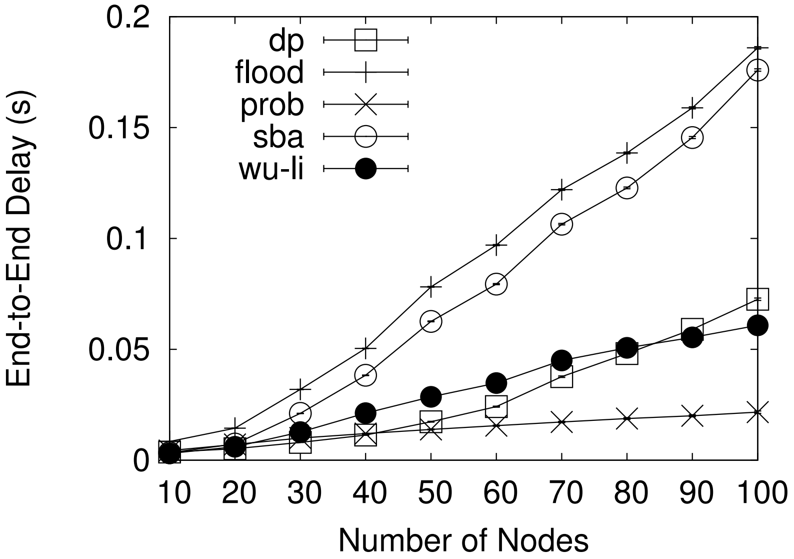


Figure 3.21. Simulation Results: End-to-End Delay of all Protocols

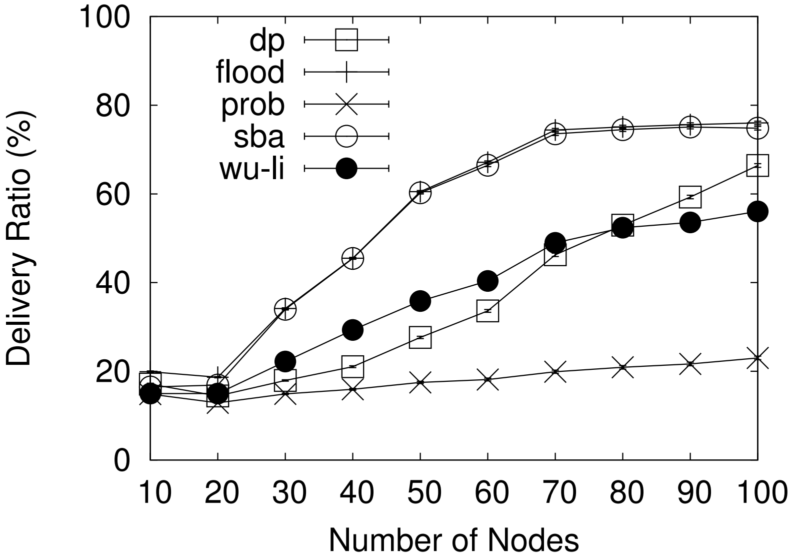


Figure 3.22. Simulation Results: Reliability of all Protocols

3.5 CONCLUSION AND SUMMARY

This chapter has introduced the broadcast problem in mobile ad-hoc networks. Existing approached were categorized and explained. Five protocols – Blind Flooding, Dynamic Probabilistic Approach [ZA05], Wu and Li's Protocol [WL99], Scalable Broadcast Algorithm [PL00] and Dominant Pruning [LK01] – were simulated under a scenario where nodes do not fail, but where they are mobile, thus inducing a network topology which is highly dynamic and frequently changing. As each of these protocols were described, known limitations were also disclosed. The choice of protocols, albeit small, included a wide array of different techniques to broadcasting, and conclusions made, therefore, can be easily generalized to most (if not all) broadcasting protocols for *MANET*s. Unfortunately, these simulation results showed that complete network coverage is never obtained, and delivery ratios never even reach 80% of the network.

In the next chapter strategies that attempt to raise the delivery ratio of protocols are presented, and the simulation results of protocols that use such strategies are discussed. More importantly, the reliable broadcasting problem will be analyzed. This is the first step in the direction of enabling complex and safety-critical applications to benefit from a mobile ad-hoc network.

CHAPTER 4

STRATEGIES FOR ENSURING SUFFICIENT COVERAGE WHILE BROADCASTING IN MOBILE AD-HOC NETWORKS

> *To effectively communicate, we must realize that we are all different in the way we perceive the world and use this understanding as a guide to our communication with others.*
>
> —ANTHONY ROBBINS

4.1 INTRODUCTION

As shown in Chapter 3, even when no nodes crashed broadcasting protocols were unable to guarantee message delivery to all nodes in the network. Even though it has been pointed out that providing total reliability for broadcasting in *MANET*s is impractical and unnecessary when the physical communication channels are error prone [LW07], poor results such as the ones presented in the previous chapter continue to push the development of strategies that attempt to raise the delivery ratio of broadcasting protocols in such a way as to ensure higher delivery rates beyond best-effort guarantees. In the *MANET* community, protocols that use such strategies are known as *reliable broadcasting protocols* and this will be the focus of this chapter.

Initially a discussion about the general aspects and behaviors of the protocols when taking reliability into consideration will be promoted. Reliable broadcasting strategies will then be presented and discussed. Although a great number of these exist, many are not adequately designed for *MANET*s, and the reason for this will be highlighted in the next section. Further on, acceptable strategies – that is, the ones that have been properly adjusted for *MANET*s – are presented. Finally, three protocols that rely on such strategies will be chosen to be simulated and analyzed under the same set of scenarios as the protocols in Chapter 3.

4.2 RELIABLE BROADCASTING STRATEGIES IN MANETS

There exists a number of papers that propose solutions that attempt to ensure higher delivery rates while broadcasting in *MANET*s. However, a great number of these have assumptions which can not be justified nor accepted. It is reasonable to say that some focus not on a practical protocol that can be implemented and used, but rather on general design and evaluation ideas. Many do not assume a real world scenario characterized by interference introduced by the environment, link instability, and transmission failures due to node movement.

- The authors of [CSM03], for example, do not indicate that any kind of mobility is assumed, on the contrary, nodes seem fixed. As is the case in [WCK02].

- In [TXZ04], an offline algorithm is proposed to predict the worst case link durations in order to detect more durable paths. Based on the results of this algorithm, a broadcast is guaranteed to deliver messages. But this is obviously not usable in a real world situation.

Few solutions to the reliable broadcasting problem require that the number of nodes that will participate in the network be previously determined or even fixed, preventing addition of new nodes or even the removal of existing nodes.

- Both [SNGC05] and [VE03] are examples of where the complete network membership is known by all nodes, and no nodes fail.
- In [PR99] and [JL02] nodes must not permanently fail as well.
- In the protocols presented in [VE04, VE05], it is assumed that there exists a fixed number of nodes in the network and they use this knowledge to build a vector to identify message distribution to every node. Once again, this limits the utilization of such protocols in realistic situations.

Other solutions to the reliable broadcasting problem even require specialized hardware such as GPS receivers, multiple antennas, multiplexing capacity of transmitting frequencies and ranges, or even the support of fixed gateways. We can list many protocols that fall under this class of requirements.

- For example, a solution using two different transmission ranges (through two different wireless interfaces) is proposed both in [WD05] and [MCS$^+$06].
- In [TXZ04], [HDY05] and [MCS$^+$06] nodes are location aware.
- Protocols described in [SHSM06], [PR97], [MCS$^+$06], [LBC04] and [PP05] control message traffic to prevent congestion and excessive message dropping by using a specific channel access protocol or by limiting message transmission through synchronous rounds.
- The middleware proposed in [CDGN05] requires that neighboring nodes be nearly synchronized in order to emulate rounds. In [BV05], in order to provide probabilistic guarantees, a similar requirement is placed on the network.
- In [KT03], sparse regions of the network are populated by mobile nodes that exhibit characteristics similar to fixed gateways. These nodes however are mobile, and move according to network necessity. That is, whenever a part of the network is experiencing low connectivity, the mobile nodes move, filling in these spots and re-establishing the connectivity.

Although creative, in our opinion, acceptable solutions must not assume any of the previous limitations. Therefore, as far as we known, the remaining reliable broadcasting protocols can be classified according to three distinct categories: *forward error correction encoding algorithms*, *link reliability concept-based* and *retransmission-based*. Generally, existing reliable broadcasting protocols combine some or all of these approaches in order to achieve higher delivery rates, and when this occurs it will be promptly noted [PR99, WCK02]. Representative examples includes Double Covered Broadcasting [LW07], Reliable Multicast Algorithm [GSPS02], Efficient Reliable Broadcasting [HTS07], Reliable Broadcasting [AVC95], EraMobile [OGA06], Mistral [PBBvR06] and Reliable Multicasting Protocol [Kun03]. Unfortunately the small number of existing reliable broadcasting protocols accurately reflects how little attention has been devoted to reliability requirements beyond best-effort in *MANET*s [BKP03].

In the next section, as each category is described, these seven protocols will also be presented. Only three of them, however, were chosen to be simulated, and they were Reliable Broadcasting, Double-Covered Broadcast and EraMobile. This choice was greatly influenced either by limitations observed or by results of simulations done by the authors. The simulation results of fail-free runs of these three protocols will be discussed, and later, a performance comparison will be presented. Exactly like Chapter 3, protocols were simulated under the same set of scenarios and with the same observations regarding failures.

4.2.1 Approaches Based on Retransmission

Retransmission-based approaches are built upon an easily understood concept: whenever a node sends a message, it retains a copy of the message until all the receivers acknowledge reception. If no acknowledgment is received within a reasonable time, the sender will continually retransmit the same message until it can be assured that all correct nodes have received the message, failed or left the network. Acknowledgments may be positive (ACK) or negative (NACK). In positive acknowledgments schemes, the receiver must notify the sender which messages were correctly received. Whereas in negative acknowledgment schemes, the receiver notifies every message received incorrectly that will need to be retransmitted.

Acknowledgment schemes, however, are known to have serious scalability problems. They often incur high communication overhead, long end-to-end delays due to acknowledgment transmissions and have increased risks related to channel congestion due to the *ACK implosion problem* [ICP00]. The ACK implosion problem occurs whenever simultaneous receivers are required to send ACKs in response to the reception of a message. While this problem occurs more frequently in positive acknowledgment schemes, it may also occur in negative acknowledgment schemes if multiple nodes fail to receive a message. NACK-based schemes additionally incur longer delays since the sender must wait until the next broadcast to determine if the previous one was successfully delivered or not [LW04].

Retransmission-based approaches also include history exchanging protocols (also known as gossip or epidemic algorithms [DGH^{+}87, GBvR02]). Whenever nodes running such protocols come into contact with each other, the history of received messages are exchanged and missing messages are requested for retransmission. The main problem with this strategy is that, in order to ensure reliability, buffering space per node would need to grow indefinitely. Since this is not applicable in a realistic scenario, the size of the buffer can impact the coverage of a broadcast.

Since every transmission uses energy, there clearly exists a need to reduce the number of redundant transmissions while reaching all possible nodes. Therefore in retransmission-based approaches, the number of retransmission attempts has to be well chosen. However, it is not trivial to determine the optimal number of retransmissions that trade-off success rate against wasting too much energy. In practical terms, a maximum number of retries has to be used to properly terminate the broadcasting process. Five protocols fall under this category, they are Reliable Broadcasting, Reliable Multicasting Protocol, Efficient Reliable Broadcasting, Double Covered Broadcasting and EraMobile.

4.2.1.1 Reliable Broadcasting

Description. Proposed in 1995, the authors of the Reliable Broadcasting Protocol (namely, RB) [AVC95] based their work on simple flooding, and can therefore be classified as a probabilistic broadcasting protocol. Whenever a node transmits a message, each receiver should return an acknowledgement to the sender, and has now the responsibility to forward the message. If the sender does not receive an ACK it retransmits the message. If a host does not receive an ACK after several retries, it assumes that the link disconnection is not transient and stops sending the message. Central to the protocol is the maintenance of a history of messages received. When two hosts meet each other – and specially when nodes detect new neighbors – it uses a handshake procedure to exchange histories. On finding a missing message, a node can ask the other to supply it.

Known Limitations. One issue not addressed in the specification of the protocol is when (and how) a node can clear its buffer of history records. Since no collection of the stability of broadcasts is conducted, these history records may grow infinitely [HTS07].

Simulation Results

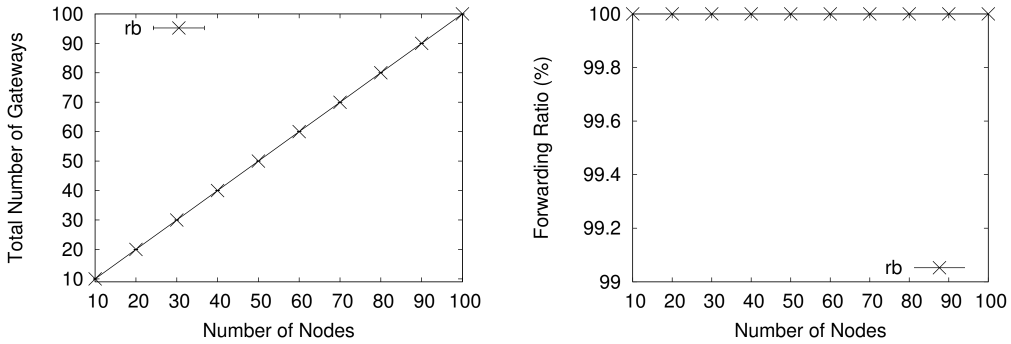


Figure 4.1. Simulation Results: Efficiency of Reliable Broadcasting Protocol

Efficiency. If the Reliable Broadcasting (RB) protocol needed to be summarized in a few words, it could be described as flooding with retransmission. Therefore, not much is expected from the efficiency results – and this is exactly what the simulation results show in Figure 4.1. The graph on the left plots the number of gateways involved for each simulation run with the specified number of nodes in the network. The graph on the right plots the forwarding ratio results. In every step of the simulation, all nodes become involved in the broadcasting process. Efficiency is neglected, as the forwarding ratio is always at 100%.

End-to-End Delay and Dropped Messages. End-to-end transmission delays and drop rates from the simulations can be seen in Figure 4.2. Delays range from 345 *seconds* to 351 *seconds*. Taking into consideration the fact that nodes broadcast during at most 400 *seconds*, this means that it takes almost all the simulation time for the last node to receive the message. Dropped messages results are even more disheartening. With only 10 nodes in the network, the number of dropped messages is approximately 173. As the number of nodes increases, so does the number of dropped messages, reaching values greater than 5300!

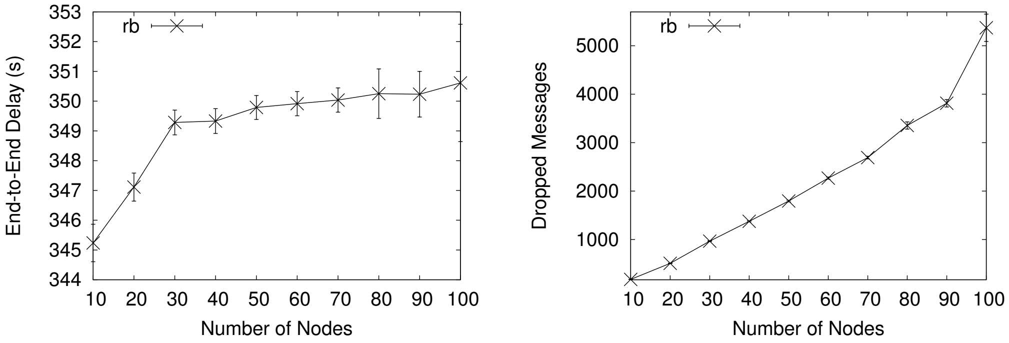


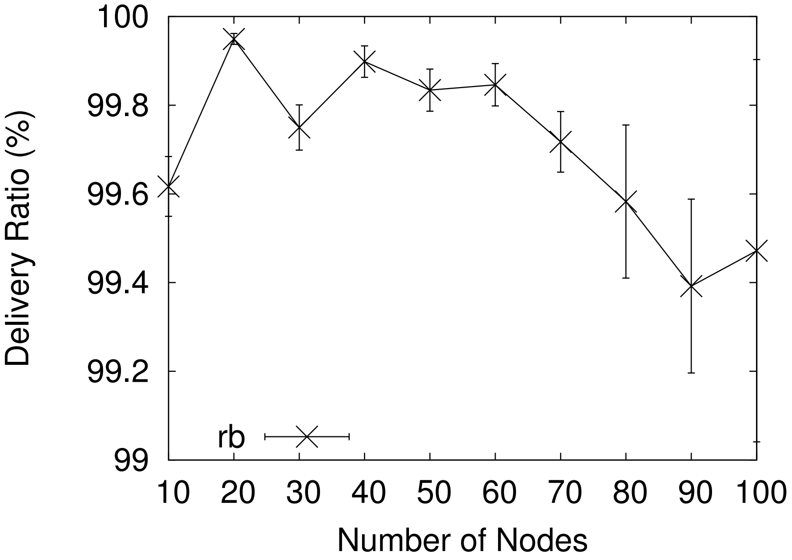
Figure 4.2. Simulation Results: Delay and Drop Rates of Reliable Broadcasting Protocol

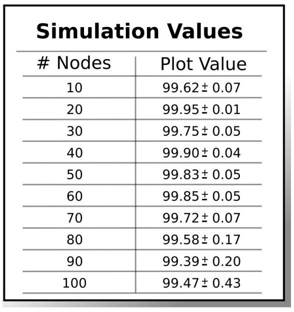
Reliability. Reliability results can be seen in Figure 4.3. Remarkably all results are higher than 99.2%. Even as node density increases and message drop rates soar, RB is capable of guaranteeing network coverage.

4.2.1.2 Reliable Multicasting Protocol

Description. Kunz's paper [Kun03] describes a practical experience of implementing a multicast routing protocol for *MANET*s. According to the author, the primary goal is to ensure as high a message delivery ratio as possible with finite resources. A reliable broadcasting protocol is also described where every node supports message retransmission by buffering the most recent messages. The buffer is implemented in round-robin fashion, storing the last unique messages a node received. Kunz's broadcasting protocol can be classified as probabilistic – as no exact delivery guarantee is assumed – but it ensures reliability by using a negative acknowledgement scheme. Every time a node receives a message, it checks whether it also received the previous sequenced number message from that source. If it did not, the node issues a NACK to the neighbors asking for retransmission.

To reduce collisions, both NACKs as well as message retransmissions utilize random forwarding jitter delays of 10 *ms*. NACKs have a timeout mechanism associated with them to ensure correct reception, and in order to reduce traffic, NACKS are re-issued





Simulation Values

# Nodes	Plot Value
10	99.62 ± 0.07
20	99.95 ± 0.01
30	99.75 ± 0.05
40	99.90 ± 0.04
50	99.83 ± 0.05
60	99.85 ± 0.05
70	99.72 ± 0.07
80	99.58 ± 0.17
90	99.39 ± 0.20
100	99.47 ± 0.43

Figure 4.3. Simulation Results: Reliability of Reliable Broadcasting Protocol

up to a certain maximum number of attempts. Additionally, if a neighboring nodes with a pending message retransmission overhears another node rebroadcasting the same message, it will cancel their redundant transmission.

Known Limitations. As previously stated, NACK-based schemes alert the sender to unsuccessful transmissions, requiring retransmissions to ensure reliability. But unless the sender actually produces a new message, one message – exactly the last one sent – if failed to be transmitted, would never be delivered as the receiver would never generate a NACK. This, however, is not observed in the protocol.

Simulation Results

This protocol was not chosen to be simulated due to the fact that it fails to detect one specific situation that can lead to unrecoverable messages, as previously noted.

4.2.1.3 Efficient Reliable Broadcasting

Description. The Efficient Reliable Broadcasting protocol proposed in [HTS07] is another probabilistic broadcasting protocol. Instead of attempting to develop a costly solution to reliable broadcasting, the authors argue in favor of a low-cost unreliable broadcast followed by additional acknowledgements and history exchanges to ensure reliability. They adopt the counter-based broadcasting protocol proposed in [NTCS99][1] as the unreliable broadcast protocol to reduce redundant broadcasts. The protocol is divided in three distinct phases denominated scattering, gathering and purging.

- The scattering phase uses the counter-based approach to broadcasting. But since this scheme cannot guarantee 100% message delivery, the protocol adopts

[1] We refer the reader to Section 3.3.2 in Chapter 3 where the protocol was fully described.

a handshaking mechanism whereby hosts exchange broadcasting histories. Unknown messages are requested through NACKs.

- In the gathering phase, to ensure reliability, the authors define that every broadcast message received must be acknowledged. To reduce the risk of the ACK implosion problem occurring, acknowledgments are combined together and returned as a single message. These combined ACKs are sent whenever a internal timer expires.

- In the final phase, called the purging phase, messages that have been acknowledged by nodes are removed from internal buffers. Every purge action is then broadcast to neighbors. While these do not need to be acknowledged, periodically nodes exchange purge histories in order to guarantee that this information is correctly disseminated.

Known Limitations. This is yet another work that covers not only the broadcast primitive but also routing and the concept of groups within the network. Therefore, there are moments that become unclear as where the broadcasting ends and another primitive begins. In the purging phase, for example, the authors are unclear as to what determines that a message is stable and we can only assume that it is related to local neighborhood information and not to the whole network. Finally, no performance evaluation is mentioned regarding the additional broadcast sent in the purging phase, nor on the cost of "purge history" broadcast.

Simulation Results

For the reasons previously listed, this protocol was not chosen to be simulated.

4.2.1.4 Double-Covered Broadcast

Description. Lou and Wu's goal when proposing the double-covered broadcasting protocol (namely, DCB) [LW07] was to reduce the number of forwarding nodes without sacrificing the broadcast delivery ratio. It is classified as a neighborhood designating protocol since it piggybacks selected gateways nodes in broadcast messages. By selecting a set of gateway nodes where not only every 2-hop node is covered, but also where all 1-hop nodes are covered by at least 2 forwarding neighbors (the sender itself and one of the selected gateway nodes), it benefits from the broadcast redundancy to improve reliability. Additionally, in DCB the re-transmission of the message by the gateway nodes serves as an ACK of correct message reception to the original sending node. This scheme avoids the ACK implosion problem. By using this method of acknowledgement no explicit ACKs need to be sent, which, according to the authors, would incur too much communication overhead and would need stable links between nodes.

If the sender fails to detect/overhear all of the re-transmissions, it assumes that a transmission failure occurred. The sender will keep re-sending the message until all forward nodes have re-transmitted or until a threshold is reached. By double-covering, DCB assumes that at least two transmissions will reach the nodes, therefore this redundancy prevents a single transmission error from interfering on message transmission and reception. But it is the retransmissions due to the lack of ACK receptions that increase the reliability of DCB, for it is this mechanism that guarantees that all 2-hop neighbors receive the message.

