A Generalized Broadcasting Technique for Mobile Ad Hoc Networks

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Abdelmajid Khelil

aus Moknine, Tunesien

Hauptberichter:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Mitberichter:	Prof. Dr.-Ing. Dr. h. c. Peter Göhner
Tag der mündlichen Prüfung:	12.04.2007

Institut für Parallele und Verteilte Systeme (IPVS)
der Universität Stuttgart
2007

Acknowledgements

I would like to take this opportunity to thank those who guided and supported me during this process. Without their contributions, this research would not have been possible.

Firstly, I would like to thank my PhD advisor, Prof. Kurt Rothermel. He taught me how to undertake this research during the past years. Besides this, he was always patient and helpful whenever his guidance and assistance were needed. Prof. Rothermel also contributed much of his valuable time to help me for improving my writing skills. Furthermore, he gave advice on how to communicate and cooperate with other researchers as a professional. Therefore, I would like to show my deep appreciation to him.

I am also grateful to Prof. Peter Göhner for serving on my PhD committee and spending time reviewing this work and giving valuable suggestions and comments on my work. I would like to thank Dr. Becker and Dr. Marrón for their valuable advice on my research work. I also appreciate valuable discussions with Jörg Hähner, Illya Stepanov, Jing Tian and other faculty members and students. Their suggestions and comments greatly encouraged me during these years. Also special thanks to all persons that reviewed this thesis or parts of it.

I would like to show my great appreciation to my parents for their continuous support during all these years. Last, but not least, my wife, Sirine, also contributed much of her time and efforts to support me during my studies.

This work was supported by the Land Baden-Württemberg. The support is gratefully acknowledged.

Abdelmajid Khelil,
Flörsheim am Main, June 16, 2007.

Abstract

Broadcasting is a major communication primitive required by many applications and protocols in Mobile Ad Hoc Networks (MANETs). It is frequently deployed for content distribution, service discovery or advertisement, and sensor data dissemination. Broadcast protocols are also a fundamental building block to realize principal middleware functionalities such as replication, group management and consensus. Broadcasting in MANETs has therefore been an active area of research recently.

Most of the research conducted on broadcasting in MANETs has primarily focused only on carefully selected application and evaluation scenarios. Consequently, the developed broadcasting schemes do not yield good performance for other scenarios. Different comparative studies show that the existing broadcasting techniques are tailored to only one class of MANETs with respect to node density and node mobility, and are unfortunately not likely to operate well in other classes.

Node spatial distribution is a key issue for the performance of broadcast protocols, since it determines the connectivity of the MANET. Our survey of potential MANET application scenarios shows a wide range of possible node spatial distributions and node mobilities. This leads to that a MANET generally shows a continuously changing network connectivity over space and time. Therefore, a generalized solution for broadcasting that accounts for the requirements of the various applications and adapts to the heterogeneous and evolving node spatial distribution and mobility is a major contribution.

In this thesis, we present *hypergossiping*, a novel generalized broadcasting technique for MANETs. Hypergossiping integrates *two* adaptive schemes and efficiently switches between them depending on local node density.

The first scheme is *adaptive gossiping*, which distributes messages within connected parts of the MANET. We adapted gossiping as follows. First, we established an analytical model for gossiping through adopting the SI mathematical model from the epidemiology. Then, we used the model to adapt the gossiping forwarding probability to local node density. As a result, we provide a simple analytical expression that nodes use to set the appropriate forwarding probability depending on the current number of neighbors. Simulation results showed that adaptive gossiping efficiently propagates messages within a network partition independent of the node spatial distribution and node mobility in that network

partition.

The second scheme is a *broadcast repetition* method, which detects partition joins using an efficient and localized heuristic and efficiently repeats the needed broadcasts upon detection of a partition join. Our approach is mobility-assisted since it exploits the mobility of nodes to efficiently deliver messages in frequently partitioned scenarios. We defined mobility metrics that simplify the design of mobility-assisted concepts, and used some of them to design a mobility-aware buffering strategy, which can significantly reduce the buffer overhead of hypergossiping.

Simulation results in the standard network simulator ns-2 show that hypergossiping outperforms all existing strategies. Hypergossiping significantly increases the delivery ratio for a broad range of MANETs with respect to node density, node mobility and network load while providing high efficiency and scalability.

Kurzfassung (German)

Eine verallgemeinerte Broadcasting-Methode für mobile Ad-Hoc-Netze

Aufgrund des schnellen Fortschritts in der drahtlosen Kommunikation nimmt die Zahl der mobilen mit einer Funkschnittstelle ausgestatteten Geräte unaufhörlich zu. Viele vorhandene Funktechnologien wie WLAN stellen neben dem infrastrukturbasierten Kommunikationsmodus einen Ad-Hoc-Kommunika-tionsmodus zur Verfügung. Der Ad-Hoc-Modus ermöglicht es den mobilen Geräten, auf eine spontane Weise direkt miteinander zu kommunizieren, wenn sie sich in Reichweite zueinander befinden.

Ein *Mobiles Ad-Hoc-Netzwerk (MANET)* ist ein Netzwerk, das durch autonome, mobile und mit einer Funkschnittstelle ausgestattete Knoten gebildet wird. Knoten sind beispielsweise Laptops, PDAs, Mobiltelefone oder Sensorknoten. Die Knoten sind üblicherweise mobil und werden entweder von Menschen oder Tieren getragen oder in Fahrzeugen wie Autos, Flugzeuge oder Fahrräder mitgeführt. Ursprünglich in der militärischen Forschung entwickelt, gewinnen MANETs in anderen Bereichen wie Handel, Akademie und Notdiensten an Bedeutung. MANETs eignen sich besonders dort, wo wenig oder gar keine Kommunikationsinfrastruktur exisitiert, oder wenn die vorhandene Infrastruktur teuer oder ungünstig zu verwenden ist. Die potentiellen Anwendungen für MANETs sind die spontane Vernetzung von Einsatzgeräten bei Militär- oder Rettungseinsätzen, die Fahrzeug-zu-Fahrzeug-Kommunikation, die computergestützte Gruppenarbeit sowie die Überwachung und Messung von schwer zugänglichen Systemen und Umgebungen.

Broadcast ist eine wichtige Kommunikationsprimitive, die es ermöglichen soll, Nachrichten von einem dedizierten Knoten, der Quelle, auf alle anderen Knoten in einem Netzwerk zu verteilen. Broadcast ist essentiell für die Realisierung zahlreicher Anwendungen und Protokolle in MANETs. Broadcast ist ein grundsätzlicher

Baustein, um Middleware Funktionalitäten wie Replikation [1], Gruppenkommunikation [2] und Konsenus [3] zu realisieren. Außerdem wird Broadcast dazu verwendet, Ressourcen zu finden und bereitzustellen. Ein Beispiel hierfür ist das Finden von Routen mit reaktiven Routing-Protokollen, wie z.B. DSR [4] und AODV [5]. Broadcast wird häufig auch eingesetzt, um Informationen wie beispielsweise Warnungen oder Meldungen auf alle Netzwerk-Knoten zu verteilen. In hoch dynamischen MANET-Szenarien, kann Broadcast als eine robuste Methode dienen, andere Kommunikationsprimitiven wie Multicast [6] zu realisieren.

Aufgrund möglicher Knotenmobilität weisen MANETs eine sehr stark variierende räumliche Knotenverteilung und folglich eine ständig variierende Konnektivität über Zeit und Raum auf. Die Vielfalt der möglichen Anwendungsszenarien für MANETs erhöhen diese Dynamik, und weiten ausserdem die Bandbreite der möglichen Netzwerkgrößen und folglich die Raumverteilung der Knoten aus. Vorherige Arbeiten wie z.B. [7] haben gezeigt, dass die räumliche Knotenverteilung einen klaren Einfluss auf die Performanz von MANET-Broadcast-Protokollen hat.

Deshalb ist eine verallgemeinerte, adaptive Broadcast-Methode erforderlich, die mit den ständig variierenden MANET-Eigenschaften und Anwendungsanforderungen ohne explizite Vorkonfiguration funktioniert. Leider gibt es eine solche Lösung in der Literatur nicht. Ziel dieser Arbeit ist, eine verallgeimeinerte, adaptive Broadcasting-Methode zu entwickeln, welche sich an die Mobilität und räumliche Verteilung der Knoten einerseits und an die Anwendungsanforderungen andererseits anpasst.

Diese Dissertationsschrift ist in Englisch verfasst und besteht aus sieben Kapiteln. In *Kapitel 1* wird eine Motivation für die Arbeit gegeben. *Kapitel 2* gibt einen ausführlichen Überblick über MANET Anwendungen und Kommunikationstechnologien, die die Bildung von MANETs ermöglichen. Danach werden die Herausforderungen und die Anforderungen an den Entwurf von Broadcast-Protokollen in MANETs im Detail diskutiert. Schließlich definieren wir unser Systemmodell. *Kapitel 3* klassifiziert die verwandten Arbeiten und beschreibt den Stand der Forschung zu Broadcast-Protokollen in MANETs. In *Kapitel 4* zeigen wir, wie man mathematische Modelle der Epidemiologie anpasst, um Broadcast-Protokolle in MANETs analytisch zu modellieren. Anschließend diskutieren wir die Anwendungsmöglichkeiten des entwickelten Modells. Eine wichtige Anwendung ist die Adaptation von Broadcast-Protokollen. *Kapitel 5* führt unsere verallgemeinerte Broadcasting-Methode ein, die für eine breite Reihe von Knoten-Raumverteilungen, Knoten-Geschwindigkeiten und Netzwerklasten effizient eingesetzt werden kann. In *Kapitel 6* definieren wir eine neuartige Klasse von Mobilitätsmetriken, um Mobilität auf einer großen Zeitskala zu quantifizieren. Dann zeigen wir, wie wir diese Metriken verwenden, um den Puffer-Overhead von Hypergossiping zu reduzieren. In *Kapitel 7* fassen wir unsere Beiträge zusammen und skizzieren einige Richtungen für zukünftige Forschungsarbeiten. In den folgenden Abschnitten dieser Kurzfassung geben wir eine Zusammenfassung der Kapitel 3 bis 7.

Stand der Forschung

In *Kapitel 3* klassifizieren wir die vorhandenen Broadcast-Protokolle in MANETs. Wir untersuchen die Stärken und Schwächen existierender Verfahren und zeigen den Bedarf einer verallgemeinerten Broadcast-Lösung.

Unsere Kriterien für die Klassifikation der existierenden Broadcast-Lösungen sind die MANET-Eigenschaften einerseits und die Anwendungsanforderungen andererseits. Aus dem Broadcasting-Blickwinkel unterscheiden wir zwischen *partitionierten* und *nicht-partitionierten* MANETs sowie zwischen Anwendungen, die gegenüber der Latenz empfindlich sind (*latenzempfindlich*) und Anwendungen, die höhere Latenzen im Bereich von Minuten oder sogar Stunden oder Tagen tolerieren (*latenztolerant*). Durch Permutation sind theoretisch *vier Klassen* von Broadcast-Lösungen möglich, die wir im folgenden genauer diskutieren.

Broadcast Lösungen für latenzempfindliche Anwendungen in partitionierten Szenarien sind nur mit einer niedrigen Nachrichten-Zustellungsrate (auch *Erreichbarkeit* genannt) möglich. Da Broadcasting per Definition eine hohe Erreichbarkeit erfordert, ist eine Lösung hier praktisch nicht möglich.

Unseres Wissens gibt es keine Broadcast-Methode, die sich die Latenztoleranz einer Anwendung in nicht-partitionierten Szenarien zu Nutze macht, obwohl eine solche Strategie die Kapazität des Netzes verbessern könnte [8]. Die Größenordnung der Verbesserung der Netzkapazität kommt auf Kosten der unvorhersehbaren Länge der Latenz. Das Fehlen einer Lösung kann durch die unvorhersagbare Bewegung der Knoten erklärt werden.

In der Literatur lassen sich zwei Hauptklassen von Broadcast-Lösungen identifizieren: Die erste Protokollklasse wurde für latenzempfindliche Anwendungen in nicht-partitionierten MANETs, und die zweite für latenztolerante Anwendungen in hoch partitionierten Szenarien entworfen. Zudem wurde ein integrierter Ansatz entwickelt, der Strategien aus diesen beiden Klassen kombiniert. Im folgenden besprechen wir kurz die beiden Klassen sowie den integrierten Ansatz.

Die erste Klasse verwandter Literatur setzt sich aus einer großen Zahl von Broadcast-Protokollen zusammen, die auf dem Flooding-Verfahren beruhen. Bei dem Flooding-Protokoll leitet ein Knoten eine Broadcast-Nachricht nach einer sehr kurzen Verzögerungszeit (im Bereich von wenigen Millisekunden) und unter einer festgelegten Bedingung zu allen seinen Nachbarknoten mittels MAC-Broadcast weiter. Der flooding-basierte Broadcast ist augenblicklich, wobei sich die Nachrichten im Raum in kürze verbreiten. Deshalb bezeichnen wir diese Protokollklasse mit *Broadcast-in-space.* Der grundsätzliche Nachteil der vorhandenen Broadcast-in-space-Verfahren ist, dass sie nur in nicht-partitionierten MANETs hohe Erreichbarkeit aufweisen können, und dass sie daher für partitionierte Szenarien nicht geeignet sind. Vergleichende Studien zeigten auch, dass diese Verfahren für spezifische (nicht-partitionierte) Szenarien optimiert sind und andere Szenarien nicht unterstützen. So eignen sich z.B. manche Protokolle für niedrige Mobilität, und andere für Szenarien mit einer niedrigen Netzlast. Deshalb wurden einigen Arbeiten gemacht, um einige der vorhandenen Verfahren an die lokale Knotendichte oder Mobilität anzupassen. Diese Anpassung vergrößert zwar die Anzahl der unterstützten Szenarien. Jedoch bleiben die adaptiven Broadcast-in-space-Verfahren für partitionierte Szenarien nicht geeignet.

Die zweite Klasse verwandter Arbeiten umfasst Protokolle, die die Broadcast-Nachrichten puffern, um sie mittels Knotenmobilität und auf Begegnungen mit anderen Knoten weiterleiten. Diese Lösungen basieren auf das allgemeine Store-and-forward-Prinzip. Da die Nachrichten Übertragungszeiten im Bereich von Minuten oder sogar Stunden benötigen können (je nach Mobilitätssmuster der Knoten) um die Zielknoten zu erreichen, nennen wir dieses Broadcast-Paradigma *Broadcast-in-time*. Anwendungen, die diese Protokolle benutzen, müssen höhere Latenzen tolerieren. Broadcast-in-time Protokolle können in hoch partitionierten Szenarien eine hohe Erreichbarkeit aufweisen. Leider sind die Broadcast-in-time Ansätze nur für hochpartitionierte MANETs optimiert und werden in nicht-partitionierten Szenarien von Broadcast-in-space-Verfahren wegen des hohen Nachrichten-Overheads übertroffen.

Das *integrierte Flooding (IF)* [9, 10] ist der erste Ansatz mit dem Ziel eine Lösung sowohl für partitionierte als auch für nicht-partitionierte MANETs zu sein. Das IF-Verfahren stellt Protokolle bereit, welche die Erreichbarkeit in partitionierten Teilen des MANETs vergrößern, und die Broadcast-Stürme in nicht-partitionierten Teilen eindämmen sollen. Die IF-Lösung integriert zwei Broadcast-in-space-Protokolle, Scoped-Flooding und Plain-Flooding [6, 9, 10], und einen Broadcast-in-time-Protokoll, Hyperflooding [6, 9, 10]. Ein Knoten schaltet zwischen diesen drei Methoden in Abhängigkeit von seiner aktuellen relativen Geschwindigkeit zu allen seinen Nachbarknoten um.

Die IF-Lösung erhöht zwar die Broadcast-Erreichbarkeit sowohl in partitionierten als auch in nicht-partitionierten Szenarien, zeigt aber die folgenden drei Nachteile. Erstens setzt Hyperflooding eine sehr einfache Broadcast-Wiederholungsstrategie ein, d.h. zu jedem neuen entdeckten Nachbar, werden alle gepufferten Nachrichten wiederholt übertragen. Wenn die Nachrichten-Pufferungszeit hoch ist, führt diese Strategie zu einer großen Zahl nutzloser und teuerer Broadcast-Wiederholungen in hoch mobilen Szenarien. Wenn die Pufferungszeit niedrig ist, zeigt die IF-Strategie eine niedrige Zustellungsrate in wenig-mobilen und partitionierten Szenarien. Zweitens, Scoped-Flooding ist nicht an die Knotendichte angepasst, was IF weniger effizient macht als adaptive broadcast-in-space-Verfahren in Szenarien mit einer heterogenen und dynamischen Raumverteilung der Knoten. Drittens benötigen die Knoten für den IF-Ansatz Geschwindigkeitsinformationen (Wert und Richtung), was wiederum eine starke Beschränkung der Anwendbarkeit vom IF-Ansatz darstellt.

Zusammenfassend eignen sich alle Broadcast-Protokolle nur für spezielle MANET Szenarien. Es gibt jedoch keine verallgemeinerte, effiziente Broadcast-Methode, die den vollständigen gewünschten Designraum abdeckt, d.h für latenzempfindliche als auch latenztolerante Anwendungen und für beliebige Raumverteilungen und Mobilitäten geeignet ist. In dieser Arbeit präsentieren wir eine effiziente Broadcasting Methode, die ein breites Spektrum von Raumverteilungen und Mobilitäten der Knoten unterstützt. Unsere Methode ist für latenzempfindliche als auch für latenztolerante Anwendungen geeignet.

Analytische Modellierung und Adaptation von Broadcast-Protokollen

Es gibt nur wenige theoretische Betrachtungen von MANETs im allgemeinen und von Broadcasting im speziellen. Dies kann darauf zurückgeführt werden, dass es bereits reife, flexible und standardisierte Simulatoren [11–14] gibt, während es an einer allgemeinen Plattform für theoretische Untersuchungen mangelt. Ebenso erschwert die volatile Natur von MANETs analytische Untersuchungen. Die vorhandenen analytischen Unterschungen für Broadcasting wurden für idealisierte Netzwerke, z.B. ideale MAC-Schicht und statische Knoten [15–17] oder für eine kleine Anzahl von Knoten [18] durchgeführt. In *Kapitel 4* präsentieren wir eine generische Methodologie, welche die analytische Untersuchung der Broadcast-Protokolle in MANETs vereinfacht.

Ein Broadcast-Protokoll versucht so schnell wie möglich alle Knoten zu erreichen. Da die Verbreitung der Broadcast-Nachrichten in MANETs anschaulich einer Epidemie, also der Verbreitung einer ansteckenden Krankheit ähnelt, beruht unserer Ansatz auf die Verwendung von etablierten, mathematischen Arbeiten der Epidemiologie. Auf diese Weise können reife Modelle aus der Epidemiologie adaptiert werden, um mathematische Modelle für Broadcast-Protokolle zu entwickeln. Diese Modelle können Aussagen über den Ausbreitungsverlauf der Nachrichten liefern, was für die Performanzanalyse und Protokolladaptation eine wichtige Rolle spielt. Obwohl sich die Ziele der Broadcast-Protokolle und der Epidemiologie unterscheiden, zeigen wir, wie die Resultate der theoretischen Untersuchung von Epidemien für das Verständnis und die Analyse von Broadcasting hilfreich sein können. Wir zeigen auch, wie man epidemische Modelle für die Adaptation von Broadcast-Protokollen in MANETs einsetzen kann.

Um die Ausbreitung einer Krankheit zu modellieren, ist es wichtig, den Verlauf der Infektion innerhalb eines Individuums und den Ansteckungsvorgang zwischen Individuen zu verstehen. In Bezug auf eine gegebene Krankheit wird ein Individuum zu einem der folgenden Verhaltenszustände assoziiert: Susceptible (S), Infectious (I), Removed (R), Exposed (E), oder iMmune (M). Abhängig von der Krankheit dürfen einige dieser Zustände nicht betrachtet werden. Mehrere mathematische Modelle wurden in der Epidemiologie (siehe z.B. [19, 20]) basierend auf den Fluss-Mustern zwischen den obengenannten Verhaltenszuständen definiert. Der Name des Modells beschreibt gleichzeitig Fluss-Muster. SI, SEI, SEIS, SIR, SIRS, SEIR, SEIRS, MSEIR, MSEIRS sind Beispiele für epidemische Modelle.

Das einfachste epidemische Modell ist das *SI-Modell*, wo ein Individuum einmal angesteckt (Verhaltenszustand S), bleibt für immer ansteckend (Verhaltenszustand I). Eine Infektion ist nicht heilbar und führt zu immer weiteren Ansteckungen. Nachdem ein Individuum kontaminiert wurde beginnt es damit andere Individuen anzustecken. Bei diesem Modell kann jedes Individuum infiziert werden. Auf lange Sicht werden auch alle Individuen infiziert.

Der Unterschied zwischen den verschiedenen ansteckenden Krankheiten besteht hauptsächlich darin, wie sich die Infektion auf ein Inviduum auswirkt und sie an andere Individuen übertragen wird. Analog besteht der Unterschied zwischen verschiedenen Broadcast-Protokollen darin, wie die empfangene Nachricht intern verarbeitet und weitergeschickt wird. Daher lassen sich einige epidemische Modelle auf bestimmte Broadcast-Protokolle übertragen. Ausschlaggebend ist das Verhalten der Knoten nach dem Empfang einer Broadcast-Nachricht, d.h., ob die Knoten 'ansteckend' bleiben oder damit aufhören, andere Knoten 'anzustecken'.

In dieser Arbeit betrachten wir zuerst das SPIN-basierte Broadcast-Protokoll [21]. Einmal durch eine Nachricht 'angesteckt' bleibt ein Knoten in diesem Protokoll solange 'ansteckend' bis die Nachricht alle anderen Knoten des MANETs erreicht hat. Entsprechend modellieren wir das SPIN-basierte Broadcast-Protokoll durch das SI-Modell [22].

Das SI-Modell basiert auf einem wichtigen Parameter, der *Ansteckungsrate* (engl. infection rate). Die Ansteckungsrate gibt die Anzahl der Neuinfektionen pro Zeiteinheit an, die von einem infizierten Knoten verursacht werden. Im Allgemeinen ist es möglich die Ansteckungsrate aus dem Mobilitätsmodell und den Broadcast-Eigenschaften analytisch zu berechnen. Jedoch haben wir diesen Ansatz nicht vertieft. Stattdessen verfolgen wir eine Strategie der Epidemiologen, die häufig ihre Modelle durch Daten kalibrieren, die aus einer realen Ausbreitung der Epidemie gewonnen werden. Wir kalibrieren unser SI-Modell mittels Simulationsdaten. Das Ergebnis der Kalibrierung ist die Bestimmung der Ansteckungsrate des betrachteten Protokolls in einem festgelegten MANET Konfiguration. Unsere Vogehensweise ist generisch und kann analog für Broadcast-Protokolle und ihre entsprechenden epidemischen Modelle eingesetzt werden.

Als ein weiteres Broadcast-Protokoll, betrachten wir das *Gossiping*-Protokoll. Gossiping ist ein Broadcast-in-space-Verfahren. Es wird auch probabilistisches Flooding genannt. Ein Knoten leitet eine Nachricht mit einer konstanten Wahrscheinlichkeit zu allen seinen Nachbarknoten weiter. Die Gossiping-Weiterleitungswahrscheinlichkeit spielt eine entscheidende Rolle für die Performanz von Gossiping. Sogenannte Broadcast-Stürme [7] (Sammelbegriff für Kollision, Kontention und Redundanz) entstehen, wenn zu viele Knoten die Gossiping-Nachricht weiterleiten. Allerdings hört der Broadcast auf, bevor die Nachricht alle Knoten erreicht hat, wenn zu wenig Knoten die Nachricht weiterleiten. Für die Bestimmung der optimalen Weiterleitungswahrscheinlichkeit spielt die Knotendichte eine wichtige Rolle.

Wir modellieren Gossiping durch das SI-Modell, da die Knoten für die kurze Zeit 'ansteckend' bleiben, in der Gossiping stattfindet. Analog zum SPIN-basierten Protokoll kalibrieren wir das SI-Modell für Gossiping durch Simulationsdaten. Wir zeigen nun, wie das epidemische SI-Modell verwendet werden kann, um Gossiping anzupassen.

Das Ziel der Adaptation von Gossiping ist den Knoten die geeignete Weiterleitungswahrscheinlichkeit zuzuordnen. Hierfür variieren wir die Weiterleitungswahrscheinlichkeit und die Knotendichte des MANETs und bestimmen für jede Kombination die entsprechende Ansteckungsrate von Gossiping. Für die Simulationen haben wir uniform verteilte Knoten generiert. Zur Bestimmung der besten Gossiping-Weiterleitungswahrscheinlichkeit für eine gegebene Knotendichte, soll die Wahrscheinlichkeit berechnet werden, für welche die Ansteckungsrate maximal ist. Hierdurch konnten wir für ein großes Spektrum von Knotendichten die entsprechenden, geeigneten Gossiping-Weiterleitungswahrscheinlichkeit bestimmen.

Da MANETs in der Regel eine dynamische und nicht-uniforme Knotendichte aufweisen, und da die räumliche Knotenverteilung (globaler Zustand) im MANET schwer zu erfassen ist, lassen wir die Knoten autonom entscheiden, welche Wahrscheinlichkeit sie für das Gossiping benutzen. Außerdem lassen wir die Knoten ihre lokale Knotendichte durch die Anzahl ihrer Nachbarknoten abschätzen. Die Anzahl der Nachbarknoten n kann sehr einfach von den Knoten über den Versand von 'HELLO'-Nachrichten ermittelt werden. Das Ergebnis unserer Anpassung ist die folgende Gossiping-Wahrscheinlichkeitsfunktion: $p = min(1.0\ ,\ 0.175 + \frac{6.05}{n})$. Beim adaptiven Gossiping leitet ein Knoten eine Nachricht mit einer Wahrscheinlichkeit weiter, die durch die obige Funktion und die aktuelle Anzahl der Nachbarknoten bestimmt wird.

Für die Evaluierung vom adaptiven Gossiping, haben wir intensive Simulationen im Netzwerk-Simulator ns-2 [11] durchgeführt. Die betrachteten Szenarien zeigen ein großes Spektrum räumlicher Knotenverteilungen. Die Simulationsergebnisse zeigen, dass das adaptive Gossiping robust bezüglich Schwankungen in der Knotendichte ist. Die Performanz vom adaptiven Gossiping ist vergleichbar mit der Performanz der besten veröffentlichten, adaptiven Broadcast-in-space-Protokolle, d.h. mit ACB (Adaptive-Counter-Based) [23] und STOCH-FLOOD (Stochastic-Flooding) [24]. Es wurde die Vergleichbarkeit hinsichtlich Erreichbarkeit, Latenz und Effizienz betrachtet. Diese Ergebnisse bestätigen die Anwendbarkeit epidemischer Modelle zur Modellierung und Adaptation von Broadcast-Protokollen in MANETs.

Unsere Vorgehensweise für die Verwendung des SI-Modells zur Adaptation von Gossiping ist allgemein. Die Adaptation weiterer Protokolle können leicht nach dem gleichen Prinzip erreicht werden. In der Biologie wird vorwiegend untersucht, wie Epidemien möglichst rasch einzudämmen sind. Diese Fragestellung lässt sich auch am Kontext von MANETs übertragen. Mit unserer Methodolgie kann nämlich die Verbreitung von Würmern und Viren in MANETs modelliert werden, um anschliessend effiziente und robuste Strategien für die Eindämmung ihrer Verbreitung zu entwickeln.

Hypergossiping: Unsere verallgemeinerte Broadcasting-Methode

Ohne Beschränkung der Allgemeinheit können wir ein MANET als eine Menge von *Netzwerkpartitionen* betrachten, die im Laufe der Zeit *fusionieren* oder *sich aufspalten*. Das *Adaptive Gossiping* verteilt die Broadcast Nachrichten innerhalb einzelner Partitionen effizient. Leider erreichen die Gossiping-Nachrichten nur Knoten innerhalb der Partition, in der die Quelle liegt. Deshalb haben wir eine neue Methode entworfen, welche die effiziente Wiederholung von Gossiping ermöglicht. Unser Ansatz besteht aus einer effizienten Heuristik zur Erkennung der Partitionsfusionierungen und aus einem Protokoll zur Wiederholung des Gossipings benötigter Nachrichten. Wir nennen diese Methode *Broadcast-Wiederholung*. Das Ergebnis der Kombination vom adaptiven Gossiping und der Broadcast Wiederholung ist eine neuartige, adaptive, verallgemeinerte Broadcast-Methode, die wir *Hypergossiping* [25] [26] nennen. Hypergossiping verwendet also zwei Komponenten: Das adaptive Gossiping zum effizienten Broadcast-in-space und die Broadcast-Wiederholung zum effizienten Broadcast-in-time.

Die Broadcast-Wiederholung startet nur, wenn das adaptive Gossiping einzelne Knoten nicht erreicht. Die Broadcast-in-time Komponente soll auf eine bestimmte Zeit begrenzt werden, die von der Anwendung toleriert und spezifiziert werden soll. Ein guter Einsatz für die Festlegung dieser Dauer ist die Berücksichtigung der Zeitgültigkeit der Daten. Diese Idee beruht auf der Tatsache, dass Daten im allgemeinen eine bestimmte Zeitgültigkeit haben [27]. Daten über Parkplatzverfügbarkeit können z.B. ein paar Minuten oder Stunden gültig bleiben. Sensordaten aber haben meist eine Gültigkeit im Bereich von einigen Sekunden, da z.B. die Updates in Kurzen Zeitabständen geschickt werden. In dieser Arbeit benutzen wir das Feld *Lebensdauer (engl. lifetime)* im Header der Broadcast-Nachrichten um die von der Anwendung tolerierte Latenzzeit zu speichern.

Eine Nachricht, die aufgrund der Netzwerkpartitionierung nicht zugestellt werden konnte, soll bei der Fusionierung von Partitionen weitergeleitet werden. Deswegen werden die Broadcast-Nachrichten bei Knoten in einem lokalen Puffer gespeichert.

Wir sagen zwei Knoten begegnen sich, wenn sie in die Kommunikationsreichweite voneinander eintreten. Die Begegnung zweier Knoten, die zweier unterschiedlichen Partitionen bis kurz vor dem Begegnen gehört haben, ist die Partitionsfusionierung der beiden Partitionen. Diese Begegnung soll daher von beiden Knoten erkannt werden, falls fehlende Pakete auf beiden Seiten ausgetauscht werden sollen. Zur lokalisierten Erkennung der Partitionsfusionierungen haben wir eine Heuristik entwickelt, die den Kern der Broadcast-Wiederholung darstellt.

Die Überprüfung, ob ein neuer Nachbarknoten in einer anderen Partition war, geschieht durch den Vergleich der Nachrichten, die sie zuletzt empfangen haben. Hierfür werden die IDs der zuletzt empfangenen Broadcast-Nachrichten in einer Liste verwaltet, die wir als *LBR-Liste* ('Last Broadcast Received') bezeichnen.

Die Heuristik für die Erkennung von Partitionsfusionierungen ist wie folgt definiert: Sobald ein Knoten einen neuen Nachbarknoten entdeckt, hängt er an seine nächste 'HELLO'-Nachricht seine LBR-Liste an. Wenn ein Knoten eine LBR-Liste empfängt, bildet er die Schnittmenge aus der empfangenen Liste und seiner LBR-Liste. Falls die Kardinalität der Schnittmenge kleiner als ein bestimmter Schwellwert ist, geht der Knoten von einer Partitionsfusionierung aus. Wir haben unsere Heuristik mittels Simulationen kalibriert, indem wir den Schwellwert bestimmt haben. Nur wenn eine Partitionsfusionierung erkannt wird, wird die Broadcast-Wiederholung wie folgt fortgesetzt.

Unmittelbar nachdem ein Knoten eine Partitionsfusionierung erkannt hat, sendet er an allen seinen Nachbarknoten die Liste der IDs aller Broadcast-Nachrichten, die er bis jetzt empfangen hat und deren Lifetime-Wert größer als 0 ist. Wir bezeichnen diese Liste als *BR-Liste* ('Broadcast Received'). Der Empfänger bildet aus der empfangenen BR-Liste und seiner eigenen BR-Liste eine Differenz-Liste. Diese Differenz-Liste enthält die IDs, die in der eigenen BR-Liste enthalten sind aber nicht in der empfangenen. Nachrichten mit diesen IDs sollen übertragen werden. Der Knoten plant alle diese Nachrichten nach einer Zufallszeit zu übertragen. Falls zwischenzeitlich keine Broadcast-Nachricht empfangen wird, deren ID sich in der Differenz-Liste befindet, werden in zufälligen Abständen alle Nachrichten mit einer ID aus der Differenz-Liste nacheinander an allen Nachbarknoten gesendet.

Wir haben intensive Simulationen für Hypergossiping in ns-2 durchgeführt. Dabei haben wir die Erreichbarkeit, die Latenz und den Nachrichten-Overhead von Hypergossiping für eine grosse Reihe von Knoten-Raumverteilungen, Knoten-Mobilitäten und Netzlasten gemessen. Diese Messungen zeigen, dass Hypergossiping für eine breite Reihe von MANETs in Bezug auf die Knotendichte, Knotenmobilität und Netzlast sehr gut geeignet ist. Hypergossiping vergrößert die Zustellungsrate auf effiziente Weise in partitionierten Szenarien. Der Vergleich der Performanz von Hypergossiping und integriertes Flooding (IF) [9, 10] zeigt die Effizienz und Überlegenheit unserer Broadcast-Wiederholungsstrategie und ihrer gegenüber der von integriertes Flooding. Wie schon bei der Beschreibung der verwandten Arbeiten erwähnt wurde, weist IF zwar eine hohe Erreichbarkeit auf aber zeigt einen nicht akzeptablen Nachrichten-Overhead insbesondere in hoch mobilen Szenarien.

Obwohl Netzwerkpartitionierung einen großen Einfluss auf die Performanz von MANET-Protokollen hat, stellt der weit verbreitete Netzwerk-Simulator ns-2 keine Werkzeuge für eine leichte Evaluierung von MANET-Protokollen hinsichtlich der Netzwerkpartitionierung zur Verfügung. Deshalb haben wir ein Framework für ns-2 entwickelt, der benötigte Informationen über Netzwerkpartitionierung zur Simulationsszeit auf einfache Weise bereitstellt [28]. Protokollentwickler können diese Information verwenden, um ihre Protokolle in Bezug auf die Netzwerkpartitionierung zu bewerten und ihre Performanz mit dem optimalen Fall zu vergleichen. Wir haben das entwickelte Framework verwendetet, um die Optimalität und Robustheit von Hypergossiping bezüglich der Netzwerkpartitionierung zu bestimmen. Dies führte zu einem besseren Verständnis der Performanz des Protokolls und zur Lokalisierung von Verbesserungspotentialen.

Mobilitätsbewusste Pufferung für Broadcasting

Mobilität spielt eine wichtige Rolle in MANETs. Obwohl sie einerseits Netzwerkfunktionalitäten wie Routing belastet, aber sie hilft andererseits, die Netzkapazität zu vergrößern [8] und Netzwerkpartitionierung zu überwinden. Um aus Knotenmobilität Nutzen zu ziehen, entsteht eine neue Klasse von MANET-Protokollen, in der die Nachrichten-Zustellung mobilitätgestützt ist. Die Anwendungen, die diese Protokolle benutzen, sollen latenztolerant sein, d.h. höhere Latenzen tolerieren. Ein Beispiel dieser Protokollklasse ist Hypergossiping. Für mobilitätgestützte Netzwerke spielt die Mobilität auf einer großen Zeitskala eine Schlüsselrolle. Bisher wird die Mobilität in MANETs nur auf einer kurzen Zeitskala untersucht. Deshalb präsentieren wir neue Mobilitätsmetriken, welche die Mobilität auf einer großen Zeitskala quantifizieren [29]. Unsere Metriken basieren auf paarweisen Begegnungen zwischen mobilen Knoten. Um einen leichten Zugang zu diesen Metriken im Netzwerk-Simulator ns-2 zu ermöglichen, stellen wir ein Framework für ns-2 Benutzer zur Verfügung.

Die Broadcast-in-time-Komponente von Hypergossiping setzt Knotenmobilität voraus, um die Erreichbarkeit in hochpartitionierten Netzwerken erfolgreich zu erhöhen. Außerdem puffern die Knoten Nachrichten und transportieren sie physisch für eine Broadcast-Wiederholung wenn sie Knoten begegnen, die diese Nachrichten noch nicht empfangen haben. Mobilität ist daher ein Schlüsselfaktor für das Design der Broadcast-in-time-Komponente von Hypergossiping im allgemeinen und der Pufferungsstrategie im speziellen.