Known Limitations. Unfortunately, the reception of the acknowledgment by the sender node does not ensure that the 2-hop neighbors received the message as well. Both the exposed terminal problem and the hidden terminal problem may defeat the reliability mechanism inherent in DCB.

Simulation Results

Efficiency. In Figure 4.4 one can note that with less than 30 nodes, DCB maintains forwarding ratio below 50%. But as the number of nodes rises to 80, there is a steep rise involving up to 65% of the network. On average network involvement is a little below 60%.

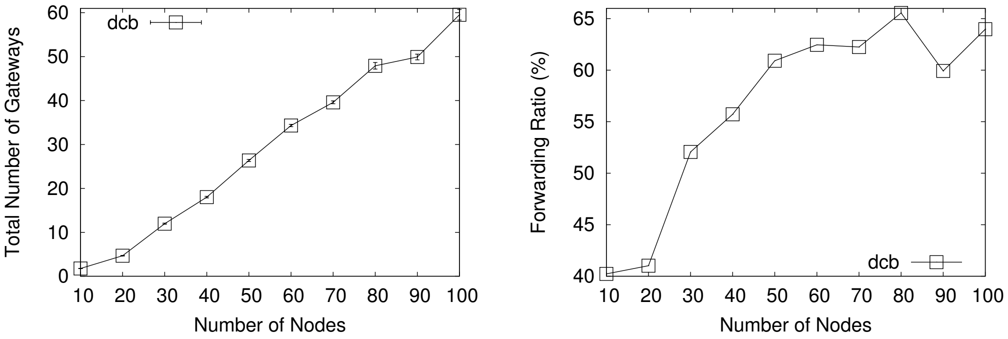


Figure 4.4. Simulation Results: Efficiency of Double-Covered Broadcast

End-to-End Delay and Dropped Messages. One might expect a longer delay coming from a broadcasting protocol classified as reliable, but simulation results show that DCB is extremely fast in delivering messages throughout the network. In Figure 4.5 simulation results show end-to-end delays kept at bay with the worst case (100 nodes) being covered under 0.3 *seconds*. And although there is an increase in dropped messages, up to 60 nodes, they remain below 200 messages. However, this value quickly rises to over 450 when 100 nodes are present in the network.

Reliability. Delivery ratio results only really start to impress when at least 60% of the network becomes a gateway. This occurs when about 50 nodes exist in the network. This can be seen in Figure 4.6. It is from this moment on that DCB starts to cover over 85% of the network. With 60 nodes, delivery rates reach over 90%. That means that the double-covered approach effectively delivers messages in a fail-free scenario.

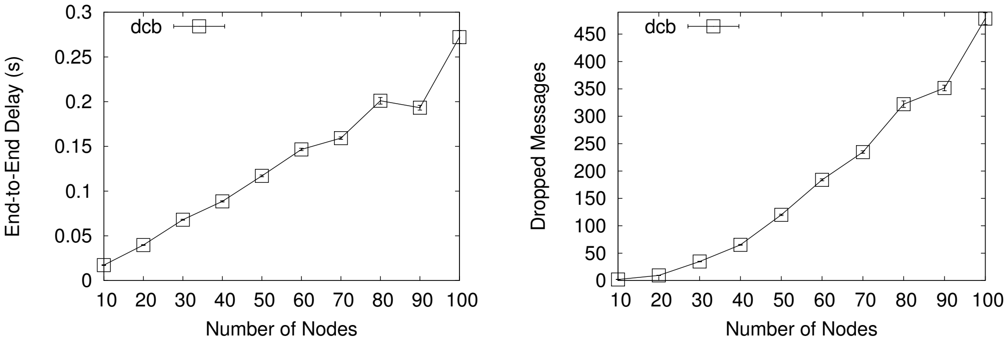
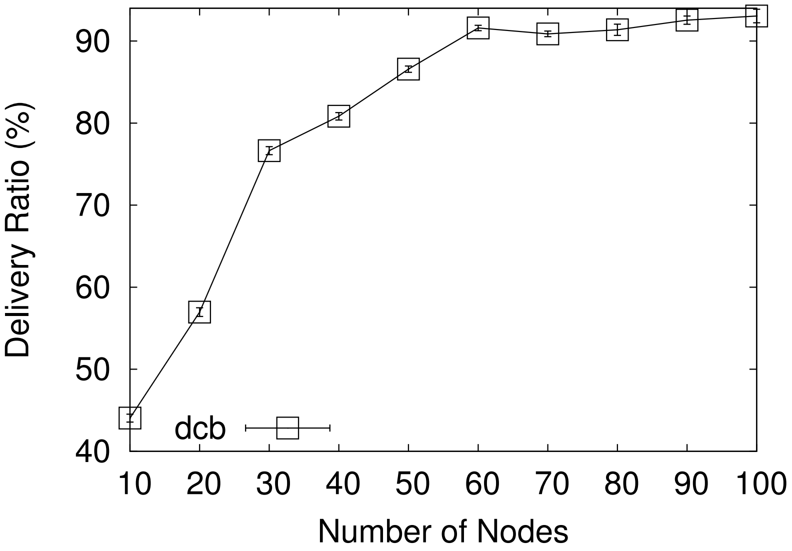
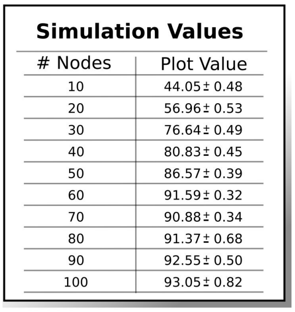


Figure 4.5. Simulation Results: Delay and Drop Rates of Double-Covered Broadcast





Simulation Values

# Nodes	Plot Value
10	44.05 ± 0.48
20	56.96 ± 0.53
30	76.64 ± 0.49
40	80.83 ± 0.45
50	86.57 ± 0.39
60	91.59 ± 0.32
70	90.88 ± 0.34
80	91.37 ± 0.68
90	92.55 ± 0.50
100	93.05 ± 0.82

Figure 4.6. Simulation Results: Reliability of Double-Covered Broadcast

4.2.1.5 EraMobile

Description. EraMobile [OGA06] is presented as a deterministic alternative to the probabilistic gossip-based broadcasting [CRB01, LEH03]. The basic idea behind gossip-based broadcasting is to have each node periodically exchange knowledge of received messages with a random set of other nodes. Missing messages can then be requested and recovered in a manner that is distributed throughout the system. Gossip-based protocols are then very resilient to node mobility and failure. The downside to the pure gossip-based approach is that, very much like all probabilistic protocols, message delivery to all nodes is not guaranteed.

Instead of randomly choosing the set of neighbors with whom the exchange will take place, the authors of EraMobile exploit the broadcast nature of wireless communication and exchange with all neighboring nodes, therefore turning EraMobile into a deterministic gossip-based protocol. That is, periodically gossip messages are exchanged with all 1-hop neighbors. Whenever a node detects that it has not received a specific message, it will request the message through a broadcast request. Neighboring nodes who receive the request will recover that message (if they still have it) and transmit it. Messages are stored within buffers during a certain time, but once removed the protocol considers that messages as lost to any nodes who still have not received it.

EraMobile is also equipped with other performance enhancing mechanisms such as random jitters on gossip sending, dynamic message buffer threshold, traffic reduction based on local node density levels, limits on the number of requests and recovers that can occur in a single gossip round, and even an adaptation mechanism for energy conservation whenever local traffic is idle.

Known Limitations. Nevertheless, EraMobile is not indicated for any time-sensitive applications since it provides high delivery rates at the cost of high end-to-end delay times. Furthermore since eventually all messages are removed from buffers, message coverage could be somewhat impacted.

Simulation Results

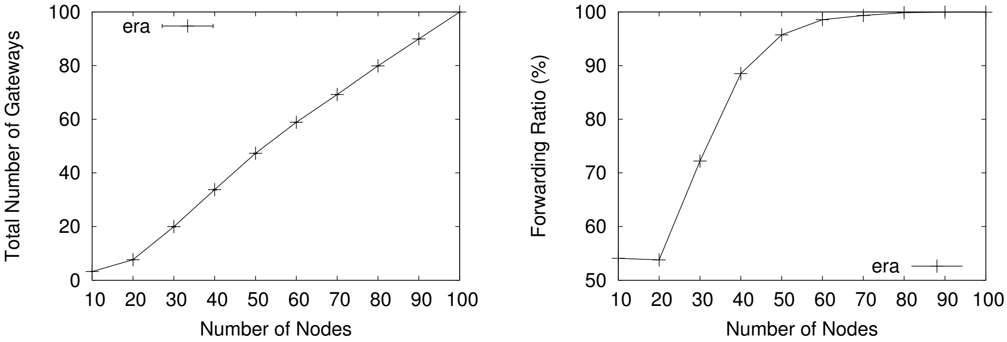


Figure 4.7. Simulation Results: Efficiency of EraMobile

Efficiency. The total number of gateways and forwarding ratio results are plotted in Figure 4.7. Since EraMobile exchanges histories, it is expected that sooner or later a large part of the network will need to retransmit a missing message that has been requested. Results corroborate this reasoning. Starting from 40 nodes on, more than 90% of the network gets involved in retransmissions. With 70 or more nodes in the network, forwarding ratio reaches 100%.

End-to-End Delay and Dropped Messages. End-to-end delays are high with EraMobile, as can be noted in Figure 4.8. With 10 nodes, for example, it takes 50 seconds to broadcast a message. With 50, almost 300. But interestingly, with more than 50 nodes, the latency value starts to drop again, resting at 150 seconds to broadcast to a 100 node network. Possibly the high node density allows a single transmission to reach many nodes, therefore reducing the number of retransmission requests and speeding overall broadcast times. Nevertheless, during the same periods, dropped messages always increase, reaching values almost as high as 1200.

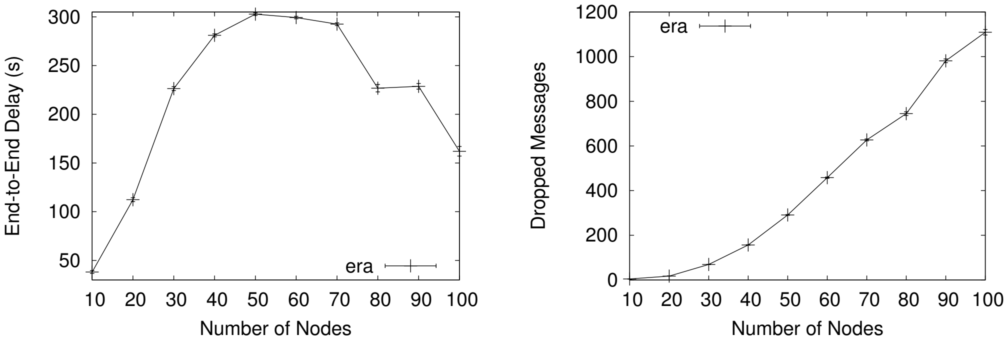


Figure 4.8. Simulation Results: Delay and Drop Rates of EraMobile

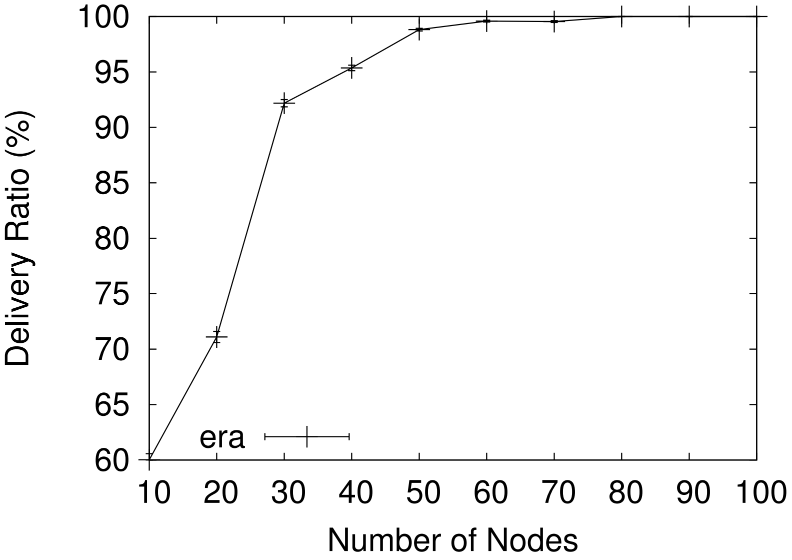
Reliability. EraMobile's reliability results a very encouraging and are plotted in Figure 4.9. More than 50 nodes already cover at least 99% of the network, and at least 30 nodes are needed to guarantee delivery to at least 90%. But with less than 30 nodes this value drops to 60%. If these results are analyzed taking into consideration the other results, then a curious situation occurs where reliability increases and end-to-end delays decrease when node density starts to grow (with at least 50 nodes).

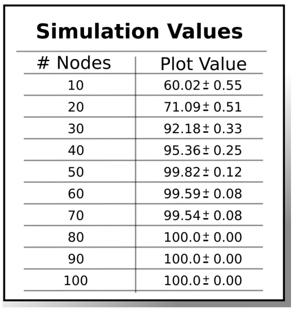
4.2.2 The Link Reliability Approach

The link reliability concept applies the idea of finding neighboring links through which message forwarding tends to suffer less mid-conversation failures, collisions, drop rates and many of the other problems that surround wireless communications. By applying this concept, broadcasting protocols are able to subjectively measure neighboring nodes and choose the ones that seem more trustworthy. Thus, instead of randomly choosing a link, the more reliable ones are preferred. Therefore, nodes would be (probabilistically) ensured of message delivery.

The concept of a reliable link however is not new. Associativity-based routing [Toh97] was one of the earliest protocols that considered link reliability prior to routing[2]. The primary idea was to prefer reliable links over transient ones, to prefer

[2] Although the authors prefer and use the term "stability" instead of "reliability".




Simulation Values

# Nodes	Plot Value
10	60.02 ± 0.55
20	71.09 ± 0.51
30	92.18 ± 0.33
40	95.36 ± 0.25
50	99.82 ± 0.12
60	99.59 ± 0.08
70	99.54 ± 0.08
80	100.0 ± 0.00
90	100.0 ± 0.00
100	100.0 ± 0.00

Figure 4.9. Simulation Results: Reliability of EraMobile

long-lived routes over shortest-path routes. Many other papers adopted a similar approach and analyzed whether the probability of a link persisting for a certain time span could be used as the reliability metric [CH05, BKS03, LSL+02, DRWT97, AASS00]. Curiously, a very limited number of broadcasting protocols use any kind of reliability information of its neighbors in order to better decide its forwarding status [BKS03, VE05, WD05].

Perhaps this is justified by the fact that all known link reliability-based protocols rely on the age of the link as a reliability metric. These works supposed that a long-lived route had a tendency to remain valid without losing connectivity over time. As a result, a better communication performance and less chances of re-routing while communicating was achieved. What is implicitly assumed is that certain nodes will group after a while, and thereafter remain moving in a similar direction and with a similar speed, while new links will tend to be always transient. However, according to [GdWFM02], these assumptions can only be justified for a few dynamic scenarios and actually not much can be presumed concerning the movement of two nodes that just came into one another's transmission range. Clearly, the duration of a link largely depends on the mobility pattern of the nodes. In fact, depending on the mobility pattern, old links will not necessarily be more stable than young ones. The Reliable Multicast Algorithm implements this approach to reliability.

4.2.2.1 Reliable Multicast Algorithm

Description. The Reliable Multicast Algorithm (namely, RMA) [GSPS02] is a multicasting protocol for *MANET*s. It assumes the existence of a group membership algorithm to manage arrivals (JOIN) and departures (LEAVE) of nodes within a specific group. But within the group, it broadcasts messages to all nodes. RMA uses the average stay time of neighboring nodes as a reliability metric, and prefers to send messages through nodes that have an older estimated lifetime. Additionally, RMA increments reliability through the use of acknowledgments from destination to source. These ACKs also contribute to disseminate link status between nodes throughout the network.

Known Limitations. Unfortunately, the approach for guaranteeing that messages are propagated across the network is not clearly described. In fact, their work focuses much more on the group membership and routing related aspects of communication. Since the authors emphasize that only local neighborhood information is used by nodes to broadcast, and that every decision is taken locally by the node, we assume that it is a deterministic neighborhood designating protocol. Furthermore, as previously identified, using age as a reliability metric is prone to failures as this depends on many factors that might not be controllable by the node.

Simulation Results

For the reasons previously listed, this protocol was not chosen to be simulated.

4.2.3 The Forward Error Correction Encoding Approach

Protocols based on forward error correction encoding algorithms (FEC) require that the sender add redundant data to messages in a way to allow the receivers to detect and correct errors without the need to ask the sender for additional data. As long as a sufficient number of messages are received, the receiver can re-assemble and correct compromised data messages. For example, n repair messages are created for every m data messages intended for transmission, but all $(n + m)$ messages are broadcast. Using a number of messages equal to m, any receiving node is capable of recovering the original m messages, safeguarding the communication between the nodes from at most n message losses. Obviously, the higher the amount of redundant information added to messages, the more message loss is tolerated.

The biggest advantage of FEC is that retransmission of data can often be avoided as no feedback is required to send back to the sender. Additionally they generally have lower recovery latencies than protocols based on retransmissions. However, on average they require higher bandwidth and increase network traffic. Furthermore, this approach needs to determine how much redundant data has to be generated in order to ensure correct message delivery, and this might not be able to be properly defined before-hand [Kun03]. Mistral is a representative protocol of this category.

4.2.3.1 Mistral

Description. The key idea behind Mistral [PBBvR06] is to extend selective flooding approaches with FEC. It is classified as a probabilistic broadcasting protocol. The authors of Mistral start off with a purely probabilistic flooding protocol, but compensate for dropped messages by periodically broadcasting compensation messages. Every one of these compensation messages encodes a set of messages that have been dropped (i.e. not rebroadcast) by the sender. For the situation that arises when nodes have lost too many messages to reconstruct missing data, Mistral's utilizes additional information within the messages to properly identify the loss. With this knowledge, nodes can rely on a secondary recovery mechanism to recover messages. Compensation messages are broadcast at regular intervals, but as soon as an internal buffer passes a certain threshold, they are also sent. After some time, compensation messages are garbage collected.

Known Limitations. According to the authors of Mistral, the protocol is capable of keeping overheads low, thus balancing message overhead against reliability. But, they acknowledge that there exists a moderate risk that some messages may fail to reach a number of nodes because of this.

Simulation Results

Due to simulation results presented by the authors – where it was shown that message delivery results were worse than flooding (even when message overhead was more than twice the number of messages that flooding generated) – this protocol was not chosen to be simulated.

4.3 PERFORMANCE COMPARISON OF RELIABLE PROTOCOLS

We match-up results from all three reliable broadcasting protocols that were simulated, namely Double Covered Broadcasting (DCB) [LW07], Reliable Broadcasting (RB) [AVC95] and EraMobile [OGA06], to compare their performance. Figure 4.10, Figure 4.11 and Figure 4.12 presents these results. Once again, we will limit comparison to three of the metrics: efficiency, end-to-end delay and reliability.

Efficiency. When all three efficiency results are plotted in the same graph, as can be seen in Figure 4.10, the large difference between DCB and the other protocols is easily noticed. Both RB and EraMobile reach the top of the graph when more than 50 nodes take part of the simulation, demonstrating no efficiency whatsoever. DCB, on the other hand, has only a slight increase as more nodes take part of the broadcast.

End-to-End Delay. RB has the worst end-to-end delay, maintaining an almost constant time of 350 seconds to disseminate a message. And while EraMobile has better results, for most part it is over the 100 second barrier. Regarding end-to-end delays, DCB is once again superior to both. This can all be seen in Figure 4.11.

Reliability. In Figure 4.12 we can note that the only protocol capable of 100% delivery ratios, 100% of the time, is RB. But this comes at a high price: all nodes get involved in the transmission and latency is extremelly high. EraMobile is bit faster and more efficient, and reaches the 90% mark with as little as 30 nodes. Finally, while DCB reduces both the number of gateways and the end-to-end delay, it depends on a denser network to guarantee message coverage.

4.4 CONCLUSION AND SUMMARY

This chapter has focused on strategies for ensuring sufficient message coverage while broadcasting in mobile ad-hoc networks. Initially, the reliable broadcasting problem

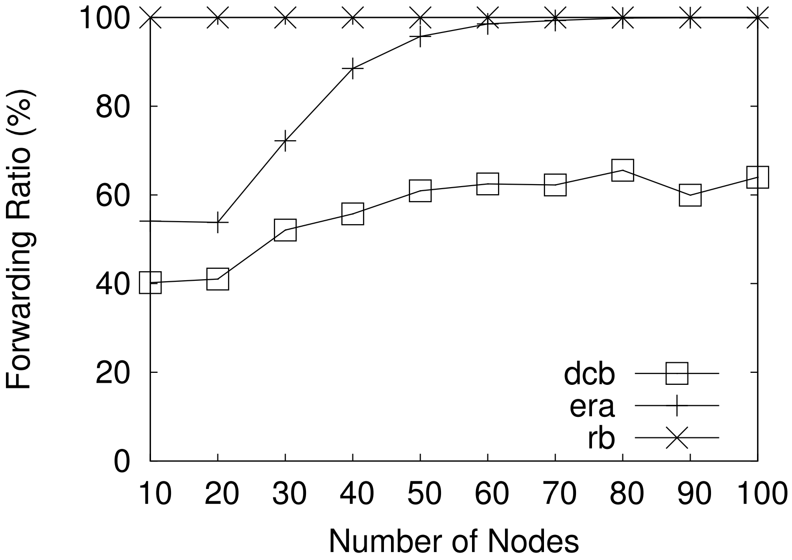


Figure 4.10. Simulation Results: Efficiency of all Reliable Protocols

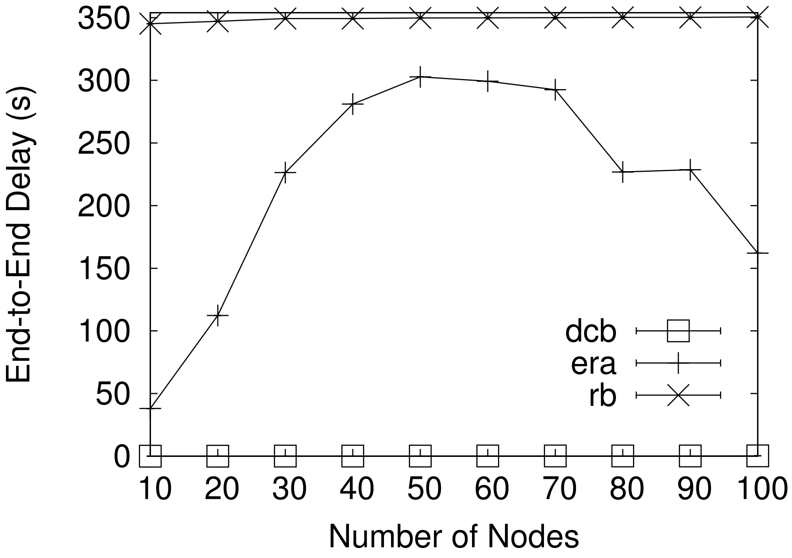


Figure 4.11. Simulation Results: End-to-End Delay of all Reliable Protocols

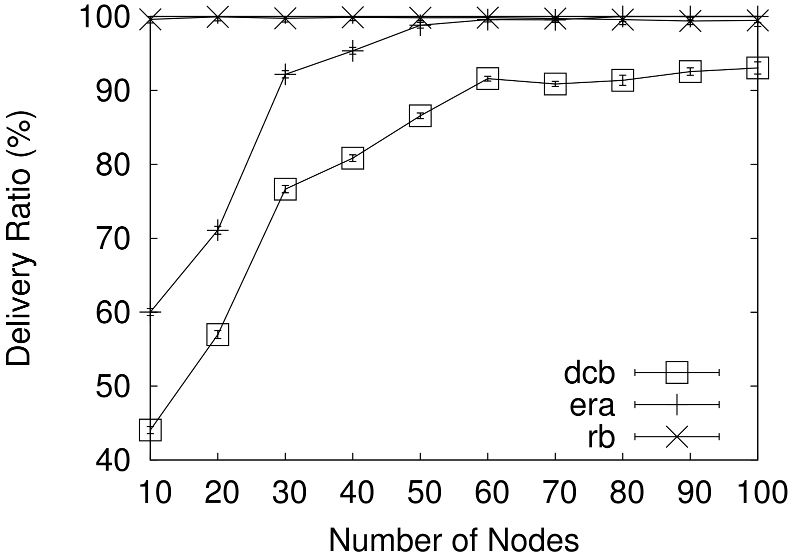


Figure 4.12. Simulation Results: Reliability of all Reliable Protocols

was described and existing strategies to solve such problem where presented. A large number of broadcasting protocols exist for *MANET*s, but as was shown, many are not acceptable for usage in a real-world scenario. The ones that are acceptable were classified and further described. Finally, three reliable broadcasting protocols were chosen to be simulated, and results were shown and analyzed. The protocols were Double Covered Broadcasting [LW07], Reliable Broadcast [AVC95] and EraMobile [OGA06].

The credibility of a broadcasting protocol lies on whether or not it is capable of delivering messages to nodes throughout the network. The most unreliable broadcasting protocols could guarantee was best-effort delivery, but reliable broadcasting protocols, on the other hand, are expected to obtain better message dissemination in spite of all difficulties regarding communication in *MANET*s. Simulation results, however, evidence that not all protocols are capable of assured message coverage, which is somewhat unsatisfactory. Worst still, up until this moment simulation scenarios were limited to fail-free situations. The next chapter introduces temporary node failures in order to place broadcasting protocols under an even more realistic scenario. Simulation results are then analyzed to better comprehend exactly how both reliable and unreliable protocols react to these failures.

CHAPTER 5

EVALUATING THE IMPACT OF FAULTS ON BROADCASTING PROTOCOLS FOR MOBILE AD-HOC NETWORKS

Do not be misled: Bad company corrupts good character.

—THE HOLY BIBLE, 1 CORINTHIANS 15:33

5.1 INTRODUCTION

The expectations of delivery ratios on broadcasting protocols for *MANET*s are pretty high: they are expected to be robust enough to handle all possible changes in the system topology and still guarantee delivery between any source-destination pair of nodes. In the previous chapters, both reliable and unreliable broadcasting protocols were simulated under similar conditions in order to observe how they behaved. In each of those chapters, however, simulation runs took place in a scenario where no nodes failed. In this chapter temporary node failures are introduced. This kind of failure helps represent scenarios where transient link availability exist; scenarios that manifest external environmental interferences that affect message propagation, and that exhibit transmission failures due to node mobility; scenarios that have been known to be considerably disruptive to communication protocols [CJWK02].

It has been well documented that changes in the environment can dramatically affect radio propagation, causing frequent network topology changes and network partitions [ER02], and this kind of unpredictability – which is commonplace in *MANET*s – must be handled by broadcasting protocols. Unfortunately, a large number of broadcasting algorithms for *MANET*s assume that during the broadcasting process there occurs none or very little topology changes and that the network remains connected. But in a real scenario, this cannot always be guaranteed. By temporarily disabling nodes, simulation runs are able to mimic such scenarios and consequently stress broadcasting protocols. Results help define a clearer picture of how broadcasting protocols truly react under realistic scenarios. This chapter starts out describing this fault model. Then simulation results of the broadcasting protocols under this model are presented.

5.2 FAULT MODEL

In Chapter 3, solutions for broadcasting in *MANET*s were classified according to their delivery guarantees. Probabilistic broadcasting protocols were introduced as having less constraints and assumptions (when compared to deterministic protocols), as being usually simpler to implement and, as normally having small memory requirements. While deterministic broadcasting protocols were described as most often using constant neighborhood set exchange between nodes in order to maintain their awareness of local topology updated.

Due to this last fact, in most deterministic broadcasting protocols a crashed node will only interfere for a short time during the broadcasting process. Such broadcasting protocols then become resilient to node crash failures. This occurs since, shortly after the failure when all neighboring nodes exchange neighborhood information, crashed nodes (who did not participate in these exchanges) are removed from their neighbors' set. In future broadcasts, crashed nodes no longer belong to any other nodes' neighborhood set, and will not be considered as possible forwarding nodes.

Using a crash failure model is therefore inadequate to analyze faults when simulating deterministic broadcasting protocols. Thus, unlike any other broadcasting analysis seen before, an omission fault model has been implemented. Although emphasized for deterministic broadcasting protocols, this model is also applicable to probabilistic ones as well.

Unfortunately, simulation runs of unreliable and of reliable broadcasting protocols had different time and space requirements. For this reason, two different fault models had to be used.

- For the unreliable broadcasting protocols, during each one of the runs an uniformly random selected set of nodes fail to send and receive *any kind of messages* for 10 seconds. When this period is over, a new group of randomly selected nodes is chosen to fail. The exact number of nodes chosen depends on the percentage of failed nodes which can be 0% (failure free), 5%, 10%, 20%, 30% and 50%.
- For the reliable broadcasting protocols, during each one of the runs an uniformly random selected set of nodes fail to send and receive *messages with large payloads, such as broadcast messages, but are still able to exchange neighborhood and control messages* for the same timespan of 10 seconds. The exact number of nodes that fail on any given run also depends on the percentage of failed nodes, but are limited to 0%, 25% and 50%.

Although high, the 50% failure scenario is important as it has been proved that to solve distributed problems with stronger reliability requirements, like consensus and atomic broadcast, this is the highest number of nodes that can fail [VE05]. It is important to note that, in parallel with the omission faults simulated by our model, other failures still keep occurring during the execution. For example, transmitted messages frequently are dropped since, after all, the radio propagation model used (in NS-2) allows for transmission errors and nodes are mobile. In order to correctly compare the protocols, comparisons between protocols will be limited to those simulated under the same set of mobility and fault patterns, including the exact same broadcast message sending times.

When defining this fault model, the importance of randomly choosing faulty nodes instead of selecting a static group of nodes was not neglected. But, since the number of failures is constant, to help better spread the failure and to represent omission faults, after a small time period another group of nodes throughout the network is picked. This also helps to stress those protocols that assume a correct behavior on the reception and transmission during a broadcast, specially by some special set of nodes such as the gateway nodes. On the other hand, this fault model favors those protocols that use additional mechanisms to properly identify message reception by neighboring nodes.

5.3 PERFORMANCE OF UNRELIABLE PROTOCOLS

The efficiency, end-to-end delay and reliability results for Blind Flooding, Dynamic Probabilistic Approach, Wu and Li's Protocol, Scalable Broadcast Algorithm and Dominant Pruning will now be presented. Simulation results will show how each one of these protocols behaved when under different failure scenarios, with node failures of 0% (failure free), 5%, 10%, 20%, 30% and 50%. Comparison results will also be shown for each metric to facilitate observations of similarities and differences between protocols.

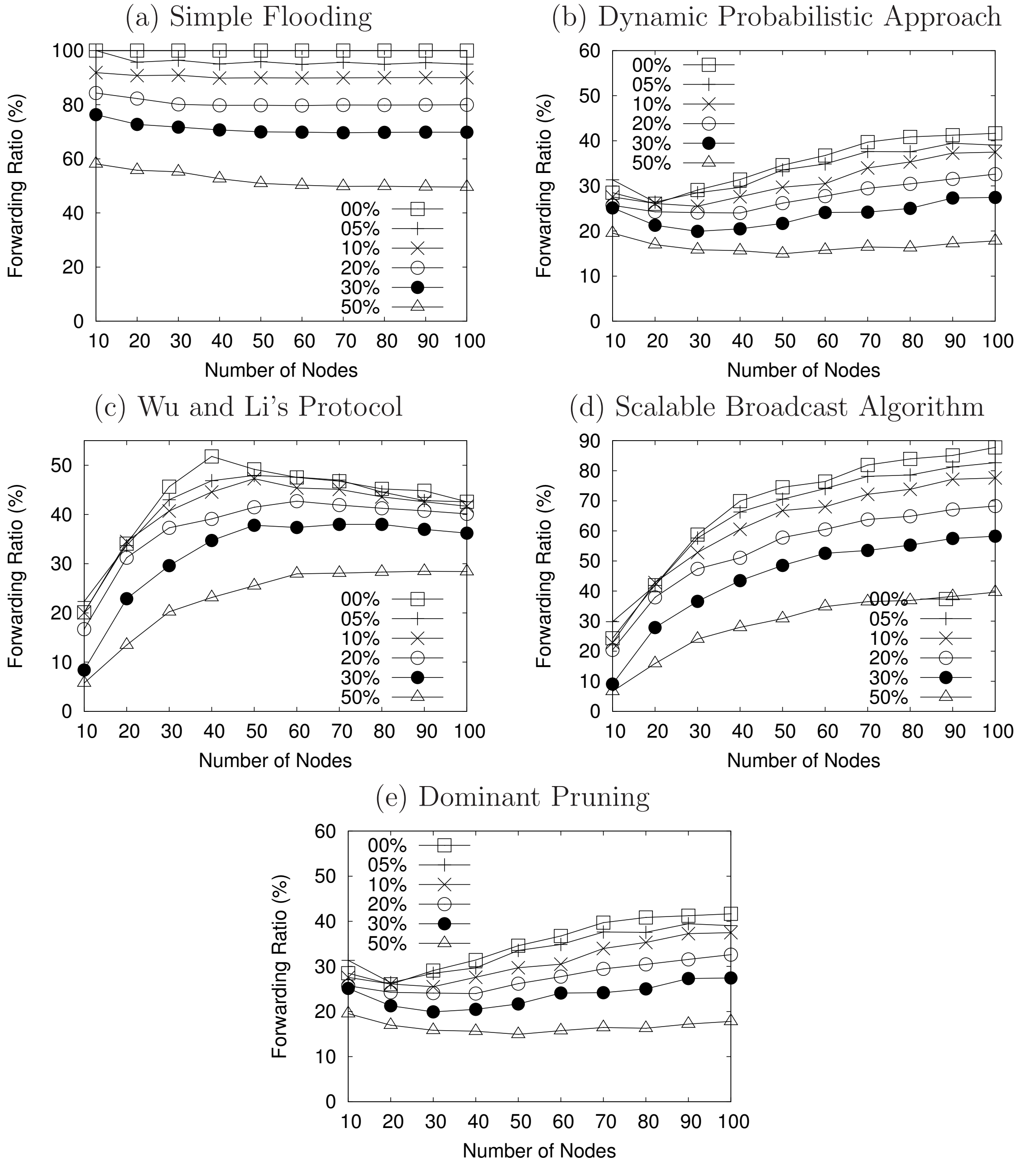
5.3.1 Efficiency

In Figure 5.1 the reader may notice how the forwarding ratio of all protocols is lowered as failures are introduced to the scenarios. That is, as more nodes fail, the smaller the number of nodes involved in the forwarding process. Flooding continues to be the most inefficient protocol, as it needs to involve almost all receiving nodes in the forwarding process. Note how even in the worst-case, a 50% node failure scenario involves between $50\% - 60\%$ of the nodes that receive a message in the forwarding process. Dynamic probabilistic obtains higher efficiency, since it compromises at most 50% of the nodes, but this drops to as low as 25% when in a 50% failure scenario. In an increasing scale, Dominant Pruning is the next most efficient protocol, as it also maintains node participation in the $25\% - 40\%$ range, but unlike Dynamic Probabilistic, this happens most of the time. In the worst-case scenario (50% node failure) this lowers to 18%. Both Wu and Li's protocol, as well as SBA, have high efficiency values when in a sparse and fault-enabled scenario, involving between $16\% - 25\%$ of the network on average, but reaching values as low as 10%. But similarities stop there. SBA's efficiency then drops sharply, involving between $60\% - 80\%$ of the network. While Wu and Li's behavior settle between $25\% - 40\%$.

Figure 5.2 show a comparison of the protocols.

5.3.2 End-to-End Delay

All protocols, as can be seen in Figure 5.3 and Figure 5.4, have a slight drop in the end-to-end delay as more nodes failed. This was the expected behavior since the node re-transmission activity ceased on all faulty nodes. SBA's backoff delay (to reduce congestion and collisions) produced a longer overall delay to transmit messages. Flooding, on the other hand, causes the broadcast storm problem which also increases the latency of the broadcast. The remaining protocols all had low delays, with both Wu and Li and DP needing some time to update 2-hop neighborhood information, while Dynamic Probabilistic's simpler approach to broadcasting maintains latency to a minimum.



(a) Simple Flooding

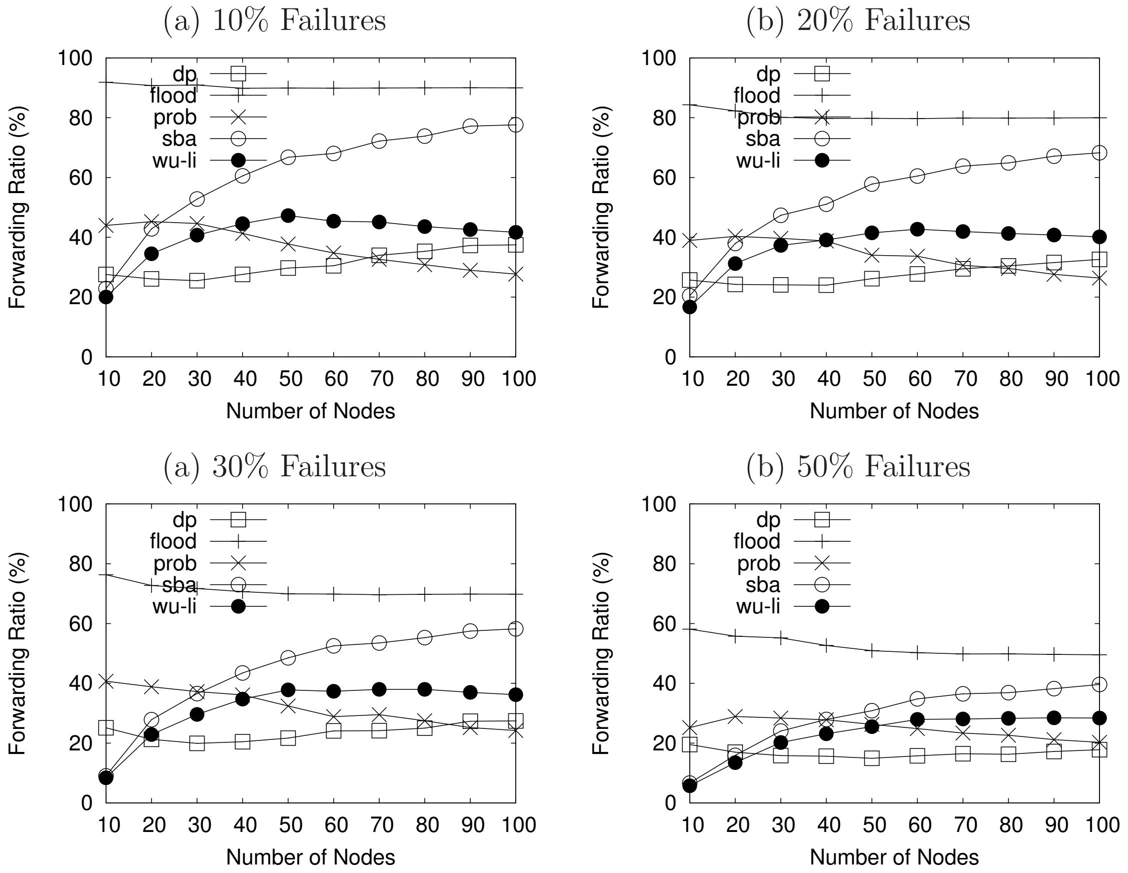
(b) Dynamic Probabilistic Approach

(c) Wu and Li's Protocol

(d) Scalable Broadcast Algorithm

(e) Dominant Pruning

Figure 5.1. Efficiency Simulation Results With Failures of All Unreliable Protocols (Individual View)

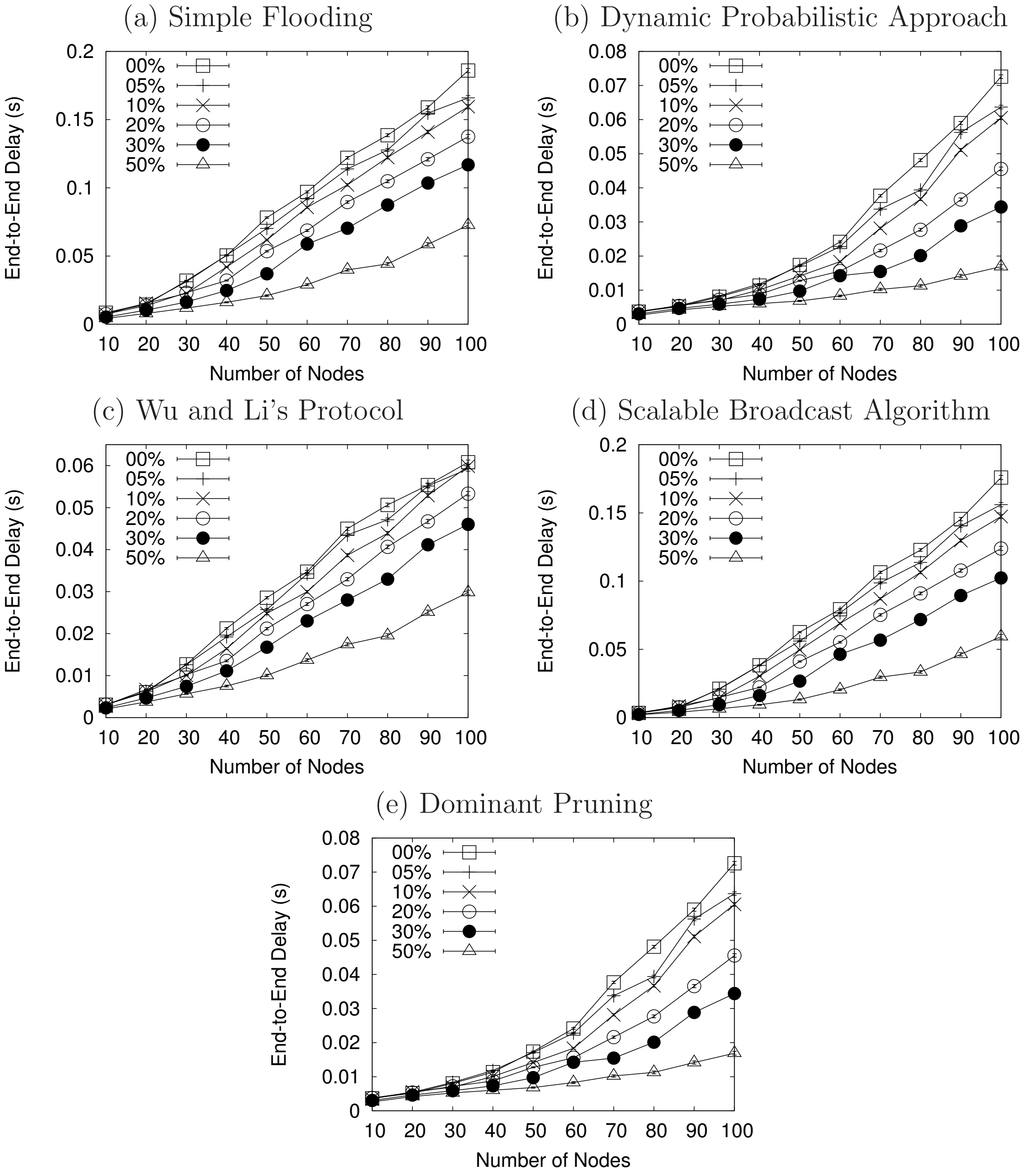

(a) 10% Failures

(b) 20% Failures

(a) 30% Failures

(b) 50% Failures

Figure 5.2. Efficiency Simulation Results With Failures of All Unreliable Protocols (Comparison View)