Die einfachste Pufferungsstrategie ist, wenn alle Knoten alle empfangenen Nachrichten puffern, so lange ihre Lebensdauer noch nicht abgelaufen ist. Diese Strategie verursacht offensichtlich einen hohen Puffer-Overhead, was den Einsatz von Hypergossiping in Szenarien mit hohem Broadcastverkehr oder hohen Werten für die Daten-Lebensdauer auf Geräten mit einem sehr kleinen Pufferspeicher erschwert. Daher ist die Reduzierng des Puffer-Overheads eine wichtige Verbesserung der Effizienz von Hypergossiping und eine Vergrößerung des Spektrums seiner Einsatzszenarien.

In dieser Arbeit präsentieren wir eine lokalisierte und effiziente Methode und skizzieren weitere Ideen für zukünftige Arbeiten, um den Puffer-Overhead von Hypergossiping zu reduzieren [30]. In unserem Ansatz schätzen die Knoten die

Nützlichkeit, eine empfangene Nachricht zu puffern, bevor sie entscheiden, diese Nachricht tatsächlich zu puffern. Je niedriger die geschätzte Nützlichkeit für eine Nachricht, desto weniger wahrscheinlich soll die Entscheidung sein, die Nachricht zu puffern.

Aus obigen Betrachtungen schliessen wir, dass der Wert der Nützlichkeit abhängig von dem Mobilitätsmuster des entsprechenden Knotens festgelegt werden soll. Die Nützlichkeit kann in Abhängigkeit von weiteren Faktoren ausgerechnet werden, wie z.B. Lebensdauer der Nachrichten. Allerdings beschränken wir uns hier auf den Aspekt der Mobilität und stellen eine lokalisierte Methode vor, um die Nützlichkeit in Abhängigkeit von Mobilitätsmustern zu schätzen.

Unser erster Schritt für die Berechnung der Nützlichkeit der Pufferung in Abhängigkeit von der Mobilität ist die Identifikation der wichtigsten Mobilitätsmuster. Intuitiv sollten die Knoten, welche die Partition öfter wechseln, mehr Nachrichten und auch für längere Zeit puffern. Es ist auch klar, dass es überflüssig ist, alle Knoten die sich als Gruppe bewegen, alle Nachrichten puffern zu lassen. In dieser Arbeit fokussieren wir auf die Bestimmung von Knoten, die sich als Gruppe bewegen. Die Gruppenbewegung wird durch eine lokalisierte Methode erkannt, die auf der Historie der letzten Begegnungen mit anderen Knoten beruht.

Simulationen in ns-2 zeigen, dass unsere lokalisierte Heuristik zur Entdeckung von Gruppenmobilität eine zufriedenstellende Genauigkeit aufweist. Deshalb zeigen die Simulationsergebnisse, dass unsere Mobilitätsbewusste Pufferung zu einer wichtigen Reduktion des Pufferspeicherbedarfs von Hypergossiping führen kann, je nachdem wie stark die Ausprägung der Gruppenmobilität ist.

Zusammenfassung und Ausblick

Broadcast ist essentiell für MANETs. Allerdings stößt der Broadcast-Design auf kritische Probleme insbesondere Broadcast-Stürme und häufige Netzwerkpartitionierung. Das Ziel dieser Arbeit ist, eine verallgemeinerte Broadcast-Methode zu entwickeln, die an die Anwendungsanforderungen und MANET-Ausprägungen angepasst ist. Wir haben eine solche Lösung entwickelt, d.h. Hypergossiping, indem wir zwei Strategien kombiniert haben: Adaptives Gossiping zum Verhindern der Broadcast-Stürmen und eine effiziente Broadcast-Wiederholungsstrategie zum Überwinden von Netzwerkpartitionierungen. Wir haben die relevanten Parameter dieser verallgemeinerten Lösung zur Knotendichte und Knotenmobilität angepasst. Unsere Strategie übertrifft alle existierende Protokolle für ein großes Spektrum von MANET-Einsatzbedingungen von sehr dicht- zu sehr dünnbesiedelten, von statischen zu hoch mobilen und von sehr kleinen zu sehr grossen Szenarien. Der Hypergossiping-Protokoll ist bedürfnislos, weil er ressourcenschonend ist und keine spezielle Knotenfähigkeiten benötigt.

In Kapitel 7 skizzieren wir einige Richtungen für zukünftige Forschungsarbeiten.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The number of mobile devices equipped with wireless network interfaces is continuously increasing due to the fast and rapid progress in wireless communication. Many existing wireless technologies such as WLAN and Bluetooth provide an ad hoc communication mode in addition to an infrastructure-based communication mode. The ad hoc mode allows mobile devices to communicate directly and in a spontaneous manner, when they enter each other's transmission area.

A *Mobile Ad Hoc Network (MANET)* is a network formed on-the-fly by autonomous mobile nodes that are equipped with wireless short-range radios. Nodes are self-computing devices such as laptops, personal digital assistants (PDAs), cellular phones or even sensor nodes. Ad hoc networks are generally single channel and a node can communicate with all other nodes that are within its transmission area. Ad hoc networking is precious in scenarios where there is little or no communication infrastructure, or if the existing infrastructure is expensive or inconvenient to use. Nodes are generally mobile and usually carried by human beings, animals or moving objects such as vehicles, airplanes or bicycles. Originated in military research, ad hoc networking is gaining importance in applications in commerce, academia and emergency services. In the next chapter, we discuss in details the most popular ad hoc networking technologies, as well as the most popular MANET application scenarios, which include vehicle-to-vehicle communication, collaborative computing, emergency operations, military scenarios, and monitoring of hardly accessible systems and environments.

An ad hoc network implies that there are *no central facilities* or fixed infrastructure, such as base stations. Nodes function independently of each other, but must coordinate and cooperate among themselves to self-organize and self-manage the network, which shows a temporary and arbitrary network topology. Considering for example the basic network task routing, a given source node wants to send a message to a certain destination node. Since the transmission range is limited, the destination may be located out of the source's transmission area. In this case, other nodes have to relay messages in order to transport the message to the destination. The source sends the message to one or more of its neighbors, which in turn forward the message to some of their neighbors, and so on until the message finally reaches the destination. Due to the cooperation of nodes, multihop communication becomes possible in the MANET. Supposing that the message must not be delivered to the destination immediately and that the source is able to predict that the destination eventually *encounters* the source, i.e. enters into the transmission area of the source, it would be possible to discharge relay nodes and save many costly transmissions.

Network-wide *broadcasting* aims at distributing messages from the source node to all other nodes in the network. Broadcasting is a major communication primitive required by many applications and protocols in MANETs. Broadcast protocols are a fundamental building block to realize principal middleware functionalities such as replication [1], group management [2] and consensus [3]. Furthermore, broadcasting is frequently used to discover and advertise resources. A simple example of resource discovery is the route discovery in many reactive routing protocols [4] [5]. Broadcasting is also frequently used to distribute content to all network participants, such as alarm signals or announcements. In highly dynamic scenarios, broadcasting serves as a robust way of realizing other communication primitives, such as multicast [6].

Traditional broadcast protocols designed for wired networks mainly follow tree-based approaches (similar to IP-multicast). These protocols are not adequate for MANETs because of the distinct characteristics of MANETs, such as the scarcity of system resources and node mobility resulting in a rapidly changing topology and a partitionable nature of the network. Broadcasting in MANETs has therefore been an active area of research recently. We distinguish two major classes of work so far: Flooding-based approaches and negotiation-based approaches.

The emphasis in *flooding-based* approaches has been on providing efficient solutions, where the source *forwards* a broadcast message to all of its neighbors, which conditionally forward the message immediately to their neighbors and so on. Thus, all nodes, which have a multihop path to the source, will be reached eventually. Broadcast messages propagate in space during a very short time period. The message spread in space therefore dominates the spread in time. We refer to this broadcast paradigm as *broadcast-in-space*. The focus of research here is to reduce the number of forwarders and relieve the *broadcast storm* problem [7] while providing maximized broadcast delivery ratio. Noteworthy is that the broadcast-in-space techniques reach only nodes that have a multihop path to the source, i.e. they are in the same network partition.

Negotiation-based approaches concentrate on the frequently partitioned networks and increase the delivery reliability by continuously advertising and requesting relevant messages (handshake). The main spreading process is carried out by means of node mobility. The messages are exchanged on favorable encounters among nodes, i.e. encounters between nodes that have the message and those that still do not have it. Depending on the spatial distribution and mobility of nodes, messages are delivered in a time-scale of minutes, hours or even days. Unlike broadcast-in-space, for negotiation-based approaches the spread in time dominates the spread in space. Hence, we refer to this broadcast paradigm as *broadcast-in-time* . In connected MANETs, the handshake mechanism causes enormous message overhead, which makes broadcast-in-time approaches not appropriate for connected scenarios. Broadcast-in-time only makes sense, if the application tolerates adequate high delays. Such applications are referred as *delay-tolerant* applications.

The remainder of this chapter is organized as follows. Firstly, we state our goal and the main problems that we investigate in this thesis. Then, we present three sections that correspond with the three main chapters describing the work carried out during the thesis. Finally, the main contributions are briefly summarized and a short outline of the thesis is given.

1.1 Problem Statement and Goal

The various application scenarios for MANETs show that a future MANET scenario can exhibit a wide range of operating conditions. Referring by *node density* to the number of nodes per unit of area, a MANET can show a wide range of node densities, i.e. a wide range of node spatial distributions and number of nodes. Consequently, the network connectivity may highly vary *over space*, depending on the number of nodes forming the MANET, the ad hoc technology used and the spatial environment that affects the signal propagation. In particular, we may encounter highly partitioned parts as well as highly connected parts in a MANET. We define *Network partitioning* as the split of the network into two (or more) disjointed groups of nodes that can not communicate with each other. *Node mobility* plays a major role in broadening the MANET operating conditions, since it induces changes in node spatial distribution causing the MANET connectivity to evolve, i.e. to vary *over time.* As different nodes may move according to different patterns and speeds, the MANET connectivity can evolve in an arbitrary way. In particular, due to mobility a MANET node can experience a continuously changing local node density and network partitions may split or join.

The broadcast-in-space techniques are not suitable for partitioned networks. Furthermore, performance comparative studies for the broadcast-in-space techniques [31–34] have shown that most of them are tailored to one class of MANETs with respect to node density and node mobility and are unlikely to operate well in other classes. This is due to that these strategies show a poor adaptation to the varying intensity of broadcast storms. For this reason, the trend in broadcast research is to broaden the range of MANET operating conditions covered for the existing solutions. The most important efforts to adapt broadcast-in-space protocols to the MANET characteristics and hence support a broader range of operating conditions are [23, 24, 35]. These protocols dynamically tune their main parameters depending on the current node density [23, 24] or node mobility [23, 35]. These efforts can however only broaden the covered scenarios within the class of connected MANETs. If the MANET is partitioned, these optimizations are no longer sufficient to perform a high delivery ratio.

The pre-configuration of one of the existing broadcasting techniques on a node, limits the number of MANET scenarios where the node can broadcast successfully. For example, pre-installing a broadcast-in-time technique, makes the node broadcast successfully only in frequently partitioned MANET scenarios or the frequently partitioned parts of the same MANET, but not in dense scenarios or parts. Thus, in a realistic MANET scenario neither a broadcast-in-space scheme nor a broadcast-in-time scheme can ensure a high broadcast delivery ratio while providing efficiency for evolving operating conditions.

Therefore, a *generalized* broadcasting technique that deals with the heterogeneous and continuously changing MANET characteristics without an explicit pre-configuration is desired. We are convinced that such a generalized solution should combine at least one adaptive approach for broadcast-in-space and another one for broadcast-in-time. An integrated approach could maximize the supported operating conditions. The framework should install different solutions and provide an efficient mechanism to switch between them, depending on the MANET characteristics or the application requirements. In order to enable successful dynamic switching, the sensing of the current MANET situation is needed. The sensing of global properties in MANETs is however very costly, if not impossible, because of the continuously changing situations in the network. Therefore, localized switching strategies have to be developed, which can estimate the global situation through local observations, with a certain level of confidence.

The first step towards a single solution that considers both connected and partitioned networks, is an integrated approach that we refer to as *integrated flooding (IF)* [9, 10]. Simulation results show that this approach does in fact broaden the covered MANET scenarios and operating conditions. However, the IF approach only covers a subset of the design space of MANETs with respect to node density and node mobility.

In Fig. 1.1, we outline the design space with respect to node density and mobility. We highlight in the design space the regions, where the discussed protocol classes as well as the IF approach are likely to operate well. The design of a solution that covers the depicted density-mobility space, presents the main objective of our research. In this thesis, we propose a solution that provides efficient broadcast-in-space for the connected parts of the MANET, i.e. network partitions, as well as efficient broadcast-in-time to distribute the messages across the MANET partitions if necessary. We refer to our strategy as *hypergossiping*. We will show that hypergossiping outperforms all existing strategies in the whole design space depicted in Fig. 1.1.

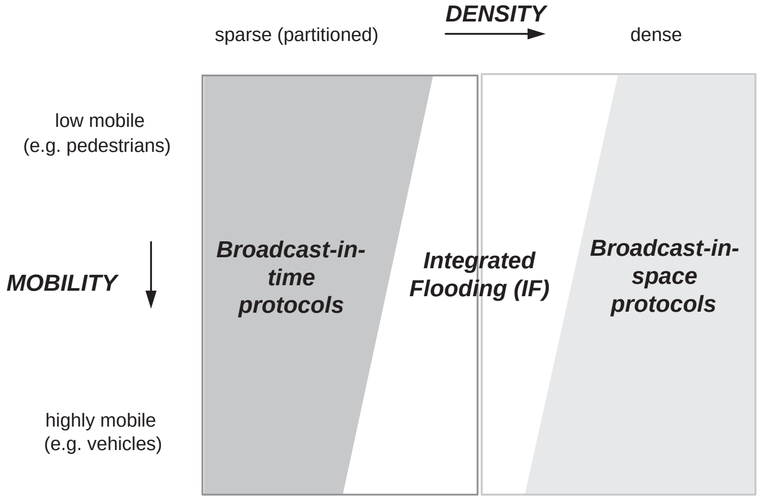


Figure 1.1: Related work in the density-mobility space

1.2 Analytical Modeling and Adaptation of Broadcast Protocols

There is currently only a few analytical work on MANETs in general and broadcasting in particular. This is in part due to the existence of flexible and standardized simulators [11–14], and in part due to the lack of a common platform to base analytical models on. Furthermore, the volatile nature of MANETs complicates analytical modeling. Most of conducted analytical research work aims at the investigation of the capacity of ad hoc networks or the scalability of ad hoc protocols [8,36–38]. For broadcasting, the analytical efforts to-date are completed for ideal network situations such as ideal MAC and static nodes [15–17] or for small size networks [18].

In order to establish analytical models for broadcasting in MANETs, we can learn from the long experience of epidemiologists, since the message spreading among mobile nodes is comparable with the spreading of infectious diseases. In Chapter 4, we show by example how to adopt mathematical epidemic models for broadcasting in MANETs. We calibrated the model using a few simulations. We adopted the so-called Susceptible-Infectious (SI) model for broadcasting in MANETs. Similarly, further models could be adopted, depending on the broadcasting strategy characteristics.

We will show how the adopted mathematical models provide an elegant way to analyze and to adapt broadcast protocols. We adopt the SI model for one broadcast-in-time scheme, i.e. the SPIN-based broadcast protocol, and for one broadcast-in-space scheme, i.e. gossiping (also probabilistic flooding). We show with the gossiping, how the epidemic model supports the adaptation of the forwarding probability to node density. This approach is generic and further studies can be easily conducted. For example, the adaptation of other protocol parameters to further MANET characteristics, or the adaptation of a further protocol to other network properties can similarly be achieved.

We show that the delivery reliability and latency as well as the efficiency of our adaptive gossiping are comparable to that of the best adaptive published strategies for broadcast-in-space, i.e. adaptive counter-based [23] and stochastic flooding [24]. This confirms the applicability of the epidemic models for adaptation of broadcast protocols.

1.3 Our Generalized Broadcasting Technique

Without loss of generality, we can consider each MANET as a set of network partitions that *join* and *split* over time. Adaptive gossiping, our strategy for broadcast-in-space, efficiently distributes messages within single network partitions. Unfortunately, the broadcast-in-space stops and only reaches nodes inside the network partition, where the source of the broadcast is located. Hence we designed a novel strategy that allows for broadcast-in-time. Our strategy consists of one efficient heuristic that detects partition joins and one protocol that rebroadcasts the appropriate messages upon detecting a partition join. Our approach for providing a generalized broadcasting technique for MANETs is to combine the following two strategies: Adaptive gossiping to efficiently broadcast-in-space and our efficient broadcast repetition strategy to broadcast-in-time if necessary. The result is a novel, adaptive, generalized broadcast algorithm, that we call *hypergossiping*. Simulations in ns-2 [11] show that hypergossiping operates well for a broad range of MANETs with respect to node density, node mobility and network load.

Although network partitioning has a great impact on the performance of MANET protocols, the widely used network simulator ns-2 does not provide utilities for an easy evaluation of MANET protocols with regard to network partitioning. Therefore, we designed a framework for ns-2 that provides partitioning information at simulation time in a simple way. Protocol developers might be interested in using this information to evaluate their protocols with respect to network partitioning and to compare their performance to the optimal case. We use the partitioning framework to evaluate our heuristic for partition join detection and to compare the performance of hypergossiping to that of the optimal broadcasting case.

1.4 Contact-Based Mobility Metrics and Buffering

Mobility plays a major role in MANETs since it stresses networking tasks such as routing on the one hand, but aids to increase the network capacity and to overcome network partitioning on the other hand. To benefit from node mobility, a new class of MANET protocols and applications are designed, which are mobility-assisted and must be delay-tolerant. They are called *mobility-assisted* since they exploit the mobility of nodes to transport messages to other nodes. Hypergossiping is an example of this protocol class. For delay-tolerant and mobility-assisted networking, mobility on a large time-scale is a key feature. So far in MANETs, the mobility is investigated on a short time-scale. That is why we present a set of novel mobility metrics that quantify a large time-scale mobility. Our approach is based on the pair-wise encounters and contacts between mobile nodes. We define the *contact* between two nodes as the history of the encounters between them. In [28], we present a detailed statistical study of our novel metrics for the widely used random waypoint mobility model [39] as an example. For the random waypoint model, we introduce in [28] an analytical model that allows protocol developers to analytically compute some of the designed metrics. In order to provide easy access to these metrics in a network simulator, we also provide a framework for ns-2.

The broadcast-in-time component of hypergossiping assumes node motion to successfully increase the delivery ratio in highly partitioned networks. Furthermore, nodes buffer messages and transport them physically before rebroadcasting them on encountering nodes, which have not received the message yet. Mobility is subsequently a major issue for the design of the broadcast-in-time component (or broadcast repetition strategy).

The simplest buffering strategy is to let all receivers store all messages as long as they are relevant. This strategy obviously causes a high buffering overhead, which limits the deployment of hypergossiping in scenarios with a high broadcast traffic or on devices with very limited buffer capabilities. Intuitively, nodes that change partition more frequently should buffer more messages and for a longer time. It is also clear that letting all nodes moving in a group store all messages, is highly redundant.

In this thesis, we present a preliminary effort to reduce the buffering overhead of hypergossiping and outline some ideas for future research. Our approach is utility and probability-based. Nodes estimate the utility of buffering a given message before deciding to actually buffer it. The lower the estimated buffering utility for a message, the lower the probability to buffer the message should be. A localized method is provided to compute this utility depending on the detected mobility patterns. Our preliminary work is based on detecting nodes that are moving in a group. Group motion is detected using a localized method that is based on the recent contacts with other nodes. Simulations show that our strategy results in an important buffer usage reduction in hypergossiping if nodes move in groups, which is a valid assumption in many MANET application scenarios.

1.5 Summary of Thesis Contributions

The main contributions of this thesis are the following:

- We show that there is a strong similarity between the spread of broadcast messages in a MANET and the spread of infectious diseases in a population of individuals. Accordingly, we adopted mathematical epidemic models to elegantly describe and obtain insight in the performance of broadcast protocols in MANETs [22].
- We further show how the adopted analytical models simplify the adaptation of broadcast protocols to key MANET properties. As example we adapt the forwarding probability of gossiping to node density [26].
- We design a solution for a generalized broadcasting technique, hypergossiping, which reduces broadcast storms, and efficiently overcomes network partitioning. Our strategy adapts to node density and node mobility, which makes it deployable in a wide range of MANET scenarios [25] [26].
- We extend the ns-2 simulator, by developing a framework, which simplifies the access to valuable network partitioning information at simulation time. We provide the framework for the ns-2 community. We use this partitioning framework, to evaluate hypergossiping with respect to network partitioning and to compare its performance to the optimal case [28].
- We define a novel set of mobility metrics based on the pair-wise encounters between nodes. These contact-based mobility metrics quantify the mobility on large time-scale and can assist in the design of mobility-assisted protocols and applications. We also provide a framework that simplifies the access of these metrics for ns-2 users [29].
- We use some of the contact-based mobility metrics to adapt the buffering strategy of hypergossiping to node mobility. We present a novel buffering strategy, i.e. mobility-aware probabilistic buffering, where nodes moving in groups cooperate to buffer the message, thus reducing the number of nodes buffering a given message at a given point in time [30] [40].

1.6 Outline of Thesis

The outline of the rest of the thesis is as follows. In Chapter 2, we first review the ad hoc communication technologies that enable the formation of MANETs as well as the potential application scenarios for MANETs. From the application scenarios we derive the main properties of MANETs. Afterwards, we investigate the major considerations for the design of broadcast protocols through discussing the challenges, the requirements and the design issues for broadcasting in MANETs. Finally, we define the system model of our work.

In Chapter 3, we first provide a taxonomy for the related work. Then, we describe the current state-of-the-art in the design of broadcast protocols in MANETs. For this, we review the existing strategies and show their limitations for use as a generalized broadcasting strategy.

In Chapter 4, using the example of the SPIN-based broadcast protocol, we show how to adopt mathematical models from epidemiology to analytically model broadcasting in MANETs. We then calibrate the epidemic model for gossiping and show, using the example of gossiping, how epidemic models can be utilized to adapt protocol parameters to MANET characteristics. Finally, we evaluate adaptive gossiping and compare it to the existing adaptive protocols using simulations for a wide range of operating conditions.

Chapter 5 introduces our generalized broadcasting technique. We show, using simulation results, that this strategy can be deployed for a wide range of node densities and speeds and a wide range of network loads. In this chapter, we present our framework that simplifies the access of network partitioning information in ns-2. A detailed evaluation of hypergossiping concerning network partitioning, using this framework, is then presented.

In Chapter 6, we define a novel set of mobility metrics to quantify mobility on a large time scale. We then show how to use these metrics to detect some mobility patterns such as nodes moving in a group in an efficient and localized way. Finally, we design an efficient probabilistic buffering strategy for hypergossiping that takes advantage of the detected mobility patterns and reduces the buffer overhead of our generalized technique.

In Chapter 7, we summarize our contributions and outline some research directions for future work.

Chapter 2

Background and Preliminaries

The specific characteristics of MANETs raise many challenges for network protocol design on all layers of the protocol stack. The physical (PHY) layer must deal with rapid changes in link qualities. The media access control (MAC) layer needs to minimize collisions and deal with hidden and exposed terminals. At the network layer, nodes need to cooperate in order to calculate paths. The transport layer must be capable of handling packet loss and delay characteristics that are very different from wired networks. Applications should be able to handle possible disconnections and reconnections.

As broadcasting in MANETs is the major concern in this thesis, we investigate in this chapter some preliminaries that are important for an appropriate design of broadcast protocols. Then, we review the typical broadcast applications and derive their requirements on broadcast protocols. Afterwards, we cover the challenges and design considerations for broadcasting in MANETs. Finally, we present the system model we consider for this thesis.

2.1 Ad Hoc Networking Technologies

In this section, we survey the existing ad hoc networking technologies that provide an ad hoc mode, which allows nodes to communicate spontaneously without the need for a communication infrastructure. We compare these technologies with regard to key parameters such as communication range and transmission rate.

2.1.1 Basics

In the following, we present a basic knowledge of the physical layer and the MAC layers designed for ad hoc networking. For more details we refer to [41].

Physical Layer (PHY)

The wireless channel is vulnerable to a variety of transmission impediments such as multipath propagation (reflection, diffraction and scattering), path loss, fading interference, doppler-shift and blockage. These factors limit the transmission range, rate and reliability. The magnitude of these factors changes depending on the environmental conditions and the mobility of the radios.

The common multiple access methods for wireless media are *OFDM (Orthogonal Frequency Division Multiplexing)* and the *spread spectrum*. Two types of spread spectrum are widely used today, namely, *FHSS (Frequency Hopping Spread Spectrum)* and *DSSS (Direct Sequence Spread Spectrum)*.

- OFDM is a multi-carrier transmission mechanism. It resembles Frequency Division Multiple Access (FDMA) in that both split the available bandwidth into a number of frequency channels. OFDM spreads the data to be transmitted over a large number of carriers that are spaced apart at precise frequencies. This spacing provides the 'orthogonality' in this technique, which prevents the demodulators from seeing frequencies other than their own.

- FHSS is a simple technique, where the transmission switches across multiple frequencies in a pseudo-random manner, i.e. the sequence of frequencies is known both by the sender and the receiver, but appears random to other nodes. The switching from one frequency to another is termed frequency hopping. FHSS is mainly used for short range radio communication.

- DSSS assigns a specific n-bit code to each node, called a chipping code. For transmitting a binary 1 or 0, the sender transmits its code or the 1'complement of its code respectively. The assigned codes are orthogonal to each other, so that transmissions are easily and uniquely extracted at the receiver. DSSS is used in all Code Division Multiple Access (CDMA) cellular systems.

The *ISM (Industrial Scientific and Medical)* band is an unlicensed radio spectrum that covers parts in the 433 MHz, 900 MHz, 2.4 GHz and 5 GHz ranges. Only the 2.4 range is unlicensed worldwide. Although this band is highly occupied (microwave oven, babyphone, ..), many ad hoc communication technologies operate on this band (WLAN, Bluetooth..).

Media Access Control (MAC)

Since wireless medium (air interface) is a shared medium, MAC developers investigated, whether they could apply media access methods designed for shared media in wired networks to wireless networks.

Let us take the example of the popular Carrier Sense Multiple Access with Collision Detection (CSMA/CD). Nodes sense the carrier and send as soon as the medium is free, while transmitting nodes listen at the carrier to detect collisions.

The Carrier Sense (CS) feature does not work well in wireless networks, since a sender might not hear another *hidden terminal*. In Fig. 2.1 node A is sending to node B. Node C wants also to transmit to node B. Since it can not hear node A it detects a free medium (CS fails) and starts to transmit to node B. The transmission of node C will collide with the transmission of node A at node B.

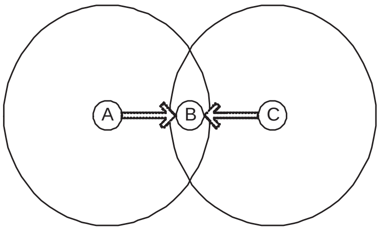


Figure 2.1: The hidden terminal problem

Collision occurs if two neighbor nodes simultaneously start to transmit. The Collision Detection (CD) feature in wired networks is extremely difficult to achieve in wireless networks, since with a single antenna, nodes can only send or receive but not both. Full-duplex radios, which send and receive at the same time, are more expensive.

The CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol avoids collisions among stations sharing the medium by utilizing a random backoff time if the station's sensing mechanism indicates a busy medium.

CSMA/CA still has the hidden-terminal problem. It also shows the so-called *exposed terminal* problem (Fig. 2.2). Node C can not send to node D as long as node B sends to node A due to carrier sense.

In order to solve the hidden-terminal and exposed-terminal problems, the Request-To-Send (RTS) and Clear-To-Send (CTS) scheme was added to CSMA/CA.

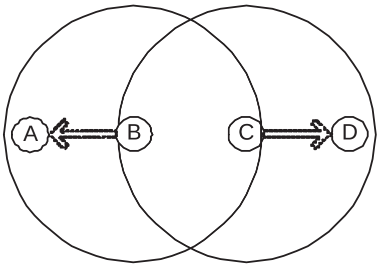


Figure 2.2: The exposed terminal problem

CSMA/CA + RTS/CTS is also known as MACA (Multiple Access with Collision Avoidance) protocol. Collisions still occur if multiple CTS are sent simultaneously.

Exposed terminals contribute to the *contention* problem. Contention occurs if the channel remains busy for a long time so that messages coming from the network layer can not be buffered any more in the MAC buffer and therefore messages have to be dropped from the MAC buffer. Contention is caused by the limited bandwith and also the limited MAC buffer size.

2.1.2 Existing Technologies

This subsection provides a short overview of the main open standards for wireless ad hoc technologies. Open standards enable economies of scale, which decreases the cost of equipment and ensures interoperability. The following emerging standards can be identified: IEEE 802.11, IEEE 802.15, IEEE 802.16, IEEE 802.20, hiperLAN and HomeRF. Each standard conventionally covers both the PHY and MAC layers. As in the literature, we classify these standards into three broad classes: Wireless Local Area Networks (Wireless LANs), Wireless Personal Area Networks (Wireless PANs) and Wireless Metropolitan Area Networks (Wireless MANs).

Wireless LANs (IEEE 802.11, hiperLAN and HomeRF)

IEEE 802.11: *IEEE 802.11* standards use CSMA/CA at the MAC layer. Due to the high bit error rate in wireless networks, 802.11 uses ACKs and a frame repetition mechanism. Optionally, the RTS/CTS scheme can be used.

Wireless technologies that belong to the IEEE 802.11 family of standards provide an ad hoc mode, in addition to an infrastructure mode . This is realized by the so-called Independent Basic Service Set (IBSS) mode or IEEE-ad-hoc-mode.

The communication range of the technologies based on IEEE 802.11 is approximately a few 100m outdoors and less than 100m indoors. Table 2.1 provides a comparison of the standards IEEE 802.11a/b/g/p.

	802.11a	802.11b	802.11g	802.11p (WAVE)
Frequency domain	5 GHz	2.4 GHz	2,4 GHz	5.9 GHz
PHY	OFDM	DSSS	OFDM	OFDM
MAC	CSMA/CA	CSMA/CA	CSMA/CA	CSMA/CA
Max radio range	100m	some 100m	some 100m	1000 m
Max transm. rate	54 Mbps	11 Mbps	54 Mbps	27 Mbps

Table 2.1: Comparison of IEEE 802.11 ad hoc technologies

IEEE 802.11p, also called *WAVE (Wireless Access in Vehicular Environments)*, is especially designed for vehicle-to-vehicle communication and for communication between the on board units (OBUs) and the Road Side Units (RSUs) in the licensed Intelligent Transportation Systems (ITS) band of 5.9 GHz. It uses the highest frequency domain in the 802.11 standards. This is because of the high mobility of nodes, which should be compensated with a high transmission power [42]. The standard resembles 802.11a, in that it uses the 5 GHz band and deploys OFDM. The main difference to the *a*-standard is the reservation of a control channel for periodic beacons. The standard prescribes that data has to be prioritized. If data with higher priority has to be transmitted, other existing data streams will be suppressed. It is intended that it will support relative node speeds of a minimum 200 Km/h and communication ranges of over 1000m. The standard is scheduled to be published in 2007.

The IEEE 802.11 standards have been widely accepted by manufacturers, as well as consumers. *Wi-Fi (Wireless Fidelity)* is the most popular wireless LAN (WLAN) technology. In essence, Wi-Fi refers to IEEE 802.11-compatible products and covers office-based and home-based LANs, as well as the publicly available hot spots. There have been many extensions to the IEEE 802.11 standard such as support of Quality-of-Service (QoS) in IEEE 802.11e, roaming in IEEE 802.11d/r, mesh networking in IEEE 802.11s, and interworking with non-802 networks in IEEE 802.11u. These optimizations increase the deployment scenarios of these standards.

A MAC broadcast in IEEE 802.11 standards can be initiated by every node. All nodes within the communication range of the broadcast initiator are able to receive the broadcast message.

HiperLAN: The European counterparts to the IEEE 802.11 standards are the *high-performance radio LAN (hiperLAN)* standards defined by the European Telecommunications Standard Institute (ETSI). Four standards have been defined for wireless networks by the ETSI, i.e. hiperLAN-1, hiperLAN-2, hiperLAN-3 (or hiperaccess) and hiperLAN-4 (or hiperlink). In the following we briefly introduce the hiperLAN standards that provide besides the infrastructure mode an ad-hoc mode, namely HiperLAN-1 and HiperLAN-2. The ad-hoc mode (also called direct mode) of communication is, however, managed by a central controller. Table 2.2 compares these different standards concerning communication range and rate, and frequency domain.

HiperLAN-1 is a standard that was introduced by the ETSI in 1995. Apart from supporting node mobility, hiperLAN-1 provides a forwarding mechanism (multi-hop routing). Topology-related data are exchanged between the nodes periodically with the help of special packets, for the purpose of forwarding. HiperLAN-1 supports node speeds up to 1.4 m/s.

HiperLAN-2 attempts to integrate WLANs into next-generation cellular sys-

	hiperLAN-1	hiperLAN-2
Frequency domain	5.15+17.1 GHz	5 GHz
PHY	GMSK	OFDM
MAC	EY-NPMA	TDMA/TDD
Max. radio range	50m	50-100m
Max. transm. rate	23.5 Mbps	54 Mbps

Table 2.2: Comparison of hiperLAN standards

tems. It aims at converging IP and ATM type services at a high data rate of 54 Mbps for indoor and outdoor applications. HiperLAN-2, an ATM compatible wireless LAN, is a connection oriented system, which uses fixed size packets and enables QoS applications to be easily implemented. HiperLAN-2 supports node speeds up to 10 m/s.

The hiperLAN standards could not penetrate the market, although they provide some interesting technical mechanisms.

HomeRF: *HomeRF* has been developed by the HomeRF working group, a consortium of mobile communication companies. HomeRF operates in the ISM band and deploys FHSS with 50-100 hops/s. For MAC, HomeRF uses CSMA/CA. The maximum data rates are 1.6, 10 or 20 Mbps. HomeRF supports both infrastructure-based and ad hoc communication. After the successful market penetration of IEEE 802.11 in 2003, the development of HomeRF has been stopped.

Wireless PANs (IEEE 802.15, IrDA)

IEEE 802.15 is concerned with personal area networks (PANs). PANs are local networks in which all of the devices are controlled by a single user or a family. IEEE 802.15 covers Bluetooth plus two other PAN standards, known as IEEE 802.15.3 and IEEE 802.15.4.

Bluetooth: Bluetooth specifies a complete protocol stack. Up to eight devices can communicate in a small network called a piconet, consisting of a master and from one to seven active slave devices. The master determines the channel (frequency-hopping sequence) and phase (timing offset, i.e. when to transmit) that shall be used by all devices on this piconet. A device in one piconet may also exist as part of another piconet (as master or slave). This form of overlapping is called a scatternet. Ten of these piconets can coexist in the same coverage range of the Bluetooth radio.

At any given point in time, the bandwidth available for all devices of one piconet is 1 Mbps. Since frequency hopping is used, no collisions take place. Collisions will occur when devices in different piconets, on different logical channels, happen to use the same hop frequency at the same time. As the number of piconets in an area increases, the number of collisions increases, and the performance degrades. In summary, the physical area and total bandwidth are shared by the scatternet.

Bluetooth provides support for three general application areas using short-range wireless connectivity: Data and voice access points, cable replacement and ad hoc networking.

Bluetooth is characterized by a shorter radio range (10m or 100m) and much higher channel establishment times (minimum 3s and typically 10s). Bluetooth is designed for personal area networking and in particular, as a cable replacement technology, where its radio range and channel establishment time are not critical. But this makes Bluetooth unsuitable for relatively dynamic networks. In addition, Bluetooth is based on its piconet topology, which limits the number of devices communicating in an ad hoc pattern to 8 devices. Although the scatternet extension in Bluetooth allows a more scalable ad hoc communication but it still shows less flexible topology. Bluetooth however, is characterized by a high energy efficiency and low chip cost.

Bluetooth is the first wireless technology which has actually tried to provide a unique communication paradigm for all household consumer electronic devices. It has been successful but it does have its limitations, such as the absence of routing and handoffs support, the bottleneck character of the master-slave architecture in terms of performance.