(a) Simple Flooding

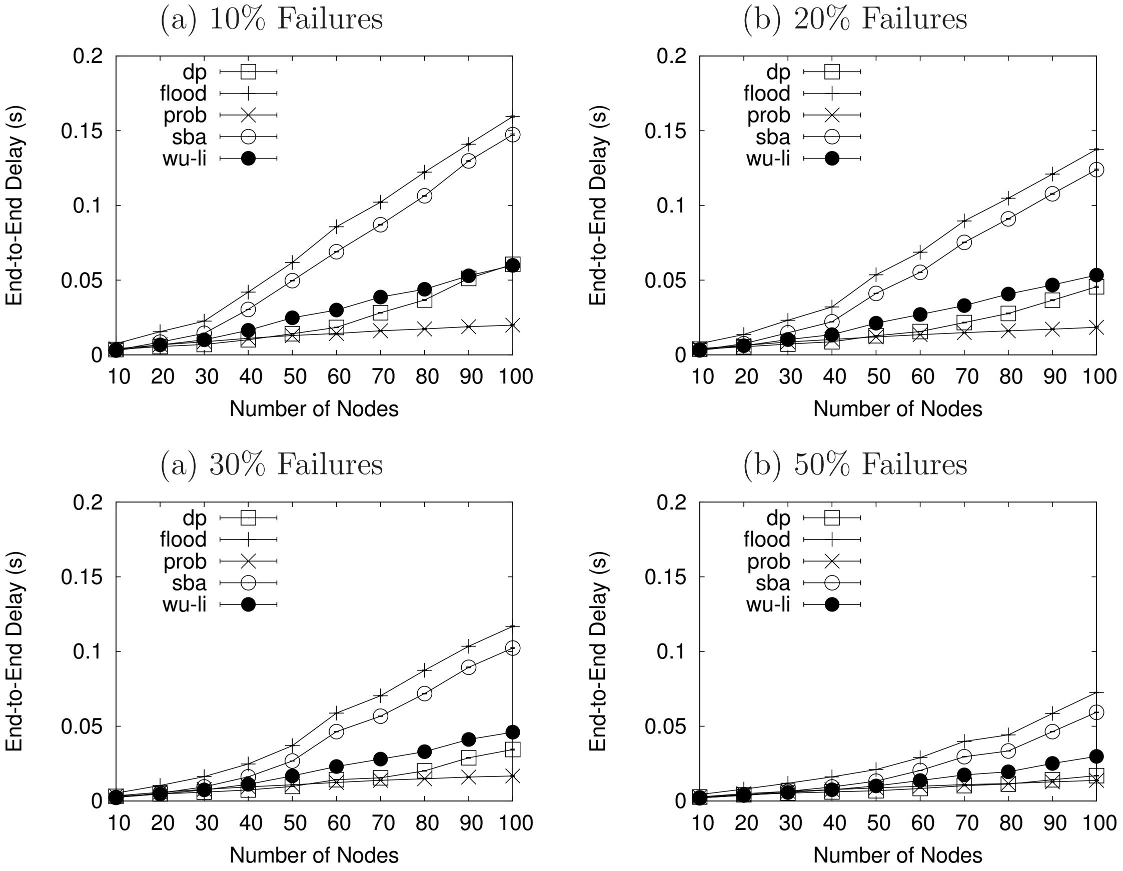
(b) Dynamic Probabilistic Approach

(c) Wu and Li's Protocol

(d) Scalable Broadcast Algorithm

(e) Dominant Pruning

Figure 5.3. End-to-End Delay Simulation Results With Failures of All Unreliable Protocols (Individual View)


(a) 10% Failures

(b) 20% Failures

(a) 30% Failures

(b) 50% Failures

Figure 5.4. End-to-End Delay Simulation Results With Failures of All Unreliable Protocols (Comparison View)

5.3.3 Reliability

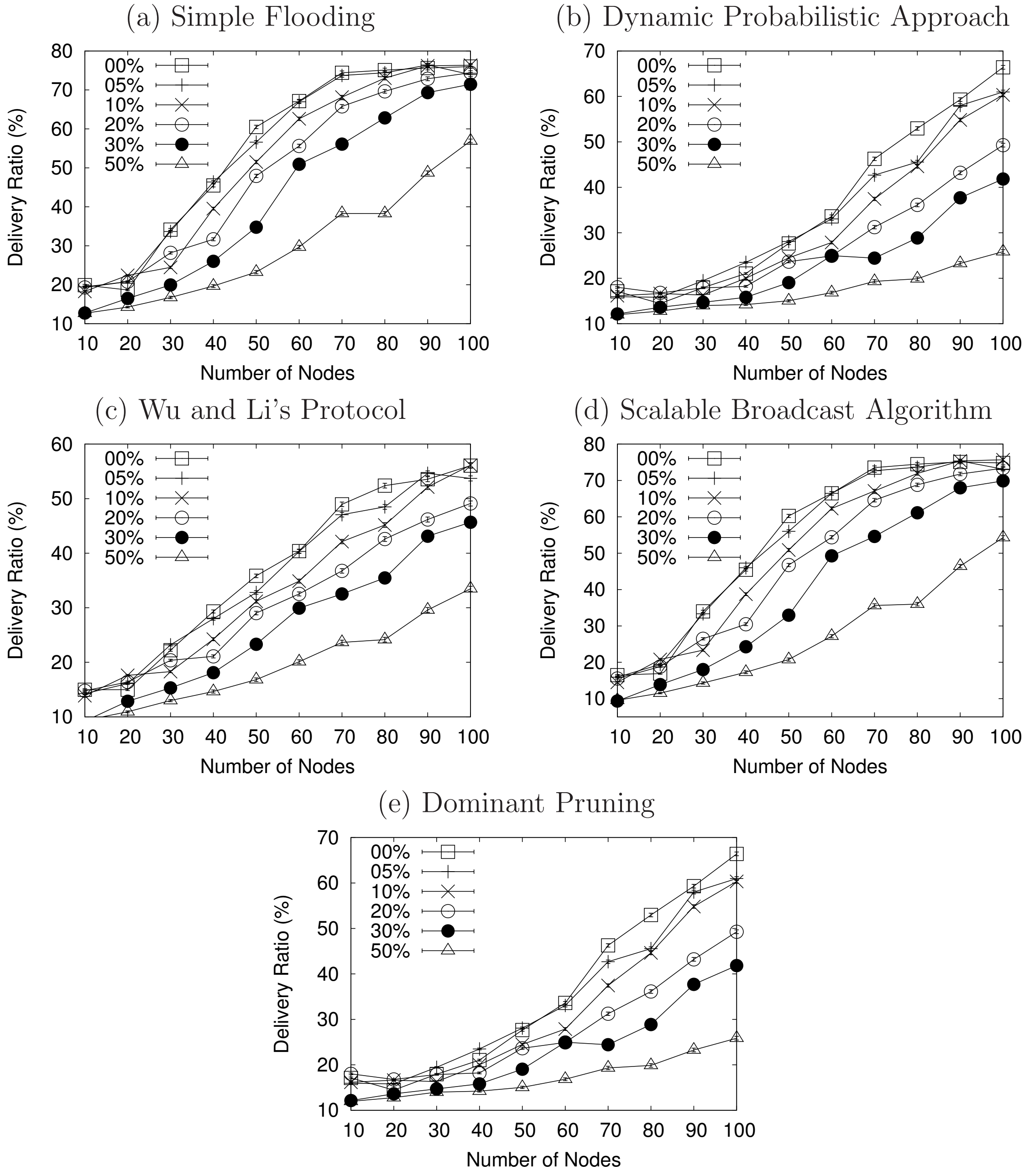
Figure 5.5 clearly shows what was expected: the reliability of all simulated broadcasting protocols lowers as the number of node failure increases. This conclusion is true to both deterministic and probabilistic approaches. Flooding and SBA were the ones with the highest delivery ratios – even when failure rates raised to 30% both protocols remained pretty stable, decreasing an average of 5% in delivery rates. On the other hand, in the worst-case scenario, where 50% of the nodes failed, even in a dense network the delivery ratio barely reached 57%. In Figure 5.6 similarities between both protocols can be seen. The fact that in simple Flooding, both reliability and fault-tolerance is *assumed* because of the high redundancy, as stated by [KMG03], is validated. Unfortunately, this does not guarantee message delivery to all nodes and only relies on the inherent redundancy to obtain coverage. As previously stated in Chapter 3, SBA's drawback is that it requires up-to-date neighborhood information. Without it, unfortunately, a node that is receiving a message will erroneously calculate its forward status. Furthermore, a node has absolutely no guarantees that the same message correctly arrived at its neighbors, and therefore cannot just assume correct reception. Simulations results help support such statements.

Dominant Pruning and Wu and Li's protocols all had similar results regarding message delivery when in a dense fail-free network (with 100 nodes). DP reached 66% of the nodes and Wu 56%. But, when failures were introduced to the simulation, DP was rapidly impacted by node failures, delivering messages to less than 50% of the network with as little as 20% node failure, and barely reaching 41% when 30% of the nodes failed. In the worst case (50% failure) it was unable to reach more than 29% of the network. Wu and Li's, on the other hand, was capable of delivering messages to 56% of the network as long as node failures remained below 10%. This value decreases to about 45% when node failures increase to 30%. The delivery ratio only drops to 33% when in a worst-case scenario. Comparisons are given in the graphs of Figure 5.6.

The authors of DP inherently assume that no errors occur during message transmission, by accepting that when a node transmits a message, all of its 1-hop neighbors correctly received the message and that, when a neighboring node forwards the message, all of its 1-hop neighbors correctly receive the message as well. But, in a fault-enabled environment this is, most often, not the case and simulations result corroborate with this as visible consequences to message delivery can be seen. Dynamic Probabilistic delivery ratios' had the lowest values of all protocols, and were all between 25% and 17%.

5.4 PERFORMANCE OF RELIABLE PROTOCOLS

The efficiency, end-to-end delay and reliability results for Reliable Broadcasting, Double-Covered Broadcast and EraMobile will be presented in this section. These reliable broadcasting protocols were simulated with node failure percentages of 0%, 25% and 50%. Simulation results are presented in a similar fashion as the previous section of unreliable protocols.


(a) Simple Flooding

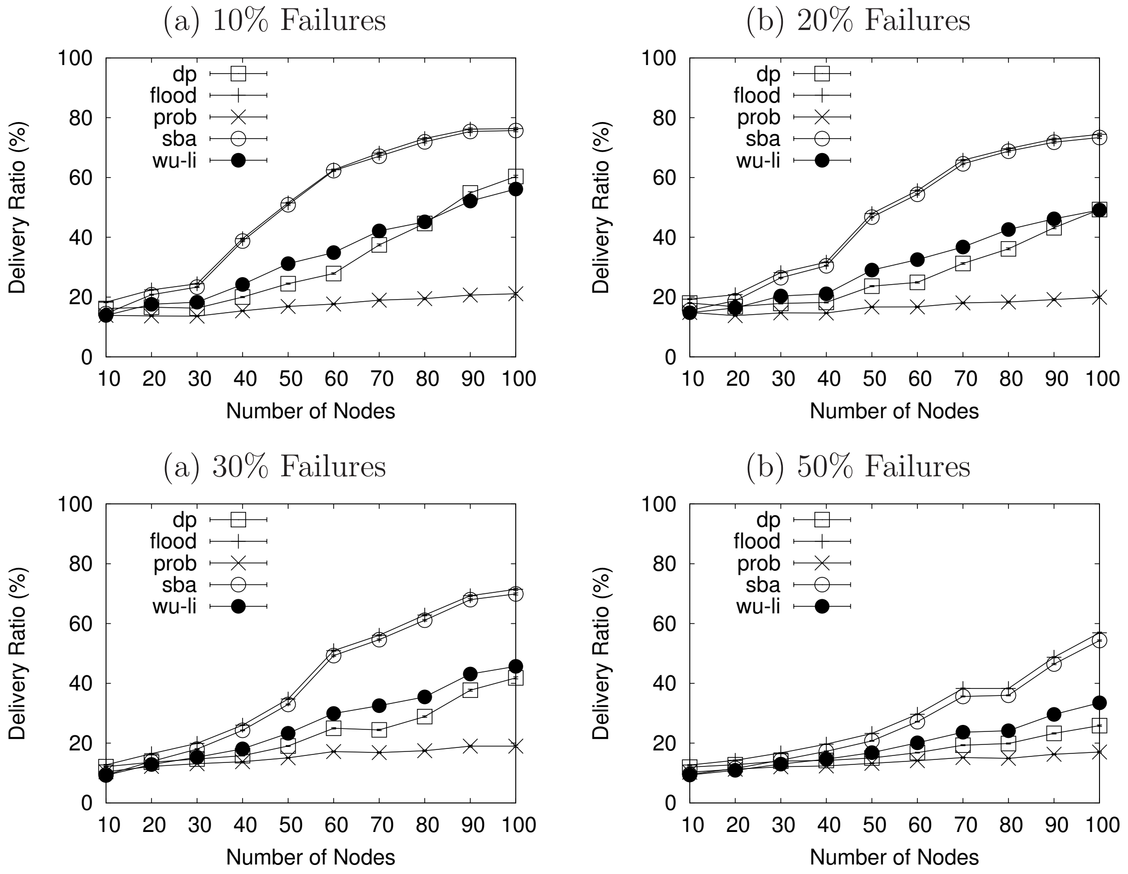
(b) Dynamic Probabilistic Approach

(c) Wu and Li's Protocol

(d) Scalable Broadcast Algorithm

(e) Dominant Pruning

Figure 5.5. Reliability Simulation Results With Failures of All Unreliable Protocols (Individual View)



(a) 10% Failures

(b) 20% Failures

(a) 30% Failures

(b) 50% Failures

Figure 5.6. Reliability Simulation Results With Failures of All Unreliable Protocols (Comparison View)

5.4.1 Efficiency

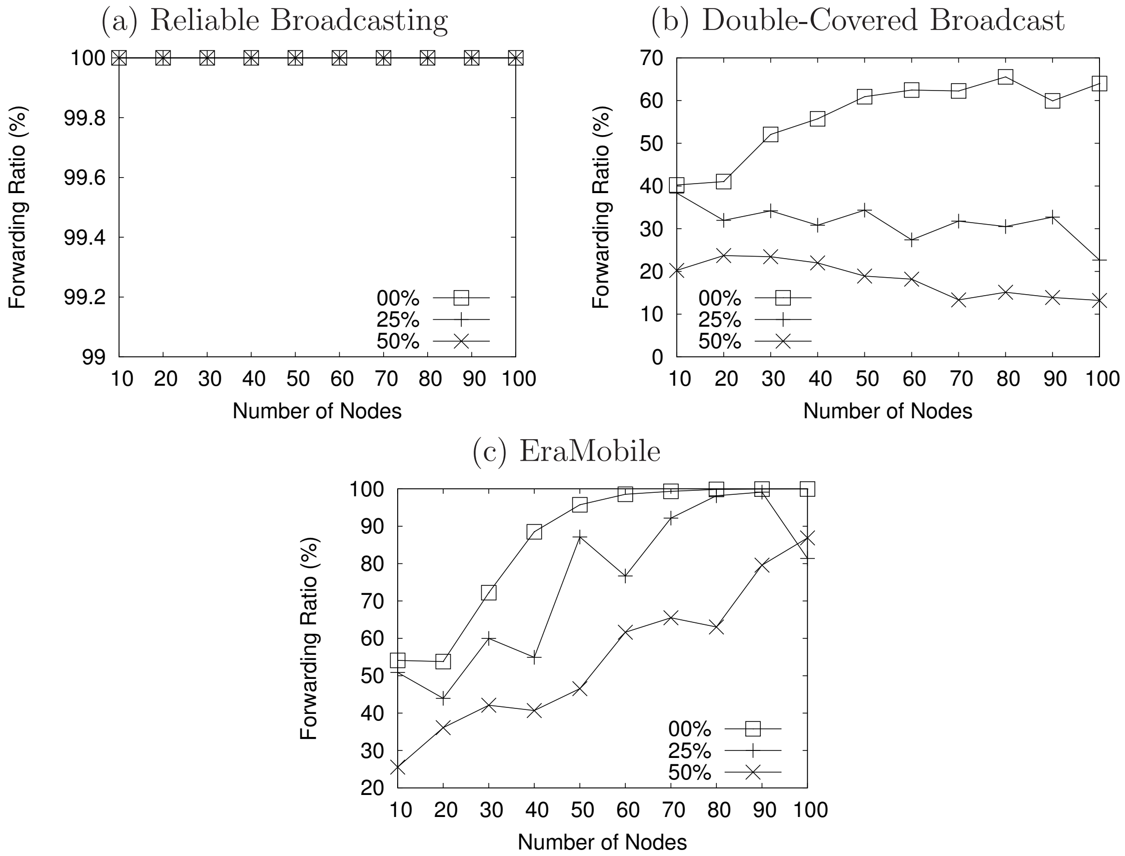


Figure 5.7. Efficiency Simulation Results With Failures of All Reliable Protocols (Individual View)

Efficiency results are shown in Figure 5.7. While RB is almost untethered by failures, maintaining a 100% forwarding ratio regardless of failures, both DCB and RB show different behaviors. DCB produces an interesting result as failures are introduced. In fail-free runs, forwarding ratios continually increase as node density grows, but with failures they go the other way. In a 25% node failure scenario, results start at 40% but finish off at less than 25%. While in a 50% node failure scenario, it starts at 20% and ends at less than 15%. Failures also impact EraMobile's results. With 25% of the network with failures and at least 50 nodes in the network, most of the time forwarding ratios are within the 80% range. With 50% of node failures most of the simulation involves less than 60%, but with 90+ nodes it rises to the $80\% - 90\%$ range. In Figure 5.8 the reader can clearly distinguish between the three protocols. DCB is always the most efficient of all.

5.4.2 End-to-End Delay

Once again DCB produces superior results when compared to the other two protocols. In Figure 5.9 and Figure 5.10 results show that DCB has delays below 0.3

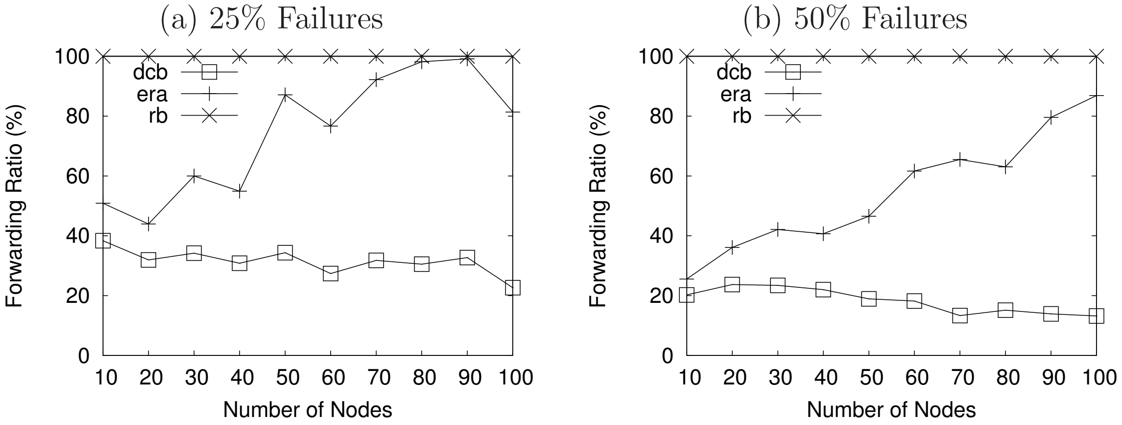


Figure 5.8. Efficiency Simulation Results With Failures of All Reliable Protocols (Comparison View)

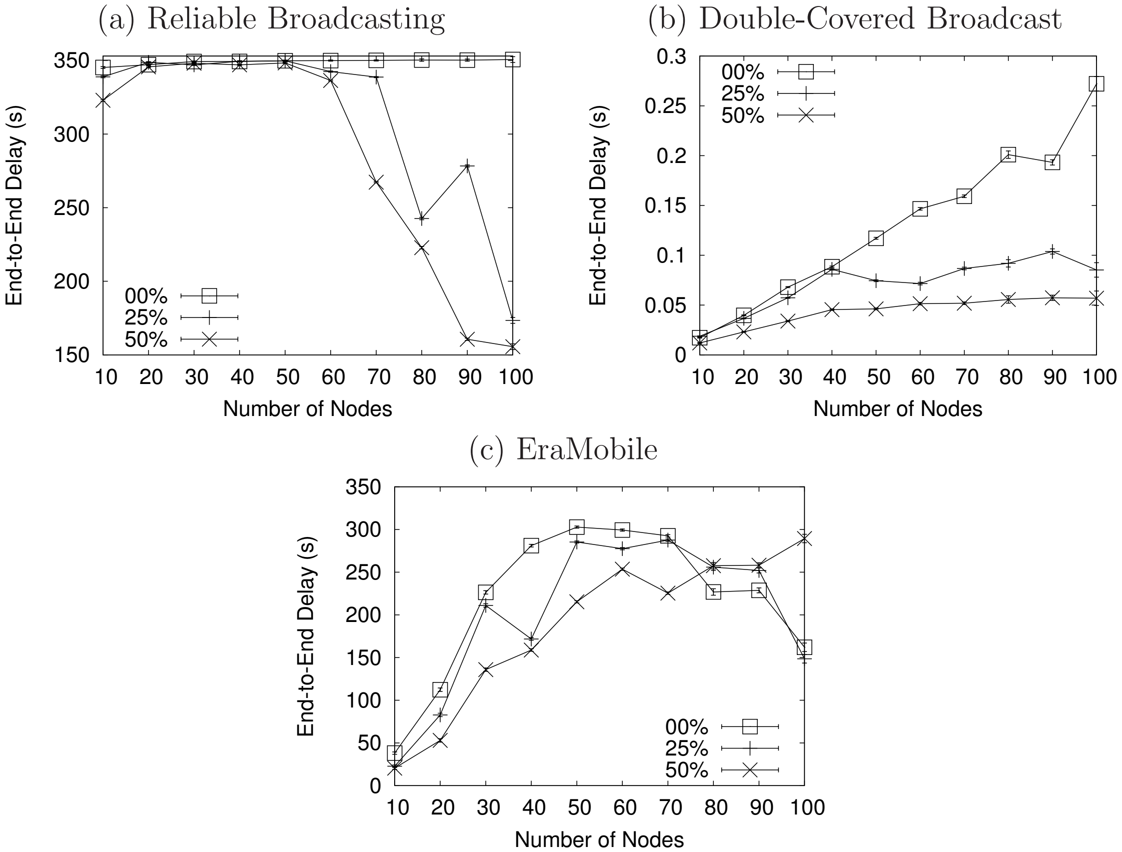


Figure 5.9. End-to-End Simulation Results With Failures of All Reliable Protocols (Individual View)

seconds all the time. RB, even in a 50% failure and 100 participanting nodes scenario takes at least 150 seconds. EraMobile's results show how quickly end-to-end delays rise. Starting off at less than 50 seconds, with 60 nodes delays are already within the 250 − 300 second range.

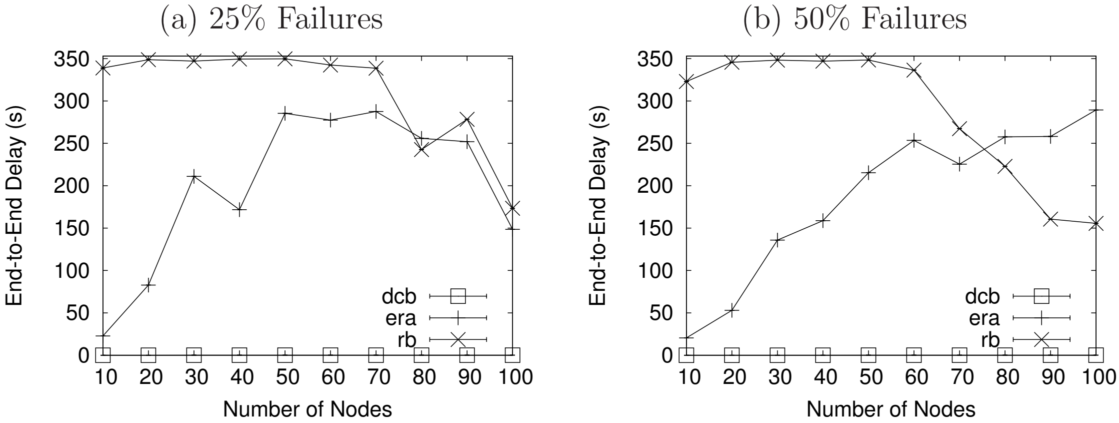


Figure 5.10. End-to-End Simulation Results With Failures of All Reliable Protocols (Comparison View)

5.4.3 Reliability

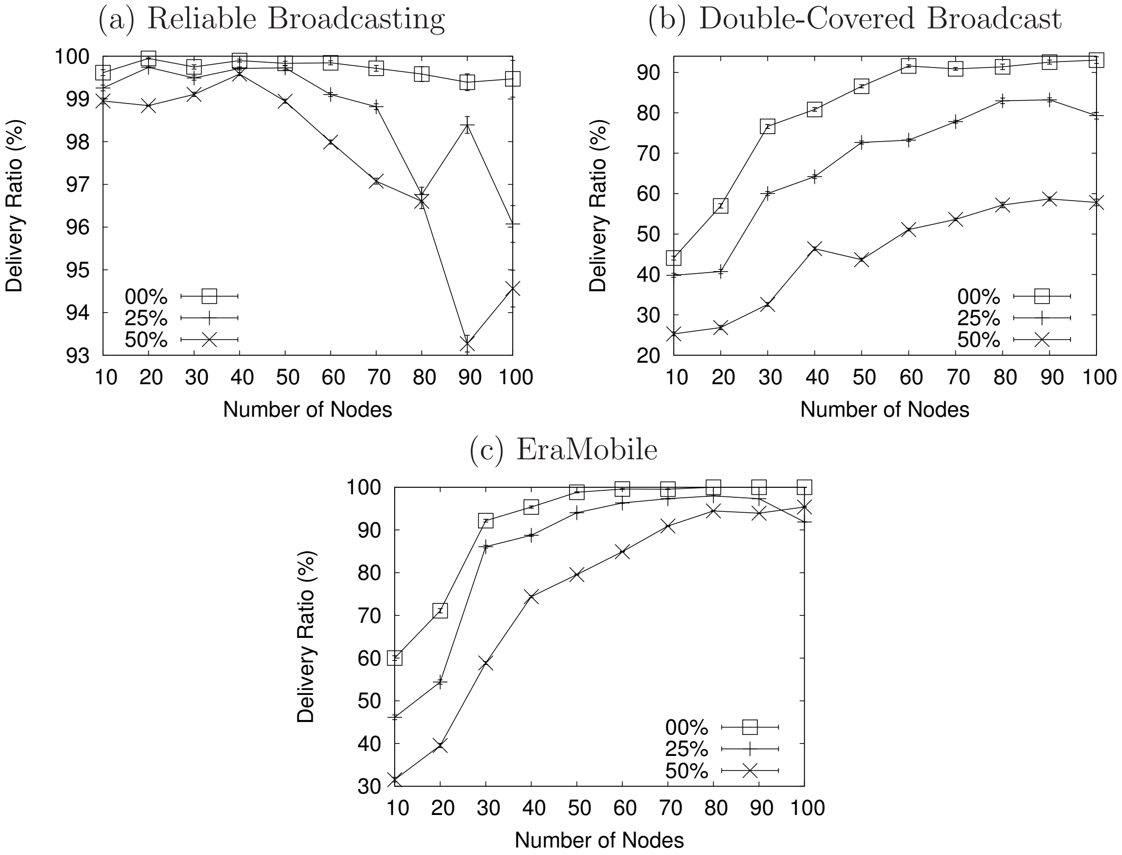
Reliability simulation results can be seen in Figure 5.11 and Figure 5.12. The worst results of RB is about 93% of network coverage, and that occurs in a 50% node failure scenario with 90 nodes in the network. For most of the other runs, reliability results are over 96%. In a sparse network those values are always between 99% and 100%. DCB's fail-free results reach 90% delivery ratios. But node failures immediatelly impact results. In a 25% scenario, at most 80% of the network is covered. In a 50% this drops to 55%. EraMobile's results are somewhat more stable than DCB, with failures dropping on average 5% when placed in a 25% node failure scenario. This rises a bit more when in a 50% node failure scenario, as results drop between 15% and 25%.

In Figure 5.12 the comparison results have been plotted so that the reader may follow the evolution of reliability results as node failures are introduced into the simulation scenarios.

5.5 LESSONS LEARNED

The performance results here presented allows us to list a few lessons learned:

- While probabilistic protocols are seen as a way to handle the lack of determinism of *MANET*s, improperly adjusting this class of protocols in order to inhibit redundant retransmissions can cause more loss than gains. While Flooding had higher delivery ratios, Dynamic Probabilistic broadcasts hardly reached the intended nodes.

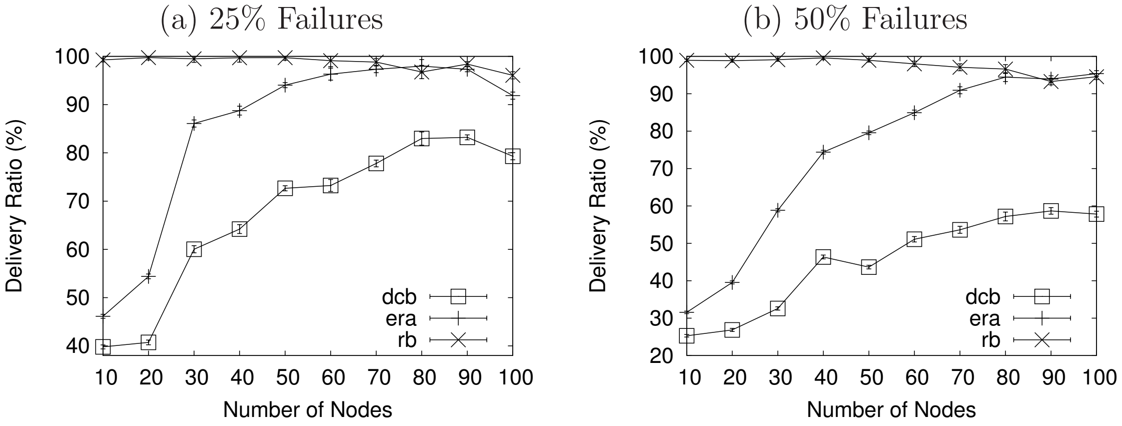


(a) Reliable Broadcasting

(b) Double-Covered Broadcast

(c) EraMobile

Figure 5.11. Reliability Simulation Results With Failures of All Reliable Protocols (Individual View)



(a) 25% Failures

(b) 50% Failures

Figure 5.12. Reliability Simulation Results With Failures of All Reliable Protocols (Comparison View)

- Most unreliable protocols handled failures up to 10% of the network without a large impact on delivery rates. This information should be used by algorithms, especially when adjusting dynamic thresholds.

- Both RB's flooding approach, as well as EraMobile's gossiping approach maintain the highest delivery ratios of all protocols, but this reliability has its cost: a large number of nodes that receive a message have to act as gateways, as demonstrated in the forwarding ratio results; furthermore, there are times that the end-to-end delays are unacceptably high.

- Most deterministic algorithms rely on correctly updated neighborhood knowledge in order to calculate forward status. But in a fail-prone scenario this information may be misleading. This seems to affect much more neighborhood designating protocols than self-pruning ones as performance results of DCB and DP indicate. However, both of these have much lower forwarding ratios and end-to-end delays.

- Clearly additional mechanisms to properly identify message reception are recommended to determine if all 1-hop and 2-hop neighbors received a message. Nodes must not just assume correct reception. While simply overhearing a retransmission by a neighboring node is one possible solution (as in DCB), its use should be limited to self-pruning algorithms where more nodes can possibly detect incorrect forwarding-related decisions. Depending on reliability requirements of the *MANET*, it might not be enough.

- For reliability to be ensured, redundancy is a must. Simulation results of the unreliable protocols show that when 60% of the network received a message, at least 25% of the network acted as gateways. And to reach 80% of the network, at least 60% of the nodes forwarded the message. Efficiency, albeit important, must not be the primary focus of a broadcasting protocol that intends to reach all correct nodes of the network. Simulation results of reliable protocols corroborate this as well.

5.6 CONCLUSION AND SUMMARY

In this chapter broadcasting protocols were introduced to a new omission fault model where temporary node failures occur. This model helped extend simulation scenarios to represent not only transient link availability with node mobility, but also external environmental interferences that are evidenced to affect message propagation, network topology changes and temporary partitions. Such a model was required in order to stress protocols with a yet more realistic simulation scenario. Although previous studies show that the broadcast protocols are very mobile resilient and support well congestion and collisions, the study conducted here show that these protocols are not fault tolerant when omission failures are taken into account. They are not capable of maintaining high delivery rate when placed in a real world scenario.

Unreliable broadcasting protocols reached 80% delivery ratios when at most 10% of the nodes failed, but barely reached 60% when a worst-case 50% scenario occurred. Reliable broadcasting protocols had superior results, but such reliability has its cost:

maxed out forwarding ratio results and unacceptable end-to-end delays. Based on these simulation results, the source of existing broadcasting problems were investigated and a list of lessons learned were presented as a step towards enhancing the capability of broadcasting algorithms to deal with omission faults in scalable scenarios.

The choice of the simulated protocols leads us to believe that the poor fault-tolerance results can be extended to most, if not all, broadcasting protocols for *MANET*s which are based on similar broadcasting mechanisms. Results, therefore, indicate a need for additional mechanisms to guarantee message delivery throughout the network. In the next chapter a simple mechanism is introduced. It is a distributed solution that extends deterministic broadcasting algorithms to help enhance a nodes' capability of dealing with omission faults. A performance comparison with existing reliable broadcasting protocols is also presented to help prove how much this mechanism raises reliability results of unreliable protocols.

CHAPTER 6

A RELIABILITY ENHANCING MECHANISM FOR DETERMINISTIC BROADCASTING PROTOCOLS

To listen closely and reply well is the highest perfection we are able to attain in the art of conversation.
—FRANCOIS DE LA ROCHEFOUCAULD

6.1 INTRODUCTION

Under a omission fault model capable of representing real-world scenarios where link failures, network partitions, topology changes and momentary node failures frequently occur, broadcasting protocols for *MANET*s – even those considered reliable – were highly impacted and were incapable of maintaining high delivery rates. Some even exhibited coverage levels that were unreasonable to accept from broadcasting protocols when placed in such a scenario. This has motivated the creation of a mechanism capable of dealing with such omission failures.

The Reliability and Stability Verification Protocol, namely *R.S.V.P*[1], will be presented in this chapter. *R.S.V.P* is an independent building block upon which any deterministic broadcasting protocol can be designed. It extends deterministic broadcasting algorithms to help enhance a nodes' capability of dealing with omission faults in a scalable scenario. To allow for scalability, the algorithm works based only on the local perception that the node has on the network and not on global information nor on a specific model of nodes behavior. In particular, it contributes to augment the reliability of those algorithms that use local neighborhood information in order to determine the forwarding node set for message retransmission. Two different versions of *R.S.V.P* are presented, one for scenarios where the rate of change of the neighborhood set is moderate, and another for when the rate of change is very high.

This chapter then applies *R.S.V.P* to three of the unreliable protocols previously described. Simulation results of these enhanced protocols will then be compared to the simulation results of the reliable protocols. Specifically, the three unreliable protocols chosen were Wu and Li's Protocol [WL99], Scalable Broadcast Algorithm [PL00] and Dominant Pruning [LK01]. While the three reliable protocols were Reliable Broadcast [AVC95], Double-Covered Broadcast [LW07] and EraMobile [OGA06]. Simulation results indicate that all enhanced protocols had an expressive delivery ratio increase and much lower end-to-end delays (when compared against the results of the reliable protocols).

[1] *R.S.V.P* means *Répondez S'il Vous Plaît* in French, and this meaning captures perfectly the rationale behind our mechanism.

6.2 EXPLORING A HYBRID APPROACH

In Chapter 4 three strategies for ensuring sufficient coverage while broadcasting in *MANET*s were presented. They were forward error correction, link reliability and retransmission.

- The forward error correction strategy piggybacks additional redundant data in messages permitting receiving nodes to re-assemble a certain number of missing messages. The biggest problem, however, is determining exactly how much redundant data has to be generated in order to ensure correct message delivery, and this might not be able to be properly defined before-hand [Kun03].

- The link reliability strategy attempts to find neighboring nodes that suffer less mid-conversation failures. Each node subjectively measures neighboring nodes and chooses the ones that seem more trustworthy. Thus, instead of randomly choosing a forwarding node, the more reliable ones are preferred. The downside to existing protocols that implement such strategy is the fact that they rely on the age of the link as a reliability metric. These works supposed that a long-lived route had a tendency to remain valid without losing connectivity over time. However, according to [GdWFM02], these assumptions can only be justified for a few dynamic scenarios and actually not much can be presumed concerning the movement of two nodes that just came into one another's transmission range. Clearly, the duration of a link largely depends on the mobility pattern of the nodes. In fact, depending on the mobility pattern, old links will not necessarily be more stable than young ones.

- Finally, the retransmission strategy determines that nodes should keep retransmitting a message until all the receivers acknowledge reception. Acknowledgment schemes, however, are known to have serious scalability problems. On the other hand, protocols that relied on this strategy where found to be better designed, while protocols from the other two strategies often lacked implementation details or demonstrated poor performance results.

By combining the last two options, a reliable broadcasting mechanism would be capable of choosing nodes that seem more trustworthy while retransmitting a message until all receivers acknowledge. But it would seem common-sense that users are not necessarily interested in how long a link might live, but rather in whether or not the link is available for the time it is needed. Therefore, instead of using age as a reliability metric, a new outlook on the concept of link reliability will be used which takes into account the correct behavior of forwarding nodes responsible for retransmitting messages. Also, in order to reduce the number of expected acknowledgements, retransmissions will be determined based only on local neighborhood replies. That is, nodes will only retransmit a message if, for some reason, neighbors up to 2-hops did not receive it.

6.3 RELIABILITY AND STABILITY VERIFICATION PROTOCOL (R.S.V.P)

The *Reliability and Stability Verification Protocol* was thought up as a mechanism capable of augmenting reliability in deterministic broadcasting protocols. It was designed as a building block upon which any existing deterministic broadcasting protocol can be defined. By applying *R.S.V.P* to a broadcasting algorithm, nodes will be able to locally identify which neighboring nodes are thought to be more reliable, according to previous observations and interactions. This information can be gathered in the topology update phase of the broadcasting protocols. Then, by using this knowledge during the gateway selection process, nodes are able to increase the chances of ensuring that their communication will suffer less mid-conversation failures and high drop rates. An overview of this can be seen in Figure 6.1.

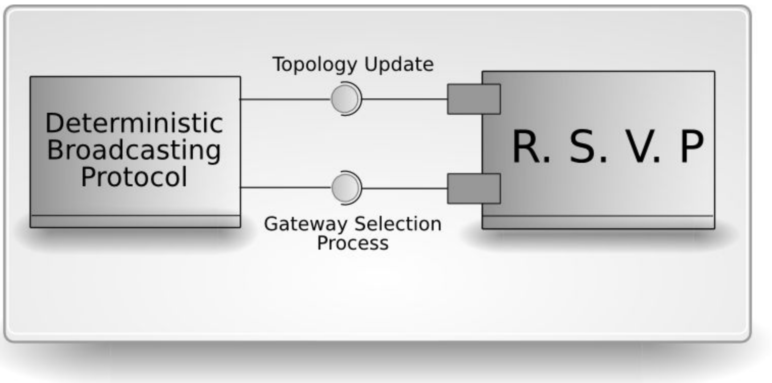

Topology Update
Deterministic Broadcasting Protocol
R. S. V. P
Gateway Selection Process

Figure 6.1. Usage of R.S.V.P by a Deterministic Broadcasting Protocol

6.3.1 Principle

The main idea behind *R.S.V.P* is to retransmit a message until all 2-hop nodes confirm reception. During this process, it is capable of rating neighboring nodes according to the number of times that a message has to be retransmitted. Although subjective, such mechanism would help assign a numerical value to a perceived observation made by any node regarding another node. In fact, this would establish a trust relationship between neighboring nodes that – much like humans – could grow or decay over a period of time. Note that *R.S.V.P* does not modify how each protocol selects forwarding nodes, instead it extends them.

To reduce the chances of incurring the *ACK implosion problem*, instead of requiring both 1-hop and 2-hop nodes to confirm reception, only 2-hop nodes must reply. From a nodes' point of view it is of the utmost importance that surrounding 1-hop nodes be the ones that receive most attention. This is due to the fact that those nodes are the ones who are to forward any future broadcasts. But in order to determine if these 1-hop nodes are actually forwarding the messages, message reception acknowledgement will have to be sent by the 2-hop neighbors. After all, messages can only reach 2-hop nodes if they are correctly forwarded by 1-hop nodes. These *ACK* messages can be piggy-backed in existing "hello" neighborhood set exchange messages that are periodically transmitted by neighboring nodes to maintain neighborhood connectivity information updated. By doing this, message transmission is further reduced. The reader is reminded that such message exchange is very common in deterministic broadcasting protocols. Such messages include the node identification and its list of neighbors. Neighbors that received the message, are then capable of learning the topology information within 2 hops. By also piggy-backing message acknowledgements, nodes will be able to confirm message reception and define node reliability values, as well as maintaining an updated perception of neighboring nodes.

Every node maintains a boolean vector of n bits (n equal to the total number of 1-hop and 2-hop neighbors) for every message sent, that indicates whether the other n nodes have received that message. This vector is called the *confirmed reception vector* (CR) and is updated every time a “hello” message arrives containing a message confirmation acknowledgment. Thus, $CR_i^m(v)$ refers to the vector of node i concerning reception of message m by node v. If a bit of this vector is set, it means that node v has confirmed reception of the message m. Since a node cannot un-confirm reception, once set it guarantees reception.

Each node also maintains a *perceived reliability vector* (PR) which contains all nodes in its 1-hop vicinity. This vector is used to establish the perceived reliability that a node has of another node. Thus $PR_i(v)$ defines the perceived reliability that node i has of node v. At any moment, any given node may compare the perceived reliability values of his neighbors, can establish nodes that he trusts more than other or even determine a minimum reliability threshold by which nodes with lower PR values are ignored.

$R.S.V.P$ requires that at every “hello” transmission, node k piggy-backs CR_k^m. When a node j receives the vector CR_k^m for the first time, it sets its own CR_j^m equal to the received knowledge vector and sets its own bit to express the reception (in other words, $CR_j^m(k) = true$). Next time node j transmits a “hello” message, it will include its updated copy of CR_j^m. By piggy-backing a node's confirmed reception vector, receiving nodes are able to detect which 2-hop nodes have not yet received that message. A node will keep retransmitting m until all 2-hop neighbors confirm reception. When this condition is finally verified, the message can be inhibited of future re-transmissions by removal from internal buffers.

6.3.2 Detailed Protocol Description

$R.S.V.P$ will now be described as can be seen in Algorithm 1. By itself, it will not have any effect on message broadcasting since we have designed it as a building block upon which any existing deterministic broadcasting protocol can be defined. The process by which neighborhood information is updated is specific to the broadcasting protocol used and therefore will not be defined in the following algorithm, but we assume that this information is available to our mechanism in order to determine neighbors' status.

Constant Symbols

The following constants will be used by the mechanism.

- $BCAST$: a generic deterministic broadcasting protocol responsible for determining a set of forwarding nodes.
- *initial_perceived_reliability*: value of the initial perceived reliability.
- *decay_value*: amount by which the perceived reliability value decays with time.
- *increase_value*: amount by which the perceived reliability value is incremented every time a new "hello" message is received.
- *decrease_value*: penalty amount value by which the perceived reliability value is decreased when nodes do not confirm reception.
- τ: internal timeout between consecutive "hello" transmissions.

Variable Symbols

The following variables are setup and used by the mechanism to define reliability.

- $N(v)$: set of directly connected (1-hop) neighbors of node v.
- $N(N(v))$: set of 2-hop neighbors of node v.
- F_v: set of forwarding nodes from nodes' v point of view calculated using $BCAST$.
- $PR_v(u)$: the perceived reliability of node u from nodes' v point of view.
- $CR_v^m(u)$: boolean reception vector which indicates reception of message m sent by node v to node u.
- X_v^m: set of neighboring nodes which have not yet confirmed reception of a message m sent by v.
- H_v: history set of messages sent by v.

When initialized, all perceived reliability values are set to *initial_perceived_reliability* (lines 2-4). The history set of messages is also cleared (line 5). Every time a new neighbor is added to the neighborhood set its value is also initialized. The value set of *initial_perceived_reliability* has to be well chosen as well. The experiments conducted by [KNG$^+$04] showed that very recent neighbors not necessarily meant a working communication link. That is why the lower the initial value, the more older neighbors are favored over new ones.

For every message m sent for the first time, the confirmed reception vector CR_v^m is initialized as well by setting all values to false to indicate that no neighbor has yet received that message, in the same manner, the set X_v^m is set to all 1-hop and 2-hop neighbors and the message m is added to the history set H_v (lines 8-12). Since

Algorithm 1 Reliability Metric Mechanism

```
Initialization procedure for node $v$:
for all $u \in N(v)$ do
   $PR_v(u) = initial\_perceived\_reliability$
end for
$H_v = \emptyset$

For every message $m$ sent by node $v$ for the first time:
for all $u \in (N(v) \cup N(N(v)))$ do
   $CR_v^m(u) = false$
end for
$X_v^m = N(v) \cup N(N(v))$
$H_v = H_v \cup \{v\}$

When “hello” is received from node $u$ by node $v$ containing $CR_u^m$:
$PR_v(u) = PR_v(u) + increase\_value$
update status of $CR_v^m(u)$ and of $X_v^m(u)$

At every interval $\tau$ at node $v$:
for all $u \in N(v)$ do
   $PR_v(u) = decay\_value * PR_v(u)$
end for
for all $message\ m \in H_v$ do
   if $X_v^m \neq \emptyset$ then
      re-transmit current message $m$
      for all $node\ u \in X_v^m\ for\ message\ m$ do
         if $u\ is\ 1\ hop\ neighbor$ then
            $PR_v(u) = PR_v(u) - decrease\_value$
         else if $u\ is\ 2\ hop\ neighbor$ then
            for all $1\ hop\ node\ t \in F_v\ connected\ to\ u$ do
               $PR_v(t) = PR_v(t) - decrease\_value$
            end for
         end if
      end for
   end if
end for
```

every "hello" message has a confirmed reception vector piggybacked, this information is analyzed in order to update the status of the neighboring nodes that have not yet confirmed reception (line 16). At this moment the perceived reliability value is also updated to reflect the fact that the neighbor is still up, by incrementing it with *increase_value* (line 15).