	802.15.1 (Bluetooth)	802.15.3 (UWB)	802.15.4 (ZigBee)
Frequency domain	2.4 GHz	3-10 GHz	2.4 GHz
PHY	FHSS	MB-OFDM	DSSS
MAC	Polling	TDMA	CSMA/CA
Max. radio range	10m or 100m	10m	10-75m
Max. transm. rate	1 Mbps	480 or 1320 Mbps	0.25 Mbps

Table 2.3: Comparison of IEEE 802.15 ad hoc technologies

The MAC broadcast in Bluetooth is different from that of IEEE 802.11, since neighbors of a node are not determined by physical approximity, but by the logical structure of the scatternet. Only the master can broadcast. The master only reaches its slaves by means of MAC broadcast.

Ultrawideband: Ultrawideband (UWB) relies on the standard IEEE 802.15.3a. The communication in UWB is based on the piconet architecture. The maximum number of devices in a piconet is 250. The standard enables the movement of massive files at high data rates over short distances. The standard operates on frequencies between 3.1 and 10.6 GHz. Until now, UWB has only offered a transmission range of about 10m at 100Mbps, 3m at 480Mbps (OFDM-UWB) or at 1320 Mbps (DS-UWB).

ZigBee: ZigBee is a new wireless communication standard for mobile devices, sensors and actuators, which allows wireless near field communication and control. Application fields include home networking, automation, security, facility management, and machine-to-machine (M2M) communication.

Similar to Bluetooth, the ZigBee standard specifies a complete protocol stack. The main design objective was to develop reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard. The technology is designed to be simpler and cheaper than Bluetooth.

ZigBee uses the standard IEEE 802.15.4 for MAC and PHY layer. The radio uses DSSS. The basic mode of channel access specified by IEEE 802.15.4 is CSMA/CA. ZigBee functions at a relatively low data rate over relatively short distances, compared to Wi-Fi. ZigBee provides transmission rates of 20, 40 and 250 Kbps using the frequencies 868 MHz (Europe), 915 MHz (USA), and 2,4 GHz respectively. The maximum transmission range is 75 m.

In ZigBee there are three types of devices: End devices (called Reduced Function Devices or RFD), Routers (called Full Function Devices or FFD) and Coordinators. End devices have to register to an arbitrary router and thus build a star-topology with routers. Routers build a tree or a mesh topology with each other. Exactly one router becomes a coordinator for the whole network.

Neighbor discovery and channel establishment in ZigBee are much faster compared to Bluetooth. Also, the maximum number of nodes in a ZigBee network ($2^{12} = 4096$) is much higher than in Bluetooth. ZigBee technology enables the coordination of (multihop-) communication among thousands of tiny sensors.

An FFD can communicate with every ZigBee device in its communication range. An RFD can only communicate with its associated FFD. Only an FFD is able to initiate a MAC broadcast, which can reach all nodes within communication range.

IrDA: The Infra-red Data Association (IrDA) specifies standards using the infrared region of the light for near field communication. Infrared communication assumes line of sight, which represents its main limitation for use in MANETs. Infrared has an effective range of some few meters and a maximal data rate of 4 Mbps. The infrared devices consume extremely lower power and cost less than Bluetooth devices.

Wireless MANs (IEEE 802.16, IEEE 802.20)

WiMAX: WiMAX is an acronym that stands for Worldwide Interoperability for Microwave Access. WiMAX is an interoperability specification based on the IEEE 802.16 standards. WiMAX is similar to Wi-Fi, in that both create hot spots, but while Wi-Fi can cover some hundreds of meters, WiMAX has a range of up to 50 Km (typically 2-5 km). Thus, WiMAX provides an alternative to cable for the last-mile broadband access or 'wireless DSL'. WiMAX also provides the capability to create ad hoc mesh networks. It will also be used as a complimentary technology to connect IEEE 802.11 hot spots to the Internet. Sony and Microsoft are working on adding WiMAX as a feature in their next generation game console. This should allow players to create ad hoc networks with other players. Initial deployment of WiMAX are in fixed locations, but a mobile version is under development (IEEE 802.16e, approved December 2005). WiMAX supports mobility of between 20 and 100 km/h.

	802.16	802.16a	802.16e
Frequency domain	10-66 GHz	2-11 GHz	2-6 GHz
PHY	OFDM-256	OFDM-256	SOFDMA
MAC	scheduling MAC	scheduling MAC	scheduling MAC
Max. radio range	5 km	50 km	5 km
Max. transm. rate	134 Mbps	70 Mbps	30 Mbps

Table 2.4: Comparison of IEEE 802.16 ad hoc technologies

IEEE 802.20: *Mobile-Fi* is based on the *IEEE 802.20* standard. Its objective is to provide Internet access to mobile users at high data rates (1 Mbps). The standard operates in the licensed bands below 3.5 GHz and supports vehicle mobility up to 250 km/h. Mobile-Fi users could enjoy broadband Internet access while traveling in a moving car or train.

Summary

In this section, we reviewed the existing communication technologies that enable the realization of MANETs.

The deployment considerations and choice of appropriate technology for a MANET are network coverage, bandwidth requirement, flexibility concerning topology, target carrier platform (e.g. mobility and device capabilities), scenario of deployment, scalability, and finally, the cost of deployment.

It would also make sense to consider integrated MANETs that consist of different MANETs that are formed using different ad hoc technologies. This requires some nodes to be equipped with different network interfaces to play the role of gateways between different MANETs.

Since most of the technologies provide, besides ad-hoc mode, an infrastructure mode, the integration of MANETs into infrastructures is simple and will open new fields of application. Mesh networking and last mile broadband access are only the first steps towards this marriage.

IEEE 802.11 is without doubt one of the most widely accepted wireless technologies today. It is a very popular technology because it resembles ethernet and because it is flexible, scalable and demonstrates comparatively high transmission rates. Although our concepts are generic, in this thesis, we will mainly evaluate them using a model similar to the IEEE 802.11 standard.

2.2 MANET Applications and Characteristics

MANETs are suitable for scenarios where an infrastructure is costly or even unavailable and communication must be deployed quickly. Ad hoc networking is originated in military research, but it is gaining more and more importance in commercial applications, academia and emergency services. Besides the adjustment of traditional applications to the ad hoc context (e.g. to reduce costs and to increase comfort), a large number of potential new services can and will be generated using the new communication paradigm. In the following, we discuss the most typical applications and projects for MANETs. Subsequently, the distinct characteristics of MANETs are highlighted.

2.2.1 MANET Applications

Vehicle Ad hoc Network: Vehicle Ad hoc NETworks (VANETs) have received increased attention in recent years. VANETs are considered to be the most popular civil application for MANETs. The network is formed by cars moving on the road, with the main aim of car-to-car communication being to improve road safety and traffic flow.

CarTalk2000 [43] is a project funded by the European Commission and aims to increase the driving safety by allowing active traffic signs such as a virtual warning triangle, or by enhancing cooperative driving such as assisted lane change. These applications are novel applications, which only became possible with the advances in ad hoc communication paradigms. The authors in [44] proposed to realize the traditional application of traffic flow control (so far infrastructure-based) using the concept of data aggregation in ad hoc networks. This should decrease the cost of deployment and maintenance.

Further important research projects are the project NOW (Network-on-Wheels) [45] funded by the German Federal Ministry of Education and Research, the project PReVENT [46], the project FleetNet [47] co-funded by the European Commission and industrial partners, and the project CarTel at MIT [48].

The Car2Car communication consortium [49] was founded by six European car manufacturers. The consortium is dedicated to the objective of further increasing road traffic safety and efficiency by means of inter-vehicle communication.

DSRC (Dedicated Short Range Communications) is a US Department of Transportation project [50], which is expected to be the first wide-scale VANET in North America. The project particularly investigates applications such as toll collection, vehicle safety services, and commercial transactions via cars. IEEE 802.11p will be used as the foundation for the DSRC project.

VANETs are usually large-scale networks, since the number of nodes constituting the network may reach thousands or millions of cars, however these networks are more sparse in nature [51]. Due to the high mobility of nodes, VANETs are characterized by a very highly dynamic network topology. In VANETs relative node speed may reach 500 km/h, which means two vehicles driving in opposite directions only stay in each other's communication range for a few seconds. Since digital maps are increasingly available in cars and more and more sensors exist on board or on the road-side such as GPS, speed and adhesion sensors, we observe an increasing number of deployment scenarios for VANETs. Other commercial scenarios related to VANETs include ship-to-ship communication.

Disaster-Rescue or Emergency Operations: Ad hoc networking can be very useful in emergency and rescue operations as well as for disaster relief efforts, e.g. in fire, flood, storms, hurricanes or earthquakes. The nature of the terrain in such applications is characterized by the absence of or by the partial or even the complete destruction of conventional infrastructure-based communication facilities and by the need for rapid deployment of a communication network. Examples are the tsunami and the New-Orleans disasters. Different teams may be formed, such as rescue teams, police teams, firefighter teams and volunteer teams. Equipped with small handheld radios, team members would be able to establish an ad hoc network and carry that network with them as they maneuver to accomplish a particular mission such as finding survivors, providing medical aids, and preventing rampage. To be effective, the networks must be self-configurable and self-organizing, as nodes move, enter or leave the field.

Wireless Sensor and Actuator Networks: Wireless Sensor and Actuator Networks (WSANs) mainly aim at simplifying the monitoring of critical or hard to observe places or situations, so that notifications and counter-measurements may be triggered at an early stage. In WSANs the nodes can be either static or mobile. [52] provides a survey of wireless sensor and actuator networks. ZigBee is a promising ad hoc networking technology for the realization of WSANs.

Examples of application are biological information acquisition and habit monitoring of wildlife species. The eye-based or video-based approach is sometimes inaccurate or even impossible. An example of the animal ad hoc network is the ZebraNET system. On the biology side, the goal of the ZebraNET project [53] is to use systems to perform novel studies of animal migrations and inter-species interaction. On the system side, ZebraNET is studying energy-aware and position-aware MANETs. Further examples include the Electronic Shepherd system [54], Whales [55], ZebraNET, the Sami Network Connectivity project [56], and Data MULEs [57].

Military Operations: Modern military equipment contains some sort of computer equipment. Ad hoc networking would allow the military to quickly maintain an information network between the equipments carried by soldiers, fleets and tanks. Two examples of military projects are the Near-Term Digital Radio (NTDR) and Joint Tactical Radio System (JTRS).

The NTDR system is a mobile packet radio network consisting of up to 400 radios and a network management terminal to serve a 20 x 30 km area. The NTDR program provided a prototype MANET to the US army. The protocols were provided by BBN and the radio hardware by ITT. These systems have been purchased and fielded by a number of other countries.

The JTRS was planned as the next-generation radio for use by the US military in field operations. JTRS is a software-defined radio (SDR) for voice and data that will be backwards compatible with a very large number of other military and civilian radio systems. It also includes wideband networking software to implement full-featured mobile ad hoc networks.

Wireless Personal Area Network (PAN): As we mentioned earlier IrDA, Bluetooth, UWB and ZigBee can replace cables and simplify the intercommunication between various personal mobile devices such as PDAs, laptops and cellular phones. We refer to such networks as PAN. PANs are characterized by a communication range of some meters, and we consider them as short-range MANETs. PANs usually consist of a few devices that are geographically close to each other. PANs are potentially a promising application field of MANETs in the future pervasive computing context.

Collaborative Computing: Another domain, in which the ad hoc wireless networks find applications, is collaborative computing. The need for a spontaneous communication with minimal configuration among a group of people at a conference, gathering or classroom can be fulfilled with an ad hoc wireless network. Consider for example, a group of students who want to share their solutions for a homework, or a lecturer distributing digital documents to the class on-the-fly. In such cases the formation of an ad hoc network can serve the purpose. Devices used in such applications could typically be laptops or PDAs with wireless ad hoc interface cards.

2.2.2 MANET Characteristics

Taking into account the diversity of MANET applications that were stated before, MANETs generally may show the following key characteristics:

Dynamic Topology: MANET nodes or a subset of them are usually mobile and can be connected dynamically in any arbitrary manner. Network links may vary over space and time, based on the proximity of hosts to each other and on the spatial environmental conditions where the MANET is deployed.

The MANET connectivity changes *over space*, due to a general non-uniform node spatial distribution in the area of interest. Whereas in over-crowded regions the connectivity is very high, and collision and contention probability thus becomes high, in under-crowded regions network partitioning takes place and communication becomes difficult if not impossible.

The MANET connectivity also changes *over time* given the node mobility, e.g. in VANETs jams appear and disappear over time. Node mobility normally demonstrates a non-uniform distribution over space (highway versus city road) and over nodes (vehicle versus pedestrians). This means network links change over time and space heterogeneously. Nodes in general move loosely from each other, which makes the topology prediction at run-time very difficult.

Topology changes are not only caused by the mobility of the nodes and the environmental conditions, but also vary with the ad hoc technology used, which show quite different properties such as communication range. The on-off usage pattern of mobile nodes and node failures such as node crash also lead to topology changes. In the open MANET scenarios where nodes can leave and join the network autonomously, the scale of the network may also vary dramatically. The penetration rate of the radios to the open application scenarios also lets the topology changes vary on the large time-scale, e.g. the number of cars equipped with radios will increase over the years.

Some applications such as tactical military scenarios may have a wide range of arrangement in terms of node mobility, and therefore the corresponding network topology changes are somewhat predictable. The *non-determinism* of topology-changes is however inherent in most of the MANET applications, since nodes function autonomously.

Summarizing, we are dealing in general with an unpredictable and continuously changing topology. In particular, *node spatial distribution* and *node mobility* play a major role in topology variation, and subsequently in the design of mobile ad hoc networks.

Scarce Resources: In most MANET applications, the most restricted resources are bandwidth, energy, storage and computation power.

Wireless links have a significantly lower capacity than wired ones. They are affected by several error sources that result in the degradation of the received signal. Therefore, bandwidth is a limited resource in MANETs.

In most MANET applications, nodes are mobile and can not be line-powered. MANET nodes are therefore generally battery-powered. As battery energy is a limited resource, one of the most important system design criterion for optimization may be saving energy.

Despite the rapid increase of computational and storage technologies for stationary devices, the miniaturization process, as well as the reduction of power usage of these products, is much slower. Consequently, mobile devices still show limited computational and storage capacities.

From all these resources, energy remains the most constrained one. Measurements showed that the network interface is the largest energy consumer [58, 59].

Heterogeneity: The large spectrum of MANET applications shows that the number of participant nodes can range from several nodes to tens of thousands of nodes. Furthermore, different scenarios may show different node mobility degrees varying from static nodes such as static sensor nodes to highly mobile nodes such as vehicles or planes. The heterogeneity in network scale and in node mobility leads to a varying degree of *topology dynamics*.

Different application scenarios may also show different levels of *scarcity of resources*. For WSANs, in particular limited energy, but also computational power and storage are the major design concerns. However, unlike most MANET applications, limited energy and limited computational power are not the issues in VANETs.

If we consider a coupling of different MANETs, we will also deal with heterogenous network interfaces with different capabilities. This heterogeneity gives rise to significant design challenges.

The heterogeneity is not only observed from one scenario to another, but also in the same scenario. Due to the high diversity of mobile devices that can be easily plugged into existing MANETs, A real MANET may have enormous variation in its device capabilities, and may show a strong heterogeneity in its node spatial distribution and node movement.

We especially emphasis that a MANET naturally shows continuously varying node density and mobility over time and space.

2.3 Broadcast Protocol Considerations

In this section, we discuss the typical broadcast applications and derive application requirements for broadcasting in MANETs. Then, we investigate the main design issues and requirements for a generalized broadcasting technique in detail.

2.3.1 Typical Broadcast Applications

Broadcast protocols are a fundamental building block to realize principal middleware functionalities such as replication [1], group management [2] and consensus [3]. They provide also a key facility for the (self)management of the MANET, e.g., the auto-configuration of node addresses [60]. Broadcasting is frequently used in MANETs with highly dynamic topology to realize other network protocols such as multicast [6]. Furthermore, we identify the following three classes of applications for broadcasting.

Content Distribution Applications: A typical application for broadcasting in MANETs is news spreading. Examples are the broadcasting of aid information in a disaster area to coordinate relief actions (e.g. fire fighting [61]), the dissemination of parking availability in a city scenario, dissemination of accident information in VANETs and the dissemination of alarms and announcements.

Further content distribution applications are publish-subscribe applications, where some nodes are subscribers to content providers. These applications typically run in the background for a few hours or even a few days. Examples are usenet-on-fly [62], latency insensitive data [63], and file sharing in a peer-to-peer (P2P) manner [64].

Resource Discovery and Advertisement: Further typical broadcast scenarios are resource (or service) discovery and advertisement. MANET nodes may have little or no knowledge at all about the capabilities and services offered by each other. Therefore, mechanisms for resource discovery or advertisement are important for these self-configurable networks. Due to the decentralized and highly dynamic nature of MANETs, service discovery and service advertisement frequently use broadcasting strategies.

An example of resources is a multi-hop routing path to a given destination. For highly dynamic topologies the route is continuously changing and the resource is so highly dynamic that maintaining a route to all nodes at every time is very costly. However, most of the time, it is not necessary to have an up-to-date route to all other nodes. Hence, a novel class of reactive routing protocols, such as DSR [4] and AODV [5], has been developed. Reactive routing protocols only set up routes to nodes they communicate with and these routes are kept alive as long as they are needed. This is realized by a route discovery mechanism, which uses broadcasting strategies to distribute control messages for route discovery.

Sensor Data Dissemination: Another important application field for broadcasting is the *sensor data dissemination.* Real-time sensor data may be disseminated to other nodes in order to realize a fully-replicated database, where every database node has a consistent view of real time events. Data consistency algorithms act on disseminated observation data [65] and chronologically order observed events in MANETs. Aggregation algorithms may use broadcasting strategies to distribute sensed values to aggregators [44].

2.3.2 Application Requirements on Broadcasting

Given the typical broadcasting applications, we now investigate their requirements on broadcast protocols. The application requirements for broadcast protocols are generally on the set of nodes that should be reached ('which'), and the timeliness of data delivery ('when'). Accordingly, we identify the following key requirements of applications on broadcast protocols:

- *Delivery Reliability*: The conventional requirement of applications for a broadcast protocol is to reach all nodes that have to be reached. We define the *Reachability* (RE) as the ratio of nodes reached by the broadcasting strategy to the total number of nodes that should have been reached. The reachability quantifies the delivery reliability of the broadcast protocol. Typically, applications require high broadcast delivery reliability, i.e. the maximization of the reachability.

- *Delivery Timeliness*: A further crucial demand of applications operating on broadcast data is the availability of up-to-date data. This can be understood as a requirement on the broadcast protocol to disseminate as fast as possible (in time).

Similar to [66], we assume in this thesis that broadcast data typically has a temporal and spatial relevance and should be disseminated within this scope. The application should pass the *time and space relevance* along with the data to the broadcast protocol. Hence nodes that have to be reached are determined by the relevance of data.

Considering the on-demand routing protocols as a broadcast application, the time relevance of broadcast data (route requests) is in the range of milliseconds or a few seconds. The consistency algorithms may require that the broadcasting of sensor data updates should be performed before the next updates are generated. Thus time relevance of updates may be set equal to the time elapsing from one update to the next. For publish-subscribe applications the time relevance of data can be set in the range of minutes, hours or even days.

In this work, we only consider the temporal relevance of data and assume that information is only valid for a certain given period of time, i.e. its *lifetime*, and becomes irrelevant after that. Lifetime is application dependent and may be in the range of seconds, minutes, or even hours. This can be explained by the following observations: News may be updated after a certain period of time, shopping offers are limited by a certain date, parking availability information may change frequently, and sensor information may be updated periodically.

2.3.3 Design Considerations for Broadcasting

In the following we discuss the design considerations for broadcast protocols in MANETs. For this, we first outline the objectives that the protocol design should follow to cope with the distinct properties of MANETs. Then, we argue for our basic decisions to design appropriate broadcast protocols in MANETs.

Design Objectives

As stated before, the highly dynamic nature of MANETs has a great impact on the protocol and application design. We believe that decentralization, adaptation, tolerance, scalability, and resource-awareness are the key design issues for applications and protocols in MANETs in general and for broadcast protocols in particular.

- *Decentralization/Localization*: Conventionally, no centralized administration entity exists to manage the operation of the MANET and centralized solutions can not really be applied in such networks without a huge overhead of communication. Therefore, self-organization is indispensable in MANETs and decentralized or localized protocols should be proposed.

- *Adaptation/Generalization*: Due to the diversity of MANET applications and the continuously changing characteristics of the MANET over space and time, a generalized solution that is applicable for most (and ideally for all) application scenarios and that adapts to the key MANET characteristics at run-time, is a major step towards the real implementation of MANETs.

- *Tolerance*: Network failures such as network partitioning are the norm rather than the exception in MANETs. Thus applications have to be tolerant, in order to deal with disruptions and unpredictable network conditions. Tolerance should be reflected in the application requirements for the underlying network protocols. Applications have to tolerate higher end-to-end delays, lower delivery ratios, temporal data inconsistencies and data incompleteness, just to mention a few.

  Many challenging research areas exploit this application tolerance to design novel networking concepts such as *opportunistic networking*, *disruption-tolerant networking* and *delay-tolerant networking (DTN)*. They are grouped under the delay-tolerant networking research field (see the Delay Tolerant Networking Research Group (DTNG) [67] [68] [69] [70]).

  Concerning delivery reliability, applications have to be *fault-tolerant* and accept that some nodes would not be reached if the application requires immediate data delivery. This is the case if the network is partitioned and no partition join takes place during the delay tolerated by the application.

  On the other hand, if the application is *delay-tolerant*, the broadcast protocol has more time to improve its delivery reliability for the application. The underlying protocols should exploit the delay-tolerance of the applications for the purpose of increasing the capacity of the network or increasing the reachability by overcoming network partitioning. They have to match the properties of the MANET, such as node spatial distribution and mobility, to the tolerated delay.

  we assume that applications tolerate a given maximum delay for broadcast delivery, and that the delay tolerated is equal to the time relevance of the application data, which is to be broadcasted.

- *Scalability*: The diversity of MANET applications show that protocol developers have to deal with networks that consist of a wide range of node numbers. The MANET scale ranges from some dozens (PAN) to some thousands or tens of thousands of nodes (e.g. VANETs or sensor networks). Accordingly, MANET protocols should also support large-scale scenarios, i.e. to be scalable.

- *Resource Efficiency*: MANET protocols and methods are supposed to run on network nodes with limited energy, computational power and memory. Consequently, the protocols should be designed to be very resource efficient or frugal.

Design Issues

The air interface is a shared medium, which shows its broadcast nature. This broadcast nature of radio channels can be exploited for simple and efficient local broadcasting. We refer by *local broadcast* or *MAC broadcast* to the capability of sending one message to all nodes within the communication range using one single transmission. This capability can be used to send network-control traffic to all neighbors (e.g. HELLO beacons) or to support broadcasting algorithms. Since in a shared medium a broadcast to all nodes in the transmission range costs the sender as much as one unicast transmission to a single neighbor, it is recommended that broadcasting strategies exploit this property.

As mentioned earlier, in this thesis we focus on the IEEE 802.11 standard. Therefore, all nodes located within the communication range are able to receive a local broadcast. Using local broadcast nodes do not use the optional RTS/CTS optimization of CSMA/CA, but transmit data (on a free channel) and without any consideration for hidden terminals. The lack of the RTS/CTS feature may also lead to the exposed terminal problem, which reduces the network capacity. Therefore, MAC broadcasting decreases the reliability of data delivery compared to sending the message to all neighbors using unicast with the RTS/CTS scheme. However, MAC broadcasting massively reduces the number of transmissions compared to the unicast-based broadcast. Accordingly, MAC broadcasting is frequently used to develop broadcast protocols in MANETs.

The most straightforward solution for broadcasting is when nodes forward a received message to all their neighbors using local broadcasts. Eventually, all nodes within the network should receive the message. This primitive strategy is like a flood and therefore, is called *plain flooding* (also called simple flooding, pure flooding, or blind flooding). Even though flooding might lead to an unnecessary message overhead, it should provide a robust basic strategy for broadcasting in networks with an unknown or changing topology. Due to its simplicity, localized nature, and topology transparency, flooding is widely used in MANETs as a basic scheme for many broadcast protocols.

However, the characteristics of MANETs still prohibit a flooding process from reaching every node. If the density of nodes is too high, the radio transmission will block out messages, since too many nodes are repeating their incoming messages. This problem is referred to as *broadcast storms* [7]. It describes three major problems that in particular occur if plain flooding is used to realize broadcasting in crowded MANETs: *redundancy*, *collisions* and *contention*. Redundancy takes place if message forwarding is useless because it only reaches nodes that already have received the message. In order to send a broadcast message, a mobile node only needs to assess a clear channel before transmitting. Therefore, collision (e.g. due to a hidden node) may occur frequently. Without further measures, a mobile node is not able to know whether a message was successfully received by its neighbors. Contention occurs if the sender has messages in its MAC buffer but it could not send them. Depending on the buffer strategy and its parameters, messages may be deleted from the queue, if the channel is blocked by other neighbors.

In order to relieve the broadcast storm problem, strategies to *restrict forwarding* to a subset of nodes have to be developed. It is obvious that for a fixed communication range the intensity of the broadcast storm problem depends on the node density, i.e. number of nodes per unit of area. Therefore, node density has a great impact on the performance of such strategies.

On the other hand, if node density is very low, the network becomes partitioned and flooding will only reach nodes in the partition containing the source. The common approach to deal with partitioning is that nodes cache messages and *repeat forwarding* at the appropriate time, i.e. partition join. We call each repetition and subsequent forward a *rebroadcast*. For this purpose we need efficient repetitive strategies.

Global view detection may become very costly in MANETs and particularly in highly mobile ones. Therefore, scenario detection is undesired in MANETs. Furthermore the same node may participate in different MANET scenarios. As a result, we require a *generalized technique* for broadcasting, which is suitable for most of application scenarios.

2.3.4 Design Requirements on a Generalized Broadcasting Technique

Since a generalized broadcasting technique is required for MANETs, we derive from the outlined MANET characteristics, the application requirements and the design issues the design requirements on a such technique. We distinguish the following five design requirements on a generalized broadcasting technique for MANETs.

Requirement 1: A generalized technique has to deal with general characteristics of MANETs and assume as *less* node capabilities as possible, since in real-world scenarios there will be a high diversity in applications and node capabilities.

Requirement 2: Due to non-uniformity of node density over the MANET area, a generalized technique should *combine* different strategies, in order to deliver messages throughout the network. Efficient methods for switching the strategies should be provided.

Requirement 3: Since node density heavily influences the performance of broadcast protocols, and because MANETs may show a wide range of node densities, the third requirement for a generalized broadcast strategy for MANETs is to *adapt* to the node density, in order to reduce broadcast storms and overcome network partitioning.

Requirement 4: Since a global state in MANETs is hard to obtain and spatial distribution of nodes may change continuously, the fourth requirement on such a strategy is that nodes adapt to *local* MANET characteristics *independently.* This means that each node should be capable to switch among different strategies or to tune the parameters of the selected one based on its *own* perspective on the network.

Requirement 5: From the point of view of protocol design, the generalized solution has to increase the *efficiency* of broadcasting for a wide range of MANET characteristics and a wide range of application requirements. The efficiency of broadcast strategies is measured based on the message overhead, which is a good indicator of energy consumption and bandwidth use, and the storage overhead.

2.4 System Model

In the following, we present our system model that consists of the node and population model as well as the network model. Roughly speaking, we consider a generalized MANET that shows a high heterogeneity and require less network and node capabilities so that our broadcasting technique is as generalized as possible.

2.4.1 Node and Population Model

We consider a generalized MANET formed by N mobile nodes, such as notebooks, PDAs and sensor node. We do not bound N, which can be in the range of tens or millions. These nodes may be either static or move according to an *arbitrary mobility model* within a geographic area of interest A. We suppose that the MANET may show *highly heterogeneous spatial distribution* of nodes, from locally very sparse to very dense, and a *highly heterogeneous node mobility pattern*, from static to highly mobile.

We denote the set of nodes that form the MANET by the node population. In this thesis, the node population is defined by fixing the geographic area A and considering the set of nodes sojourning there. We assume that nodes are uniquely identified, e.g. using their MAC addresses. Without loss of generality we enumerate the N nodes using integers $\in [0, N-1]$.

In this thesis, we allow nodes to select autonomously their operating energy modes. Our algorithms do not actively switch the energy modes of nodes, but tolerate that nodes may change their energy modes at arbitrary times.

In general, we assume the *autonomy* of the users and therefore their mobile devices. Nodes may join or leave the MANET at arbitrary times, e.g. due to the on-off usage pattern for users. We also assume that devices do not change their movement trajectories for networking purposes.

Since MANETs are highly error-prone we tolerate that nodes may fail due to the depletion of batteries or software or hardware failures.

Nodes are not required to have special capabilities such as the knowledge of their position or speed. We do not require synchronized clocks but require local clocks with a bounded drift comparable to that of ordinary quartz clocks, which is the case of the majority of commercial mobile devices. We assume that nodes are aware of their neighboring nodes. Nodes detect neighbors by periodically broadcasting a HELLO control message. Every node maintains a list of neighbors and regularly updates the list based on the HELLO messages it receives.

Security issues for broadcast protocols are not the concern of this work. We assume that nodes are neither selfish nor malicious.

2.4.2 Network Model

For this thesis, we consider an ad hoc networking technology that is similar to the IEEE 802.11 standards. We assume that the communication technology allows a MAC broadcast to all nodes within communication range. A transmitted message reaches every node within the transmitter's transmission range. A transmission is successful if the message is the only one reaching the receiver, and the receiver itself is not transmitting at the same time. We do not require that node can dynamically adjust their communication range R.

Due to the error-proneness of MANETs, our algorithms consider a wide range of network failures that may occur while broadcasting and handle them as a normal network situation rather than abnormality. The main failures that we tolerate are frequent network topology changes, collision, contention and network partitioning.

The *network* considered within the scope of this work is a set of nodes in an environment, which all can communicate with each other either directly or through multi-hops within a certain time interval. As mentioned earlier, without loss of generality, we model a MANET as a set of network partitions that join and split over time. *Network partitioning* is the split of the network into two (or more) disjointed groups of nodes that can not communicate with each other. We refer to these groups as *network partitions*. In the simplest case, a partition may consist of only one isolated node. A partition is uniquely identified by its constituting nodes and the time that the partition is formed. *Partition join* is the combination (coalescing) of two (or more) partitions into one bigger partition. Partition join occurs when two partitions come within the communication range of each other.

Due to the frequent network partitioning in MANETs, we define the network connectivity over a certain observation time interval T_{obs}: Two nodes are said to be connected in T_{obs}, if both nodes belong to the same partition for some time in T_{obs}. We refer to this connectivity as *connectivity-in-time*. If the length of T_{obs} is 0, the connectivity-in-time goes into a *connectivity-in-space*. Two nodes are said to be connected-in-space at a certain time t, if they belong to the same partition at time t.

Chapter 3

Related Work

In this chapter, we first introduce our taxonomy for the existing research work on MANET broadcasting. We then briefly review the existing broadcasting techniques and discuss their performance by pinpointing the scenarios where the techniques perform well and those where they show deficiencies.

3.1 Taxonomy

We follow a taxonomy that takes into consideration the MANET characteristics on the one hand and the application requirements on the other. With regard to broadcasting, we observe two broad classes of MANET characteristics and two further classes of application requirements. Concerning MANET characteristics, we distinguish between the non-partitioned MANETs and the partitioned ones. Concerning application requirements, we distinguish between the applications that are delay-critical and those that tolerate higher delays in the range of minutes or even hours or days.

Solutions for delay-critical applications in partitioned networks (Q1 in Fig. 3.1) are only possible with a poor reachability due to the infrastructureless nature of MANETs. Since high values of reachability take priority in broadcasting, we conclude that solutions in Q1 are impossible.

To the best of our knowledge, no broadcasting strategy has been designed in Q4, i.e. a strategy that exploits delay-tolerance of applications in non-partitioned networks, although such a strategy would help to increase the capacity of the network. In [8], the authors provide the theoretical foundations for such solutions. [36] showed that the per node throughput in a multihop network drops to zero for a large number of nodes. However, the authors in [8] showed that the per node throughput becomes $O(1)$, independent of the number of nodes. This capacity is achieved as relay nodes store the message and forward it on encountering the destination. The gain in network capacity comes at the cost of the unpredictable end-to-end delay. The lack of broadcasting techniques in Q4 is possibly due to the volatile nature of node mobility in MANETs, where the nodes move in an unpredictable way.

In the literature we identify two classes of solutions for broadcasting. The first class of solutions is designed for delay-critical applications in non-partitioned MANETs (Q2). These solutions however show poor performance in partitioned MANETs. The second class of solutions is developed for delay-tolerant applications in highly partitioned networks (Q3).

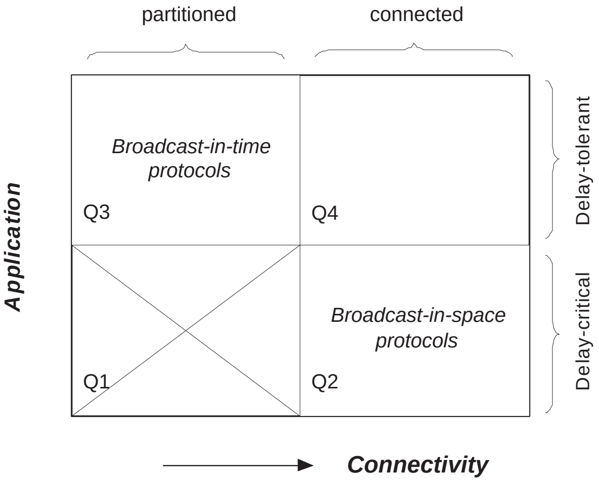


Figure 3.1: Taxonomy of related work

The first class of related work includes a large number of broadcast protocols that have been developed for connected MANETs. These protocols are based on flooding, where nodes forward messages after a very short delay in range of few milliseconds (cut-through principle). Therefore, these protocols are suitable for delay-critical applications, where the message has to be distributed within milliseconds or a few seconds to all network nodes. These strategies take advantage of only the connectivity-in-space. As mentioned in Section 1.1, we refer to this broadcast paradigm as *broadcast-in-space.*

The second class of related work is composed of protocols that cache broadcast messages and forward them through node mobility on favorable encounters with other nodes. These solutions rely on the general store-and-forward concept. Since messages need longer time periods in the range of minutes or even hours (depending on the movement patterns) to reach the destination, we refer to this broadcast paradigm as *broadcast-in-time.* The applications using these protocols have to be delay-tolerant. Applications should tolerate delay values that allow for the connectivity-in-time of the MANET. Broadcast-in-time protocols perform well in highly partitioned networks. Unfortunately, they show a poor performance in connected ones due to their high message overhead.

The following two sections present a brief review of existing solutions for both of these classes and discuss their pros and cons.

3.2 Broadcast-in-space Protocols

Most of the existing broadcast-in-space solutions are strategies to reduce the broadcast storm problem [7]. These approaches are designed for delay-critical applications. In this section, we classify these protocols and briefly review them. Finally, we discuss their main advantages and shortcomings by summarizing the results of the existing comparative studies.

3.2.1 Classification

In the following, we present a skeleton solution for the broadcast storm problem, on the basis of which we classify the broadcast-in-space protocols.

Solution Skeleton for Broadcast Storms

Broadcasting in ad hoc networks has traditionally been based on flooding. To relieve the broadcast storm problem protocol designers developed mechanisms to reduce collision probability and reduce redundancy. The most effective network-wide broadcasting protocols try to limit contention and the number of collisions by limiting the number of forwarding nodes in the network.

The following general approach can be deployed to realize these strategies. When a node n receives the first copy of a broadcast message, n sets a timer. When the timer expires, the node determines whether to forward the message or not. After the forwarding decision, nodes should simply ignore duplicate messages. There are two design issues with the approach above. The first issue is how should nodes set the waiting time (timeout). The second issue is how should nodes decide whether to forward the message or not.

Waiting Time: The first design issue seems relatively simple. This issue determines the waiting time until the forwarding decision. Protocol designers may exploit this issue to assign different values to different nodes. We distinguish four objectives to assign different values.

The main objective of different values is to avoid simultaneous forwarding at neighbor nodes, in order to reduce the probability of a collision. One simple solution that is realized in most broadcast-in-space protocols is a random waiting time, called *Random Assessment Delay (RAD)* .

A second objective is to influence the forwarding time depending on the estimated additional coverage. In [71], the authors suggest fixing the timeout value depending on the distance to the sender. The idea behind this is to allow a node covering more new area to forward the message earlier than a node covering less new area. In [72], the authors propose a method to dynamically adjust the RAD based on a node's relative neighbor degree (ratio of node's degree to the maximum degree of all its neighbors).