Since a typical deterministic broadcasting protocol needs to update its neighborhood view periodically, we assume that this occurs after every interval τ. It is also at this moment that the perceived reliability value is updated. Initially to reflect the fact that the perceived reliability should decay with time, we apply a *decay_value* (lines 19-21). Every time a neighbor does not confirm reception, a sender node can conclude that either it is his fault or it is the receivers fault. Thus the perceived reliability value must be updated. If it is a 1-hop neighbor u that does not confirm reception of a message sent by v, $PR_v(u)$ will decrease by *decrease_value* (line 27). If it is a 2-hop neighbor, then *every* 1-hop neighbor that is directly connected to the 2-hop node and belongs to the set of forwarding nodes will be penalized by decreasing the perceived reliability of that node by *decrease_value* (line 30). This is yet another point of the algorithm that depends on the $BCAST$ protocol, this time for the selection of gateway nodes. This is necessary since our mechanism needs to establish which 1-hop neighbors are reliable, and if this gateway neighbor seems unable to correctly forward a message, it must be penalized.

6.3.3 Correctness Arguments for R.S.V.P

Consider an execution where a message m is sent at time t_0 and where the sender v is correct. Assume that there exists a stable two-way communication channel between all 1-hop neighbors and that the network meets the network connectivity requirement. Additionally, assume that between any 2-hop neighbors there exists a multi-hop stable communication path.

Proposition 6.1 *(Guarantee to Establish Reliability Values) Any* R.S.V.P*-enabled deterministic broadcasting protocol is guaranteed to eventually be able to establish reliability values to all* 1*-hop neighbors known at time* t_0 *even if nodes fail, in spite of neighborhood topology changes, as long as only previously connected* 1*-hop neighbors from time* t_0 *who continue to be* 1*-hop neighbors are expected to be assigned reliability values.*

Proof In the described scenario, the network does not permanently partition and all 1-hop neighbors (who were also 1-hop neighbors at time t_0) are able to eventually send and receive messages with success. Therefore, eventually the sender will receive "hello" messages containing updated neighborhood sets and confirmed reception vector from all 1-hop neighbors. From the updated neighborhood sets received, the sender will detect if any nodes have been removed or added from the original set of nodes present at time t_0. New nodes are ignored, while nodes that are no longer in the local neighborhood will be removed from both the senders' neighborhood set as well as from X_v^m. Due to failures, a node may be incorrectly removed from this set, but this will only be temporary since eventually the sender will detect and restore both sets. Through the confirmed reception vectors received, the sender is able to determine the reception status of all nodes belonging to its' 2-hop neighborhood, including the status of the 1-hop nodes. With this information in hand, the sender will update the perceived reliability value of those 1-hop nodes. Therefore, reliability values can be established for all 1-hop neighbors known at time t_0 that continue to be 1-hop neighbors through the *R.S.V.P* algorithm. ▮

Proposition 6.2 *(Local Delivery Guarantee) If node v is executing a* R.S.V.P*-enabled deterministic broadcasting protocol and broadcasts a message at time t_0, then all local (up to 2-hops) nodes known by v at time t_0 that are executing the same protocol eventually deliver that message, as long as the communication channel is valid.*

Proof Since nodes remain local and there exists a stable communication channel, at a later time t_1, the neighborhood set of v and X_v^m will still contain all of them. Even if a node is temporarily detected as failed, eventually an updated "hello" message will correctly restore the node to the sets. But as long as X_v^m is not empty, the sender will continue to retransmit the message. Since there exists a multi-hop stable communication path, eventually these nodes will receive the message and acknowledge reception through the confirmed reception vectors. At that moment, the sender will be assured that all local nodes delivered the message. ▮

6.3.4 Usage of R.S.V.P in Very Dynamic Topology Scenarios

What is implicitly assumed by the *R.S.V.P* algorithm is that the perceived reliability of neighboring nodes can eventually be determined. In other words nodes must move in a correlated manner or according to a group mobility model. The perceived reliability is a value that is determined over time by observations and interactions. Through message exchange this value will eventually detect those nodes which are reliable. It does not make sense to try to compute the perceived reliability value in any other scenarios which cannot guarantee this stability. In an unstable scenario for example, new nodes will constantly be added to the neighborhood set due to mobility or failure, as will old nodes be constantly removed. Therefore depending on the rate of change of the neighborhood set, this value will not reflect anything meaningful.

Although not implemented, a second version of *R.S.V.P* was envisioned for extremely mobile scenarios where link breakage constantly occurs. This can be seen in Algorithm 2. In this version, instead of continuously maintaining an updated perceived reliability vector, every unique message will induce a snapshot of the reliability of neighboring nodes. This is due to the fact that we are now assuming a less stable scenario in which the set neighbors are constantly changing. As a consequence, the perceived reliability vector is not used.

The following additional symbols are used:

- *reliability_distrust*: warning message sent by a node that contains the list of nodes that are thought to be unreliable;
- N_v^m: counter of number of times a message m has been sent by node v.

Algorithm 2 Reliability Metric Mechanism for Unstable Scenarios

```
For every message sent by node $v$ for the first time:
for all $u \in (N(v) \cup N(N(v)))$ do
    $CR_v^m(u) = false$
end for
$N_v^m = 0$
$X_v^m = N(v) \cup N(N(v))$

Every time $\tau$ at node $v$:
for all $message\ m \in H_v$ do
    if $N_v^m \leq maximum\ number\ of\ retransmission\ attempts$ then
        re-transmit current message $m$
        $N_v^m = N_v^m + 1$
    else
        for all $node\ u \in X_v^m\ for\ message\ m$ do
            if $u\ is\ 1\ hop\ neighbor$ then
                $reliability\_distrust \leftarrow u$
            else if $u\ is\ 2\ hop\ neighbor$ then
                $reliability\_distrust \leftarrow all\ 1\ hop\ node\ t\ who\ are\ gateways\ and\ connected\ to\ u$
            end if
        end for
        send $reliability\_distrust$ message
        stop transmitting this message
    end if
end for

When node $v$ is gateway and receives $reliability\_distrust$ message:
for all $u \in reliability\_distrust$ do
    remove $u$ from Gateway set
end for
restart Gateway selection process

When "hello" is received from node $u$ by node $v$ containing $CR_u^m$:
update status of $CR_v^m(u)$ and of $X_v^m(u)$
```

The main idea behind this version is to re-send a message a certain number of times. When that number is reached, the nodes which have node yet confirmed reception are determined. The 1-hop neighbors that *should* have had re-transmitted the message (but, for whatever reason where unable to) are considered unreliable for that message and a warning is raised. All nodes that receive that warning will remove the nodes contained in the warning message from their neighborhood set and re-calculate the forwarding node set.

Each unique message will be re-transmitted up to a maximum number of times (line 10). If this number is reached and there are still nodes which have not confirmed, the sending node will assume transmission failures have occurred, will determine which 1-hop forwarding nodes are not reliable (lines 15-18) and, finally will broadcast a *reliability_distrust* message informing nearby nodes that they should ignore those nodes (line 21). Once sent, the node will cease to re-transmit (line 22).

When any node receives a *reliability_distrust* message – *for that message only* – it will re-compute the forwarding node set and determine if it should re-transmit the message or not (lines 26-30). This measure helps raise broadcast redundancy in order to improve the local delivery ratio.

6.3.5 Related Work

Some of the ideas behind *R.S.V.P* were based on previous works which are now presented. In their search for a solution to the consensus problem in *MANET*s, Vollset and Ezhilchelvan [VE05] introduced three crash-tolerant broadcasting protocols capable of tolerating up to a pre-defined number F of failed nodes. Instead of forwarding a message only once, their protocols require the retransmission of the same message until at least $n - F$ nodes (where n is the number of nodes in the network) confirm reception. This requirement, however, limits the utilization of the protocols in realistic situations. Nonetheless, of the three protocols, the simple mechanism behind the Proactive Dissemination Protocol (PDP) was found to be the most interesting and one particular idea was used during the development of the *R.S.V.P* mechanism. Each node using the PDP protocol must know of every other node present in the MANET. This is necessary since every node must maintain a boolean vector of n bits that indicates whether the other n nodes have received a message m. This vector is called the *knowledge vector* by the authors. Any reader can easily notice the resemblance between their vector and the *confirmed reception vector* used in *R.S.V.P*. We refer the reader to [VE05] for a more complete description of PDP.

6.4 PERFORMANCE OF ENHANCED PROTOCOLS

The performance results of Wu and Li's Protocol, Scalable Broadcast Algorithm and Dominant Pruning enhanced with *R.S.V.P* will now be presented. The enhanced results will be compared to the original versions of the respective protocol. Since latter the protocols will be compared to the reliable protocols, the fault model used will be modified accordingly, i.e., during each one of the runs an uniformly random selected set of nodes will fail to send and receive *messages with large payloads, such as broadcast* *messages, but will still be able to exchange neighborhood and control messages* for the timespan of 10 seconds. The exact number of nodes that fail on any given run also depends on the percentage of failed nodes, but are limited to 0%, 25% and 50%.

Since the *R.S.V.P* mechanism rates neighboring nodes, this information can be used by the broadcasting protocols in order to remove those nodes that are below a reliability threshold. The enhanced protocols were then modified to use this information. Furthermore, a maximum number of retries was used to properly terminate the broadcasting process. In Table 6.4, the exact values used during simulations are shown.

Simulation Parameters for *R.S.V.P*	
Initial Perceived Reliability	300
Max. Retransmission Attempts	5
Decay Value	0.9
Increase Value	30
Decrease Value	60
Reliability Threshold	210

Table 6.1. Simulation Parameters

6.4.1 Wu and Li's Protocol

Efficiency. In Figure 6.2 the efficiency results of Wu and Li's Protocol enhanced with *R.S.V.P* can be seen. Note how the enhanced version of the protocol uses almost 4 times as many gateways. The forwarding ratio which was always below the 50% mark, jumped to at least 90% , even involving at moments all 100% of the receiving nodes.

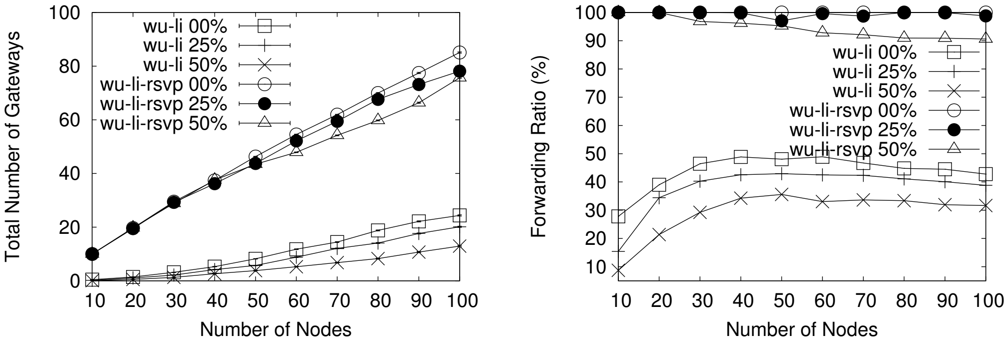


Figure 6.2. Simulation Results: Efficiency of Wu and Li's Protocol with and without *R.S.V.P*

End-to-End Delay and Dropped Messages. Much like the efficiency metric, end-to-end delays also increase when enhanced with *R.S.V.P*, especially when in a sparse network. This can be seen in Figure 6.3. Delays, however, are lowered to the $8-15$ second range when node density increases (starting from 60 nodes in the network). Curiously for most of the time the delays of the enhanced protocol are almost the same, even with node failures. The number of dropped messages also rise with *R.S.V.P*, but only in a sparse network. With 70 nodes in the network, drop rates of the enhanced protocol are smaller than the non-enhanced protocol.

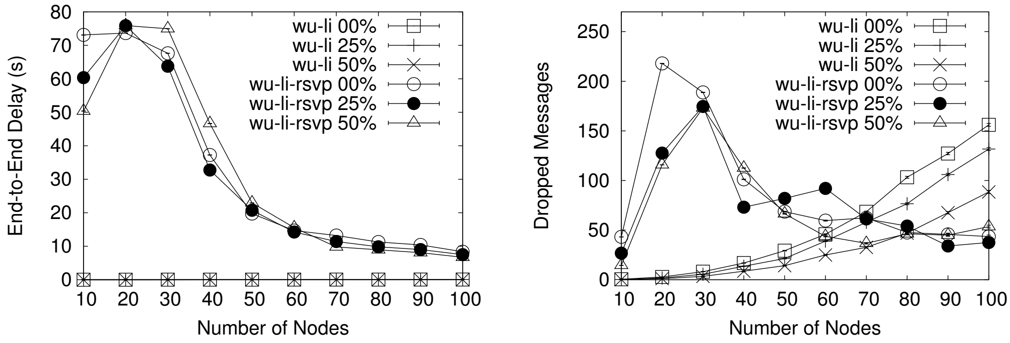


Figure 6.3. Simulation Results: Delay and Drop Rates of Wu and Li's Protocol with and without *R.S.V.P*

Reliability. In Figure 6.4 the differences between Wu and Li's protocol running with and without the usage of R.S.V.P can be seen. While the R.S.V.P-enabled version had delivery rates up to 100% percent, without it they barely reached 60%. On average, the enhanced version when on a network with at least 50 nodes was able to deliver messages to 79% − 91% of the nodes. The enhanced version was also more resistant to failures.

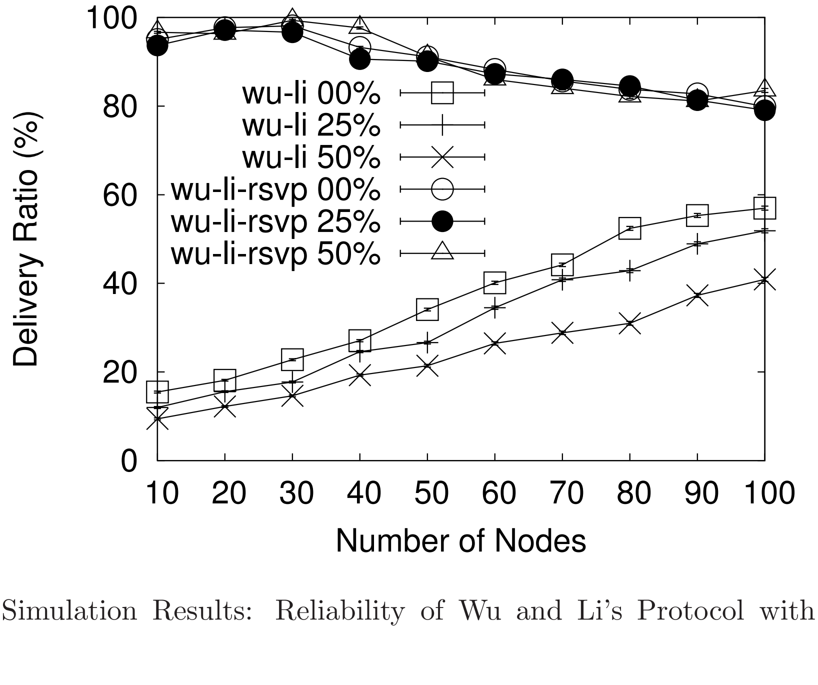


Figure 6.4. Simulation Results: Reliability of Wu and Li's Protocol with and without *R.S.V.P*

6.4.2 Scalable Broadcast Algorithm

Efficiency. SBA's efficiency results are not exactly good, as even the non-enhanced protocol involves between 60% − 80% of the network. With *R.S.V.P* this values soars to 100%, even in a very sparse network with only 10 nodes. This behavior occurs regardless of the number of failures. Results are plotted in Figure 6.5.

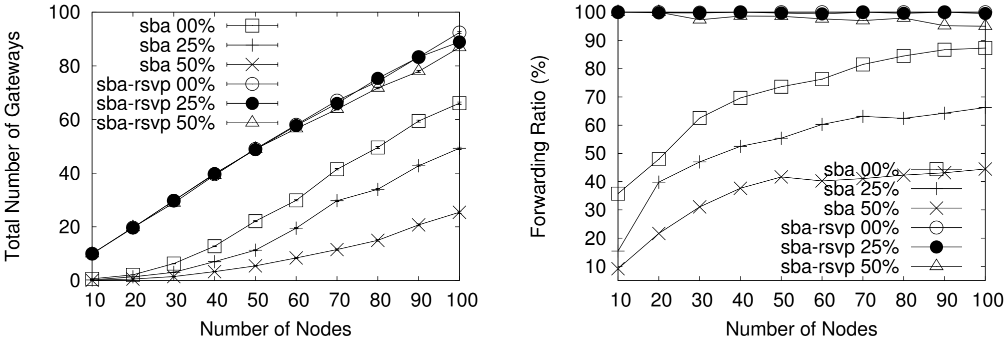


Figure 6.5. Simulation Results: Efficiency of SBA with and without *R.S.V.P*

End-to-End Delay and Dropped Messages. Both delays and dropped messages also rise in the enhanced version of SBA. In Figure 6.6 the reader can see that in a network with 20 − 40 nodes, delays stay between 70 and 75 seconds. However, as more nodes are added to the network, delays drop to as little as 20 seconds. Dropped messages also follow a similar pattern. With more than 80 nodes, dropped messages for the enhanced protocol are almost the same as the non-enhanced protocol.

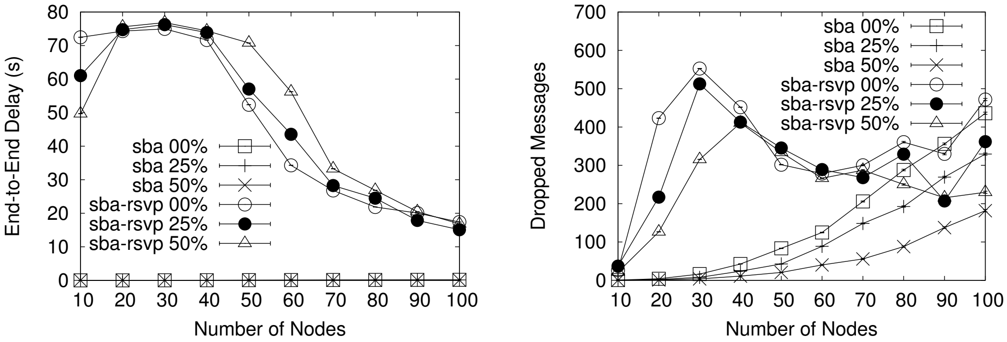


Figure 6.6. Simulation Results: Delay and Drop Rates of SBA with and without *R.S.V.P*

Reliability. Once again we can see, in Figure 6.7, how delivery ratio increases for the enhanced SBA protocol. In all runs, delivery was guaranteed to at least 90% of the nodes. Also, it is interesting to note that node failure does not have such a negative effect when relying on *R.S.V.P.* In the original SBA, in a fail-free scenario, rates reach 75% but drop to 57% with 50% node failures.

6.4.3 Dominant Pruning

Efficiency. Unlike previous protocols, neither the forwarding ratio nor the number of gateways is that much impacted, at least in a dense network, where results are

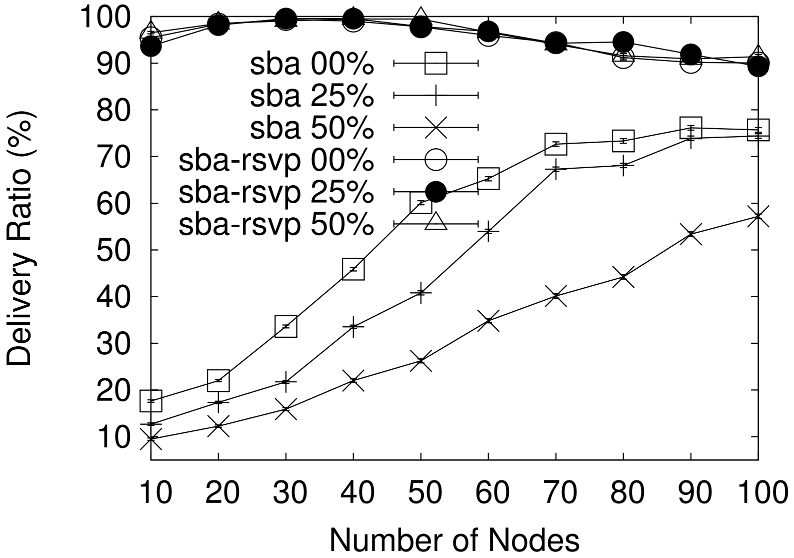


Figure 6.7. Simulation Results: Reliability of SBA with and without *R.S.V.P*

at times identical. Figure 6.8 demonstrates this. The total number of gateways for the enhanced version of the protocol shows only a slight rise at times. While the forwarding ratio in a sparse network rises to almost 80%, it quickly drops to at most 45% when more nodes take part of the broadcast. These results are mostly due to DP being a neighborhood-designating protocol.

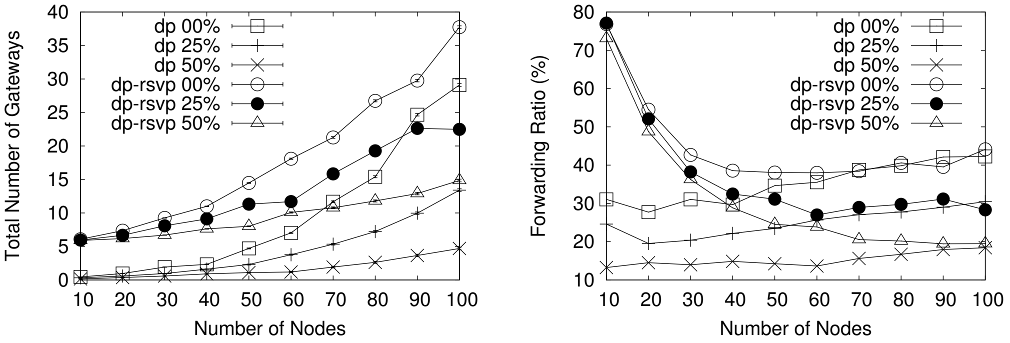


Figure 6.8. Simulation Results: Efficiency of DP with and without *R.S.V.P*

End-to-End Delay and Dropped Messages. While end-to-end delays rise, of all three protocol, enhanced DP has the smallest and most uniform delays. In Figure 6.9 results show that it takes on average 5 seconds to broadcast a message. Results are also unaffected by failures. Dropped messages, however, show a increase in the enhanced version of DP.