A third objective is to control the time the packet remains buffered at the network layer. Until the forwarding decision is taken, the broadcast packet remains cached at the network layer, before it is either delivered to the MAC layer for forwarding, or purged. For example, the authors of [31] introduced a simple adaptation of this timeout value to the network load level.

A fourth objective is to allow sufficient time for the forwarding decision component to decide, whether or not to forward, for example by counting redundant messages.

Forwarding Decision: The second design issue aims at prohibiting nodes from forwarding a message if the forwarding would be redundant. Most of the schemes presented in the literature are called according to the strategy used to realize this second design issue. The classification of broadcast-in-space protocols is also based on this issue.

Classification

In [32], the authors classify broadcast-in-space schemes into heuristic-based (also called threshold-based) and topology-based approaches. The authors in [73] identify a third class, the class of energy-efficient protocols. [31] sub-classifies the heuristic-based class into (plain flooding,) probability-based and area-based (Fig. 3.2).

Some of the broadcast-in-space techniques have been adapted to some local MANET characteristics such as node density. Therefore, we start to review the existing non-adaptive protocols and then discuss the few adaptive protocols.

3.2.2 Review of Existing Protocols

In the following, we first review some representatives of non-adaptive broadcast-in-space protocols and then the few adaptive protocols known from the literature.

Non-Adaptive Broadcast-in-space Protocols

Now we review the non-adaptive broadcast-in-space protocols belonging to the heuristic-based, topology-based and energy efficient classes. Finally, we discuss the performance of these protocols.

- Heuristic-Based Approaches: This class covers the probability-based and the area-based protocols.

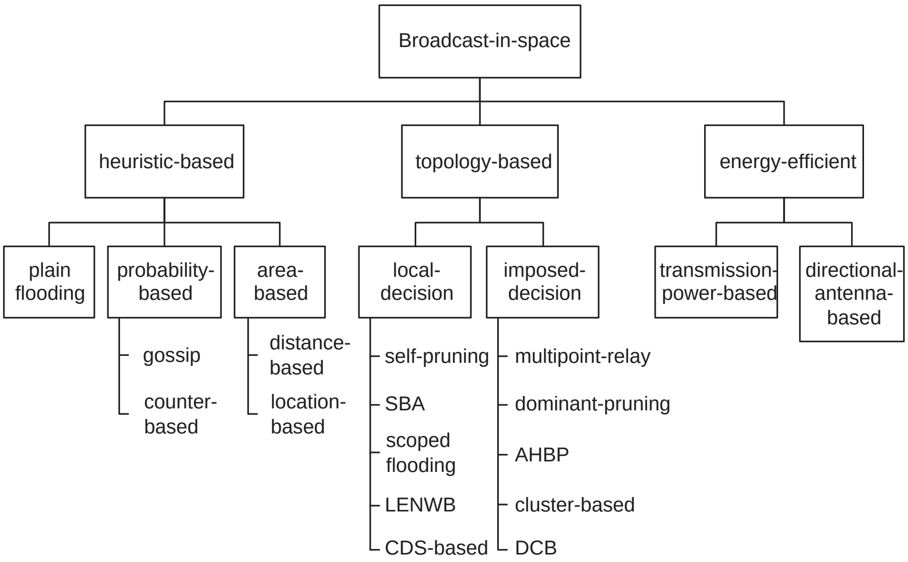


Figure 3.2: Classification of broadcast-in-space protocols

`Probability-Based Approaches:`

Two probabilistic approaches have been presented in the literature: Gossiping and counter-based.

In *gossiping* [7,74], each node forwards a message with a certain probability p and drops the message with a probability of $1 - p$. Therefore, gossiping is also known as the probability-based flooding.

In the *counter-based* scheme [7], before forwarding the first copy of a message, a mobile node initiates an RAD and counts the number of received copies of the current message. When the RAD expires, the node only forwards the message if the counter does not exceed a threshold value C_{th}. Otherwise the forward is canceled. The basic idea behind the counter-based scheme is that the more copies a node receives, the higher the chance that its neighbors have already received the same message, and the more likely that a forward is redundant.

`Area-Based Approaches:`

A node following an area-based approach estimates the additional coverage area obtained, if the message would be forwarded and forwards the message if the estimated area is higher than a fixed threshold. In the literature, two area-based schemes have been proposed in [7]: The distance-based and location-based techniques.

A *distance-based* forwarding decision is taken, based on the distance of the node itself to all its neighbors that have forwarded the message. The distance is calculated from the position of the node itself and the position of the forwarding neighbors indicated in the packet header. Alternatively, the authors of [7] proposed to use signal strength to approximate distances. Nodes only forward a message if the distance to all nodes from which they received the message, is larger than a certain threshold for the distance value.

The *location-based* approach estimates the additional coverage area more precisely and instead of using the distance of nodes as a measure for the additional area covered, the authors of [7] proposed a method, which directly calculates the expected area covered, from the positions of previous senders. This calculation of the additional coverage becomes complicated when several copies of the same packet are received. The location-based approach also requires the assistance of location devices for every node, which is hard to realize in many MANET applications.

- Topology-Based Approaches: We identify two sub-classes: The *local-decision-based approaches* and *imposed-decision-based approaches*. In local-decision-based approaches (also called receiver-based approaches, or reactive approaches), each node determines on its own, whether or not to forward a broadcast message. However, in imposed-decision-based approaches (also called sender-based approaches, proactive approaches, or neighbor-designated approaches), the forwarding decision is imposed by other nodes such as the previous relay node or the cluster-head.

  `Local-Decision-Based Approaches`

  The basic idea of these approaches is that a node exploits neighborhood connectivity and history of the nodes that the message has already visited, in order to decide on its own, if it is a forward node or not.

  The authors of [75] proposed a generic scheme that covers most existing local-decision-based approaches. The scheme is based on two conditions, namely on neighborhood connectivity and history of the nodes already visited. Each node builds information about its k-hop neighborhood by exchanging (k-1)-hop information with its 1-hop neighbors, by means of periodic HELLO messages. Information about a node's property, such as ID or node degree, and a list of nodes already visited is added to the broadcast messages. Based on this information a node decides whether or not to forward a message.

  The simplest local-decision-based method is *flooding with self-pruning* [76] or the *neighbor coverage scheme*. The sender piggybacks a list of its 1-hop neighbors on each transmitted broadcast message and a receiver only forwards the message immediately if it can cover some additional nodes to those of the sender.

The *scalable broadcast algorithm* (SBA) [72] uses the same forwarding strategy as the neighbor coverage scheme, with the following two main differences. First, nodes insert the list of their 1-hop neighbors to HELLO beacons and not to data messages. Secondly, nodes do not forward immediately, but initiate an RAD. For each neighbor that forwards during the waiting period, the node re-calculates its additional coverage. If the RAD expires and the additional coverage set is not empty, the node is a forward node. In SBA, RAD is not constant, but it is adapted depending on a node's relative neighbor degree.

Scoped flooding [6, 9, 10] is a variant of the SBA protocol, which severs the forwarding condition. Upon the expiration of RAD, a node forwards the message, if more than a fixed ratio of its neighbors are still not covered. The authors of [10] propose to use 15% as the ratio value. Scoped flooding behaves like SBA for 0% ratio value.

The *lightweight and efficient network-wide broadcast* (LENWB) [77] is similar to the neighbor coverage scheme, but nodes acquire 2-hop neighborhood information by the periodic sending of HELLO beacons that contain the list of 1-hop neighbors. Upon receiving a broadcast message from a sender, the receiver computes the coverage of its 1-hop neighbors that received the message and have a higher node degree. Only if all receiver's neighbors are covered by higher degree nodes, is the forward canceled.

Several of the local-decision-based approaches are based on *connected dominating sets* (CDS). [78] proposed an algorithm, which only requires 2-hop neighbor information. A node belongs to the dominating set if two unconnected neighbors exist. Only nodes that belong to the CDS forward the message. Unlike [78], in [79] 1-hop neighbor information is sufficient if nodes are aware of their positions in order to determine if two neighbors are connected . Under the assumption that each node knows its accurate position, connected dominating sets and the concept of planar subgraphs are used in [80] to reduce the communication overhead for broadcast messages.

Imposed-Decision-Based Approaches

The basic idea is that each node selects a subset of its 1-hop neighbors for forwarding the message, such that all 2-hop neighbors can be reached by this subset.

In *multipoint relay* (MPR) [81], nodes insert the list of their 1-hop neighbors into their HELLO beacons, so that nodes are aware of their 2-hop neighborhood. The sending node selects forwarding nodes from its 1-hop neighbors, so that all 2-hop neighbors are covered by the set (the selection rule is defined in [81]) . Nodes piggyback the forwarding list in their HELLO beacons. Only nodes in this list forward broadcast messages.

Like MPR, nodes in *dominant-pruning* [76] acquire 2-hop neighbor knowledge using HELLO beacons, and senders select the designated forwarders using the same MPR rule. Unlike MPR, receivers select the forwarding set depending on MPR selection rule and additionally depending on the knowledge of which neighbors have already been covered by the senders broadcast. The forwarding set is selected from the 1-hop neighbors that are not neighbors of the previous relay. The forwarding list of the same node may therefore differ from message to message. The forwarding list is piggybacked on the broadcast message. In [82], Lou and Wu present total dominant pruning and partial dominant pruning, two improvements that utilize neighborhood information more efficiently.

The *ad hoc broadcast protocol* (AHBP) [83] is similar to MPR, but piggybacks the forward designation onto the broadcast message. Nodes that receive a broadcast message from a node that is not listed as a neighbor, always forward the message.

In *cluster-based* schemes, the decision is imposed by the cluster formation algorithm. The general idea of cluster-based schemes has been introduced by Li et al. in [7]. The proposed scheme in [7] assumes that clusters have been formed in the MANET and are maintained regularly by the underlying cluster formation algorithm. It also assumes that the cluster head's forwarding covers all nodes of that cluster. The cluster-based scheme proposes that only cluster heads and gateway nodes (nodes that can communicate with nodes from other clusters) forward the broadcast messages using any broadcast technique such as gossiping. Further cluster-based broadcast protocols have been defined in [84, 85].

In [86], the authors propose a broadcast algorithm, called *double-covered broadcast* (DCB). The sender selects forwarding nodes in such a way that first the sender's 2-hop neighbors are covered and secondly the sender's 1-hop neighbors are either a forward node, or a non-forward node but covered by at least two forwarding neighbors. The retransmissions of the forward nodes are received by the sender and serve as an acknowledgement. If the sender does not detect all its forwarding nodes' retransmissions, it will resend the packet until the maximum number of retries is reached. Simulation results show that DCB provides good performance for a broadcast operation under a high transmission error rate environment.

- Energy-Efficient Approaches:

  The authors in [73] classify the existing power-efficient broadcast techniques into transmission-power-based approaches and direction-antenna approaches.

  `Transmission-Power-Based Approaches:`

These protocols adjust the radio range to realize an efficient network-wide broadcast. In [87], the authors proposed a *broadcast incremental power* (BIP) algorithm that constructs a tree starting from the source node and adds a node in each step, which is not yet included in the tree, but which can be reached with minimal additional power from one of the tree nodes. [88] considered the minimum energy broadcasting problem and proposed a localized protocol, where each node only requires the knowledge of its own position and those of its 1-hop neighbors. The algorithm presented in [89] constructs a static routing tree, which maximizes network lifetime by accounting for residual battery energy at the nodes. The algorithm however, does not really maximize the possible network lifetime, if nodes are mobile. [90] presented a distributed topology control algorithm, which extracts network topologies that increase network lifetime by reducing the transmission power.

`Direction-Antenna Approaches:`

Directional antennas are used to improve the performance of broadcasting by reducing interferences, contention, etc. It was shown in [91] that MAC protocols, which utilize directional antennas can improve the performance of broadcast traffic in ad hoc networks. In [92], each node is assumed to have a beam-width of 90 ° and packets are only forwarded in the 270 ° direction from that in which the packet arrived. If nodes are aware of their neighborhood through HELLO messages, nodes may explicitly send the packet to nodes that are farthest from the current node. In [93], directional antennas are used to transmit broadcast packets to all neighbors in a connected planar subgraph of the complete network graph, namely the relative neighborhood graph.

Discussion of Non-Adaptive Protocols: Due to the increasing number of broadcast-in-space techniques, several performance comparison studies have been conducted. A comparative study for some representatives of heuristic-based and topology-based protocols can be found in [31]. The authors of [32] presented a comparison of the performance of some topology-based protocols in ad hoc Networks. In [33], the authors compare the performance of some broadcast protocols based on self-pruning. A comparison of power-efficient broadcast algorithms has been presented in [34]. A comparison study of the performance of various directional antenna algorithms is provided in [94]. The main conclusions of these performance comparisons can be summarized as follows.

Roughly speaking, non-adaptive broadcast protocols show for some selected scenarios a good performance providing for high reachability and efficiency. However for other scenarios they show significant deficiencies.

Non-adaptive heuristic-based protocols use heuristics with predefined fixed thresholds to reduce broadcast storms. They do not adapt to the MANET situations, which vary over time and space and show quite different levels of broadcast storms. This means these algorithms are not flexible enough to cope with a wide range of network scenarios. They are very sensitive to the chosen threshold value and may perform well in some scenarios and very poorly in others.

Non-adaptive topology-based show high reachability and save many forwards in stable network topologies, i.e. if nodes are static or move very slowly. However these protocols require accurate topology information, which is hard to acquire in the case of frequent topology changes or frequent collisions. That is why the performance of these protocols suffers significantly in congested networks, as the topology information becomes inaccurate and in highly mobile networks, as the frequent topology changes induce an excessive or even prohibitive amount of control traffic [31]. On the other hand, it was shown that heuristic-based approaches are more robust against frequently changing topologies.

Some techniques such as CDS-based and cluster-based do not consider the history of the nodes already visited to determine forward nodes. This implies that the relay nodes are the same for all broadcast traffic as long as the network topology does not change. This may lead to the abuse of certain nodes, which may run out of energy faster. On the other hand, these protocols may let non-forwarding nodes switch to sleep mode without affecting the overall network operation, thus prolonging the network lifetime.

Energy-efficient approaches construct a power-efficient network structure, which requires a large computational and communication overhead. This overhead may be beneficial in a static network, where the structure has to be established only once. In a mobile network however, it may be either impossible to maintain this structure at all or only with an intolerable amount of energy consumption. Since, we assume that nodes do not require specific capabilities such as directional antennas or the capability to dynamically adjust the communication range, we will not consider the energy-efficient approaches in the remainder of this thesis.

The lack of determinism is inherent in MANETs, inducing the deployment of probabilistic-based and randomized design to combat indeterministic topologies. Gossiping is the simplest flooding scheme and is topology-independent. Although gossiping does not consider the nodes previously visited, it does not abuse certain nodes by forwarding, because of its probabilistic nature.

Adaptive Broadcast-in-space Protocols

Some non-adaptive broadcast-in-space techniques have been adapted to local MANET characteristics. The basic idea of adaptive topology-based approaches is to manage node mobility better, for the purpose of avoiding stale topology information [23, 35]. Adaptive heuristic-based protocols however, adapt the heuristics to the number of neighbors [23, 24].

`Adaptive Topology-Based Protocols:`

In [35] authors proposed to use two different communication ranges in order to cope more efficiently with mobility. One range for the topology management (determination of forwarders) and another for the data transmission. They recommend selecting a shorter range for topology management and to adapt the difference between the two ranges to node mobility, which requires speed information. They further proposed a mechanism to ensure consistency between the different views of different nodes on the network. In [23], the authors proposed one adaptive topology-based scheme, called the adaptive neighbor-coverage scheme. The authors adapted the neighbor-coverage scheme by dynamically adjusting the HELLO interval to node mobility reflected by neighborhood variation, so that the required 2-hop topology information is more accurate. Despite these optimizations, the adaptive topology-based schemes still have the main drawback that neighborhood information may be inaccurate in congested networks.

`Adaptive Heuristic-Based Protocols:`

In [23] the authors also proposed two adaptive heuristic-based schemes, called adaptive counter-based (ACB) and adaptive location-based (ALB). By means of simulations, the authors derived the best appropriate counter-threshold and coverage-threshold for ACB and ALB respectively, as a function of the number of neighbors. The authors showed that these adaptive schemes outperform the non-adaptive schemes and recommend ACB if location information is unavailable and simplicity is required. We will compare our protocols to ACB in Sections 4.5.7 and 5.5.5. Cartigny et al. [24] adapted the forwarding probability of gossiping to the local number of neighbors n and called their adaptive scheme stochastic flooding (STOCH-FLOOD). Nodes use the following forward probability: $p = min(1, 11/n)$. We will also compare our protocols to stochastic flooding in Sections 4.5.7 and 5.5.5.

Discussion of Adaptive Protocols: Adaptive topology-based and heuristic-based schemes are shown to outperform non-adaptive broadcast-in-space schemes. ACB, ALB, adaptive neighbor-coverage and stochastic flooding support a broad range of node densities and speeds, however they show poor reachability in partitioned networks. In [24], the authors adapted gossiping experimentally. We show how to use analytical models to adapt this scheme.

3.3 Broadcast-in-time Protocols

In partitioned MANETs nodes may not be able to communicate if they belong to different partitions. In this case message broadcasting is no longer trivially achievable by simple flooding. The main research focus of broadcast-in-time protocols is on highly partitioned networks, where the applications should be delay-tolerant and the broadcasting strategies should exploit the mobility of nodes to overcome network partitioning.

The first approach to cope with this situation is based on handshake procedures. Nodes use handshake mechanisms to 'say' which messages they already received, so that other nodes carrying other messages can forward missed message to them. These strategies are grouped under the *negotiation-based* class.

The second approach is *repetitive-flooding*. A flooding phase is likely to stop before all nodes are reached. Repeating the message transmission over a period of time, which is referred to as hyperflooding [6, 9, 10], can help to cope with network partitions.

3.3.1 Negotiation-Based Protocols

The general idea behind these protocols relies on a handshake procedure. Nodes exchange advertisement messages (ADV), reply with data request messages (REQ), and finally exchange the appropriate DATA. These protocols are shown to be appropriate for highly partitioned and low mobility networks. In the literature we find the following schemes: SPIN-based protocols [21] and NADD [95].

- *SPIN-Based Protocols:* In [21], the authors presented *Sensor Protocols for Information via Negotiation* (SPIN). This protocol family is based on a three-way-handshake mechanism. We will investigate this protocol in more details in Section 4.3.

- *NADD Protocol:* In [95], the authors presented a *Negotiation-based Ad hoc Data Dissemination protocol* (NADD), a protocol for data dissemination in frequently partitioned MANETs. The protocol implements a three-way-handshake mechanism, in which a node advertises the IDs of a subset of locally stored messages to all its neighbors. Receivers of the advertisement message reply with a request message, where they indicate the IDs of the messages they have missed. The advertising node can then transmit the requested data. The advertising is triggered by the reception of the first copy of a message or by the discovery of a new neighbor. Since the number of cached messages becomes very large for large update rates, the authors discussed different advertising and selection strategies for the data to be advertized.

Negotiation-based protocols are robust against network partitioning since they significantly increase the delivery ratio in highly partitioned networks. By communicating with each other about the messages they still need to obtain, nodes are better able to cope with network partitioning. Due to the robustness of handshake mechanisms against network partitioning, we will utilize them to realize our generalized broadcasting technique.

The main shortcoming of the negotiation-based protocols is that they are tailored for highly partitioned networks and low mobility networks. Therefore, they are less efficient in connected or highly mobile MANETs, compared to the broadcast-in-space protocols. For very dense networks the negotiation-based protocols lead to a higher message overhead and longer delivery delay.

Negotiation-based protocols show high end-to-end delays (NADD shows delays up to some hours [95]). Therefore, these protocols are well suited to delay-tolerant applications. In contrast, they are unsuitable for delay-critical applications.

3.3.2 Hyperflooding

In hyperflooding [6, 9, 10], nodes store messages for a fixed time period and rebroadcast them on discovering new neighbors. The negotiation feature is missing in this approach, which presents one of the main shortcomings of hyperflooding.

Intuitively, this solution relies on a highly primitive partition join detection mechanism, which we expect to perform very poorly in highly mobile and in dense networks. Furthermore, the caching strategy allows each node to shortly cache each newly received message independently from the time until next encounter of unreached nodes. This makes the solution inefficient for highly partitioned and low mobility networks, where the caching time is not long enough until the encounter of unreached partitions.

Simulation results show that hyperflooding is suitable for scenarios, where the network is partitioned but connectivity-in-time is achieved in a short time. These scenarios are a kind of grey-zone with respect to network connectivity. Hyperflooding shows a higher end-to-end delay than broadcast-in-space algorithms, but not as high as negotiation-based protocols.

3.4 Integrated Flooding (IF)

The first step towards a generalized broadcasting solution in MANETs was the integrated scheme presented in [9, 10]. We refer to this scheme as Integrated Flooding (IF).

Nodes switch between three flooding schemes at run-time, namely, plain flooding, scoped flooding, and hyperflooding. The authors recognize mobility as the main cause of broadcast partitioning [96] and switch between these schemes according to the relative node mobility [9]. To this end, nodes include velocity information (speed and direction) in HELLO beacons. If a node's current value of relative velocity to its neighbors is higher than a high_threshold, the node switches to hyperflooding mode. If the relative velocity is below a low_threshold, scoped flooding is used. Otherwise, the node switches to plain flooding. Alternatively the same authors suggest in [10] that IF switches between the three schemes based on network load. The authors use MAC layer collisions as an indicator of network load. If a node detects a current number of collisions higher than a high_threshold, the node switches to scoped flooding mode. If the number of collisions is lower than a low_threshold, hyperflooding is used. Otherwise, the node switches to plain flooding. However, the authors do not mention how to combine both switching criteria. Furthermore, these switching criteria may lead to opposite decisions. For example, in low mobility networks with low traffic, the relative velocity based switching would install scoped flooding, but the network load based switching would install hyperflooding. For these reasons and because we mainly focus on a wide range of network conditions concerning node density and node mobility, and are less concerned with network load, we only consider the relative velocity based switching criteria for IF in this work.

To the best of our knowledge, IF is the single existing adaptive MANET broadcast protocol that considers both connected and partitionable networks. Unfortunately, IF shows the following three drawbacks. Firstly, hyperflooding deploys a very simple broadcast repetition strategy, i.e. on each discovery of a new neighbor, all cached packets are rebroadcasted. If the buffering time of packets is high, the strategy leads to a high number of useless and costly rebroadcasts in highly mobile networks. If the buffering time is low, the strategy shows a poor delivery ratio in low mobility partitioned networks. Secondly, scoped flooding uses a predefined forward threshold, which makes IF less efficient than the adaptive heuristic-based and topology-based schemes in highly dense scenarios. Thirdly, relying on velocity information presents a strong limitation of the deployment of the IF protocol.

IF is the closest work from the literature to ours. We will compare our solution to IF in Section 5.5.5.

3.5 Conclusions

The common drawback of the existing broadcast-in-space techniques is that they show a poor reachability in partitioned ones, since broadcasts reach only the nodes of the partition containing the broadcast source. Comparative studies also show that these schemes are optimized for specific scenarios and do not support a broader range of MANET situations.

Therefore, some research work has been conducted in order to adapt some of the existing schemes to local MANET characteristics such as the number of neighbors [24] [23] or the neighborhood change rate [23]. The adaptive schemes support a broad range of connected MANETs. However, they still show a poor delivery ratio in partitioned networks.

Broadcast-in-time approaches are tailored for partitioned MANETs and are outperformed by the broadcast-in-space algorithms in non-partitioned scenarios.

Integrated flooding is the first strategy, whose objective is a generalized solution for both partitioned and connected MANETs. The IF approach presents solutions to increase delivery ratio in partitioned parts of the MANET and to relieve the broadcast storm problem in connected parts. Unfortunately, this strategy provides a poor partition join detection method, as we are going to show in Chapter 5, which makes the IF protocol perform poorly in low mobile and highly partitioned, and in connected and highly mobile networks.

In Fig. 3.3, we roughly depict our related work in the density-mobility-delay sphere.

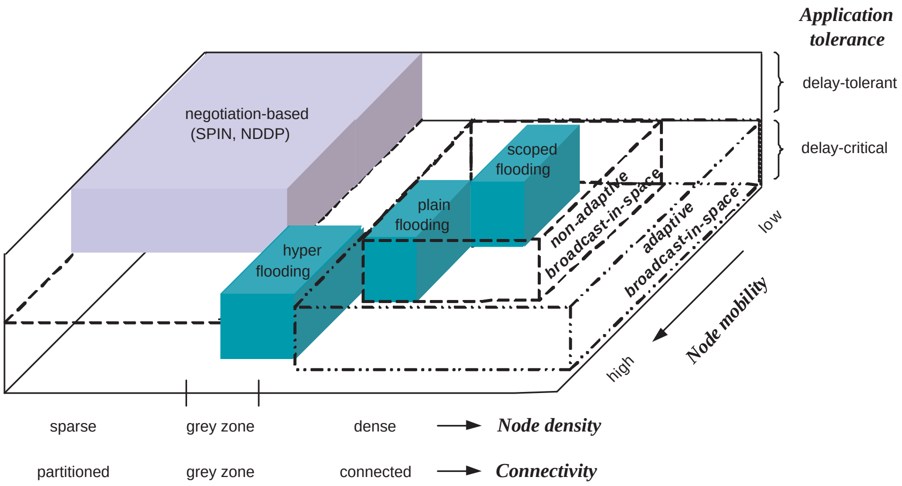


Figure 3.3: Related work in the density-mobility-delay sphere

Summarizing, different broadcasting strategies are suitable for different situations. However, there is no single strategy, which performs well in the complete depicted space with respect to node density and node mobility. This emphasizes the strong need for a strategy that is suitable for the entire scope of desired parameters. In Chapter 5, we introduce a technique that covers a wide range of node densities, speeds and network loads, and which supports both delay-critical and delay-tolerant applications. We call our technique *hypergossiping*.

Chapter 4

Analytical Modeling and Adaptation of Broadcast Protocols

After a short motivation, we give an overview of our approach to provide analytical models for MANET broadcasting. Then, we show how to adopt the so-called SI mathematical compartmental model from epidemiology to analytically model broadcast protocols in MANETs. For this purpose, we first consider the SPIN-based broadcast-in-time protocol [21] as example. Furthermore, we calibrate the SI model for gossiping and use the results to adapt the gossiping forwarding probability to node density. Afterwards, we evaluate adaptive gossiping and compare it to related work. Finally, we briefly summarize the results of this chapter.

4.1 Introduction

Many researchers analytically model the message spreading among network nodes for fundamental purposes, such as for understanding the spreading mechanism (performance evaluation) and for protocol adaptation. A further use of this analytical modeling is the design of counter measures if a fast message spreading is not desired, such as in the case of computer worms and viruses. Since the message spreading among network nodes is similar to the spreading of infectious diseases, some researchers denote their protocols to be epidemic and rely on mathematical models from epidemiology to establish analytical models for message propagation.

Examples for the use of the mathematical epidemiology outside the biological field are the modeling of how ideas propagate [97] and the analysis of the algorithms designed for maintaining replicated databases [98]. IBM Research also applied mathematical epidemiology to model the spreading of computer viruses on the Internet and to derive anti-virus strategies [99].

In MANETs nodes are carried by mobile users or by mobile objects such as human beings, animals or vehicles. The MANET population is similar to the populations considered in epidemiology, composed of different species of human beings or animals. The message spreading among MANET nodes can be seen as a 'message disease'. The difference between the infectious diseases consists mainly of how the infection is stored and transmitted to other individuals. Similarly, the difference between the MANET broadcasting techniques consists of how the message is processed and forwarded to other nodes.

These observations show that mathematical epidemiology may play a major role in modeling and designing broadcasting techniques for MANETs. In this thesis, we show how to adopt the results of mathematical epidemiology to the field of broadcasting in MANETs.

4.1.1 Mathematical Epidemic Models

Mathematical modeling is an essential tool in studying infectious diseases. Basic aims in studying the disease spread, are to gain a better understanding of transmission mechanisms and those features that are most influential in that spread, so as to enable predictions to be made, and to determine and evaluate control policies. Thus mathematical models have a particularly important role to play in making public health decisions, to ensure the control of infectious diseases is better informed and more objective. To cite a successful example, adopting a culling proportion calculated based on a mathematical model, Foot-and-Mouth disease, in the UK in 2001 was controlled successfully. With the outbreak of SARS in 2003-2004 and avian flu in 2005-2006, epidemic modeling has taken on even more significance from the perspective of public health and policy making.

To model a given disease, it is important to understand the course of infection within an individual and the patterns of infection between individuals. Based on the compartmental analysis, many mathematical models have been proposed (see for example [19, 20]). With respect to a given disease, one individual is mapped to one of the following states (also called compartments): Susceptible (S), Infectious (I), Removed (R), Exposed (E), or iMmune (M). Depending on the disease, some of these states may be not considered. A series of deterministic compartmental models have been defined in the epidemiology literature based on the flow patterns between the above compartments. The acronym of the model describes these flow patterns such as: SI, SIS, SEI, SEIS, SIR, SIRS, SEIR, SEIRS, MSEIR, MSEIRS. For example in the MSEIRS model, immune individuals first become susceptible, then exposed, then infectious, then removed with a temporal immunity, and finally susceptible again. The simplest model is the *SI model*, where individuals once infected, remain infectious forever. With this model every individual can be infected. After an individual was infected it begins to infect other individuals. In the long term all individuals are infected.

4.1.2 Objectives

Existing mathematical models that describe the epidemic spreading process can be as useful for us as they are for medical researchers. Medical researchers use epidemic models both to describe the spread of disease within a population and to take preventive or treatment measures. We use epidemic models both to *describe* and to *adapt* broadcasting in MANETs.

With regard to broadcasting, protocol designers are interested in understanding the nature of the spreading depending on the *protocol parameters* and on the *MANET properties.* The quality of broadcasting can be expressed in the spreading progress, both in time and in space. In this work, we focus on the spreading progress in time. We define for a given message the *spreading ratio* at time t as the ratio of the number of nodes that received the message up to time t to the total number of nodes N. Let $i(t)$ denote the spreading ratio at time t, with $0 \leq i(t) \leq 1$. The most relevant factors which affect the characteristics of message spreading are the broadcast protocol and the network connectivity over space and time. The network connectivity over space and time is mainly determined by the node spatial distribution, node mobility, communication parameters (e.g., transmission range and rate), and the total number of nodes.

To obtain the spreading ratio i over time t for a given broadcast protocol and a given MANET configuration, *simulations* can be used. This requires an appropriate simulator that models the MANET configuration in a sufficiently accurate way. The broadcast protocol should then be implemented in this simulator. Simulations can be run for different MANET properties and protocol parameters, to understand the impact of these parameters on the broadcast protocol performance, mainly given by the spreading ratio $i(t)$. Simulations can only be done for discrete values of the parameters of interest. Simulations could also be time-consuming, which limits their use for large-scale scenarios. Simulation results are usually provided as data sets, e.g., $table[time\ t_i \mid spreading\ ratio\ at\ t_i]$ (Fig. 4.1). Consequently, simulation results are difficult to manipulate and administrate, since they can not be easily generalized. Furthermore, each change in the MANET or protocol parameters requires running new simulations to obtain the corresponding spreading ratio.

Analytical models however provide the spreading ratio as a mathematical expression, e.g. $spreading\_ratio = i(t)$, which represents an elegant way to describe the spreading ratio over time i(t). If i(t) can be expressed as a function of the relevant MANET properties and protocol parameters, the performance of the corresponding broadcast protocol is available with minimal cost for arbitrary parameter values (Fig. 4.1). However, analytical models are hard to develop and especially for MANETs, given their indeterministic and unpredictable nature. Our approach for analytically modeling broadcast protocols in MANETs consists in adjusting existing mathematical models from the epidemiology to MANET broadcasting as we are going to discuss in details in Sections 4.2 and 4.3.

Since MANETs show a wide range of operating conditions, the adaptation of the broadcast protocols on-the-fly is desired. We show how an adopted analytical epidemic model can simplify the adaptation of broadcast protocols. In this work, we use the SI model to adapt the forwarding probability of gossiping to the local

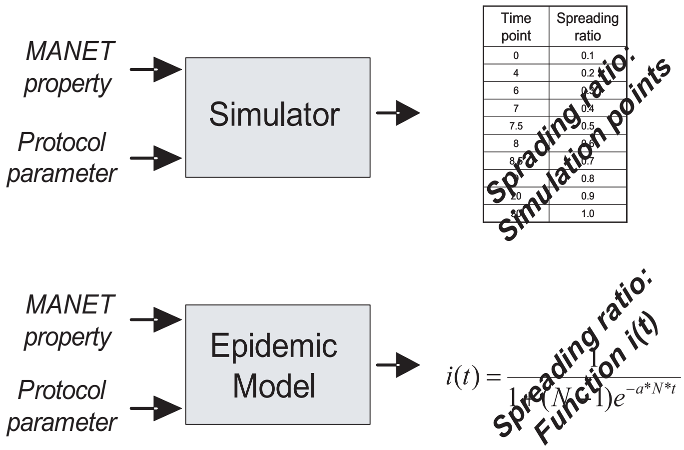


Figure 4.1: Simulation versus analysis

MANET density.

4.2 Overview of our Approach for Analytical Modeling

Our approach to develop analytical models for broadcast protocols is based on the adjustment of the established mathematical models from the epidemiology to MANET broadcasting. In this work, we focus on the SI model and show in Section 4.3 that this model is suitable for modeling the SPIN-based broadcasting strategy and for modeling and adapting gossiping in Section 4.4. We show step by step how to adopt the SI analytical model for the SPIN-based protocol.

In general, a model depends on a few parameters whose values should be determined by observations. Table 4.1 gives the parameters of the SI model and the corresponding implications for broadcasting in MANETs.

In the SI model the entire progress of $i(t) = \frac{I(t)}{N}$ can be described by an analytical expression that only depends on the total number of nodes N, time t and on an important model parameter, which is the infection rate a. The infection rate gives the number of the new infections per time unit which are caused by an infectious node. We point how one basically can calculate the infection rate analytically from the mobility model and the broadcast properties. Nevertheless, we will not follow this approach and proceed, instead, similarly to the epidemiologists who often calibrate their models using data collected from a real epidemic outbreak.

In this work, we obtain the observations needed for the calibration of the model using simulations. First, we evaluate the performance of the considered broadcast algorithm in the chosen MANET scenario using simulations. In particular, we

	Epidemiology	**Broadcasting**
N	Number of individuals	Number of nodes
$S(t)$	Number of susceptible individuals at time t	Number of nodes that are interested in the message at time t
$I(t)$	Number of infectious individuals at time t	Number of nodes that carry the message at time t
e	Contacts per unit of time and per individual	Encounters per unit of time and per node
β	The probability of transmission in a contact between an infective and a susceptible individual	The probability of transmission in an encounter between a node that carries the message and one other that is interested in the message
$a = \beta \frac{e}{N}$	Infection rate	Infection rate
$\alpha = aI(t)$	Force of infection: The probability per unit of time for a susceptible to become infective	Force of broadcast: The probability per unit of time for a node to receive the message

Table 4.1: Epidemiology versus broadcasting

focus on the spreading ratio. Then, we fit the points obtained to the analytical expression provided by the SI model, in order to identify the corresponding infection rate for the chosen protocol and MANET configuration (Fig. 4.2). Our methodology is generic and can be used similarly for other broadcast protocols and the corresponding epidemic models.

Since the *node density* strongly impacts the performance of broadcast protocols [7], the modeling of a broadcasting strategy should account for the density of nodes. Therefore, we explore the impact of node density on the infection rate a. Repeating the calibration procedure for some node densities and interpolating the points (density, infection rate), we derive an analytical expression for the infection rate depending on the node density.

As stated before, the dynamics in MANETs lead to a constantly changing density over time and in space. As a result, even incorporating the density of the MANET alone does provide a suitable criteria for the dynamic selection of broadcasting strategies or the adaptation of the parameters of one particular broadcasting strategy. In order to provide adaptation to node density, we propose to investigate the impact of node density and the core protocol parameters on the infection rate and select the protocol parameter value that maximizes the infection rate.