Reliability. In Figure 6.10 the performance of DP in its original and R.S.V.P-enabled version can be seen. DP is the protocol with the lowest delivery ratios of all three simulated protocols. In the non-enhanced version, delivery ratios reach the maximum of 68% when in a fail-free scenario, but lower to only 25% when in a 50% scenario. In the extended version, however, coverage is always between 63% − 85%.

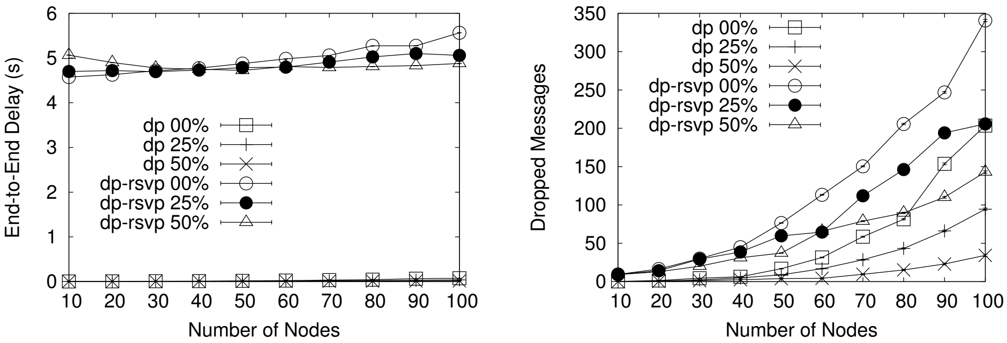


Figure 6.9. Simulation Results: Delay and Drop Rates of DP with and without *R.S.V.P*

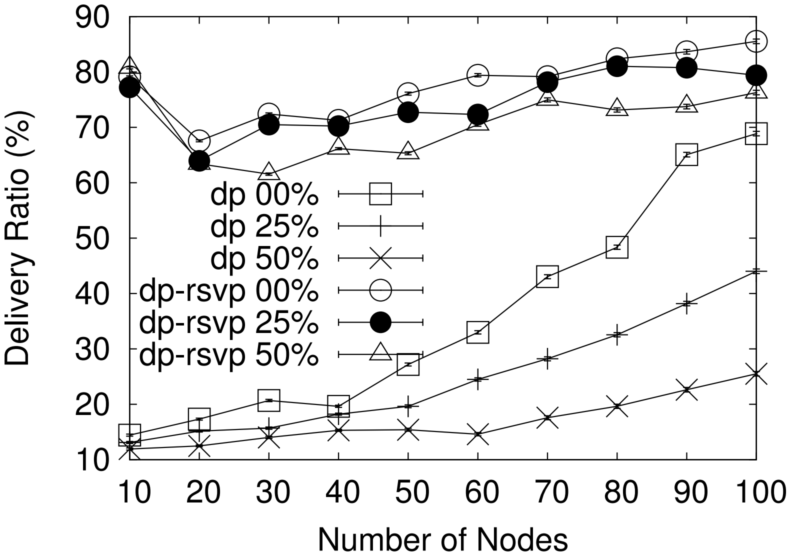


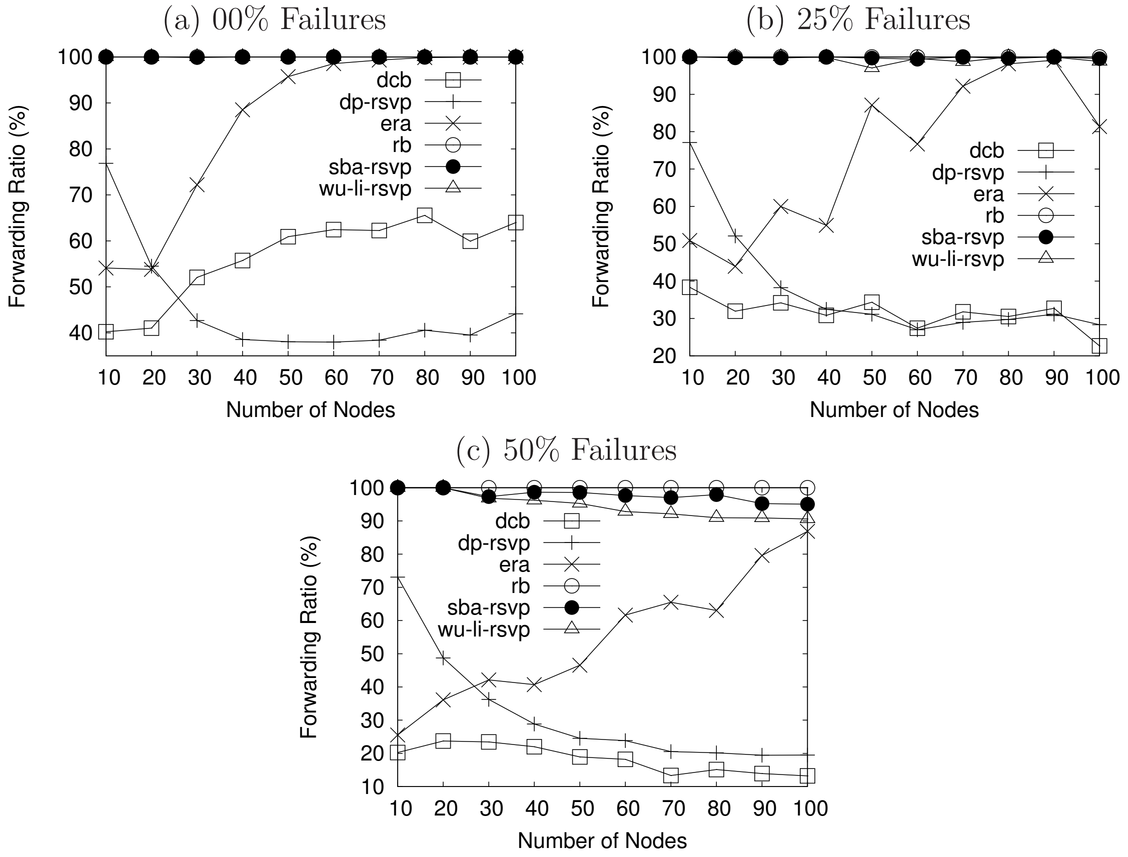
Figure 6.10. Simulation Results: Reliability of DP with and without *R.S.V.P*

6.4.4 Enhanced vs. Reliable Protocols

In this section the simulation results of the enhanced versions of Wu and Li's Protocol, Scalable Broadcast Algorithm and Dominant Pruning will be compared to the results of Reliable Broadcast, Double-Covered Broadcast and EraMobile.

Efficiency. In Figure 6.11 the efficiency results of the comparison between enhanced and reliable protocols can be seen. In a fail-free scenario both DCB and enhanced DP achieve the best efficiency. Enhanced DP for most of the time involves at most 40% of the network, while DCB 60%. All the other protocols always involve all nodes in the broadcasting process. As failures as introduced in the network the overall picture slightly changes. While enhanced DP is almost unchanged with 25% node failures, with 50% node failures both enhanced SBA and WU finally involve less than 100% of the nodes. Wu is a bit more efficient than SBA, involving at times only 90% of the network.

End-to-End Delay and Dropped Messages. Regarding end-to-end delays, all protocols that relied on *R.S.V.P* had results that took less time than both RB and EraMobile. This was reproduced in all failure scenarios, as can be seen in Figure 6.12. Of the reliable protocols, only DBC achieved lower delays. Enhanced SBA had the worst delay times, with WU coming in second. Enhanced DP was the protocol with the best delay times of all three.


(a) 00% Failures

(b) 25% Failures

(c) 50% Failures

Figure 6.11. Efficiency Simulation Results With Failures of All Enhanced vs. Reliable Protocols

Reliability. Reliability results from the simulations can be seen in Figure 6.13. Of all enhanced versions of the protocols, SBA was the one most benefited from relying on *R.S.V.P.* Regardless of node failures, it achieved at least 90% delivery ratios. In fact, as more failures were induced in the network, the better the reliability results produced. Enhanced Wu had also very stable results, even when placed under different failures scenarios. With less than 50 nodes, reliablity values were always better than or equal to 90% . But with higher node densities, this drops to around 80%. Enhanced DP had the worst reliability results of the three, reaching as low as 60% of the network with 30 nodes in a 50% node failure scenario. But, on average, its' delivery ratio reached

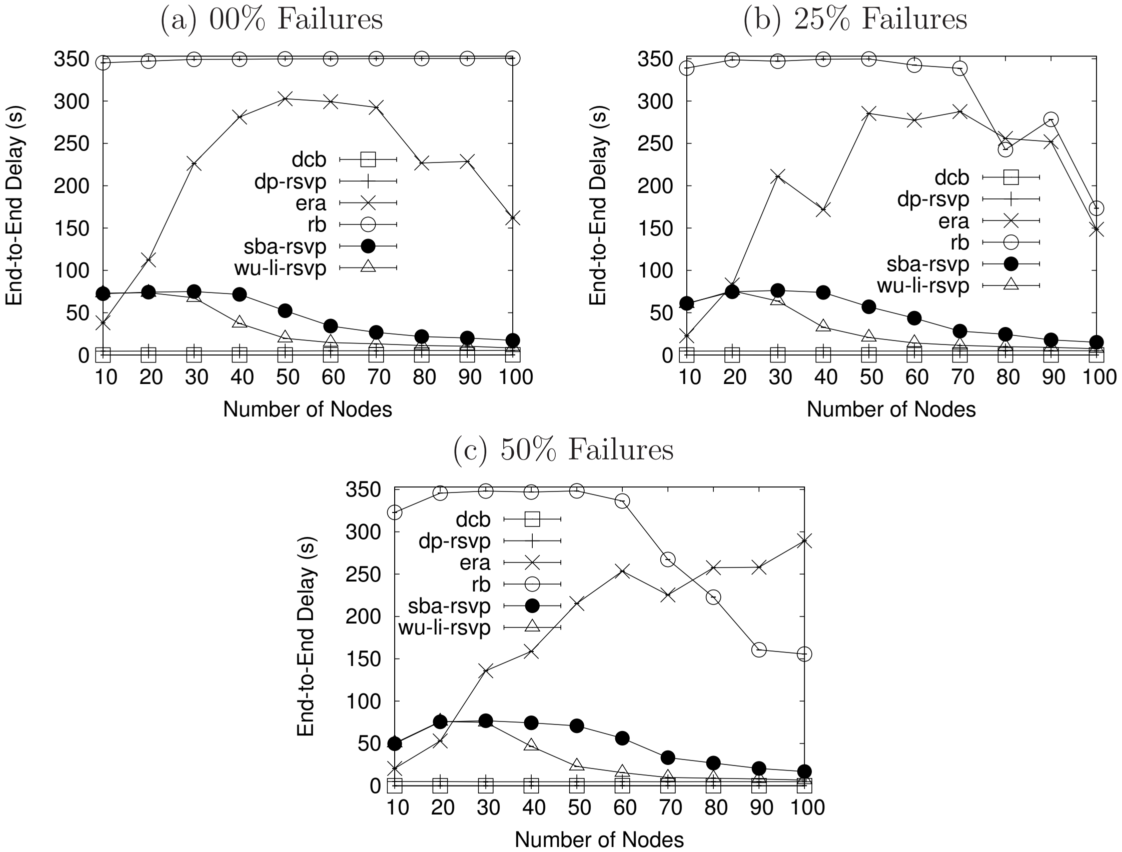


Figure 6.12. End-to-End Simulation Results With Failures of All Enhanced vs. Reliable Protocols

70% of the nodes of the network. However, in light of the forwarding ratio and end-to-end delays results previously discusses, enhanced DP has a stunningly good overall behavior.

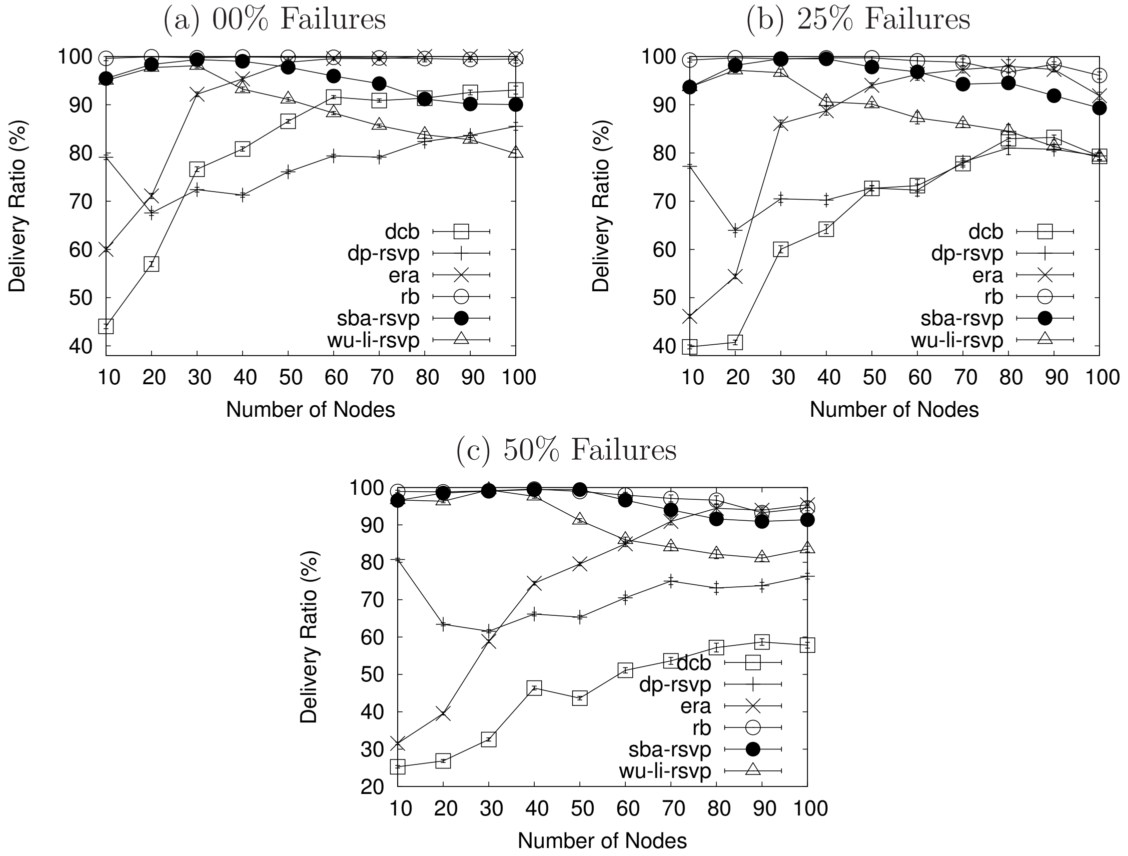


Figure 6.13. Reliability Simulation Results With Failures of All Enhanced vs. Reliable Protocols

6.5 CONCLUSION AND SUMMARY

In this chapter a simple reliability metric mechanism called *R.S.V.P* was proposed by which any node is capable of identifying more reliable links prior to transmission. It is a distributed solution that extends deterministic broadcasting algorithms to help enhance a nodes' capability of dealing with omission faults in a scalable scenario. In order to evaluate the performance and behavior of *R.S.V.P*, enhanced versions of unreliable protocols were simulated and compared against reliable protocols. In all those protocols *R.S.V.P* was used by incorporating the supplied reliability values into the forwarding node set selection process that already exist in each individual algorithm. It is interesting to note that when the protocols relied on the *R.S.V.P* mechanism, the reliability ratios were significantly raised, allowing for over a 96% network coverage at the expense of message re-transmissions and a higher end-to-end delay. The performance comparison with existing reliable broadcasting protocols indicate how efficient R.S.V.P is. Results show that the unreliable protocols enhanced with $R.S.V.P$ exhibited end-to-end delays which are still reasonable when compared against the end-to-end results of the reliable broadcasting protocols. Based on the simulations results here presented, it can be concluded that deterministic broadcasting protocols will greatly benefit from the use of this mechanism since correct message reception by neighboring nodes can be verified, instead of just assumed.

CHAPTER 7

SUMMARY AND CONCLUSIONS

Dissertations are not finished; they are abandoned.
—FRED BROOKS

This work evaluates how well a significant number of broadcasting protocols for *MANET*s behave when under a realistic scenario of momentary failures and topology changes, which is represented by an omission fault model. The study conducted here show that many protocols are highly impacted by node failures and are not capable of maintaining high delivery rate. Some even exhibit coverage levels that are unreasonable to expect from broadcasting protocols when placed in a real world scenario. Reliable protocols – which use additional mechanisms to ensure higher delivery rates beyond best-effort guarantees – have been proposed, but while they reach higher delivery ratios, they exhibit unacceptably high end-to-end delays. As a result of the study conducted, a new mechanism that helps to enhance the reliability of deterministic broadcasting protocols was proposed. The mechanism allows for scalability, and is capable of ensuring good delivery rates (in spite of failures) while maintaining lower end-to-end delays. Simulation results demonstrate the efficacy of the mechanism.

This chapter summarizes the contributions made throughout this work. Future research areas are also described.

7.1 SUMMARY

In Chapter 2 some of the main characteristics of *MANET*s were briefly outlined, as was shown the fundamental problems that surround wireless communications. This chapter also defined the system model that is assumed in the rest of this work. Some observations on the simulation environment were also made. In particular, the default values were exposed. The metrics that were evaluated during all simulation runs were described as well.

Broadcasting is a fundamental building block for dealing with routing and reaching consensus [SCS03, LK00, WD05, WC02, VE03, MGL04], and this was the focus of Chapter 3, where the broadcast problem in *MANET*s was introduced. Existing approaches were categorized and explained. Five protocols – Blind Flooding, Dynamic Probabilistic Approach [ZA05], Wu and Li's Protocol [WL99], Scalable Broadcast Algorithm [PL00] and Dominant Pruning [LK01] – where simulated under a scenario where nodes do not fail, but where they are mobile, thus inducing a network topology which is highly dynamic and frequently changing. As each of these protocols were described, known limitations were also disclosed. Unfortunately, simulation results showed that that complete network coverage is never obtained, and delivery ratios never even reach 80% of the network.

Chapter 4 then focused on strategies that attempt to raise the delivery ratio of protocols in order to ensure sufficient message coverage while broadcasting in mobile ad-hoc networks. Initially, the reliable broadcasting problem was described and existing strategies to solve this problem were presented. Although a large number of broadcasting protocols exist for *MANET*s, it was shown that many are not acceptable for usage in a real-world scenario. The ones that were acceptable were classified and further described. Finally, three reliable broadcasting protocols were chosen to be simulated, and results were shown and analyzed. The protocols were Double Covered Broadcasting [LW07], Reliable Broadcast [AVC95] and EraMobile [OGA06].

In Chapter 5 broadcasting protocols were introduced to a new omission fault model where temporary node failures occur. This model helped extend simulation scenarios to represent not only transient link availability with node mobility, but also external environmental interferences that are evidenced to affect message propagation, network topology changes and temporary partitions. Such a model was required in order to stress protocols with a yet more realistic simulation scenario. Unreliable broadcasting protocols reached 80% delivery ratios when at most 10% of the nodes failed, but barely reached 60% when a worst-case 50% scenario occurred. Reliable broadcasting protocols had superior results, but such reliability has its cost: maxed out forwarding ratio results and unacceptable end-to-end delays. Based on these simulation results, the source of existing broadcasting problems were investigated and a list of lessons learned were presented as a step towards enhancing the capability of broadcasting algorithms to deal with omission faults in scalable scenarios.

Such poor performance results indicated a need for additional mechanisms to guarantee message delivery throughout the network. This is why in Chapter 6 a reliability metric mechanism called *R.S.V.P* was proposed by which any node is capable of identifying more reliable links prior to transmission. It is a distributed solution that extends deterministic broadcasting algorithms to help enhance a nodes' capability of dealing with omission faults in a scalable scenario. Unreliable deterministic broadcasting protocols can use *R.S.V.P* by incorporating the supplied reliability values into the forwarding node set selection process that already exist in the broadcasting algorithm. In order to evaluate the performance and behavior of *R.S.V.P*, enhanced versions of unreliable protocols were simulated and compared against reliable protocols. It is interesting to note that when the protocols relied on the *R.S.V.P* mechanism, the reliability ratios were significantly raised, allowing for over a 96% network coverage at the expense of message re-transmissions and a higher end-to-end delay. Based on the simulation results presented in this chapter, it can be concluded that deterministic broadcasting protocols will greatly benefit from the use of this mechanism since correct message reception by neighboring nodes can be verified, instead of just assumed.

7.2 CONCLUSIONS

The credibility of a broadcasting protocol lies on whether or not it is capable of delivering messages to nodes throughout the network. Broadcasting protocols are expected to guarantee message dissemination in spite of all difficulties regarding *MANET*s. Simulation results, however, evidence that not all protocols are capable of assured message coverage, which is somewhat unsatisfactory. Although previous studies show that the broadcast protocols are very mobile resilient and support well congestion and collisions, the study conducted here show that these protocols are not fault tolerant when omission failures are taken into account. They are not capable of maintaining high delivery rates when placed in a real world scenario. However, when unreliable broadcasting protocols used the *R.S.V.P* mechanism, message delivery ratios were significantly increased at the expense of message re-transmissions and a higher end-to-end delay. Nevertheless, simulation results even show that these enhanced protocols had better overall performance than many reliable protocols.

7.3 FUTURE WORK

The current definition of our fault model takes into consideration the importance of randomly choosing faulty nodes throughout the network and representing failures characterized by interference introduced by the environment, link instability and transmission failure due to node movement. Nevertheless, it still represents a rather peculiar failure model where a certain number of nodes fail (and then start working correctly) at exactly the same time. Despite this fact, the performance results presented throughout this work are valid since they still model a scenario much more realistic than the simpler fail-stop model. Nevertheless, as a future work we can envision an even more realistic fault model.

In order to truly stress broadcasting protocols, a real ad-hoc network would be needed. Unfortunately, this was not accessible to us during the timespan of this research. Test-bed implementations would surely help uncover a series of unexpected issues that are far from being discovered with current state-of-the-art simulation work. Also, it has been shown that a protocol as simple as Blind Flooding could get qualitatively different results if implemented on different simulators [CSS02], but unfortunately we did not have enough time to do this. Furthermore, a large number of mobility models have been proposed – each one modeling very specific scenarios – and to correctly understand the behavior of broadcasting protocols in *MANET*s would require that any one protocol be applied to a wide range of different scenarios.