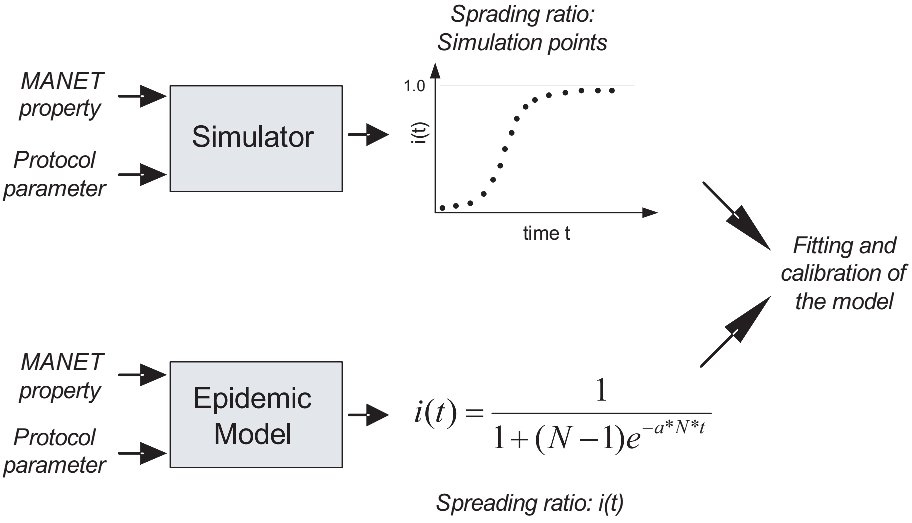


Figure 4.2: Overview of our approach

In Section 4.4, we adopt the SI model for the gossiping protocol. We then compute the infection rate in dependency of the MANET density and the forwarding probability of gossiping. To adapt the forwarding probability to node density, we select the probability value that maximizes the infection rate for a given node density. The result is adaptive gossiping, where nodes set the appropriate forwarding probability depending on the number of neighbors. The performance comparison of adaptive gossiping with related work shows the competitiveness of adaptive gossiping. This proves the applicability of our established modeling methodology for the adaptation of broadcast protocols in addition to their clear usefulness for the concise description of protocol performance.

4.3 Analytical Modeling of the SPIN-Based Protocol

After motivating the need for analytical models that model broadcasting in MANETs and after giving a brief overview on our approach, we now establish the SI analytical model for the SPIN-based protocol. Then, we calibrate the model and provide an analytical expression for the infection rate, the SI model's core parameter, in dependency on node density. Finally, we present similar approaches and discuss the uses of our model. In this section we present our work that we have published in the proceedings of MSWiM'02 [22].

4.3.1 The SPIN-Based Broadcast Protocol

We consider a broadcasting strategy that follows the protocols for information dissemination in sensor networks (Sensor Protocol for Information via Negotiation: SPIN-1 [21]). When a mobile node discovers other mobile nodes, it advertises a summary of its messages. The listening nodes then request the messages which they are interested in. Finally, the advertising node sends the requested data. For more details see [21]. Noteworthy is that nodes store messages in local databases and do not purge them.

4.3.2 Establishing the Analytical Model

According to the broadcast protocol and assuming that every node is interested in the disseminated message, a node follows a two-state *compartmental model*: It either carries the message or not, and once infected by the message, a node remains infectious. Let $S(t)$ denote the number of *susceptible* nodes, and $I(t)$ the number of *infective* nodes. Thus we consider the two-state mathematical model shown in Fig. 4.3. Each letter in a rectangle refers to a *compartment* in which a node can reside.

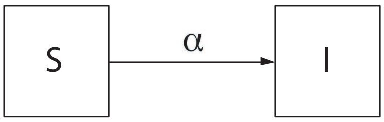


Figure 4.3: Compartment diagram for the SI model

Hereby, α is the *broadcast force* in the MANET. This parameter indicates the strength of the broadcasting process and has the dimension $1/time$.

We focus on the behavior of large scale populations so we use a deterministic compartmental epidemic model. For small populations a stochastic model should be used [20]. To develop the solution, we need to write the mass balance equations for each compartment:

$$\begin{cases} \frac{dS(t)}{dt} = -\alpha * S(t) \\ \frac{dI(t)}{dt} = \alpha * S(t) \end{cases}$$

The problem is that α is not a constant, but depends on the number of susceptible and infectious nodes and the probability of transmitting the message upon encounter. In the following we compute α.

As mentioned before, we say that two nodes *encounter* each other if they are in each other's communication range. We define the *encounter rate* as the average number of encounters per node and per unit of time, and denote it by e. Therefore, each susceptible node makes e encounters per unit of time that are of the message transmitting type. Thus in total, the susceptible nodes make $e * S(t)$ encounters per unit of time. Hence we assume that nodes move autonomously, we can assume that the encounters are at random with members of the total population: $N = S(t) + I(t)$. Then, only the fraction $I(t)/N$ of the encounters are with infectious individuals. Let β be the probability of message transmission in an encounter between an infectious and a susceptible node. Then the rate of susceptible nodes that become infectious is

$$\beta(e * S(t))\frac{I(t)}{N} \; .$$

Thus the broadcast force is

$$\alpha = \frac{\beta * e}{N} I(t) \; .$$

We substitute

$$a = \frac{\beta * e}{N} \tag{4.1}$$

and call it *infection rate*. Thus

$$\frac{dI}{dt} = a * S(t) * I(t) \tag{4.2}$$

With $S(t) = N - I(t)$ we get:

$$\frac{dI}{dt} = a * I(t) * [N - I(t)] = a * N * I(t) - a * I^2(t) \; .$$

This first order ordinary differential equation has the following general solution:

$$I(t) = \frac{N}{1 + C * N * \exp(-a * N * t)} \; ,$$

where C is a constant of integration that depends only on the initial conditions. C is computed as follows: At the beginning ($t = 0$) we assume that we have just one node that carries the message. So $I(t)$ should fulfill $I(0) = 1$.

$$I(0) = 1 \Rightarrow \frac{N}{1 + C * N} = 1 \Rightarrow C = \frac{N-1}{N} \; .$$

Thus the final solution of (4.2) is:

$$I(t) = \frac{N}{1 + (N-1) * \exp(-a * N * t)} \; .$$

According to the definition of the spreading ratio (see Section 4.1.2):

$$\boxed{i(t) = \frac{I(t)}{N} = \frac{1}{1 + (N-1) * \exp(-a * N * t)}} \tag{4.3}$$

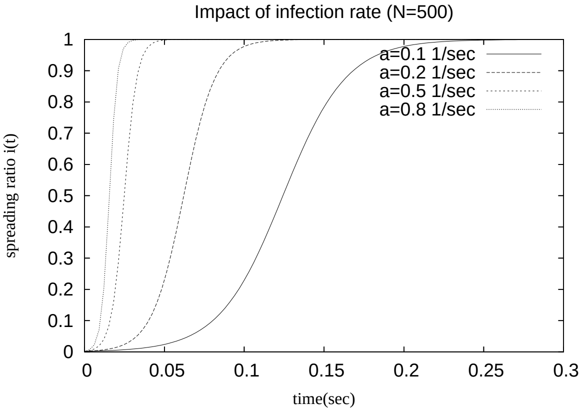


Figure 4.4: Impact of the infection rate a on i(t)

In Fig. 4.4, i(t) is plotted for different values of the infection rate a, while fixing the number of nodes to $N = 500$.

We note here again that the quality of broadcasting mainly depends on both the MANET properties and the broadcast protocol. The MANET property most relevant for broadcasting is the network connectivity over space and time, which primarily reflects the node spatial distribution, node mobility and the used communication technology (PHY and MAC layers).

In our simple mathematical model the infection rate a and the number of nodes N are the single parameters that determine the quality of the message propagation in a given MANET. Thus we conclude that the infection rate a models the impact of the MANET properties and the broadcast protocol characteristics on the broadcasting quality. Therefore, the infection rate is a core model parameter that plays an important role for the performance analysis of broadcasting in MANETs.

Equation (4.1) shows that a depends on the total number of nodes N, the encounter rate e, and the probability β of message transmission, given an adequate encounter. We note here that the encounter rate e depends on node spatial distribution, node mobility and the ad hoc communication technology. β captures the impact of the communication and broadcast protocol parameters on the message propagation. This shows that our modeling approach is *hierarchical* which allows us to proceed *modularly* to further develop the analytical model by providing an analytical expression for a depending on the MANET properties and the broadcast protocol parameters. This task can be reduced to the determination of e from the mobility and communication models and the calculation of β from the broadcast algorithm and the communication model.

In [29], we investigated the encounters between nodes in more details. There, we defined a set of mobility metrics based on node encounters and presented a detailed statistical and analytical analysis of these metrics for the random waypoint mobility model [39] as example. In [29], we provided an analytical expression of the encounter rate (e) for the random waypoint mobility model and assuming that nodes can communicate if their geographical distance is lower than the communication range:

$$e = R * (v_{max} - v_{min}) * d$$

Where R, v_{max}, v_{min} and d are the communication range in m, the maximum node speed in m/s, the minimum node speed in m/s and the node density in $1/m^2$ respectively.
We substitute e and $d = N/A$ in Eq. (4.1) and obtain

$$a = [\beta] * [R * (v_{max} - v_{min})/A] \tag{4.4}$$

The analytical computation of e depends on the complexity of the considered mobility and network models. The analytical computation of the probability of message transmission given an adequate encounter (β) can be easily achieved given an appropriate analytical model for the MAC layer. In this work, we will not further consider the analytical computation. Instead of that, we use an empirical approach to calibrate our analytical model.

4.3.3 Calibration of the Analytical Model

In the following we calibrate our model by comparing simulation results with the analytical results. We show the applicability of our analytical methodology in obtaining insights into the performance of broadcast protocols for MANETs.

Protocol Evaluation using Simulation

Before we proceed with the calibration of the SI model, we briefly investigate the performance of the SPIN-based protocol by doing a few simulations. For this purpose, we use our own simulator, which is written in Java and implements a MAC layer based on the IEEE 802.11b standard.

The area of interest is a 1000m x 1000m two-dimensional field. Let N be the number of nodes in this area and d its node density (measured in $1/km^2$). To vary the node density d, we fix the size of the deployment area A and vary the total number of nodes N. We use the random waypoint mobility model [39] with node speed between 3 Km/h and 70 Km/h and a pause time between 0s and 100s.

At the beginning of the simulation one node generates a single message. We assume that every node is interested in this message. Table 4.2 summarizes the simulation parameters.

Fig. 4.5 illustrates the spreading ratio for the following node densities: 150, 180 and 220 $1/km^2$. The spreading ratio obtained using simulations (Fig. 4.5) looks

Parameter	Value
Area	1000m x 1000m
Number of nodes	N ∈ [50 , 5000]
Communication range	R = 75m
Bandwidth	r = 2048 Kbps
Message size	510 bytes
Mobility model - Max speed - Pause	Random waypoint - Uniform betw. 3 and 70 km/h - Uniform betw. 0 and 100s

Table 4.2: Simulation settings for the evaluation of the SPIN-based protocol

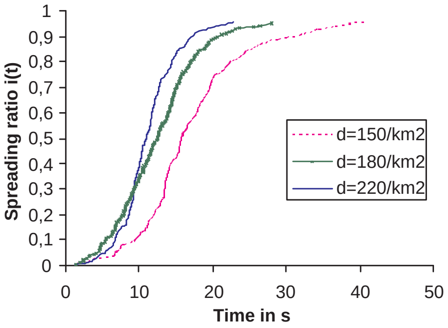


Figure 4.5: Spreading ratio of the SPIN-based protocol (simulation)

very similar to the spreading ratio gained from the analytical model (Fig. 4.4). This demonstrates the applicability of existing epidemic models to describe broadcasting in MANETs.

Computation of the Infection Rate

In the following we present our approach to determine the infection rate a. We proceed similarly to the epidemiologists who assume the availability of some experimental data that roughly describe the spreading of the infectious disease to calibrate the corresponding epidemic model. We rely on a few simulations to calibrate our epidemic model.

First of all, we determine the spreading ratio of the considered broadcast protocol for the considered MANET scenario using simulations. Fig. 4.6 shows the spreading ratio of the SPIN-based protocol for the MANET scenario described in the simulation settings above and with a node density equal to $d_i = 100/km^2$.

Afterwards, we use the *least squares method*, a procedure for finding the best fitting curve to a given set of points, to fit the simulation results to the formula (4.3). We use the software package `mathematica` [100] to perform this fitting procedure (Fig. 4.6).

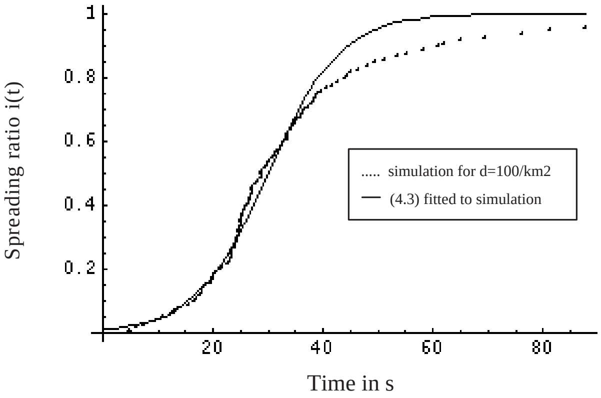


Figure 4.6: Fitting simulation results to formula (4.3)

The result of fitting is the determination of the infection rate a_i for the SPIN-based protocol in the considered MANET with the node density d_i. Through substituting the infection rate a by the computed value a_i in Eq. (4.3), we obtain an analytical expression that describes the message propagation using the SPIN-based protocol in the considered MANET scenario. By this way, we established a concise way to describe the performance of a broadcast protocols in MANETs: The infection rate and the formula (4.3) instead of set of points.

Impact of Node Density on the Infection Rate

Now, we show the applicability of our epidemic model to carry out a handy performance analysis of broadcast protocols in MANETs. We are concerned in investigating the impact of node density on the performance of the SPIN-based broadcast protocol as example. According to our analytical model, it is sufficient to investigate the impact of node density d on the infection rate a. For this study, we use the same settings as fixed in Table 4.2.

A few simulations are run to get enough points within the space $a = a(d)$. Then, we repeated the above fitting procedure for node densities between 50 and 5000 $1/km^2$ and obtained the infection rate of the SPIN-based broadcast protocol for the different node densities in the considered MANET. Using interpolation we can then establish an analytical expression for the infection rate as function of node density $a(d)$. The results are shown in Fig. 4.7.

We also used `mathematica` to interpolate the resulting points and to provide an analytical expression for the infection rate depending on the node density $a = a(d)$. For the interpolation we used the *series expansion* ansatz. The results are listed below and depicted in Fig. 4.8.

- $d \leq 620\ 1/km^2$:

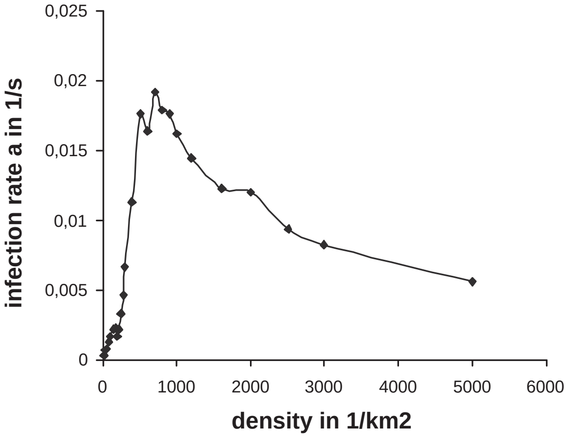


Figure 4.7: Infection rate depending on node density

Ansatz: $a = a_0 + a_1 * d + a_2 * d^2$

Result:

$$a = -7.18 * 10^{-4} + 1.77 * 10^{-5} * d + 1.95 * 10^{-8} * d^2 \qquad (4.5)$$

where a is measured in $1/s$ and d is measured in $1/km^2$.

- $d \geq 620\ 1/km^2$:

  Ansatz: $a = a_0 + \frac{a_1}{d} + \frac{a_2}{d^2}$
  Result:

  $$a = 6.4 * 10^{-4} + \frac{25.5}{d} - \frac{9081.6}{d^2} \qquad (4.6)$$

  with a in $1/s$ and d in $1/km^2$.

It is clear from Fig. 4.8 that with increasing node density the infection rate a at first increases, reaches a peak and then decreases. There exists therefore an optimum node density d_{opt}, which maximizes the infection rate and consequently maximizes the propagation speed (or minimizes broadcast delay). The infection rate a is maximized for $d = d_{opt} \approx 620\ 1/km^2$. We explain this behavior as follows. If d is too small, the network is very sparse and frequently partitioned. Only through the movement of infective nodes between partitions the message spreading can continue. If d is too large, then contention and collisions become dominant and the message propagates more slowly.

Considering Eq. (4.4), we conclude that for the considered scenarios, β first increases if the node density increases. This is due to the increase of the number

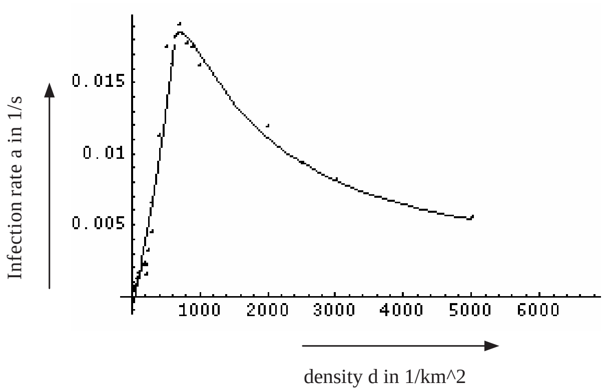


Figure 4.8: Interpolation of infection rate

of neighbors of nodes, which leads to a higher probability of a successful infection, since one single transmission reaches more nodes. However, if node density exceeds the optimum value, the probabilities of collision and contention increase. Therefore, the probability of infection and consequently the infection rate starts to decrease. This proves that the broadcast-in-time scheme is less appropriate for connected scenarios as we stated in Chapter 3.

These results are in line with previous work. Kleinrock et al. [101] showed that the optimum number of neighbors for a node in a fixed ad hoc network is $n_{opt} = 6$. The authors of [102] investigated the optimum number of neighbors in MANETs for different mobility models. In [103], the authors investigated the effect of transmission range on network throughput and discovered that, from a certain communication range the throughput decreases asymptotic to a constant different from zero. Using the percolation theory, the authors in [51] proved the existence of a critical node density below which the considered infinite population consists of an infinite number of bounded partitions and above which there is a unique infinite connected component. If we consider an increase in the number of neighbors and an increase in communication range as an increase in node density, the results reported in [51,101–103] confirm our results. Since we used the random waypoint mobility model for the simulations, the node spatial distribution is almost uniform. Consequently the optimal mean number of neighbors for our settings is $n_{opt} = \pi R^2 d_{opt} = 10.95$.

The current study proves that the developed analytical model provides a succinct method to investigate the impact of MANET properties on the performance of broadcast protocols. Our methodology simplifies the identification of key protocol behavior. Such key observations are harder to conclude from a large set of simulation data. The work carried above is generic and can be applied to investigate the impact of further MANET or protocol parameters on the message propagation.

4.3.4 Relevance of the Epidemic Models

In the following we discuss how broadcast protocol developers benefit from the analytical models to both describe and adapt their protocols. We also propose further uses for these models.

Analytical models for broadcast protocols allow compact description of message spreading in MANETs by means of analytical expressions that use very few parameters. Our analytical SI model offers a comparable accuracy to that of the simulation and a high compactness. To understand the impact of a relevant parameter such as node density on the performance of the protocol, only a few simulations are needed to calibrate the model. Using interpolation and extrapolation an analytical expression can then be established. The spreading ratio of the protocol is then easy to compute for continuous values in a larger interval.

The approach presented here can be applied for any broadcasting technique that shows an SI compartmental behavior, i.e. once a node receives a broadcast message it remains ready to forward this message for other nodes until all nodes receive the message. Other strategies that show a different compartmental behavior, can similarly be modeled by the corresponding epidemic model. Therefore, our work provides a *generic platform* for the community to analytically model broadcasting.

Since a certain node density exists, for which the SPIN-based broadcast protocol performs optimally, and because MANETs may show a large spectrum of node densities, we conclude the need to adapt the SPIN-based broadcast protocol to node density. In order to provide adaptation, key protocol parameters should be identified and tuned in an appropriate way so that the infection rate remains maximized.

An alternative approach for adaptation is to dynamically switch between different inter-operating strategies. The presented analytical model could be used by mobile nodes to dynamically switch between different broadcasting strategies. Assuming a node can perceive its environment (e.g. local node density and total number of nodes), it can easily process the infection rate of the considered SI broadcasting strategy by using the corresponding analytical expressions (4.5) and (4.6). Running different broadcasting strategies, for which the node has an SI model, the node would be able to predict the progress of message spread over time for these strategies (i.e. using (4.3)). The node could then switch between these strategies to adapt broadcasting to the evolving demands of the application and to the current situation in the MANET, e.g. the local node density.

The analytical epidemic model presented here can be also used to design counter-measures to prohibit the spreading of undesired data such as worms or viruses. Recent projects discuss such uses for the epidemic models that we adopted to the broadcasting in MANETs. The authors of [104] study the effectiveness of mitigation techniques against worms in MANETs using epidemic analytical models. In [105], the authors modeled the spread of a worm over

VANETs, using the SIR epidemic model. The model is then used to analyze patching scenarios, which curb the worm outbreak.

Similar Approaches In [18], a simple stochastic epidemic model is used to analytically determine the delay before data has spread to all mobile devices. This model is a *pure birth process*, which is a simple continuous-time Markov process. The authors also assumed two compartments S and I for a node, but they only considered small populations (5 nodes), for which stochastic models are recommended. We are interested in a wide range of network densities, for which deterministic compartmental epidemic models are more suitable. In our work we investigate the entire progress of propagation over time and analytically express the model parameter in terms of node density.

In [106], the authors studied message spreading in one direction along one road. They focussed on the spatial propagation in highly mobile VANETs. The message propagation is modeled as a stochastic renewal reward process.

In [15–17] analytical models for broadcast-in-space have been developed. However these models assume ideal channel conditions, such as ideal MAC.

4.4 Modeling and Adaptation of Gossiping

In this section, we demonstrate the utility of epidemic models to adapt broadcast protocols in MANETs. For this we discuss gossiping in detail, model it with an SI epidemic model, and finally use this model to adapt gossiping to the local node density.

4.4.1 The Gossiping Protocol

Wikipedia states that ”Gossip is the act of spreading news from person to person, especially rumors or private information”. This is the linguistic and social metaphor of gossiping. Gossiping in MANETs is simply defined as probabilistic flooding. On receiving the first copy of a given message, gossiping forwards the message, with a probability p, to all nodes in the receiver's communication range. In order to reduce the collision probability, nodes delay forwarding for a random time between 0 and *fDelay*. We assume that each node stores the list of IDs of messages received or originated, in a so-called *broadcast_table* for a time period in the range of a few seconds. Thus nodes are able to decide, whether a received copy of a given message is the first one. The pseudo-code description for gossiping is given by Algorithm 1. We denote by *random(x)*, a function that returns a random float value $\in [0, x]$.

Algorithm 1 Gossiping (p)

```
Var: p, fDelay
List: broadcast_table
# On receiving a DATA message M: do gossiping(p):
if M.ID $\notin$ broadcast_table then
    # M is received for the first time
    deliver M
    add {M.ID} to broadcast_table
    if $random(1.0) \leq p$ then
        wait (random(fDelay))
        send M to all neighbors using MAC broadcast
    end if
else
    discard M
end if
```

According to this protocol, on average, only $p * N$ nodes forward the message. Thus the number of saved message forwards is $(1 - p) * N$. To maximize the number of saved forwards, we have to reduce the probability p. But how much can we reduce it? In [74], the authors investigated gossiping in more depth. In order to reduce the number of redundant messages, every node will forward a message based on a uniform probability p. The authors showed that gossiping exhibits a bimodal behavior. There is a threshold value p_0 such that, in sufficiently large random networks, the gossip message quickly dies out if $p < p_0$ and the gossip message spreads to the entire network if $p > p_0$. Thus, ideally we would set the gossiping probability close to p_0 (slightly higher), and therefore save approximately a ratio of $1 - p_0$ forwards compared to plain flooding.

In [74], the authors identified an optimum value of $p_0 = 0.65$ for their test scenarios. Intuitively, an optimal probability for one node density may be suboptimal for other densities, so this value is not likely to be globally optimal. Furthermore, since the node density varies over time and space, we have to adjust the probability to the local node density. This is what we aim to do, using an epidemic model for gossiping.

4.4.2 Epidemic Model for Gossiping

The propagation delay using gossiping is in the range of milliseconds or a few seconds at maximum, depending on the current network parameters and load. During this period the nodes remain almost in the same position. During the spreading of a message to other nodes, infected nodes do not encounter new neighbors. Therefore, we can simply assume that infected nodes remain infective during gossiping until the end of the propagation. Consequently, we can also model gossiping using the SI compartmental model. Therefore, we apply the same equations to gossiping as in Section 4.3. For the calibration of the gossiping epidemic model we also rely on simulation results. If the network is partitioned, we set the delay for unreachable nodes to be ∞. Therefore, the infection rate is approximately 0, for highly partitioned MANETs.

The infection rate clearly depends on the gossiping probability p. If this probability is 0, the infection rate will be also 0. If p increases, the infection rate also increases. However, if the network is very dense and all nodes forward every newly received message, contention and collisions increase, so that delay increases, and subsequently the overall infection rate will decrease.

That is why we investigate the impact of both node density and gossiping probability on the infection rate in more details in the next section. This investigation allows the selection of the appropriate probability depending on node density.

4.4.3 Adaptation of Gossiping Forwarding Probability

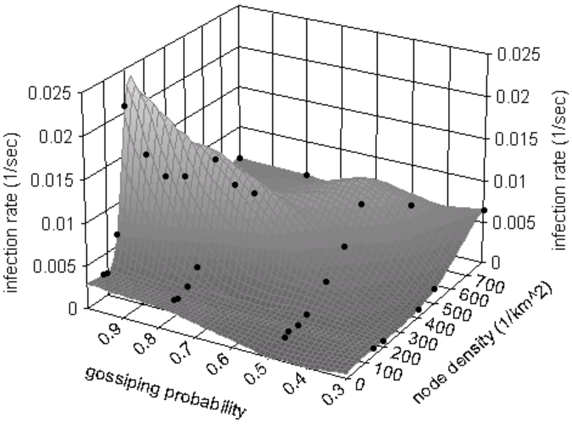
The goal of adapting gossiping is to achieve higher efficiency by reducing the number of forwarders, but without sacrificing the reachability or experiencing any significant degradation. Since the intensity of the broadcast storm depends on the local node density and may vary over time and space, we should adapt the gossiping probability p to the node's current number of neighbors, which reduces forward redundancy, contention, and collisions.

In this section, we adapt gossiping to the local node density by determining the appropriate gossiping probability as a function of the number of neighbors. For this study, we use the same parameters as given in Table 4.2 with a $100m$ communication range and $3m/s$ maximum speed. We continue to use the random waypoint mobility model for simulations, since the model shows an almost uniform node spatial distribution. This property simplifies the conversion of node density to number of neighbors and vice versa. Given the number of neighbors n and the communication range R, a node easily computes its local density by:

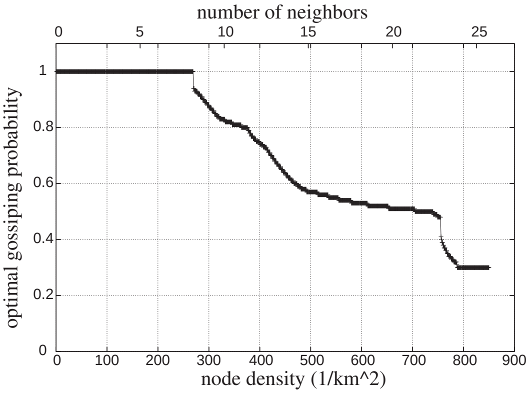
$$d = \frac{n+1}{\pi R^2} \Leftrightarrow n = \pi R^2 d - 1 \tag{4.7}$$

Adaptation Using the SI Epidemic Model

According to the SI model, the infection rate determines the spreading ratio and therefore it is a measure for delivery reliability and delay. The higher the infection rate, the lower the mean delay. In the following we show how we used these results to adapt gossiping. In the case of gossiping, the infection rate mainly depends on the node density of the MANET and the gossiping forwarding probability p. In order to adapt the forwarding probability to the node density, we should select the gossiping probability that maximizes the infection rate. We vary node density and the forwarding probability p and compute the corresponding infection rate for some parameter combinations. Fig. 4.9 a) shows the measured infection rates and their interpolation. Fig. 4.9 b) shows the optimal probability, which should be used for gossiping depending on the MANET node density.



(a) Infection rate



(b) Optimal probability

Figure 4.9: Adaptation of gossiping using the infection rate

Consistent with our fourth requirement of generalized broadcasting strategy (see Section 2.3.4), we let every node set the gossiping probability *locally* and *independently*. A node i can easily estimate its local node density d_i using Eq. (4.7), given its number of neighbors n_i. According to the value of d_i the node sets on-the-fly the forwarding probability p_i for gossiping.

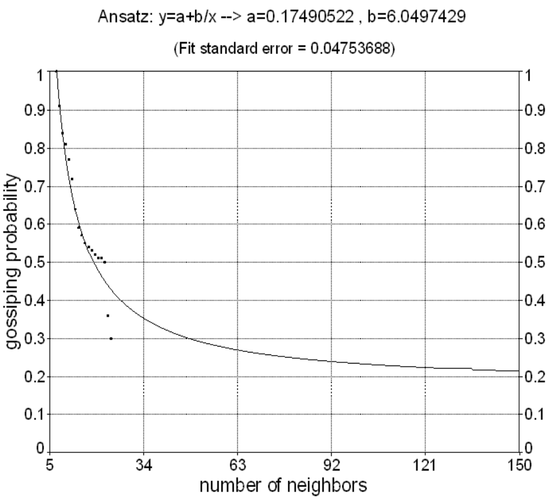
To avoid the computation of local node density, which also assumes that nodes know their communication range R, we propose that nodes select the gossiping probability depending on the current number of neighbors n. By scaling the x-axis of Fig. 4.9 b) using formula (4.7), we get the optimal gossiping probability p as a function of n. We could now provide the discrete values of this curve as a lookup table that maps the number of neighbors to the probability values. At run-time, nodes could then access this lookup table in order to set the gossiping probability dynamically, depending on their current number of neighbors.

Nevertheless, in order to elegantly present our adaptation results for the community, we analytically express the gossiping probability depending on the number of neighbors. To ensure adaptation for higher dense networks, we extrapolate the gossiping probability value to higher number of neighbors. We use the following series expansion ansatz: $p(n) = a + b/n$. The fitting process using the least squares method, recommends $a = 0.175$ and $b = 6.050$. The fitting standard error is about 4.75% (Fig. 4.10). Summarizing, our gossiping probability is analytically given by the following equation:

$$\begin{cases} p = 1.0, & if\ n \leq 7 \\ p = 0.175 + 6.05/n & if\ n \geq 8 \end{cases}$$

Summarizing we analytically express the forwarding probability for adaptive gossiping as follows:

$$\boxed{p \ = \ min\ (\ 1.0\ ,\ 0.175 + \frac{6.05}{n}\ )} \tag{4.8}$$


Ansatz: y=a+b/x --> a=0.17490522 , b=6.0497429

(Fit standard error = 0.04753688)

Figure 4.10: Extrapolation of gossiping probability

Relevance of Epidemic Models for Protocol Adaptation

We show the relevance of the analytical epidemic models for the adaptation of broadcast protocols through investigating alternative approaches for the adaptation.

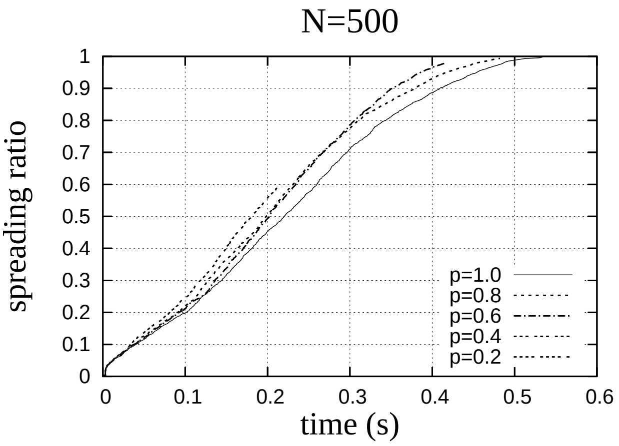


Figure 4.11: Adaptation of forwarding probability (Simulation-based approach)

Fig. 4.11 shows the spreading ratio of gossiping over time for 500 nodes and different forwarding probabilities. We conclude that only probabilities higher than 0.6 provide a reachability close to 100%. We also conclude that the forwarding probability 0.6 provides faster propagation than higher probabilities. This is due to broadcast storms if more than 60% of nodes forward the packet. Thus, investigating the spreading ratio obtained from simulations provides an alternative approach to fix the appropriate gossiping probability.

However, the selection of the probability is achieved visually and therefore it is not handy and error-prone. Furthermore, the approach requires running simulations for probability values as fine as possible to increase the accuracy of adaptation. Comparing the simulation-based approach with the approach relying on the epidemic model we note the simplicity of the last approach, which provides an automated method for the selection of the appropriate forwarding probability depending on node density, using only fewer simulations.

4.5 Evaluation of Adaptive Gossiping

We now evaluate the adaptive gossiping protocol with scenarios that show a wide range of node spatial distributions and node speeds. Additionally, we study the impact of communication range and mobility patterns on the performance of adaptive gossiping. We also compare adaptive gossiping with STOCH-FLOOD and ACB. Our evaluation approach is simulation-based. For this purpose, we first present our simulation model and simulation settings.

4.5.1 Simulation Model

For the remainder of this thesis, we use the widely used network simulator ns-2 [11]. Ns-2 is a discrete event simulator targeted at wired and wireless networking research. For efficiency reasons ns-2 is implemented in OTcl and C++. OTcl is an object-oriented extension of the interpreted language Tcl (the Tool Control Language). Ns-2 uses OTcl for control and C++ for data manipulation, since OTcl permits an easy and dynamic simulation configuration and C++ a fast and efficient manipulation of data and implementation of protocols.

In this section, we present the PHY and MAC models implemented by ns-2 and the mobility that we will use for the performance evaluation.

PHY and MAC Models

Spatial environment and node mobility affect the wireless channel characteristics. The transmitted signal is vulnerable to reflections, diffractions and scatterings. Therefore, the receiver often receives a superposition of the direct-path (also line-of-sight) signal and a multipath signal.

Path loss models are used to estimate the received signal level as a function of distance. With the help of such models we can predict signal power at the receiver. Network simulators use these models to estimate the signal power at all potential receivers. In this way, simulators are able to predict collisions and determine, which nodes receive the transmitted frame correctly, which ones receive it with errors and which nodes do not receive it at all.

The *free space* is the simplest propagation model, where only the direct-path signal is considered. The *two-ray* or *two-path* model is another popular propagation model. This model considers both the direct-path and the ground reflection path. It is shown that this model gives more accurate predictions at long distances than the free space model, but does not give a good result for short distances. Hence a combination of both models is recommended: Up to a distance threshold (called cross-over distance) the free-space model is used and from the cross-over distance the two-ray model is considered. The model is simple while providing a high propagation accuracy. This model is implemented in ns-2 and it is known as the *TwoRayGround* propagation model. We use the TwoRayGround model like most of existing simulation work in ns-2. Nodes in ns-2 deploy omni-directional antennas.

The ns-2 simulator implements a MAC layer according to the IEEE 802.11 standard. The radio channel readily supports MAC broadcast. We presented more details of the IEEE 802.11 standard in Section 2.1.

Summarizing, in ns-2 two nodes can communicate when their spatial distance is less than or equal to the fixed communication range R. The resulting topological graph of the MANET is therefore an undirected graph. An edge between two nodes is added if their distance is below R.

Mobility Models

Mobility plays a key role in MANETs. That is why many models have been developed in the last few years. Surveys on mobility modeling for wireless networks can be found in [107] and [108]. In [108, 109], the authors classify the existing models into three main classes: Random models, models with spatial dependencies and models with geographic restrictions. In the following we detail the most common representative of each class, i.e. random waypoint, reference-point group mobility and graph-based mobility respectively.

Random Waypoint Mobility Model: The *random waypoint* mobility model is defined in [39] as follows. Initially, each node chooses a destination randomly uniform in the considered area and a speed randomly uniform between v_{max} and v_{min}, and moves there with the chosen speed. Then the node pauses for a random period of time (*pause*), before repeating the same process.

The random waypoint has simple (mathematical) characteristics, which makes it easier to the understand the impact of node mobility on the performance of the protocol under consideration. Therefore, the random waypoint model has been intensively investigated in the literature [110–113], and it is often used for MANET simulations.

Reference-Point Group Mobility (RPGM): In MANETs, mobile nodes often move as a team or group, e.g. rescue teams, military convoys, wild-life and sea-life. It is thus desirable to model group motion to enable a more accurate evaluation of MANETs.

In [114], the authors introduced the *Reference-Point Group Mobility (RPGM)*. Initially group members are uniformly distributed in the neighborhood of the group center. The subsequent speed and directional choices of members are derived by random deviation from that of the group center. The group centers move according to the random waypoint model.

This variant of the RPGM model is also known as the nomadic community model. Other variants of RPGM have been also defined: Column model [107] and pursue model [107].

The RPGM model can be readily applied to many MANET application scenarios. RPGM can be used to model the mobility of soldiers or rescue teams. Here each group has a logical center (group leader) that determines the group's motion behavior. Moreover, with a proper choice of parameters, RPGM can be used to model several proposed mobility models.

Graph-Based Mobility Model: The authors in [115] presented the *graph-based mobility model*, which reflects the spatial constraints of movement given by the spatial environment in the real world. The model uses a graph to represent the movement constraints imposed by the infrastructure. The vertices of the graph represent locations that the users might visit and the edges model the connections between these locations, e.g. streets or train connections. Nodes always move along the edges of a graph. Each mobile node is initialized at a random on the graph. The node randomly selects another vertex as a destination and moves towards this destination with a randomly selected speed, using the shortest possible path. At the destination, the node makes a pause for a randomly selected period. It then picks out another random destination and repeats the same process.

The graph-based model can be used to model VANETs. Vehicles move on roads, which can be modeled as graphs.

Selection of Mobility Models for Performance Evaluation: It is imperative to use a rich set of mobility models to thoroughly evaluate MANET protocols. As we aim at developing a generalized solution for MANETs, we will consider all three models discussed above. We emphasis here that the three models show a wide range of spatial distributions from uniform (Random Waypoint) to very heterogeneous (RPGM) and wide range of mobility patterns from uncorrelated (random waypoint) to strong correlated (RPGM). This allows for simulating a wide range of MANET properties with respect to node density and and node mobility, thus covering a wide range of MANET application scenarios.

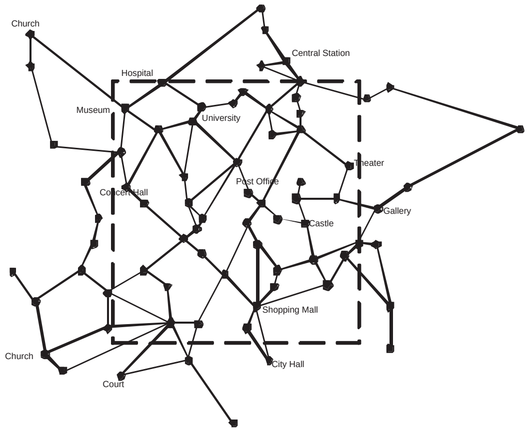


Figure 4.12: The Stuttgart graph

For the generation of the corresponding movement traces we use the following tools. Ns-2 provides its own tool, i.e. `setdest`, to generate movement traces following the random waypoint model. We use `bonnmotion` [116] to generate RPGM scenarios and `canuMobiSim` [117] to generate stuttgart-graph mobility scenarios. In this thesis, we use a road map of Stuttgart city to generate an example of graph-based mobility scenarios (Fig. 4.12).

Parameters	Value(s)
Simulation area	1000m x 1000m
Number of nodes	N $\in$ {50, 100, 200, 300, 500}
Comm. range	$R \in \{50, 100, 200, 300\}$m
Bandwidth	$r = 1$ Mbps
Data packet size	280 bytes
Mobility models - Max speed - Pause	Random waypoint, RPGM, graph-based - $v_{max} \in [0,30]$ m/s - Uniform betw. 0 and 2s
fDelay	10ms

Table 4.3: Simulation parameters for the evaluation of adaptive gossiping

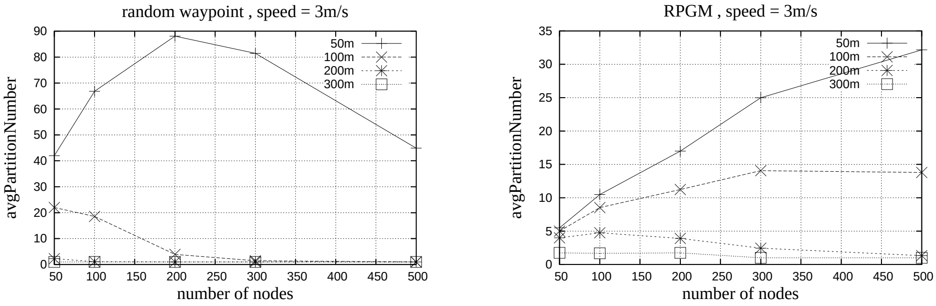
4.5.2 Simulation Settings

We generate N mobile nodes in a 1km x 1km two-dimensional field, where these nodes move according to an arbitrary mobility model. The mobility models we consider for the evaluation of adaptive gossiping are the random waypoint (rw) model, the reference point group mobility model (RPGM) and the graph-based mobility model. The movement area considered for the graph-based model is the 1km x 1km focussed area in Fig. 4.12. For all mobility models, we vary the node speed between 0 m/s and a maximum speed value (v_{max} m/s), and select a pause time uniformly between 0 and 2s. For the RPGM mobility model, nodes are generated in groups consisting of 10 ± 5 nodes. The group members are geographically grouped, since nodes are located at a maximum of 10m away from the group center. The simulation parameters are summarized in Table 4.3.

As mentioned before, nodes acquire neighborhood information by means of HELLO beaconing. For all simulations in this work we use a random beaconing period between 0.75s and 1.25s. A neighbor is removed from the neighbor list, if during 2s no beacon is received from this neighbor.

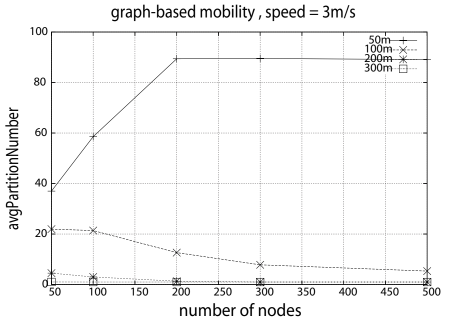
We use the following traffic model. At the beginning of the simulation, namely between the first and the third second, s senders send a single message. The simulation time selected for all scenarios in this chapter is 20s. We set the number of senders to $s = 25$.

Since the knowledge of the partitioning of the MANET is important for understanding the performance of adaptive gossiping and other broadcast-in-space algorithms, we computed the number of partitions for the different scenarios, which we will use in this subsection (Fig. 4.13). We use for this computation our own framework, which we introduce in Section 5.5.6 in detail.



(a) Random waypoint

(b) RPGM



(c) Graph-based

Figure 4.13: Average number of partitions

4.5.3 Performance Metrics

In Section 2.3.2, we identified two basic requirements of the applications on a broadcasting protocol, i.e. delivery reliability and delivery timeliness. In order to evaluate broadcast protocols with respect to delivery reliability and timeliness, the performance metrics reachability and delay respectively are commonly used in the broadcast community. In the following we define these both metrics. With respect to a given broadcast message, we denote by *#Forwd* the number of nodes that forwarded the message and by *#Reach* the number of nodes that received the message after the termination of the protocol.

REachability (RE): The ratio of nodes receiving the message to the total number of nodes, i.e. $RE = \frac{\#Reach}{N}$ $(\in [0,1])$. The reachability metric measures the delivery reliability of the broadcast algorithm.

Delay: Average end-to-end delay over all receivers. Denoting the origination time of the message by t_s and the arrival time of the message at node i by t_i, we calculate the delay as follows: $delay = \frac{1}{\#Reach}\sum_{reachedNode_i}(t_i - t_s)$.

To evaluate the efficiency of broadcast protocols the message complexity is a key factor. An efficiency metric for broadcast protocols that takes the message complexity into account and that is commonly used by the community is the following metric :

MNF: Mean Number of Forwards per node and message. $MNF = \frac{\#Forwd}{N}$.

As we were using the spreading ratio for describing the quality of a broadcast protocol, we differentiate the metrics above to the spreading ratio. Both metrics RE and $delay$ are easily gained from the spreading ratio. Given the spreading ratio as a time function $i(t) \in [0, 1]$. The RE is the maximum value of the spreading ratio (reached when the broadcast protocol terminates), or $RE = \max(i(t))$. The delay is calculated as follows: $\frac{1}{RE} \int_0^{RE} i^{-1}(x)\,dx$, where $i^{-1}(t)$ is the inverse function of $i(t)$.

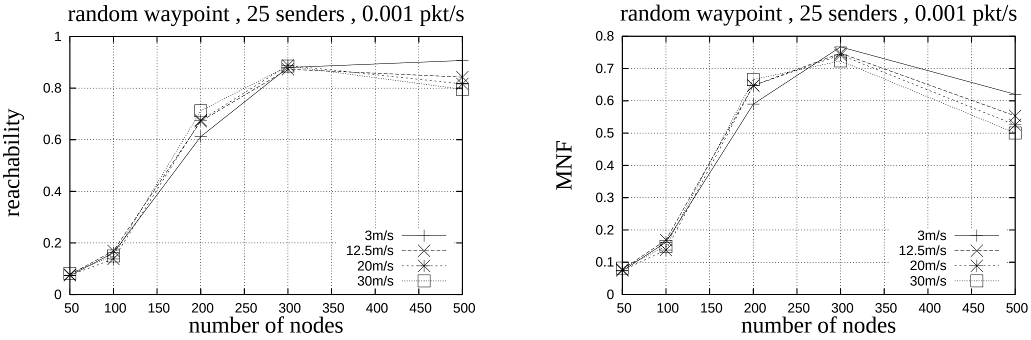
4.5.4 Impact of Node Density and Node Mobility

For this study, we vary the node density by tuning the total number of nodes and keeping the area unmodified. We vary the node mobility and the node spatial distribution by considering different mobility models, i.e. random waypoint, RPGM and graph-based, and by varying the maximum node speed.

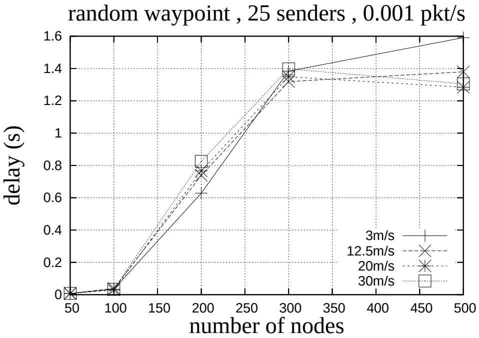
Random Waypoint: First of all, we consider the random waypoint mobility model and vary the number of nodes and the maximum node speed. From Fig. 4.14 a), we observe that the reachability of adaptive gossiping first increases with node density, reaches a maximum and then starts to decrease. We qualitatively explain this effect as follows: Obviously, gossiping can only reach nodes that belong to the partition, which contains the source node. For random waypoint, the mean number of partitions decreases with the increasing number of nodes (Fig. 4.13 a), 100m range). This means that the average partition size is increasing. Therefore, reachability increases with the increasing number of nodes. For high number of nodes, collision probability becomes higher and the reachability begins to decline.

The impact of node speed is marginal. However, we present three observations. Firstly, for very sparse networks the mobility has no impact on the reachability. Secondly, for scenarios that are neither very sparse nor connected (e.g. 100 nodes), the mobility may help to overcome network partitioning and the reachability increases with higher speeds. Thirdly, for dense scenarios, reachability decreases with higher speeds. This effect is also observed for plain flooding [96]. The reason is that a node may sense a free carrier and starts to transmit; but while moving very fast it disturbs other ongoing transmissions (CSMA operates on inconsistent topologies!).

In Fig. 4.14 b), we show the message overhead (MNF) of adaptive gossiping. Assuming a uniform spatial distribution of nodes, we can estimate the MNF of gossiping as follows: $MNF \approx p * RE$. For random waypoint, we can assume a uniform node distribution for the short scenario time (20s) used for current simulations. This explains the behavior of MNF, which shows a strong similarity to that of reachability. For lower number of nodes, p is probably 1.0 and



(a) Reachability (b) MNF



(c) Delay

Figure 4.14: Impact of node density and speed on the performance of adaptive gossiping (random waypoint)

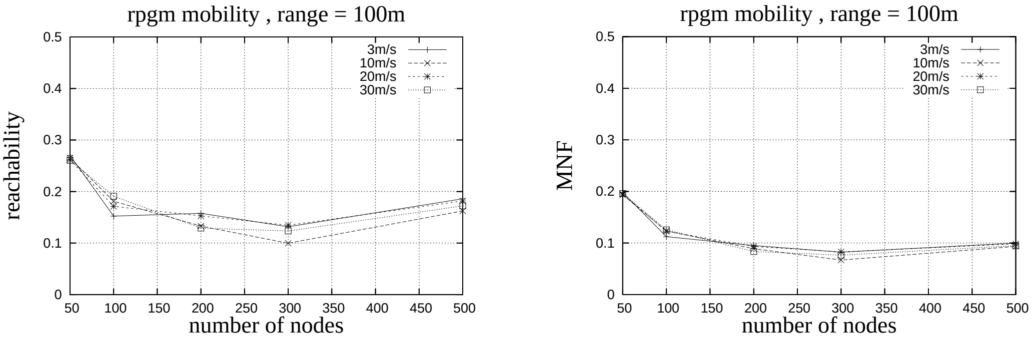
$MNF \approx RE$. For higher number of nodes, nodes use lower forwarding probabilities, thus increasing the number of saved forwards, and therefore $MNF < RE$.

The delivery delay increases with increasing number of nodes since the number of traversed hops to the destination and the buffering time of messages at the MAC layer increase (Fig. 4.14 c)).

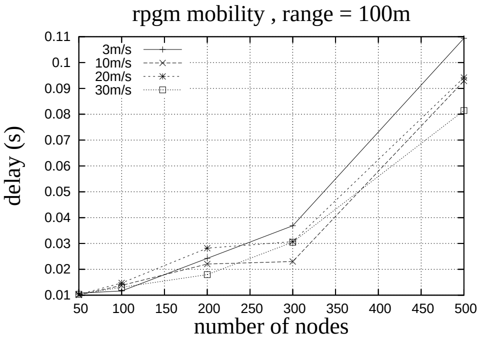
RPGM: In Fig. 4.15, we show the performance of adaptive gossiping for the RPGM mobility model.

For the RPGM mobility model, nodes are generated in groups consisting of 10 ± 5 nodes. In this way, the mean group size is 10. The group members are also geographically grouped since nodes are located at a maximum of 10m away from the group center. Therefore, in very sparse MANETs (such as $N = 50$ nodes in 1 km^2 and $R = 100m$) groups can not communicate with each other, or only very a few of them can do. In this way, the group set forming the MANET is probably the same as its partition set.

According to the observations above, in very sparse MANETs, the mean partition size is almost equal to the mean group size. Therefore, the reachability of



(a) Reachability (b) MNF



(c) Delay

Figure 4.15: Impact of node density and speed on the performance of adaptive gossiping (RPGM)

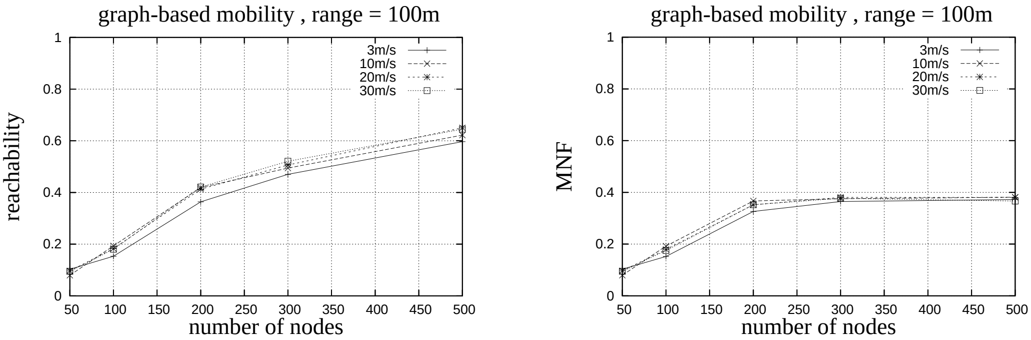
gossiping is about $10/N$, which proves that with an increasing number of nodes, the reachability of gossiping will temporarily decrease. This is valid until the generated groups become close to each other so that a network partition becomes composed of more than one group (Fig. 4.13 b)). As a result, the reachability of gossiping starts to increase. This explains the dependency of the gossiping reachability on the number of nodes (Fig. 4.15 a)).

Taking into account the approximation $MNF \approx p * RE$, we can easily explain the behavior of MNF (Fig. 4.15 b)) using the behavior of reachability (Fig. 4.15 a)).

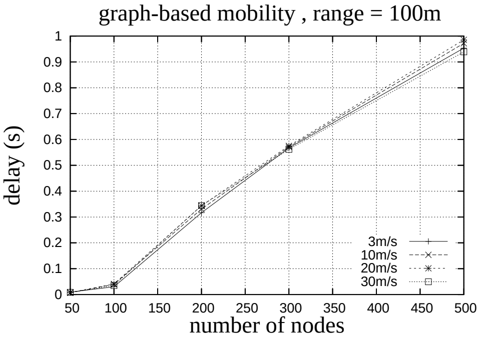
In Fig. 4.15 c), we observe that, similar to the random waypoint, the delay increases with the number of nodes, due to the increase in the number of traversed hops and the increase in time spent by broadcast messages at the MAC layer. The impact of node mobility is marginal similar to that for the random waypoint. RPGM shows a much lower delay than random waypoint, since most of groups members are reached by the first transmission of the source node.

Graph-Based Mobility Model: In Fig. 4.16, we show the performance of adaptive gossiping for the graph-based mobility model.

The reachability increases with an increasing number of nodes. This is due to the decrease of number of partitions with an increasing number of nodes (Fig. 4.13 c), 100m range). The increase of the reachability with N is however remarkably slower than that for the random waypoint mobility model (Fig. 4.14). The reason, is that nodes moving on a line (road) are more likely to show network partitioning [51]. Therefore, the graph-based mobility model shows more partitions than the same node configuration for random waypoint (Fig. 4.13 a)), in most of cases. This can be explained intuitively as follows. Random waypoint shows a statistically almost uniform node distribution. For random waypoint, broadcast-in-space therefore has many possibilities to circumvent directions, in which the broadcast stops. If the node movement is restricted to a graph, nodes approach each other on the one hand, but on the other hand, the broadcast-in-space now has less spreading possibilities (directions), probably only backwards or forwards along the graph edges.



(a) Reachability (b) MNF



(c) Delay

Figure 4.16: Impact of node density and speed on the performance of adaptive gossiping (graph-based)

MNF (Fig. 4.16 b)) and delay (Fig. 4.16 c)) behaviors can be explained similarly to the random waypoint and RPGM mobility models.

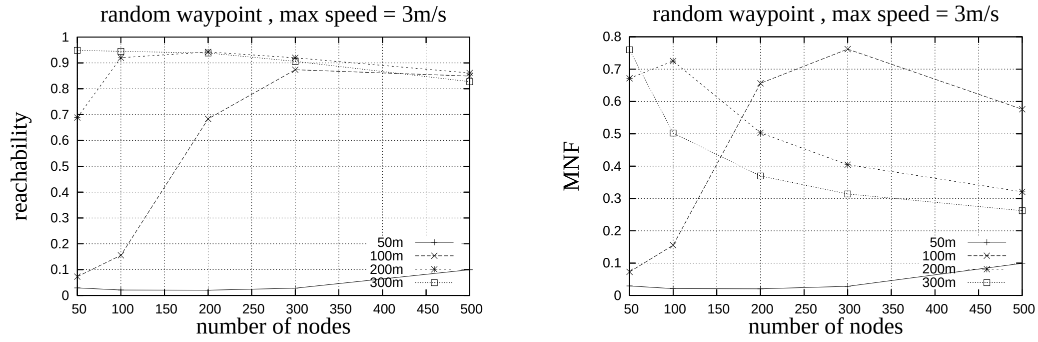
4.5.5 Impact of Transmission Range

In this study, we investigate the performance of gossiping for different communication ranges $\in \{50, 100, 200, 300\}m$. We notice that an increase in communication range can be interpreted as an increase of node density. We also consider the three different mobility models random waypoint, RPGM and graph-based. We hereby fix the maximum node speed at 3m/s.

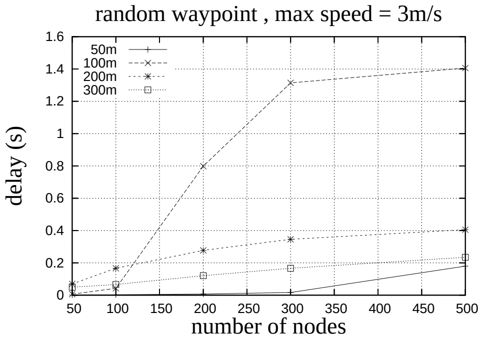
Random Waypoint: The reachability of adaptive gossiping increases with the communication range (Fig. 4.17 a)). For very low communication ranges, the reachability decreases with increasing number of nodes and reaches a minimum (by 200 nodes for a 50m range), and increases for higher numbers of nodes. We explain this decrease of reachability as follows. For very sparse MANETs, an increase of number of nodes, leads to a decrease in the ratio of partition size to the total number of nodes. Consider as an example the following extreme case. Nodes are isolated. So the reachability of gossiping is $1/N$. If we increase the number of nodes by δN and all nodes remain isolated, the reachability of gossiping is $1/(N + \delta N)$. Therefore, the reachability of gossiping decreases with increasing number of nodes in highly sparse MANETs.

For higher communication ranges, the curve of reachability however shows a maximum. The reachability slightly decreases for higher numbers of nodes. This behavior is also observable for RPGM, but it is more clear for the graph-based mobility model. We explain this decrease by the increasing number of collisions. The number of collisions increases since most of source nodes are within each other's communication range. Therefore, one broadcast has more impact on the other broadcasts taking place almost simultaneously. Gossiping has not been adapted to network load. Consequently, for higher network loads the reachability of gossiping is likely to decrease. We will not adapt gossiping to the network load and delegate this optimization to the generalized broadcast strategy. This effect is confirmed by the performed delay (Fig. 4.17 c)), which increases with the communication range and especially for a 300m range.

For discussing the message overhead, we first consider the communication range 100m (Fig. 4.17 b)). MNF first increases with the number of nodes, reaches a maximum and then decreases. The maximum is reached, when almost all nodes forward broadcast messages, i.e. gossiping goes into plain flooding. MNF reaches its maximum, when the MANET starts to be constituted of one large partition and a few small partitions. If the MANET node density increases, adaptation of gossiping runs and saves a number of forwards, which is reflected by the decrease of MNF. For the 200m communication range, the maximum is reached for 100 nodes. For a 300m communication range the maximum moves to the left of 50 nodes and is no longer observed for our experiment settings. For a 50m range,



(a) Reachability (b) MNF



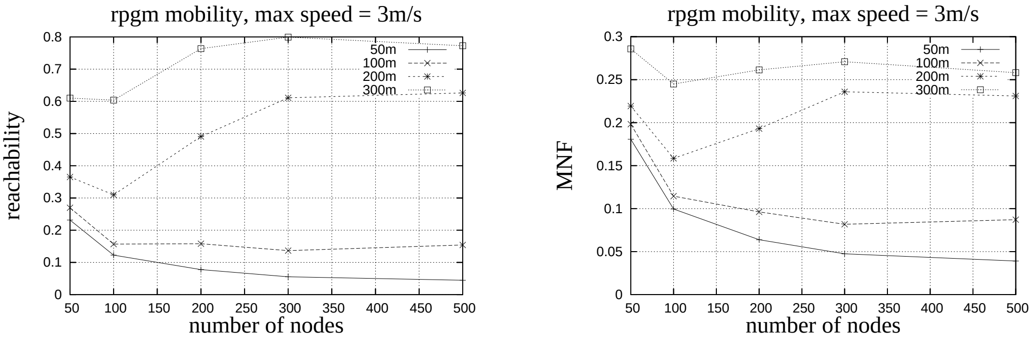
(c) Delay

Figure 4.17: Impact of transmission range on the performance of adaptive gossiping (random waypoint)

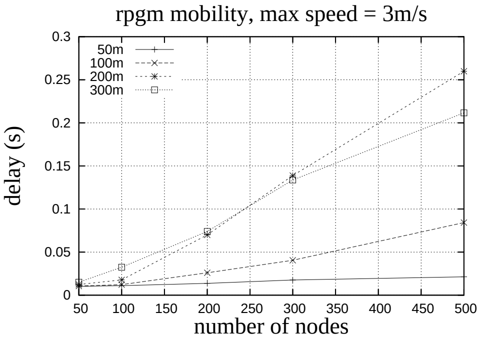
MNF is very close to reachability, since the node density is very low and almost all receivers forward messages. The maximum is reached for a number of nodes that is higher than 500 nodes.

RPGM: For RPGM, the reachability of adaptive gossiping increases with the communication range (Fig. 4.18 a)). For a communication range of 50m the gossiping reachability does not reach its minimum for N up to 500 nodes. For a 100m range the minimum reachability can be observed for N higher than 300 nodes. For higher communication ranges, the minimum reachability is reached with a lower number of nodes, as groups start to be able to directly communicate with each other.

Graph-Based Mobility Model: For the graph-based mobility model, the reachability of adaptive gossiping increases with the communication range (Fig. 4.19 a)). The gossiping reachability behaviors very similar to the reachability for the random waypoint. However, for a 100m range, the reachability for random waypoint is clearly higher. This is due to the difference in node spatial distribution,



(a) Reachability (b) MNF



(c) Delay

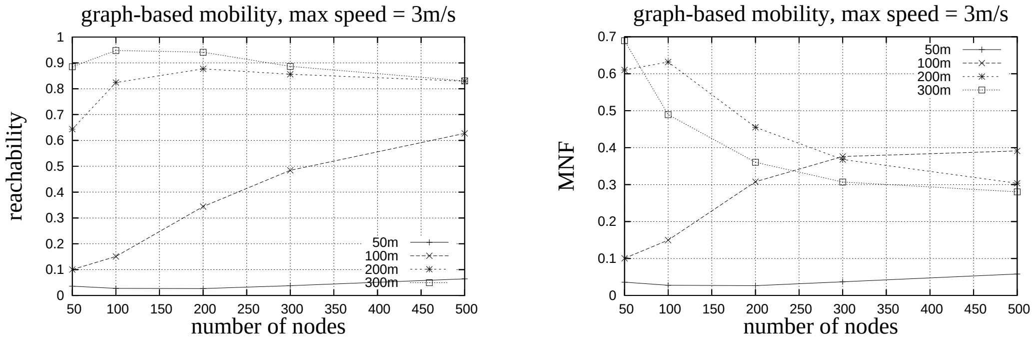
Figure 4.18: Impact of transmission range on the performance of adaptive gossiping (RPGM)

which leads to different partitioning composition for the same communication range. After a certain communication range, the partitioning composition of the MANET remains constant and reachability saturates (Fig. 4.13).

4.5.6 Comparison of Adaptive Gossiping to the Optimal Case

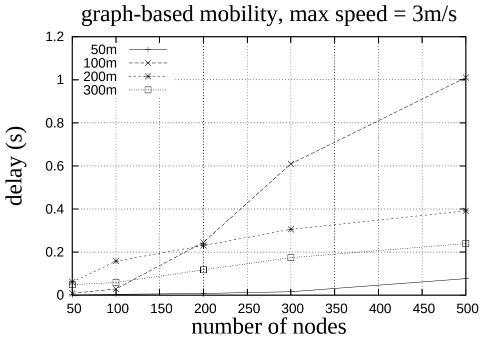
From the studies above, we realize the strong need for a global view with respect to network partitioning in the MANET for a better understanding of the protocol performance. In [28], we presented the tools required for ns-2 users, in order to simplify the access to this global view. In the following, we present the global evaluation of adaptive gossiping. Adaptive gossiping aims to efficiently reach all nodes in the partition where the broadcast source is located. In this section, we aim to investigate in more details the delivery reliability of adaptive gossiping. In particular, we define the optimal gossiping reachability (OG_RE) as the ratio of the size of the partition containing the gossiping source node to the total number of nodes: $OG\_RE = \frac{partition\_size}{N}$.

The reachability of adaptive gossiping should correlate with the partition size.



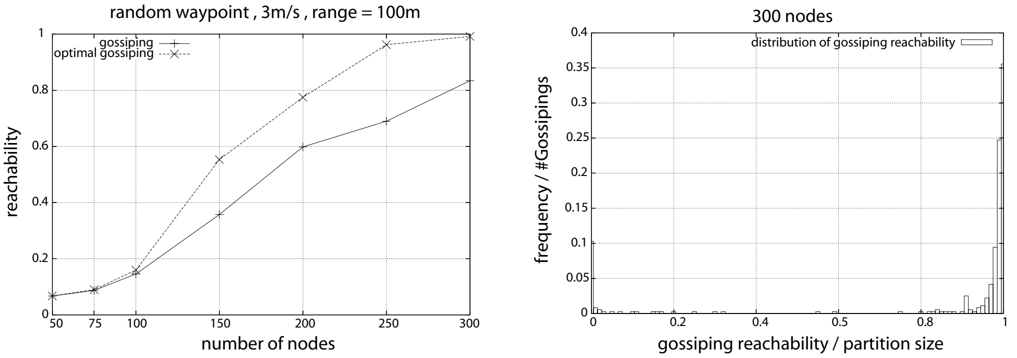
(a) Reachability

(b) MNF



(c) Delay

Figure 4.19: Impact of transmission range on the performance of adaptive gossiping (graph-based)



(a) Reachability

(b) Frequency histogram

Figure 4.20: Comparison of adaptive gossiping to the optimal case

Fig. 4.20 a) shows that the gossiping reachability is slightly below the optimal gossiping reachability. This is due to collisions, which prohibit gossiping from progressing, and which become more frequent with increasing number of nodes. Fig. 4.20 b) shows the frequency histogram of the ratio of the number of nodes reached by gossiping to the sender's partition size. This figure shows that in most of cases gossiping reaches either more than 90% of the partition nodes or less than 10% of nodes.

This study demonstrates the importance of the partition size information our framework provides.

4.5.7 Comparison to Related Work

In the following, we compare the performance of our adaptive scheme to that of the Adaptive Counter Based scheme (ACB) [23] and of stochastic flooding (STOCH-FLOOD) [24]. We vary the mobility model (study 1) and the communication range (study 2). In the studies 1 and 2, we consider $s = 25$ senders and a maximum node speed of 3 m/s. In both studies, we vary the total number of nodes.

Preliminaries

The adaptive thresholds are shown in Fig. 4.21. For ACB we use the dynamic threshold given in [23]. ACB uses a random time span to count redundant packet receptions and possibly forwards the message after this span. This time period is comparable to the random forwarding delay of gossiping ($fDelay$) and stochastic flooding. We choose the same value for these parameters, i.e. 10ms, which is also used in [31].

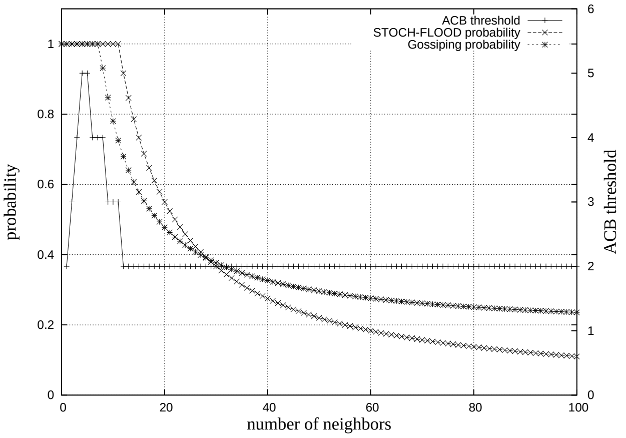
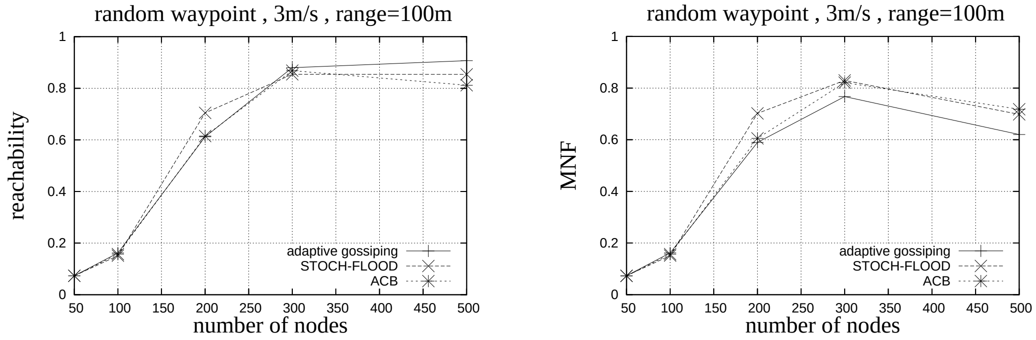


Figure 4.21: Adaptive thresholds

The comparison of adaptive gossiping to stochastic flooding can be intuitively undertaken based on the comparison of probability functions used by each protocol (Fig. 4.21). Adaptive gossiping starts decreasing the forwarding probability for a number of neighbors equal to 8 or higher. However stochastic flooding starts decreasing the probability from 11 neighbors. Up to 28 neighbors gossiping uses a lower probability than that of stochastic flooding. Therefore, we expect that the reachability of both protocols will be comparable but that the MNF of adaptive gossiping should be lower than the MNF of stochastic flooding for lower communication ranges.

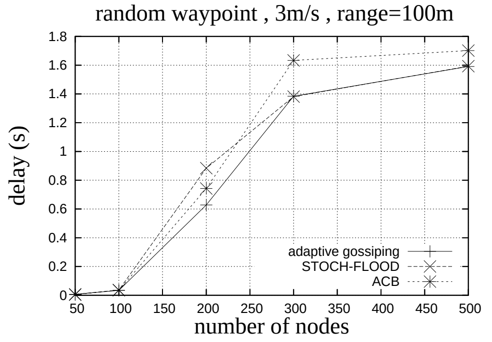
Study 1: Comparison with Regard to Mobility Models

As expected, simulation results show that both adaptive gossiping and stochastic flooding perform very comparably wit respect to reachability and delay. Except that the MNF of stochastic flooding is slightly higher than that of adaptive gossiping. This is valid for the random waypoint model (Fig. 4.22), the RPGM mobility model (Fig. 4.23), and the graph-based mobility model (Fig. 4.24).



(a) Reachability (rw)

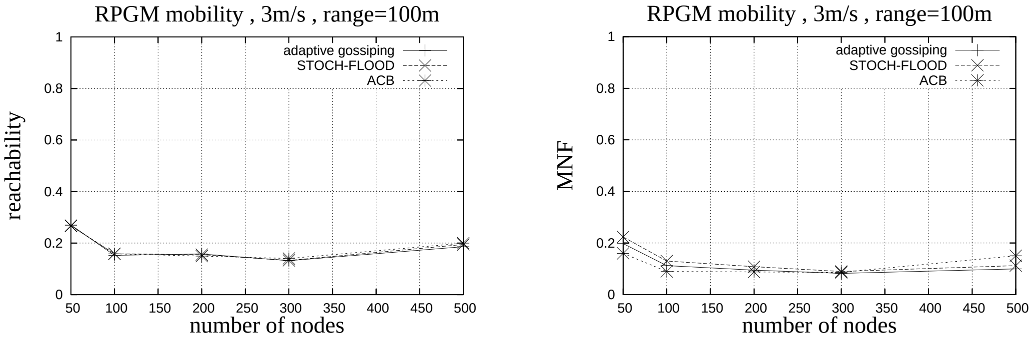
(b) MNF (rw)



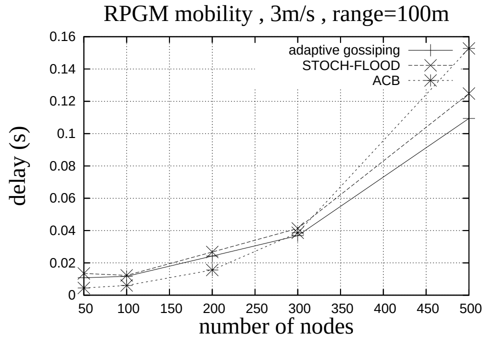
(c) Delay (rw)

Figure 4.22: Comparison of adaptive gossiping to related work (random waypoint)

In Fig. 4.24, we observe that for graph-based mobility ACB has a slightly higher reachability particularly for higher numbers of nodes (about 10% for 500 nodes). The message overhead is however much higher: For 500 nodes, ACB



(a) Reachability(RPGM) (b) MNF (RPGM)



(c) Delay (RPGM)

Figure 4.23: Comparison of adaptive gossiping to related work (RPGM)

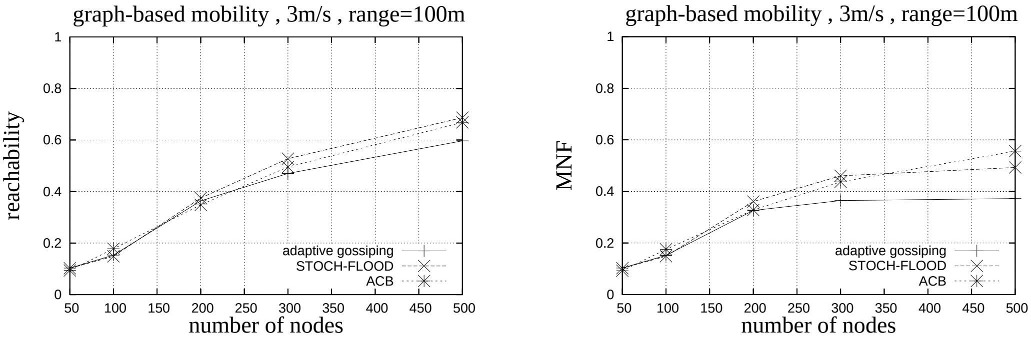
shows MNF=0.58 and gossiping shows MNF=0.39. For 500 nodes, the average delivery delay of ACB is 1.8s and that of gossiping is only 1s.