Finally, the simulation results from Dominant Pruning enhanced with *R.S.V.P* demonstrate that end-to-end delays can be kept at a minimal. Further research in this area might lead to solutions that would allow the usage of *R.S.V.P* in applications that have timing constraints.

BIBLIOGRAPHY

[AASS00] Sulabh Agarwal, Ashish Ahuja, Jatinder Pal Singh, and Rajeev Shorey. Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks. In *ICC (3)*, pages 1697–1701, 2000.

[ASD$^+$06] Muneeb Ali, Umar Saif, Adam Dunkels, Thiemo Voigt, Kay Romer, Koen Langendoen, Joseph Polastre, and Zartash Afzal Uzmi. Medium access control issues in sensor networks. *SIGCOMM Comput. Commun. Rev.*, 36(2):33–36, 2006.

[Ass99a] IEEE Standards Association. Ieee 802.11, 1999 edition: Information technology – telecommunications and information exchange between systems – local and metropolitan area network – specific requirements – part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. 1999.

[Ass99b] IEEE Standards Association. Ieee 802.11b-1999 supplement to 802.11-1999: Wireless lan mac and phy specifications: Higher speed physical layer (phy) extension in the 2.4 ghz band. 1999.

[Ass03a] IEEE Standards Association. Ieee 802.11g-2003: Information technology – telecommunications and information exchange between systems – local and metropolitan area networks – specific requirements – part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications – amendment 4: Further higher-speed physical layer extension in the 2.4 ghz band. 2003.

[Ass03b] IEEE Standards Association. Ieee 802.15 part 15.3: Wireless medium access control (mac) and physical layer (phy) specifications for high rate wireless personal area networks (wpan). 2003.

[Ass03c] IEEE Standards Association. Ieee 802.15 part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans). 2003.

[Ass05] IEEE Standards Association. Ieee 802.15 part 15.1: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpans). 2005.

[AVC95] S. Alagar, S. Venkatesan, and J. Cleveland. Reliable broadcast in mobile wireless networks. In *Proc. of Military Communications Conference (MILCOM'95)*, pages 236–240, November 1995.

[BBF$^+$01] Béatrice Bérard, Michel Bidoit, Alain Finkel, François Laroussinie, Antoine Petit, Laure Petrucci, and Philippe Schnoebelen. *Systems and Software Verification. Model-Checking Techniques and Tools.* Springer, 2001.

[BCB99] Stefano Basagni, Imrich Chlamtac, and Danilo Bruschi. A mobility-transparent deterministic broadcast mechanism for ad hoc networks. *IEEE/ACM Trans. Netw.*, 7(6):799–807, 1999.

[BGH$^+$07] Sven Burmester, Holger Giese, Stefan Henkler, Martin Hirsch, Matthias Tichy, Alfonso Gambuzza, Eckehard Munch, and Henner Vocking. Tool support for developing advanced mechatronic systems: Integrating the fujaba real-time tool suite with camel-view. In *ICSE '07: Proceedings of the 29th International Conference on Software Engineering*, pages 801–804, Washington, DC, USA, 2007. IEEE Computer Society.

[BKP03] C. Basile, M. Killijian, and D. Powell. A survey of dependability issues in mobile wireless networks. Technical report, LAAS CNRS Toulouse, France, February 2003.

[BKS03] Vartika Bhandari, Nupur Kothari, and Dheeraj Sanghi. A reliable on-demand routing paradigm for mobile ad hoc networks. 2003.

[BMJ$^+$98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[Bri01] L. Briesemeister. *Group Membership and Communication in Highly Mobile Ad Hoc Networks.* PhD thesis, School of Electrical Engineering and Computer Science, Technical University of Berlin, Germany, November 2001.

[BT02] Jeffrey Q. Bao and Lang Tong. A performance comparison between ad hoc and centrally controlled cdma wireless lans. *IEEE Transactions on Wireless Communications*, 1(4):829–841, 2002.

[BV05] Vartika Bhandari and Nitin H. Vaidya. Implementing a reliable local broadcast primitive in wireless ad hoc networks. Technical report, Dept. of Computer Science, Dept. of Electrical and Computer Eng. and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, July 2005.

[CDG$^+$05] Gregory Chockler, Murat Demirbas, Seth Gilbert, Nancy A. Lynch, Calvin C. Newport, and Tina Nolte. Reconciling the theory and practice of (un)reliable wireless broadcast. In *ICDCS Workshops*, pages 42–48. IEEE Computer Society, 2005.

[CDGN05] Gregory Chockler, Murat Demirbas, Seth Gilbert, and Calvin Newport. A middleware framework for robust applications in wireless ad hoc networks. *Allerton Conference 2005: Forty-Third Annual Allerton Conference on Communication, Control, and Computing*, September 2005.

[CDK94] George Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed Systems: Concepts and Design.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[CH05] Sungsoon Cho and John P. Hayes. Impact of mobility on connection in ad hoc networks. In *Proceeding of the IEEE Wireless Communications and Networking Conference (WCNC 2005).*, volume 3, pages 1650–1656, 2005.

[CJWK02] Kwan-Wu Chin, John Judge, Aidan Williams, and Roger Kermode. Implementation experience with manet routing protocols. *ACM SIGCOMM Computer Comm. Review*, 32(5), 2002.

[CRB01] R. Chandra, V. Ramasubramanian, and K. Birman. Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks. In *ICDCS '01: Proceedings of the The 21st International Conference on Distributed Computing Systems*, pages 275–283, Washington, DC, USA, 2001. IEEE Computer Society.

[CSM03] Song Yean Cho, Jin Hyun Sin, and Byung In Mun. Reliable broadcast scheme initiated by receiver in ad hoc networks. In *Proceedings of the 28th Annual IEEE Conference on Local Computer Networks (LCN 2003)*, pages 281–282. IEEE Computer Society, October 2003.

[CSS02] David Cavin, Yoav Sasson, and André Schiper. On the accuracy of manet simulators. In *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 38–43, New York, NY, USA, 2002. ACM Press.

[CWKS97] Brian P. Crow, Indra Widjaja, Jeong Geun Kim, and Prescott T. Sakai. Ieee 802.11 wireless local area networks. *Communications Magazine, IEEE*, 35(9):116–126, 1997.

[CZI03] Apichet Chayabejara, Salahuddin Muhammad Salim Zabir, and Norio Shiratori Research Institute. An enhancement of ieee 802.11 for efficient operations in manet. In *PDCS 2003: Proceedings of The IASTED Conference on Parallel and Distributed Computing and Systems.* IASTED: International Association of Science and Technology for Development, 2003.

[DGH+87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *PODC '87: Proceedings of the sixth* *annual ACM Symposium on Principles of distributed computing*, pages 1–12, New York, NY, USA, 1987. ACM Press.

[dOCGS06] Talmai Brandão de Oliveira, Victor Franco Costa, Fabíola Greve, and Leizer Schnitman. Evaluating the impact of faults on broadcasting protocols for manets. In *VII Workshop on Fault-Tolerant Computing, with (SBRC) Symp. on Computer Networks*, pages 49–60, Curitiba, Brazil, May 2006.

[DRWT97] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks. *IEEE Personal Communication*, 1997.

[DW03] Fei Dai and Jie Wu. Distributed dominant pruning in ad hoc networks. In *Proceedings of the IEEE 2003 International Conference on Communications (ICC 2003)*, volume 1, pages 353–357, 2003.

[DW04] Fei Dai and Jie Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-prunning. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):1027–1040, 2004.

[dW07] Christian de Waal. A mobility scenario generation and analysis tool. www.informatik.uni-bonn.de/IV/BonnMotion/, June 2007.

[EE01] Jeremy Elson and Deborah Estrin. Time synchronization for wireless sensor networks. In *Proc. of the 2001 International Parallel and Dist. Processing Symposium (IPDPS), Workshop on Parallel and Dist. Computing Issues in Wireless Networks and Mobile Computing*, volume 1, pages 1965–1970, 2001.

[Eph02] A. Ephremides. Energy concerns in wireless networks. *IEEE Wireless Communications*, 9(4):48–59, 2002.

[ER02] Jeremy Elson and Kay Romer. Wireless sensor networks: A new regime for time synchronization. Technical report, UCLA, July 2002.

[FA02] Adam Fulford and Andrew Alleyne. An embedded mechatronic system for wireless servo control. In *Proceeding of the 2002 American Control Conference*, volume 2, pages 1658–1659. IEEE Computer Society, 2002.

[FLP85] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382, April 1985.

[Fri46] H.T. Friis. A note on a simple transmission formula. In *Proceedings of the IRE*, volume 34, pages 254–256, 1946.

[GBvR02] Indranil Gupta, Kenneth Birman, and Robbert van Renesse. Fighting fire with fire: using randomized gossip to combat stochastic scalability limits. *Special Issue of Quality and Reliability of Computer Network Systems, Journal of Quality and Reliability Engineering International*, 18(3):165–184, 2002.

[GC04] Gregor Gaertner and Vinny Cahill. Understanding link quality in 802.11 mobile ad hoc networks. *IEEE Internet Computing*, 8(1):55–60, 2004.

[GdWFM02] Michael Gerharz, Christian de Waal, Matthias Frank, and Peter Martini. Link stability in mobile wireless ad hoc networks. In *LCN '02: Proc. of the 27th Annual IEEE Conf. on Local Computer Networks*, pages 30–42, Washington, DC, USA, 2002. IEEE Computer Society.

[GK96] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. In *European Symposium on Algorithms*, pages 179–193, 1996.

[GKW$^+$02] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical Report CSD-TR 02-0013, UCLA, February 2002.

[GSPS02] Thiagaraja Gopalsamy, Mukesh Singhal, D. Panda, and P. Sadayappan. A reliable multicast algorithm for mobile ad hoc networks. *icdcs*, 00:563, 2002.

[GW02] A. J. Goldsmith and S. B. Wicker. Design challenges for energy-constrained ad hoc wireless networks. *IEEE Wireless Communications Magazine*, 9(4):8–27, August 2002.

[HDY05] Hossam Hassanein, Hongyan Du, and Chihsiang Yeh. Robust route establishment in high mobility manets. In *1st International Computer Engineering Conference New Technologies for the Information Society*, December 2005.

[HJR04] Q. Huang, C. Julien, and G. Roman. Relying on safe distance to achieve strong partionable group membership in ad hoc networks. *IEEE Transactions on Mobile Computing*, 3(2):192–205, April-June 2004.

[HT94] Vassos Hadzilacos and Sam Toueg. A modular approach to fault-tolerant broadcasts and related problems. Technical Report TR94-1425, 1994.

[HTS07] Chih-Shun Hsu, Yu-Chee Tseng, and Jang-Ping Sheu. An efficient reliable broadcasting protocol for wireless mobile ad hoc networks. *Elsevier's Ad Hoc Networks Journal*, 5(3):299–312, 2007. Article in Press.

[ICP00] M. Impett, M. S. Corson, and V. Park. A receiver-oriented approach to reliable broadcast in ad hoc networks. In *Proc. of Wireless Comm. and Networking Conf. (WCNC)*, volume 1, pages 117–122, September 2000.

[JL02] Ming-Yu Jiang and Wanjiun Liao. Family ack tree (fat): a new reliable multicast protocol for mobile ad hoc networks. In *IEEE International Conference onCommunications*, volume 5, pages 3393–3397, 2002.

[JM04] Alpár Jüttner and Ádám Magi. Tree based broadcast in ad hoc networks. *MONET Special Issue on WLAN Optimization at the MAC and Network Levels*, 2004.

[KA05] Paul A. Kawka and Andrew G. Alleyne. Stability and feedback control of wireless networked systems. In *Proceeding of the 2005 American Control Conference*, volume 4, pages 2953–2959. IEEE Computer Society, 2005.

[Kin03] Patrick Kinney. Zigbee technology: Wireless control that simply works. www.zigbee.org/resources, October 2003. Whitepaper.

[KMG03] Anne-Marie Kermarrec, Laurent Massouli, and Ayalvadi J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. Parallel Distrib. Syst.*, 14(3):248–258, 2003.

[KNE03] David Kotz, Calvin Newport, and Chip Elliott. The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dept. of Computer Science, Dartmouth College, July 2003.

[KNG+04] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *Proc. of the 7th ACM Int. Symp. on Modeling, Snalysis and Simulation of Wireless and Mobile Systems (MSWiM '04)*, pages 78–82, New York, NY, USA, 2004. ACM Press.

[KSD06] C. Rama Krishna, Chakrabarti Saswat, and Datta Debasish. A modified backoff algorithm for ieee 802.11 dcf-based mac protocol in a mobile ad hoc network. In *TENCON 2004: IEEE Region 10 Conference*, volume 2, pages 664–667. IEEE Computer Society, 2006.

[KT03] Zhenqiang Ye Srikanth V. Krishnamurthy and Satish K. Tripathi. A framework for reliable routing in mobile ad hoc networks. volume 1, pages 270–280, 2003.

[Kun03] Thomas Kunz. Multicasting in mobile ad-hoc networks: achieving high packet delivery ratios. In *Proc. of the 2003 conference of the Centre for Advanced Studies on Collaborative research (CASCON'03)*, pages 156–170. IBM Press, 2003.

[LBC04] Justin Lipman, Paul Boustead, and Joe Chicharo. Reliable optimised flooding in ad hoc networks. In *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, volume 2, pages 521–524, June 2004.

[LEH03] Jun Luo, Patrick Th. Eugster, and Jean-Pierre Hubaux. Route driven gossip: Probabilistic reliable multicast in ad hoc networks. In *Proc. of INFOCOM 2003*, 2003.

[LEH04] J. Luo, P. Eugster, and J. Hubaux. Pilot: Probabilistic lightweight group communication system for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 3(2):164–179, April-June 2004.

[LH03] Ben Liang and Zygmunt J. Haas. Predictive distance-based mobility management for multidimensional pcs networks. *IEEE/ACM Trans. Netw.*, 11(5):718–732, 2003.

[LK00] Hyojun Lim and Chongkwon Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proc. of the 3rd ACM Int. Workshop on Modeling, Analysis and Simul. of Wireless And Mob. Sys. (MSWIM '00)*, pages 61–68. ACM Press, 2000.

[LK01] Hyojun Lim and Chongkwon Kim. Flooding in wireless ad hoc networks. *Computer Comm.*, 24(3–4):353–363, 2001.

[LMAH04] J. Lee, K. Morioka, N. Ando, and H. Hashimoto. Cooperation of distributed intelligent sensors in intelligent environment. *IEEE/ASME Transactions on Mechatronics*, 9(3):535–543, 2004.

[LMK07] Yang Liu, Milosz Mazurkiewicz, and Marek Kwitek. A study towards reliability- and delay-critical wireless communication for robocup robotic soccer application. In *Proceeding of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, pages 633–636, Shanghai, China, 2007.

[LSL$^+$02] Geunhwi Lim, Kwangwook Shin, Seunghak Lee, H. Yoon, and Joong Soo Ma. Link stability and route lifetime in ad-hoc wireless networks. In *Proc. of the 2002 Int. Conf. on Parallel Processing Workshops (ICPPW '02)*, page 116, Washington, DC, USA, 2002. IEEE Computer Society.

[LW02] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Trans. on Mobile Computing*, 1(2):111–123, 2002.

[LW04] Wei Lou and Jie Wu. Double-covered broadcast (dcb): A simple reliable broadcast algorithm in manets. In *23rd Joint Conference of the IEEE Computer and Comm. Soc. (INFOCOMM)*, volume 3, pages 2084–2095, 2004.

[LW07] Wei Lou and Jie Wu. Toward broadcast reliability in mobile ad hoc networks with double coverage. *IEEE Trans. on Mobile Computing*, 6(2):148–163, 2007.

[MCS$^+$06] Mansoor Mohsin, David Cavin, Yoav Sasson, Ravi Prakash, and André Schiper. Reliable broadcast in wireless mobile ad hoc networks. In *Proc. of the 39th Hawaii Int. Conf. on Syst. Science (HICSS '06)*. IEEE Computer Society, 2006.

[MGL04] P. Mohapatra, C. Gui, and J. Li. Group communications in mobile ad hoc networks. *IEEE Computer*, 37(2):52–59, February 2004.

[MW05] Patricia Mcdermott-Wells. What is bluetooth? *Potentials, IEEE*, 23(5):33–35, 2005.

[NS-07] NS-2. The network simulator, http://www.isi.edu/nsnam/ns/, June 2007.

[NTCS99] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. 5th ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pages 151–162. ACM Press, 1999.

[OGA06] Öznur Özkasap, Zülküf Gen, and Emre Atsan. Epidemic-based approaches for reliable multicast in mobile ad hoc networks. *SIGOPS Oper. Syst. Rev.*, 40(3):73–79, 2006.

[OVT01] Katia Obraczka, Kumar Viswanath, and Gene Tsudik. Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks*, 7(6):627–634, 2001.

[PBBvR06] Stefan Pleisch, Mahesh Balakrishnan, Ken Birman, and Robbert van Renesse. Mistral:: efficient flooding in mobile ad-hoc networks. In *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, pages 1–12, New York, NY, USA, 2006. ACM Press.

[PL00] Wei Peng and Xi-Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proc. 1st ACM international symp. on Mobile ad hoc networking & computing (Mobihoc)*, pages 129–130, Piscataway, NJ, USA, 2000. IEEE Press.

[PM04] Asad Amir Pirzada and Chris McDonald. Establishing trust in pure ad-hoc networks. In *Proc. of the 27th Australasian Conf. on Comp. Sci. (ACSC '04)*, pages 47–54, Darlinghurst, Australia, 2004. Australian Computer Society, Inc.

[PP05] Seungjin Park and Roopesh R. Palasdeokar. Reliable one-hop broadcasting (rob) in mobile ad hoc networks. In *PE-WASUN '05: Proceedings* *of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 234–237, New York, NY, USA, 2005. ACM Press.

[PR97] E. Pagani and G. P. Rossi. Reliable broadcast in mobile multihop packet networks. In *MobiCom '97: Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 34–42. ACM Press, 1997.

[PR99] Elena Pagani and Gian Paolo Rossi. Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks. *Mob. Netw. Appl.*, 4(3):175–192, 1999.

[Rap01] Theodore Rappaport. *Wireless Communications: Principles and Practice, 2nd Edition.* Prentice-Hall, Upper Saddle River, New Jersey, 2001.

[RCS05] S. Ray, J.B. Carruthers, and D. Starobinski. Evaluation of the masked node problem in ad hoc wireless lans. *IEEE Trans. on Mobile Computing*, 4(5):430–442, 2005.

[Rit03] Michael W. Ritter. The future of wlan. *Queue*, 1(3):18–27, 2003.

[Rze03] George Rzevski. On conceptual design of intelligent mechatronic systems. *Mechatronics. The Science of Intelligent Machines. An International Journal*, 13(10):1029–1044, 2003.

[SAL$^+$03] John A. Stankovic, Tarek Abdelzaher, Chenyang Lu, Lui Sha, and Jennifer Hou. Real-time communication and coordination in embedded sensor networks. In *Proceedings of the IEEE, 91(7)*, 2003.

[Sat01] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, pages 10–17, August 2001.

[SCS03] Y. Sasson, D. Cavin, and A. Schier. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. *IEEE Wireless Communications and Networking*, 2:1124–1130, March 2003.

[SHSM06] Fred Stann, John Heidemann, Rajesh Shroff, and Muhammad Zaki Murtaza. Rbp: robust broadcast propagation in wireless networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 85–98, New York, NY, USA, 2006. ACM Press.

[SNGC05] Kulpreet Singh, Andronikos Nedos, Gregor Gaertner, and Siobhan Clarke. Message stability and reliable broadcasts in mobile ad-hoc networks. In *Proc. of ADHOC-NOW*, volume 3788 of *Lecture Notes in Comp. Sci.*, pages 297–310, 2005.

[TK75] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii – the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on Communications*, COM-23(12):1417–1433, 1975.

[TNS03] Yu-Chee Tseng, Sze-Yao Ni, and En-Yu Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Transactions on Computers*, 52(5):545–557, 2003.

[Toh97] Chai-Keong Toh. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communications*, 4(2):103–139, 1997.

[TXZ04] Jian Tang, Guoliang Xue, and Weiyi Zhang. Reliable routing in mobile ad hoc networks based on mobility prediction. *IEEE Int. Conf. on Mob. Ad Hoc and Sensor Syst.*, 32(5), October 2004.

[VB00] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[VE03] E. Vollset and P. Ezhilchelvan. An efficient reliable broadcast protocol for mobile ad-hoc networks. Technical Report CS-TR: 822, School of Comp. Sci., Univ. of Newcastle, December 2003.

[VE04] Einar Vollset and Paul Ezhilchelvan. Scribble: an efficient reliable manycast protocol for ad-hoc networks. *IEEE Int. Conf. on Mobile Ad Hoc and Sensor Syst.*, pages 561–563, October 2004.

[VE05] Einar W. Vollset and Paul D. Ezhilchelvan. Design and performance-study of crash-tolerant protocols for broadcasting and reaching consensus in manets. In *24th IEEE Symposium on Reliable Distributed Systems (SRDS 2005)*, pages 166–175, 2005.

[WC02] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proc. of the 3rd ACM Int. Symp. on Mob. Ad Hoc Networking & Computing*, pages 194–205. ACM Press, 2002.

[WCH06] Xudong Wang, Sunghyun Choi, and Jean-Pierre Hubaux. Guest editorial - wireless mesh networking: Theories, protocols, and systems. *IEEE Wireless Communications*, 13(2):8–9, April 2006.

[WCK02] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. Psfq: a reliable transport protocol for wireless sensor networks. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 1–11, New York, NY, USA, 2002. ACM Press.

[WD04] Jie Wu and Fei Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Trans. Computers*, 53(10):1343–1354, 2004.

[WD05] Jie Wu and Fei Dai. Efficient broadcasting with guaranteed coverage in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(2):259–270, 2005.

[WL99] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *DIALM '99: Proc. of the 3rd Int. Workshop on Discrete algorithms and Methods for Mobile Computing and Comm.*, pages 7–14, New York, NY, USA, 1999. ACM Press.

[WYY06] Jiawei Wang, Hongnian Yu, and Suiran Yu. Investigation of a mobile manipulator over wired/wireless control system. Technical report, 2006.

[YHE04] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.

[ZA05] Qi Zhang and Dharma P. Agrawal. Dynamic probabilistic broadcasting in manets. *Journal of Parallel and Distributed Computing*, 65(2):220–233, 2005.

[ZL00] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283. ACM Press, 2000.

[ZL04] Jianliang Zheng and Myung J. Lee. Will ieee 802.15.4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard. *Communications Magazine, IEEE*, 42(6):140–146, 2004.