Study 2: Comparison With Regard to Communication Range

For this study, we fix the mobility model to RPGM and consider different communication ranges.

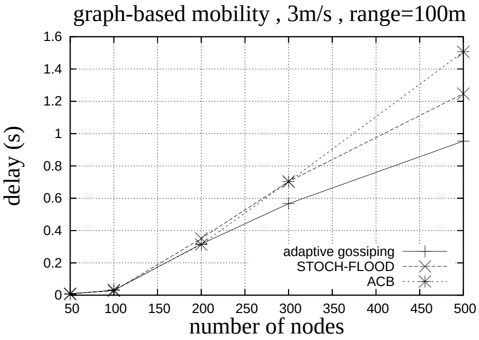
Simulation results that we do not include here show that for communication ranges less than 100m, the three protocols show almost the same performance. However, for higher range values , the performance differences become more clearly (Fig. 4.25). ACB shows in general slightly lower reachability and much higher MNF and delay compared to both stochastic flooding and adaptive gossiping. This is due to the fact that the dynamic ACB threshold is the same for all scenarios, where nodes have more than 12 neighbors (Fig. 4.21), which is obviously not well done.

Comparing stochastic flooding and adaptive gossiping, we conclude that adaptive gossiping saves slightly less forwards and reaches slightly less nodes. Noteworthy is that for scenarios with high communication range and number of neigh-



(a) Reachability

(b) MNF



(c) Delay

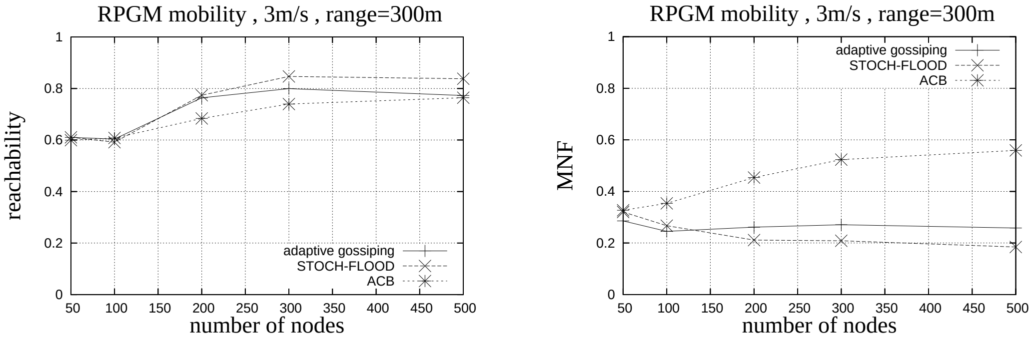
Figure 4.24: Comparison of adaptive gossiping to related work (graph-based)

bors, the number of neighbors of a certain node is possibly higher than 28. This corresponds to approximately 100 nodes for 300m communication range. Starting from this value adaptive gossiping begins to use higher probabilities than stochastic flooding.

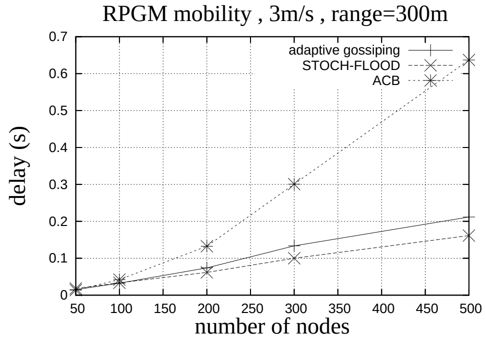
Summarizing, we can roughly conclude that adaptive gossiping shows a very comparable overall performance to stochastic flooding and that both protocols outperform ACB and particularly in highly dense scenarios. Between adaptive gossiping and stochastic flooding, we identify the following marginal differences. In extremely dense networks, stochastic flooding saves more forwards and reaches slightly more nodes than adaptive gossiping. However, in less dense scenarios adaptive gossiping saves more forwards and reaches slightly less nodes than stochastic flooding.

4.6 Summary

In this chapter, we applied the SI mathematical epidemic model for two simple broadcast algorithms, the SPIN-based protocol and gossiping. We showed that



(a) Reachability (b) MNF



(c) Delay

Figure 4.25: Comparison for 300m transmission range

MANET broadcasting techniques, using these simple epidemic algorithms, can be well modeled by the SI compartmental epidemic model and can be described by its main parameter: The infection rate. This analytical model is also suitable for other broadcast protocols, where nodes once received the message, they remain ready to forward it. Our approach to adopt the existing mathematical models for modeling broadcasting in MANETs presents a generic platform for the analysis of broadcasting in MANETs. This platform has gained a great success, since many works are referencing it [27, 35, 64, 66, 118–127].

We believe that this methodology is useful in obtaining insights into the performance implications of different broadcast schemes and MANET properties. Our methodology requires only a few simulations for the calibration of the model. Our main contribution is the establishment of a common platform to base further theoretical investigations on. A further contribution is the proof-of-concept of the applicability of epidemic analytical models for the adaptation of broadcast protocols.

We showed with the example of gossiping, how to use epidemic models to adapt broadcasting strategies in MANETs. We used the analytical epidemic model developed for gossiping to adapt the main parameter of gossiping, i.e. the forwarding probability, to the most relevant MANET characteristic, i.e. node density. The result is an adaptive broadcast protocol that adapts locally to the continuously changing node spatial distribution. Gossiping dynamically adjusts the forwarding probability only based on the number of neighbors, a locally available information, and without requiring any particular information, such as distance, position, or velocity.

Intensive simulations show that the dynamic selection of the forwarding probability reduces the total number of nodes forwarding a certain message, thus effectively alleviating the broadcast storm problem. Adaptive gossiping performs very comparably to the few adaptive broadcast schemes known from the literature. This shows the applicability of the analytical platform we developed for adaptation of MANET broadcast protocols. Similarly, our approach can be used to adapt other broadcasting strategies to the MANET characteristics that are relevant for the considered protocol.

Chapter 5

Hypergossiping: Our Generalized Broadcasting Technique

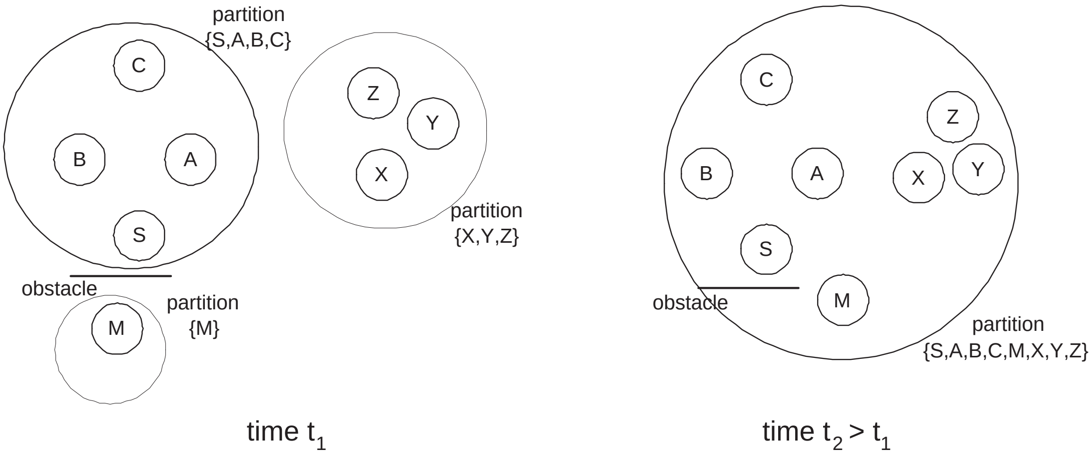
In this chapter, we first give a short motivation stating our objectives and the problems that we will cope with in order to provide a generalized broadcasting technique. Then, an overview of our solution fulfilling our requirements discussed in Chapter 2 is outlined. Afterwards, we present our primary contribution in this chapter, i.e. a novel broadcast repetition strategy. We provide the pseudo-code for hypergossiping and evaluate our approach for different mobility models reflecting a wide range of node densities and speeds. We also present a framework for ns-2 that simplifies the evaluation of MANET protocols with regards to network partitioning. We then use this framework to evaluate hypergossiping in more details and to compare its performance to the optimal case.

5.1 Motivation

As mentioned in Chapter 3, most broadcasting techniques are tailored to one class of MANETs with respect to node density and mobility and are not likely to operate well in other classes. The diversity of MANET applications and the continuously changing MANET connectivity, however, require a generalized broadcast strategy for a wide range of MANET operating conditions.

The adaptive gossiping approach that we developed in previous chapter presents a real step towards a generalized solution for broadcasting, since it efficiently distributes the messages in a broad range of connected MANETs with respect to node density and mobility. As gossiping shows a poor reachability in partitioned scenarios, gossiping can be complemented with a broadcast-in-time component that repeats broadcasting, in order to distribute messages across the network partitions. The main content of this chapter is a novel broadcast repetition strategy, which we combine with adaptive gossiping to realize our generalized broadcasting technique for MANETs. Our technique dynamically switches between both components depending on the local density, reflected by the number of neighbors.

Next we present a motivation scenario, and discuss the problems that our approach has to deal with. We consider the pedestrian scenario in Fig. 5.1, where a source node S broadcasts a message at time t_1, which we assume to remain relevant for the application up to a time point after t_2. At time t_1 we observe three partitions {S,A,B,C}, {X,Y,Z} and {M}. We assume that a broadcast-in-space scheme such as adaptive gossiping is implemented for broadcasting. If S initiates a broadcast at time t_1, it sends the message to all its neighbors using MAC broadcast. Receivers of this message or a subset of them (e.g. A) may *forward* the message. In this way, the broadcast reaches nodes A, B and C but does not reach the nodes X, Y, Z, and M. Now we assume at time t_2 node A *encounters* node X and node M encounters node S resulting in the join of all three partitions. If node A has buffered the message, node A should share it with node X, i.e *rebroadcast* it to node X. Similarly S should rebroadcast the message to node M. Finally, node X should *rebroadcast* the received message to its neighbors Z and Y according to the installed broadcast-in-space scheme.



(a) MANET is partitioned (b) MANET is connected

Figure 5.1: Motivation scenario for broadcast repetition

Besides network partitioning, broadcast storms may stop the broadcast from continuing. For example, if S and C start to send simultaneously, collisions may happen at A and B. In this case, the broadcast of S dies out at and the source S should rebroadcast the message. Noteworthy is that partition joins have to be detected only if there are messages to be rebroadcasted.

It is a challenging task to detect the appropriate partition joins and to rebroadcast the appropriate messages upon the detection of an appropriate join. In the following we present our novel approach.

5.2 Overview of the Hypergossiping (HG) Approach

Considering the MANET as a set of partitions that may join or split over time, and following our second requirement (Section 2.3.4), a generalized broadcast solution should combine one scheme for broadcast-in-space and one other for broadcast-in-time.

The analysis of the broadcast storm problem [7] and the performance comparison of adaptive broadcast-in-space techniques in previous chapter suggest adaptive gossiping for its simplicity and its high reachability and efficiency in connected MANETs independent from their node spatial distribution and mobility. Thus we select *adaptive gossiping* for broadcast-in-space.

The broadcast-in-time scheme should allow for an efficient broadcast repetition on partition joins. We call this strategy *broadcast repetition*. Our strategy consists of one heuristic that detects partition joins and one protocol that rebroadcasts the appropriate messages upon detecting a partition join.

After broadcast repetition the broadcast-in-space scheme can continue to distribute the message to the joining nodes (Fig. 5.2).

The broadcast application specifies the maximum delay it tolerates by setting the lifetime of the broadcast messages. Nodes have to buffer messages during their lifetime and to rebroadcast these messages or a subset of them on partition joins. Depending on the mobility of nodes, the node spatial distribution and the lifetime value, messages will succeed in reaching other partitions or not.

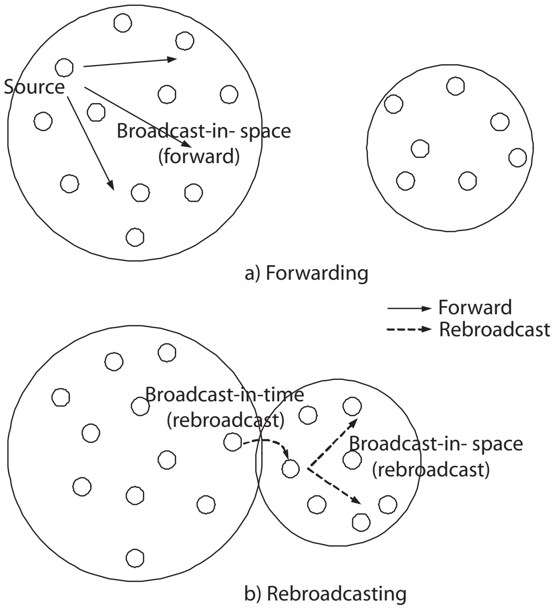


Figure 5.2: Hypergossiping approach

Accordingly, we extend the entries of the broadcast_table with the corresponding remaining lifetime. Nodes continuously decrement the lifetimes of originated or received packets. Nodes purge entries from the broadcast_table and possibly from the buffer, when the corresponding lifetime expires. When a message is forwarded or rebroadcasted, the remaining lifetime is included in the message.

5.3 Broadcast Repetition

The common strategy to overcome network partitioning is the repetition of forwarding, i.e. rebroadcasting. For this purpose nodes need two facilities: One to detect *when* to rebroadcast and a second to decide *what* to rebroadcast. In the following we introduce our novel partition join detection heuristic and rebroadcasting protocol.

5.3.1 Partition Join Detection

We allow nodes to share the IDs of recently received or locally originated messages with their neighbors. We call this list 'Last Broadcast Received' or LBR-list (Fig. 5.3). The rational behind this, is that two neighboring nodes that belong to the same partition should have received the same broadcasts that had taken place in this partition. By this way, if two nodes encounter each other, they are able to conclude whether they have been populating different partitions before the encounter. If a node receives an *LBR* that 'sufficiently' differs from its own LBR, the node can conclude with a certain confidence, that it is joining a new partition. We denote the maximum allowed size of an LBR by *maxLBRlength*. A node triggers rebroadcasting only if the overlap between the received LBR and its own LBR does not exceed a given percentage of its own LBR. We denote this percentage threshold as 'intersection threshold' (*IS_threshold*). In order to provide an accurate detection of partition joins, maxLBRlength and IS_threshold have to be dimensioned appropriately. In Subsection 5.5.3 we show how to calibrate maxLBRlength and IS_threshold.

The strategy above is suitable for detecting both causes of broadcast interruption: Network partitioning and broadcast storms. Firstly, if two partitions, say P1 and P2, join, some nodes of partition P1 will receive LBRs from other nodes belonging to the former partition P2. In this way nodes are able to detect the partition join event. Secondly, if a broadcast stops progressing within a partition due to collision or contention, nodes that received the broadcast may detect this on receiving the LBR of one neighbor that has not yet received the packet.

Our strategy is also suitable for MANETs, where nodes may disconnect and connect due to for example on-off usage or reboot. These nodes miss broadcasts taking place while they are unavailable. If a node reconnects, its LBR-list is empty. Therefore, a neighboring node, whose LBR list is not empty, is able to detect this kind of partitioning.

In order to save bandwidth, nodes exploit the existing HELLO beaconing to share their LBRs. Exchanging the LBRs is only necessary if a new neighbor is detected. Thus we do not include the LBR in each HELLO beacon but only in the beacon that directly follows the discovery of a new neighbor. This delays the broadcast repetition until the next discovery of a new neighbor, in the case of broadcast interruptions caused by broadcast storms.

5.3.2 Rebroadcasting

In Order to allow rebroadcasting, nodes must buffer messages that need to be rebroadcasted. If not otherwise stated, we assume that nodes buffer all received and originated messages as long as they are relevant, i.e. during their lifetimes. Accordingly, we suppose that $m = n$ in Fig. 5.3.

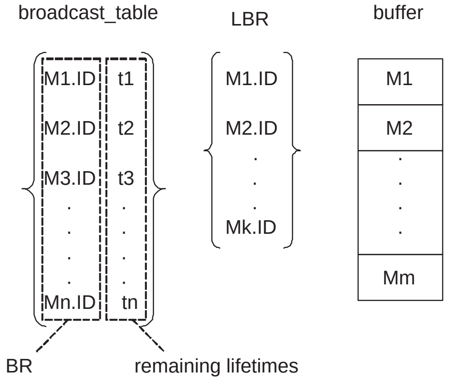


Figure 5.3: Definition of BR, LBR and buffer

A node triggers rebroadcasting by MAC-broadcasting a list of the broadcast IDs the node has received so far. We call this list 'Broadcast Received', or *BR-list* for short (Fig. 5.3). Thus neighbors know which packets the sender has already received and can select from the buffer the packets that this sender missed. On receiving these new packets the sender gossips them, so they can reach all joining nodes. To increase rebroadcasting efficiency, nodes do not rebroadcast immediately upon the reception of a BR-list, but schedule the rebroadcasting for a random time between 0 and *rDelay*. Nodes cancel rebroadcasting if one neighbor starts to rebroadcast before the scheduled time. To reduce the probability of collisions nodes do not rebroadcast all packets at once but wait a random time between 0 and fDelay before rebroadcasting the next packet.

5.4 Pseudo-Code for Hypergossiping

The pseudo-code description for hypergossiping is given by Algorithm 2.

Algorithm 2 Hypergossiping (HG)

```
Var: p, n, fDelay, IS_threshold, maxLBRlength, lifetime, rDelay
List: myLBR, myBR, broadcast_table
            ______On receiving a message (msg) M ______
    • if M is DATA do gossiping(p):
if M is received for the first time then
  deliver M
  insert {M.ID, remaining lifetime} to broadcast_table
  insert a copy of M to buffer
  if $myLBR.length < maxLBRlength$ then
    insert {M.ID} to myLBR
  else
    use FIFO to insert M.ID to myLBR
  end if
  $n \Leftarrow$ current number of neighbors
  $p = min(1.0, 0.175 + \frac{6.05}{n})$
  if $random(1.0) \leq p$ then
    wait (random(fDelay))
    send M to all neighbors
  end if
else
  discard M
end if
    • if M is HELLO with LBR do partition join detection:
$is \Leftarrow card(myLBR \cap recvLBR)/card(myLBR)$
if $is \leq IS\_threshold$ then
  send BR to neighbors
end if
    • if M is HELLO with BR do rebroadcasting:
set $timeout \Leftarrow random(rDelay)$
On receiving a DATA msg with $ID \in (myBR - recvBR)$ before timeout:
$exit()$
On timeout:
for all buffered msg M with $ID \in (myBR - recvBR)$ do
  $wait(random(fDelay))$
  rebroadcast M
end for
            ______On discovering a new neighbor ______
insert myLBR to next HELLO beacon
            ______On expiration of lifetime of msg M ______
if $M \in buffer$ then
  delete M from buffer
end if
if $M.ID \in myLBR$ then
  delete M.ID from myLBR
end if
delete the entry {M.ID, remaining lifetime} from broadcast_table
```

We denote by *card(X)*, a function that returns the number of elements of set X. The function *random(x)* returns a random float value $\in [0, x]$.

Lines 3-20 mainly describe adaptive gossiping. The code is similar to Algorithm 1 with some modifications. The forwarding probability p is now calculated as a function of the current number of neighbors n following Eq. 4.8 (Lines 12-13). The message entry for the broadcast_table contains in addition to the message ID the remaining lifetime of the message (Line 5). Furthermore the ID of the message is inserted to the LBR-list according to the FIFO approach (Lines 7-11). The message is also buffered (Line 6).

The broadcast repetition component of hypergossiping is mainly described by the rest of the code, i.e. Lines 21-40. The pseudo code for local partition join detection consists of Lines 21-24 and Line 33. The rebroadcasting protocol is given by the Lines 25-32. In Lines 34-40 we describe the tasks required if a message's lifetime expires.

5.5 Performance Evaluation

In this section, we first introduce the simulation settings and define our performance evaluation metrics. Afterwards, we calibrate the partition join detection. Then, we study the performance of hypergossiping and compare it to related work for a wide range of node spatial distributions, node mobilities and network load.

The performance evaluation metrics qualitatively describe the performance of the broadcast strategy. The determination of the distance to the optimal case is not possible, since the values of these metrics in the optimal case are impossible to compute without global partitioning information. Unfortunately, the network simulator ns-2 does not provide utilities for an easy evaluation of MANET protocols with regard to network partitioning. Therefore, we present in this section a framework that simplifies the access to the valuable partitioning information for ns-2 users. We then use the partitioning framework to evaluate our broadcast repetition strategy and to compare the performance of hypergossiping to that of the optimal broadcasting case.

5.5.1 Simulation Settings

We use the same simulation model as that we used for the evaluation adaptive gossiping (Section 4.5.1). The simulation settings are similar to that stated in Section 4.5.2 and summarized in Table 4.3. However, we fix the communication range to $100m$. Table 5.1 summarizes the supplemental simulation parameters of the experiments in this chapter.

We use a slightly modified communication load model. At the beginning of the simulation s from N nodes initiate broadcasting at a random time between

Parameters	Value(s)
Communication range	$R = 100$m
rDelay	10ms
Lifetime	$\in$ [5,1800] s
Packet rate	$\in$ [0.001,1] pkt/s
Number of senders	$s = 25$

Table 5.1: Simulation parameters for the evaluation of hypergossiping

1 and 3 seconds, and continue to send packets at a constant packet rate. This load model is suitable for the different MANET application scenarios, where data sources may send updates with a frequency that ranges from one update every hour to one update every second. For example, a room-temperature sensor may trigger an update of the room's temperature every hour, since the temperature normally does not vary so frequently in rooms. A vehicle, however, has to trigger an update of its velocity more frequently in order to allow jam recognition on a highway [44]. That is why we suppose that packet rates ranging from 0.001 to 1 packet/s (pkt/s for short) cover a wide range of our operation scenarios. We use a fixed lifetime value during a simulation, i.e. all senders use the same lifetime for all packets they generate.

For the performance evaluation we consider the first s packets generated; the subsequent packets generate background traffic. Simulations stop some seconds after the lifetimes of the first s packets expire. For the same simulation scenario, we ran 10 passes with 10 different movement traces and considered the average.

5.5.2 Performance Metrics

In Section 4.5.3, we defined the basic evaluation metrics for broadcast protocols. For the evaluation of hypergossiping we continue to use the reachability and the delay metrics to measure the delivery reliability and timeliness respectively. For the measurement of efficiency of hypergossiping we slightly modify the MNF metric to account for the additional number of transmissions implicated by the broadcast repetition strategy, i.e. rebroadcasts. We call the modified metric MNFR and define it as follows:
MNFR: Mean Number of Forwards and Rebroadcasts per node and message.

In order to allow for a separate evaluation of the two components building hypergossiping, i.e. adaptive gossiping and the broadcast repetition strategy, we define the following five sets of nodes with respect to a given broadcast message:

1. *Forwd*: Nodes that forward the message.

2. *Reb*: Nodes that rebroadcast the message.

Metric	Symbol	Value
Gossiping		
REachability	G_RE	$= \frac{card(R(G))}{N}$
Mean Number of Forwards	MNF	$= \frac{card(Forwd)}{N}$
Average end-to-end delay	G_delay	$= \frac{1}{card(R(G))} \sum_{i \in R(G)} (t_i - t_s)$
Hypergossiping		
REachability	HG_RE	$= \frac{card(R(HG))}{N}$
Mean Number of Forwards and Rebroadcasts	MNFR	$= \frac{card(Reb)+card(Forwd)}{N}$
Average end-to-end delay	HG_delay	$= \frac{1}{card(R(HG))} * \sum_{i \in R(HG)} (t_i - t_s)$
Rebroadcasting gain	gain	$= \frac{card(R(H))}{card(Reb)}$

Table 5.2: Evaluation metrics

3. *R(G)*: Nodes receiving the packet during the first round of the broadcast, i.e. only using forwarding and without any rebroadcasting.
4. *R(H)*: Nodes reached by means of rebroadcasting.
5. *R(HG)*: Nodes reached by HG, i.e. by means of either forwarding or rebroadcasting. This results in $R(HG) = R(H) + R(G)$.

Accordingly, we define reachability, delay and message efficiency of gossiping and hypergossiping separately. Given these values the corresponding values for the broadcast repetition are easy to conclude.

We define the following new performance metric to measure the benefit from broadcast repetitions: *Gain* is the mean number of additionally reached nodes per rebroadcast. Gain quantifies the benefit in reachability from one rebroadcast.

In Table 5.2 we illustrate our performance metrics for both gossiping and hypergossiping. We denote the origination time of the message by t_s and the arrival time of the message at node i by t_i.

5.5.3 Calibration of Partition Join Detection

To evaluate and calibrate the partition detection heuristic, we may use the global view given by the simulator. A metric for the accuracy of the heuristic is the ratio of correct detections to all detections. This ratio has to be maximized while dimensioning the heuristic. In this work, we evaluate the partition join detection by measuring the efficiency of rebroadcasting. A suitable efficiency metric for rebroadcasting is the mean number of additionally reached nodes per rebroadcast, i.e. its gain. The higher the gain, the more efficient the rebroadcasting is. In order to dimension the partition join detection parameters, i.e. IS_threshold and maxLBRlength, we select the values with maximum gain.

N	50	100	200	300	500
n	0.57	2.14	5.28	8.42	14.7
1 pkt/s	25%, 100	25%, 100	0%, $\geq$ 50	0%, 100	0%, 100
0.1 pkt/s	50%, 100	25-50%, 100	25-50%, $\geq$ 25	$\leq$ 50%, $\geq$ 25	0%, $\geq$ 50
0.001 pkt/s	25-75%, $\geq$ 25	$\leq$ 75%, $\geq$ 25	$\leq$ 75%, $\geq$ 10	$\leq$ 50%, $\geq$ 25	0%, $\geq$ 25

Table 5.3: Calibration of partition join detection

For calibration we arbitrarily use the random waypoint model, fix the lifetime at 60s and the maximum speed at 30m/s, and we vary the IS_threshold in {0%, 25%, 50%, 75%, 100%} and the maxLBRlength in {1, 5, 10, 25, 50, 100}. For every combination, we compute the gain and select the combination that maximizes the gain. The LBR-list of a node serves someway for the identification of its network partition until the next partition join. An identification should consider the size of the partition (reflected by number of nodes) and the number of broadcasts originated per unit of time in that partition (reflected by the packet rate). That is why we repeated the calibration process for a wide range of number of nodes and packet rates. The combinations, for which the gain is maximized, are listed in Table 5.3. We repeated these steps for lifetime values of 600s and 1800s and concluded that these combinations are almost independent from the lifetime value. We explain this as follows: Longer lifetimes mean a higher number of partition joins within the lifetime period. Since gain is a relative metric that measures the mean efficiency of the broadcast repetition strategy over all partition joins and if we assume that this efficiency remains almost constant for every join, we can conclude that the mean efficiency is independent from the number of joins and therefore independent from the lifetime. A variation of the maximum speed can be also interpreted as a variation of the lifetime, with respect to the calibration process. Since both variations lead to a variation in the number of partition joins within the lifetime period. We therefore conclude that the maximum speed also has no significant impact on the calibration process.

In Table 5.3 we observe that the denser the network or the more congested it is, the smaller the IS_threshold, but the higher the maxLBRlength that should be selected. In this work, we use a simplified calibration of partition detection. For MANETs with densities higher than 200 $nodes/km^2$, we use the tuple (0%, 100), otherwise we use the tuple (25%, 100). This calibration is suitable for most of simulated scenarios in Table 5.3. Consistent with our second requirement, we let every node select the IS_threshold value locally and independently : At run-time a node sets the IS_threshold to 25%, if its current number of neighbors is lower than 6, and 0% otherwise.

5.5.4 Performance of Hypergossiping

After adaptation and calibration we now evaluate hypergossiping for a wide range of node densities, mobilities and packet rates. Finally, we compare the performance of hypergossiping to that of IF, ACB and stochastic flooding. If maximum speed is 0 m/s, the nodes are static and do not discover new neighbors. Consequently, they do not trigger broadcast repetition and hypergossiping alters to simple gossiping.

Impact of Node Density and Mobility:

In this section, we investigate the performance of hypergossiping for a wide range of node densities and node mobilities.

Higher node mobility may lead to more frequent network partition joins. It then follows that the higher the mobility, the higher the reachability and the lower the delay. Fig. 5.4 shows that the impact of node mobility on reachability is more significant for lower lifetimes. For very short lifetimes, hypergossiping reachability is similar to that of simple gossiping. Fig. 5.4 illustrates that reachability saturates at around 80% for 50 nodes. We explain this as follows. At a packet rate of 0.0005 pkt/s every sender originates only 1 packet within lifetimes up to 2000s. Thus within the lifetimes considered in Fig. 5.4 only 25 messages are relevant in the MANET at any given time. LBRs can store all IDs of these messages. The partition condition (*overlap* $\leq 25\%$) becomes stronger over time and some partitions could not be detected. In [28] and Section 5.5.7, we investigate the accuracy of our partition join detection heuristic in more detail using the global view given by the simulator.

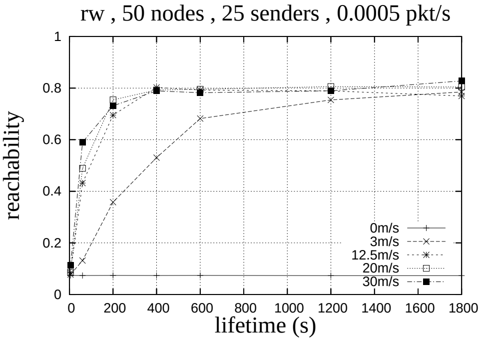
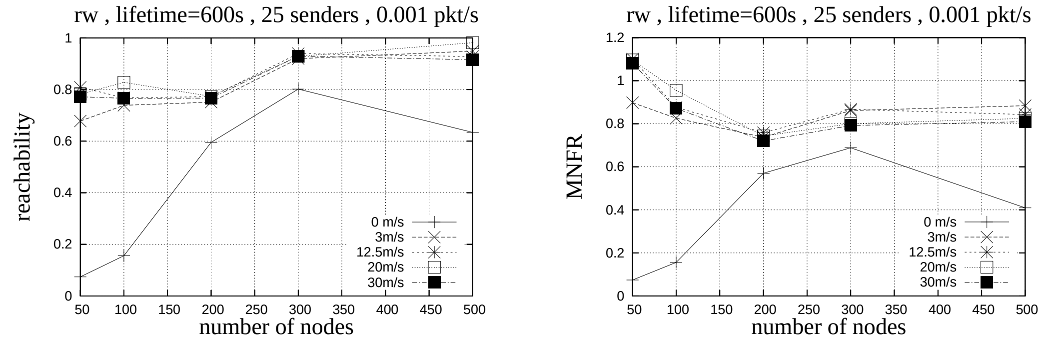


Figure 5.4: Impact of lifetime on reachability

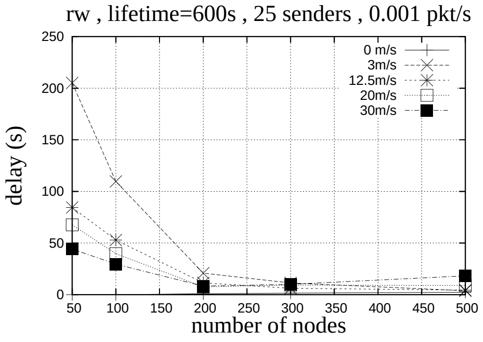
We now arbitrarily set the lifetime value to 600s and the packet rate to 0.001 *pkt/s*. This means that the packet's lifetime expires before the subsequent packet originates. Fig. 5.5 a) shows that rebroadcasting can strongly increase reachability in sparse MANETs; For 50 nodes and 3m/s, reachability increases from 8%

to 68%. Hypergossiping also increases reachability if gossiping reachability drops because of collisions; Reachability increases from 63% to 92% for 500 nodes and 3m/s. Hypergossiping keeps the MNFR very low while increasing reachability; Nodes forward and rebroadcast a given message maximum 1.1 times (Fig. 5.5 b)). The bend in reachability and MNFR at 200 nodes, in Fig. 5.5 a) and b), is due to our simple calibration of partition join detection, where IS_threshold jumps from 25% to 0% by node densities around 200 $nodes/km^2$.



(a) Reachability

(b) MNFR



(c) Delay

Figure 5.5: Impact of node density and mobility

Impact of Network Load:

In this section, we discuss the performance of hypergossiping for a wide range of packet rates. For this study, we consider 50, 300 and 500 nodes and a maximum speed of 30 m/s. The lifetime is arbitrarily set to 200 s. We vary the packet rate from 0.001 to 1 pkt/s.

Fig. 5.6 shows that even for high network traffic hypergossiping provides a high reachability. Furthermore, the reachability surprisingly increases if packet rate increases. Similar to the saturation effect in Fig. 5.4, we explain this effect as follows. If the packet rate increases, nodes update their LBRs at a faster rate,

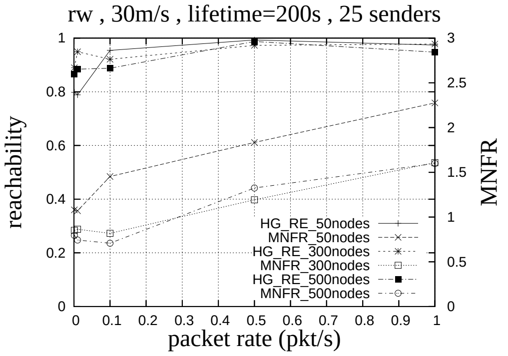
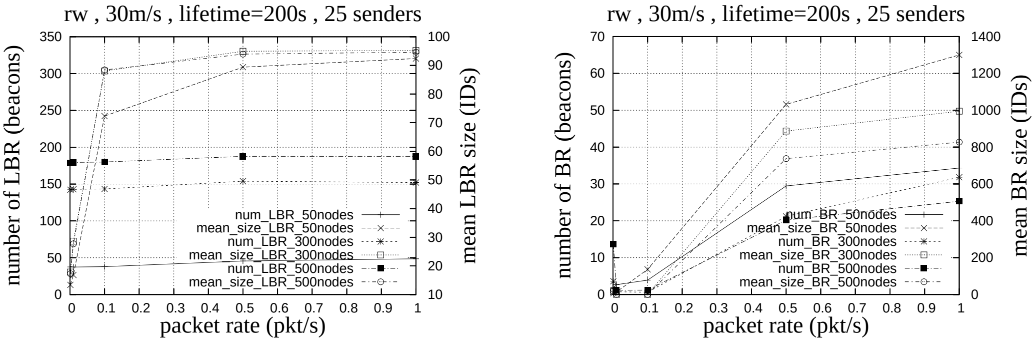


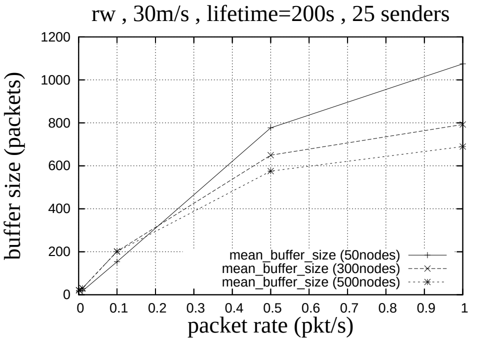
Figure 5.6: Hypergossiping RE and MNFR versus packet rate

thus the partition join detection condition gets weaker and the saturation effect less important. This also leads to higher MNFR.

Message and Storage Overhead:



(a) Number and size of LBR beacons (b) Number and size of BR beacons



(c) Buffer size

Figure 5.7: LBR, BR, and buffer overhead

We now investigate buffer size, BR size, LBR size, number of BR beacons and number of LBR beacons. For this study, we arbitrarily set lifetime to 200s and maximum speed to 30m/s, whilst varying packet rate and number of nodes. The number of BRs and the number of LBRs are counted within the simulation time, which is set to 250s. The buffer size and BR size are metrics for the storage overhead. BR- and LBR-size and -number quantify the additional bandwidth use, caused by the exchange of BR and LBR beacons.

The number of LBR beacons depends on the frequency of discovering new neighbors, which in turn depends on the node density and mobility of nodes (Fig. 5.7 a)). The number of LBR beacons increases with node density. It increases slightly with higher send rates because of more frequent collisions, which makes neighborhood information less accurate. The mean size of LBR increases with the increasing packet rate and node density but is limited by the maxLBRlength value, i.e. 100 IDs.

Similar to LBR, the size and number of BRs increase with the increasing packet rate (Fig. 5.7 b)). In contrast to LBR, size and number of BRs decrease with increasing node density. This is due to the calibration of partition join detection, which uses stronger detection conditions (0%) for dense networks and thus triggers fewer BR-beacons. Fewer BR beacons leads to higher delays and thus lower mean BR sizes.

Fig. 5.7 c) illustrates buffer size versus packet rate. The mean buffer size is computed over time and over all nodes. The buffer size increases with packet rate. For high packet rates the buffer size decreases with node density but for lower send rates it increases with density. This is due to collisions, whose number increases with increasing packet rates.

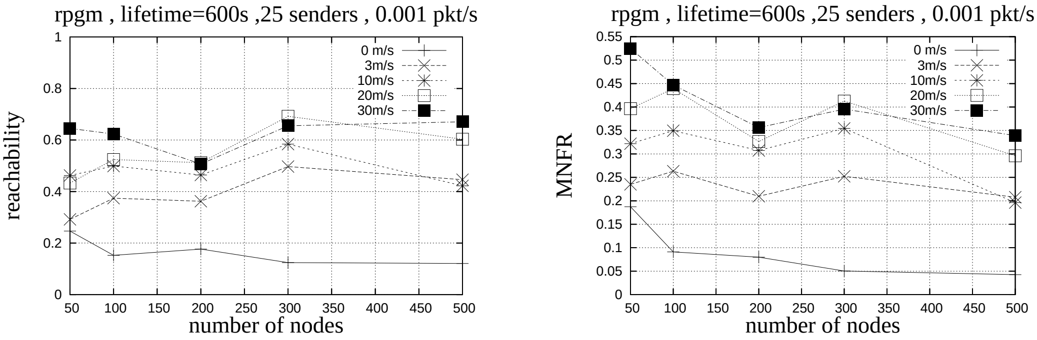
In Chapter 6, we show how we reduce buffer overhead, by designing a mobility-aware buffering strategy for hypergossiping.

Impact of Mobility Models:

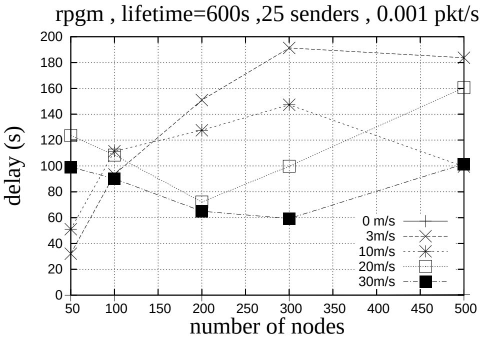
We now investigate the performance of hypergossiping for further mobility models. We first present the performance for the RPGM model and then for the graph-based mobility model. For all studies, we arbitrarily set the packet rate to 0.001 pkt/s.

RPGM: Fig. 5.8 illustrates the performance of hypergossiping for MANETs where nodes move according to the RPGM mobility model. Also for RPGM, hypergossiping succeeds to increase the reachability with a reasonable number of broadcast repetitions, since the MNFR reaches a maximum of 0.55. The increase in reachability depends on node speed, but even for low speed values, we observe an increase in reachability.

Compared to the random waypoint model, hypergossiping shows a lower reachability for almost all number of nodes and speeds. To explain this effect, we



(a) Reachability (b) MNFR



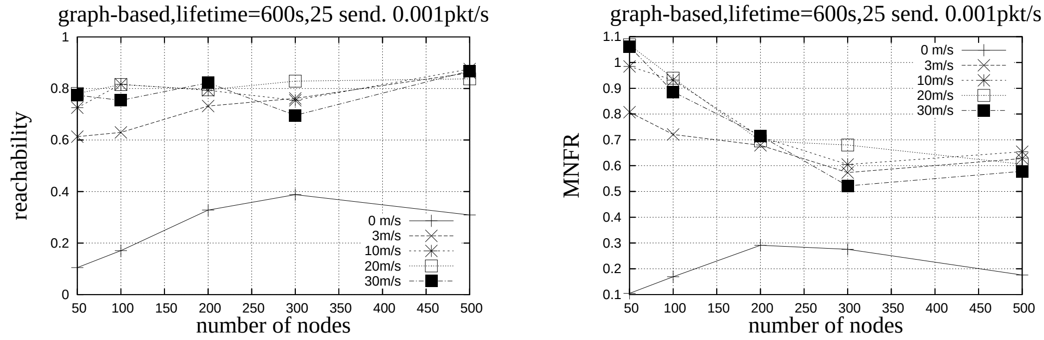
(c) Delay

Figure 5.8: RPGM mobility

consider the number of partitions presented in Fig. 4.13 a)b) for a 100m communication range. In contrast to the random waypoint where the number of partitions decreases with the increase of number of nodes, for RPGM the number of partitions increases with the increase of number of nodes. Also we note that for RPGM, nodes of a group are at maximum 10m far from the group center and therefore the group members detect each other as neighbor for a communication range of 100m. Since the average number of group size is 10, nodes have in average at least 10 neighbors, therefore they set 0% for the IS_threshold, which makes the partition join detection impossible for two joining partitions, which have received at least one common broadcast message.

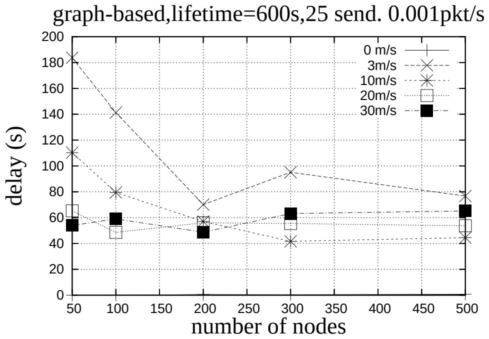
Graph-Based Mobility Model: Fig. 5.9 presents the performance of hypergossiping for the graph-based mobility model. Also for the graph-based mobility model the broadcast repetition strategy of hypergossiping improves massively the performance of adaptive gossiping. The number of rebroadcasts required for this improvement remains acceptable, since MNFR reaches a maximum of 1.07.

Hypergossiping performs very comparably for both the random waypoint model and the graph-based model. Except for a high number of nodes, hypergossiping



(a) Reachability

(b) MNFR



(c) Delay

Figure 5.9: Graph-based mobility

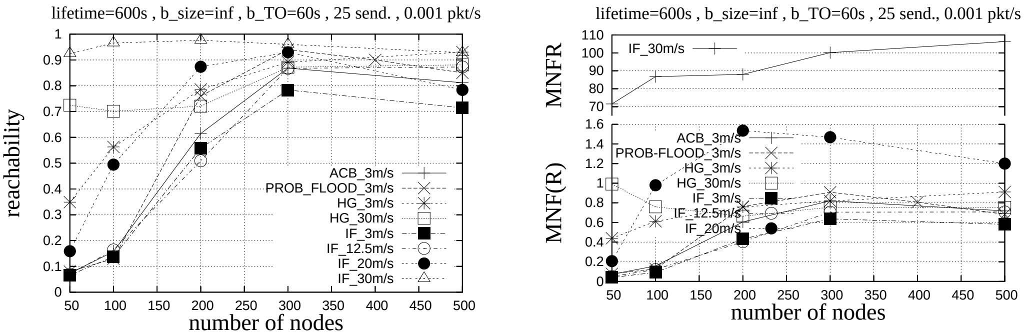
reaches slightly less nodes in graph-based mobility model. This is due to the fact that the graph-based model shows more number of partitions than the random waypoint model for the same scenario configuration.

5.5.5 Comparison to Related Work

Now we compare hypergossiping to the stochastic flooding (STOCH-FLOOD) [24], to the adaptive counter-based scheme (ACB) [23] and to the integrated flooding (IF) [9, 10]. For this study, we set the packet rate at 0.001 pkt/s. We arbitrarily use the random waypoint mobility model.

For IF, we use the following parameters (stated in [10]): In scoped flooding a node forwards a newly received message, if at least 15% of its neighbors are not covered by neighbors that have forwarded the message. We note here that scoped flooding is a topology-based scheme that requires 2-hop topology information, which means that a node has to include its neighbor list in each HELLO beacon. Hyperflooding holds packets for a fixed time period and rebroadcasts them on discovering a new neighbor. Nodes install scoped flooding if their relative speed to all their neighbors is lower than $10m/s$. If the relative speed is higher than

25m/s hyperflooding is selected. Otherwise plain flooding is deployed. We note that for maximum speed values up to 12.5m/s, nodes will never reach the higher switching threshold of IF (i.e. 25m/s) and thus hyperflooding will never be installed in such a configuration. For higher speed values some nodes may install hyperflooding mode, which should increase the reachability of IF. For maximum speed equals 3m/s, the maximum relative node speed is 6m/s, that is why only scoped flooding is installed by IF.



(a) Reachability

(b) MNF(R)

Figure 5.10: Comparison of hypergossiping to related work

For hypergossiping, we fix the lifetime to 600s and use the same buffering strategy as IF. IF buffers all broadcast messages for a given fixed time, called *buffer_timeout*. We arbitrarily set the value of buffer_timeout to 60s. ACB and STOCH-FLOOD demonstrate almost mobility-independent performance and thus we only present the results for 3m/s for these protocols.

We can easily conclude from Fig. 5.10 a) that hypergossiping reachability outperforms STOCH-FLOOD and ACB reachability for sparse networks and highly dense networks whilst maintaining MNFR below 1. This is because HG remedies both causes of broadcast interruption: Network partitioning and broadcast storms.

Comparing HG and IF, we conclude that IF does not provide an efficient partition join detection strategy: IF provides a very high reachability for highly mobile MANETs, but MNFR ranges from 72 to 106 rebroadcasts per packet and node. Simulation results, which we do not present here, show that even for very low buffer timeout values, MNFR is very high for highly mobile scenarios: For example, for *buffer_timeout* $= 5s$ and $N = 100$ nodes, MNFR is 11. For lower mobility, IF installs only scoped flooding and subsequently performs similarly to ACB and STOCH-FLOOD for sparse networks but worse than these schemes for dense networks, as there is a lack of adaptation on the part of scoped flooding to node density.

5.5.6 Providing Global Partitioning Information for ns-2

In this section, we compute network partitioning information for arbitrary movement patterns and provide an interface, which satisfies the most important needs of developers of (partition-aware) MANET protocols and applications developers.

Approach

GOD (General Operations Director) is a central omniscient instance of ns-2 [11]. GOD stores global state information. The GOD instance implemented by the current ns-2 version manages the shortest path information between nodes. This global information is used by MANET routing protocol developers.

We follow a similar approach to that for generating GOD information for MANET routing. We first annotate movement trace files with basic partitioning information. The GOD instance then loads this information at the simulation begin. During simulation GOD aggregates partitioning information and generates dynamic partitioning information. For MANET developers GOD provides a generic interface that simplifies the use of partitioning information during simulation (Fig. 5.11).

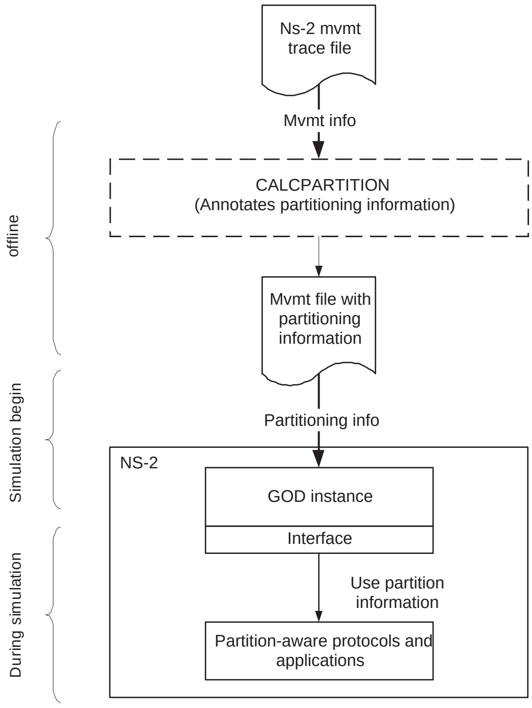


Figure 5.11: Providing partitioning information for ns-2

The annotation is performed offline. This increases the reusability of trace files, since the annotation only depends on the node movement, provided that the communication range remains constant. This also reduces simulation run-time, since calculating network partitioning information can be time-consuming. The partitioning information generated is calculated based on a fixed communication model similar to that of ns-2, where we assume that nodes can communicate if their distance is below a given constant value R. This makes the annotated partitioning information invalid, if nodes fail or change their communication range at run-time.

Annotation

An annotation is a set of OTcl commands for ns-2. We have to annotate complete partitioning information, while keeping the size of trace file as small as possible. However, we also have to generate as many statistics as possible offline, in order to keep simulation time as short as possible.

The tool `calcpartition` is able to annotate every ns-2 movement trace file, independently from the mobility model and from the generation tool. The path of the movement trace file, as well as the nominal communication range are provided as arguments for `calcpartition`. The tool generates partitioning information according to the ns-2 network model, i.e. an undirected graph, where an edge between two nodes is added if their distance is below R.

One approach for providing partitioning information is to annotate the partitions constituting the MANET at the beginning of the scenario and at each subsequent time, when this set of partitions will change, i.e. at each future join or split event. In order to reduce the size of this information we proceed as follows. We first annotate the initial partition composition of the MANET and then we annotate only the event type and the resulting partition(s).

We annotate the initial partitions constituting the MANET by means of OTcl commands in the following format:

$god_ set-part <partition-list>
*<partition-list>= {<node-list>} ({<node-list>})**
*<node-list> = <node-ID>(, <node-ID>)**

Set-part is a function that we added to the GOD class. The subsequent partition join and split events are annotated using the following commands respectively:

$ns_ at <time> "$god_ set-join <new-partition>"
$ns_ at <time> "$god_ set-split <partition-list>"

Set-join and set-split are two functions, which we also added to the GOD class. For each event, we only annotate the nodes that are concerned with that event. This decreases the size of the annotated trace files.

Consider the following example of annotation:

```
$god_ set-part { 0, 1 } { 2 } { 3, 4 }
$ns_ at 4.26 "$god_ set-join { 0, 1, 2 } "
$ns_ at 6.54 "$god_ set-split { 3 } { 4 } "
```

This example shows that at the beginning of the scenario the MANET is formed by three partitions. The first partition contains nodes 0 and 1. The isolated node 2 represents the second partition. Nodes 3 and 4 form the third partition. At time 4.26s, the partitions { 0, 1 } and { 2 } join. At time 6.54s, the partition { 3, 4 } splits into two partitions.

At the end of the scenario file, `calcpartition` lists some helpful statistics such as the number of joins, the number of splits, the average, minimum and maximum partition size, the average number of partitions, and the average time to next split or join (in ms). These statistics are valid for the movement scenario time and are printed out as OTcl comments.

GOD Interface

After loading the annotated partitioning information from the scenario file, GOD prepares this information for protocol developers. In order to be able to do this, GOD provides an interface for ns-2 users (Fig. 5.11).

This interface is generic and easy to use. It provides sufficient functions to satisfy the needs of the ns-2 users. Protocol developers need to debug and evaluate their protocols by observing important indicators or by defining optimal protocols to compare their performance to the optimal case.

For the evaluation of hypergossiping the following partitioning information is valuable. As we showed earlier partition size is crucial for understanding the performance of adaptive gossiping. The partition join events and the nodes that realized the join are important to evaluate the broadcast repetition strategy of hypergossiping.

We distinguish two main classes of possible needs of protocol developers. A developer may be interested in certain instantaneous or statistical values or in certain split or join events. Therefore, our GOD interface for partitioning information provides two user modes. Firstly, the Query-Interface allows users to query instantaneous or statistical partitioning information. Secondly, the Subscribe-Interface allows nodes to subscribe to and then to unsubscribe from partition join or split events.

Query-Interface: The query-interface processes the requests of nodes for partitioning information. Developers may need instantaneous or statistical information.

- Instantaneous partitioning information describes the current MANET partitioning topology. Currently, the following functions are implemented:
  - *getNumberOfPartitions()*: returns the current number of partitions.
  - *getNodesOfPartition(node$_i$)*: returns the IDs of nodes that constitute the partition that contains $node_i$.

  - *getPartitionSize(*$node_i$*)*: returns the size of the partition containing $node_i$.
  - *belongToSamePartition(*$node_1$*,*$node_2$*)*: checks whether $node_1$ and $node_2$ belong to the same partition.

- Statistical partitioning information can be calculated over time, partitions or nodes [128]. Statistics over time are performed between two past points of time t_1 and t_2. We currently provide the following functions:

  - *getAverageNumberOfPartitions(*t_1*,*t_2*)*: returns the average number of partitions during the time interval between t_1 and t_2.
  - *getAveragePartitionSize(*t_1*,*t_2*)*: returns the average size of partitions over the time interval between t_1 and t_2.
  - *getMinPartitionSize(*t_1*,*t_2*)*: returns the minimum partition size between t_1 and t_2.
  - *getMaxPartitionSize(*t_1*,*t_2*)*: returns the maximum partition size between t_1 and t_2.
  - *getPartitionChangeRate(*$node_i$*,*t_1*,*t_2*)*: returns the partition change rate of $node_i$, i.e. the number of join or split events that the partition of $node_i$ experiences between t_1 and t_2.
  - *getAveragePartitionChangeRate(*t_1*,*t_2*)*: returns the average partition change rate over all nodes.
  - *getSeparationTime(*$node_i$*,*$node_j$*,*t_1*,*t_2*)*: returns the cumulative time between t_1 and t_2, during which the nodes $node_i$ and $node_j$ do not belong to the same partition.
  - *getConnectionTime(*$node_i$*,*$node_j$*,*t_1*,*t_2*)*: returns the cumulative time between t_1 and t_2, during which the nodes $node_i$ and $node_j$ belong to the same partition.
  - *getNumberOfJoins(*t_1*,*t_2*)*: returns the number of partition joins between t_1 and t_2.
  - *getNumberOfSplits(*t_1*,*t_2*)*: returns the number of partition splits between t_1 and t_2.

Subscribe-Interface: This interface propagates partition events to the interested nodes. Nodes can subscribe to or unsubscribe from partition join or split events. The current subscribe interface provides the following major functions:

- *subscribeJoin()*: Allows nodes to subscribe to all join events. The subscribers receive a notification each time a partition join occurs. The join notification mainly contains the nodes constituting the joining partitions, and the IDs of both nodes that realized the join.

- *subscribeSplit()*: Allows nodes to subscribe for all split events. The split notification mainly contains the nodes constituting the resulting partitions.
- *unsubscribeJoin()*.
- *unsubscribeSplit()*.

5.5.7 Evaluation of HG Using Global Partitioning Information

So far, we have discussed the partitioning framework. We now show the usability of global partitioning information provided, for the evaluation of protocols and applications in ns-2. For this, we experiment with the behavior of hypergossiping, if the GOD information is used to provide nodes with perfect partitioning information. We use global partitioning information in order to determine the optimality of the broadcast repetition strategy, concerning network partitioning.

Global Information Needed: We require two kinds of global information for the global evaluation of hypergossiping. Firstly, we need a global view concerning network partitioning. Gossiping is mainly interested in the partition size information, since its reachability is dependent on the partition size, where it takes place. The broadcast repetition strategy of hypergossiping is mainly interested in partition joins and in the nodes that caused the joins. These nodes eventually have to initiate the rebroadcasting protocol. This first global view is easily provided by the framework presented in Subsection 5.5.7. Secondly, we need a global view in terms of the broadcasting state, i.e. the knowledge about the spreading of broadcast messages at every point of time. We require this global view for hypergossiping, firstly, to validate whether gossiping reaches all nodes within a single partition, and secondly, to determine the messages that should be rebroadcasted on partition join. This knowledge is also required for the evaluation of the partition join detection heuristic. A partition join should only then be detected by the heuristic if some rebroadcast messages need to be rebroadcasted on this join.

Evaluation of Broadcast Repetition: Using global view in terms of network partitioning and broadcasting state, we now perform two studies to evaluate the broadcast repetition strategy of hypergossiping. The goal of the first study is to determine the optimal values for the performance metrics (reachability, Delay and MNFR) and thus to show the quality of the broadcast repetition strategy. The purpose of the second study is to count the correct, wrong and redundant decisions of the broadcast repetition strategy.

- Study 1: Optimal Broadcast Repetition

For this study, we define the following new protocol: HG with optimal broadcast repetition, *HG-OBR* for short. The gossiping implementation is not modified. Optimal broadcast repetition means optimal partition join detection and optimal rebroadcasting protocol. Optimal partition join detection is easily given by the partitioning GOD interface. Nodes simply have to subscribe to join events. Optimal rebroadcasting is given if both nodes, which triggered the partition join, rebroadcast the messages that the other node has not yet received. Optimal rebroadcasting does not send messages through MAC, but calls the receive procedure of the opposite node and waits for *fDelay* until the next call for next message. Note that optimal rebroadcasting prohibits redundant rebroadcasts.

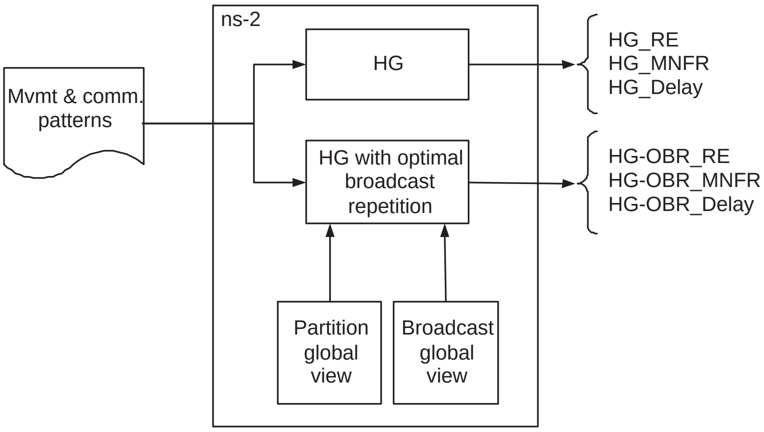


Figure 5.12: Optimal broadcast repetition (study 1)

We define HG-OBR_RE, HG-OBR_MNFR and HG-OBR_Delay as the values of the RE, MNFR and Delay of HG-OBR, respectively. Thus the broadcast repetition strategy of HG can be compared qualitatively to the optimal case. To enable a fair comparison we use the same movement and communication patterns for both protocols (Fig. 5.12). For random number generation we use the same seed in order to increase the similarity of both scenarios.

This approach has two main advantages. Firstly, we do not require new evaluation metrics. Secondly, the approach provides an aggregated (coarse-grained) measurement of the distance to the optimal broadcast repetition.

Nevertheless, this approach shows two main drawbacks. Firstly, it does not allow an easy back trace to the deficiencies of the broadcast repetition strategy; it is not clear if the difference to the optimal case is due to the partition join detection heuristic or to the rebroadcasting protocol. Secondly, two simulation runs are required; One run for HG and another run for HG-OBR.

- Study 2: Hypergossiping Observation

This approach is an online monitoring of HG with respect to network partitioning (Fig. 5.13). Using the partitioning and the broadcast global view, we validate each decision made by the broadcast repetition strategy, i.e. each decision made by the partition join detection and each decision made by the rebroadcasting protocol.

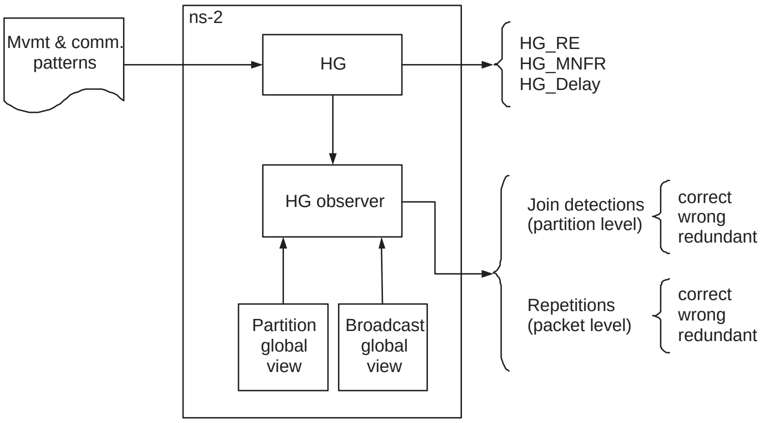


Figure 5.13: Hypergossiping observation (study 2)

This approach allows for a fine-grained evaluation of the broadcast repetition strategy. It also permits an easier back trace of the weak points of the broadcast repetition strategy, since separate evaluations of the partition join detection heuristic and the rebroadcasting protocol are given. Furthermore just one single simulation run is needed.

For this study, we define new metrics. We consider two levels for the definition of these metrics: the partition level and the packet level. On the partition level we define metrics that validate the decisions made by the partition join detection heuristic. On the packet level the metrics validate the decisions made by the rebroadcasting protocol.

1. Partition level: At this level we count the correct and wrong partition join detection decisions as well as the redundant decisions, i.e. decisions made by more than one node from each partition. We assume two partitions, say P_1 and P_2, join and that node n_1 from P_1 and node n_2 from P_2 triggered the join. We define the following three metrics:

    a) *Correct Detections (CD)*: We increment CD, if node n_1 (resp. n_2), has to rebroadcast a list of packets HL_1 (resp. HL_2) and at least one node of P_1 (resp. P_2) detects the partition join. We also increment CD, if node n_1 (resp. n_2) has no packets to rebroadcast and no node of P_1 (resp. P_2) detects the join.

    b) *Wrong Detections (WD)*: We increment WD, if node n_1 (resp. n_2), has to rebroadcast a list of packets HL_1 (resp. HL_2) and no node of P_1 (resp. P_2) detects the partition join. We also increment WD, if node n_1 (resp. n_2) has no packets to rebroadcast and at least one node of P_1 detects the join.

   c) *Redundant Detections (RD)*: We increment RD if more than one node makes a correct detection.

2. Packet level: At this level we validate each decision to rebroadcast or not a buffered message, whether it is correct, wrong or redundant. Similar to the metrics at partition level, we define the following three metrics.

   a) *Correct Repetitions (CR)*

   b) *Wrong Repetition (WR)*

   c) *Redundant Repetition (RR)*

Algorithm 3 shows the pseudo-code for the hypergossiping observer, which monitors hypergossiping and increments the metrics CD, WD, RD, CR, WR and RR.

Algorithm 3 HG observer

```
On partition join $P_1$ with $P_2$:
for all $i \in \{1, 2\}$ do
  if $P_i$ have to rebroadcast a set of packets HLi then
    if Pi_detected_join then
      CORRECT_DETECTION++
      for all $packet \in HLi$ do
        if rebroadcasted then
          CORRECT_REPETITION++
        end if
        if redundant_rebroadcasted then
          REDUNDANT_REPETITION++
        end if
        if not_rebroadcasted then
          WRONG_REPETITION++
        end if
      end for
    else
      WRONG_DETECTION++
    end if
  else
    if Pi_detected_join then
      WRONG_DETECTION++
      if a packet is rebroadcasted then
        WRONG_REPETITION++
      end if
    else
      CORRECT_DETECTION++
    end if
  end if
end for
```

Simulation Results

Our simulation settings mainly rely on the settings shown in Table 5.1. For this study, however, we arbitrarily use the random waypoint mobility model. We set the packet rate to 0.01 pkt/s, and the lifetime value to 200s. The simulation time is fixed at 250s.

In the following, we present the simulation results for the global evaluation of hypergossiping. We also show the usefulness of information that we provide in GOD instance, to understand the behavior of hypergossiping and to identify opportunities for improvement in terms of hypergossiping.

Next, we show the simulation results for the global evaluation of the broadcast repetition strategy.

Study 1 - Optimal Broadcast Repetition Fig. 5.14 shows HG_RE and the values for HG_RE, if the broadcast repetition is optimal, i.e. HG-OBR_RE. For 30 m/s and starting from 100 nodes, the HG-OBR_RE slightly decreases for increasing numbers of nodes; this is due to collisions (we use the real implementation of gossiping). As expected the HG-OBR_RE is higher than the HG_RE. However, for 300 nodes, where the MANET consists of a very large partition and some isolated nodes, hypergossiping reaches only a few more nodes than hypergossiping with optimal broadcast repetition. This is due to the fact that the broadcast repetition of hypergossiping is able to remedy the gossiping breaks caused by collisions, in addition to overcoming network partitioning [25].

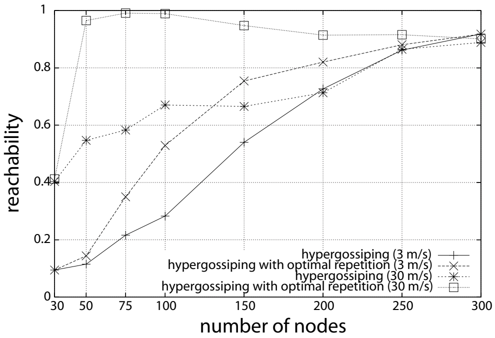
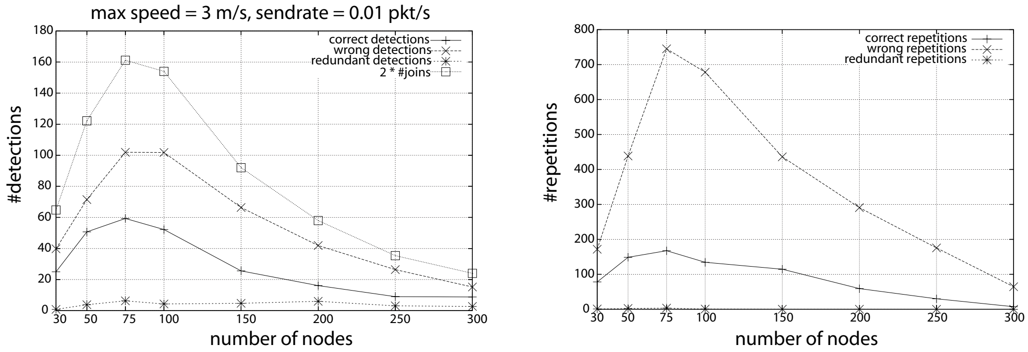


Figure 5.14: Optimal broadcast repetition

The comparison with the optimal case shows that there are some possible improvements, in order to increase the reachability of hypergossiping in sparse MANETs. The improvement potential is higher for higher mobility.

Study 2 - Hypergossiping Observation: Fig. 5.15 a) shows the number of the correct, wrong and redundant detections of the broadcast repetition strategy, as well as the doubled number of joins. In Fig. 5.15 b) we present the number of correct, wrong and redundant broadcast repetitions. Fig. 5.15 a) demonstrates that the number of correct and wrong detections correlates well with the doubled number of joins. We notice that the number of wrong detections is relatively high. Simulation results show that these are mainly the joins that were not detected by the heuristic. We also notice that the number of redundant detections and the number of redundant broadcast repetitions respectively, are very low. This means that our detection heuristic and suppression mechanism, work well in minimizing redundant detections and repetitions respectively.



(a) Correct, wrong and redundant detections

(b) Correct, wrong and redundant repetitions

Figure 5.15: Hypergossiping observation

The observation of the broadcast repetition shows that there are some possible improvements for the partition join detection heuristic. It is necessary to increase the number of correct detections, in order to increase the reachability of hypergossiping efficiently.

5.6 Summary

In the current chapter, our generalized broadcast techniques hypergossiping has been sketched. Hypergossiping provides an adaptive broadcast strategy combining a broadcast-in-space technique, so as to reduce broadcast storms, and a broadcast-in-time repetitive strategy, in order to overcome the problems associated with network partitioning. In contrast to the IF scheme [9,10], which adapts to node mobility, hypergossiping uses the local node density as the main criteria for adaptation.

The observation of the increasing need for partitioning information while developing MANET protocols and applications, encouraged us to provide generic partitioning information for the widely used network simulator ns-2. We provide a utility, `calcpartition`, to annotate arbitrary movement files with partitioning information and an interface to easily use this information during simulation.

We release the source code of `calcpartition` and a patch for the required ns-2 modifications for ns-2 community.

We showed the applicability of the provided information using the example of hypergossiping. This global evaluation showed some improvement potentials for hypergossiping, which can be considered in future work.

Chapter 6

Contact-Based Mobility Metrics and Buffering

In this chapter, we first define a set of novel mobility metrics that can be very useful for mobility-assisted, delay-tolerant ad hoc networking. Then, we use some of these metrics to efficiently detect relevant mobility patterns and to accordingly adapt the buffering strategy of hypergossiping to the mobility of nodes.

6.1 Introduction

As mentioned earlier mobility of network nodes plays a major role in MANETs. On the one hand, it stresses networking tasks by disrupting routes, changing the signal propagation characteristics and causing network partitioning. On the other hand, *mobility-assisted* protocols take advantage of mobility to transport messages to other nodes (store-and-forward), in order to increase network capacity or to overcome the problem of network partitioning. Nodes buffer messages and monitor the network condition, in order to retransmit them whenever the destination becomes (more easily) reachable. To allow for this commination paradigm, the applications have to be *delay-tolerant*.

Hypergossiping is a broadcast protocol that profits from the mobility benefits to maximize the reachability and efficiency for delay-tolerant applications by overcoming network partitioning.

So far, we have suppressed the storage constraints on mobile nodes and investigated the performance of hypergossiping, assuming sufficient buffering space. Currently, hypergossiping uses the following simple buffering strategy. The protocol allows all nodes to buffer all received messages as long as they are still relevant, i.e. for the residual lifetime. This strategy, however, may produce a high buffering overhead, which is not practical for devices with limited resources or applications that frequently initiate broadcasting of data with high lifetimes. Therefore, we need new strategies that reduce the buffer overhead of hypergossiping. Our overall objective is to design a buffering strategy with lower demands on buffer space, without or with low degradation of reachability.

Although node mobility is a key issue for mobility-assisted protocols, we observe the lack of mobility metrics that help understanding the mobility on a larger time scale. These metrics can simplify the design and adaptation of mobility-assisted ad hoc protocols and applications. The metrics should quantify the mobility of nodes on a large time scale, i.e. for time periods in the range of minutes, hours, or even days.

Existing mobility metrics, e.g. link change rate, have been primarily designed for non-delay-tolerant ad hoc routing protocols. They model the mobility instantaneously and do not quantify it on a large time scale.

Consider a campus scenario consisting of students and staff members. The variety of movement patterns of both groups on a large time scale, e.g. one working day, can be easily observed. Students frequently roam between different departments, faculties, libraries and cafeterias. Staff members however show a much lower move-to-pause ratio than students and intensively interact locally with neighboring staff members and sometimes with visiting students. To detect such large time scale mobility patterns, we have to investigate the mixture of mobile nodes, i.e. the relative mobility of nodes on a large time scale.

6.2 Contact-Based Mobility (cbm) Metrics

The main content of this section is the definition of novel mobility metrics that quantify large time-scale mobility and can help network protocol developers to easily evaluate their delay-tolerant protocols or to adapt them to node mobility. Our approach is based on the pair-wise encounters between mobile nodes and defines mobility metrics that quantify the mixture level of mobile nodes. In order to provide easy access to these metrics in a network simulator, we finally provide a framework for ns-2.

6.2.1 Existing Mobility Metrics

In this section, we briefly revise the existing mobility metrics. We classify existing metrics into three groups: First, *velocity-based* metrics, such as average speed and average relative speed between all nodes; secondly, *link-based* metrics, such as link change rate [129] and link duration [130, 131]; and thirdly *route-based* metrics, such as route change rate [132], route duration [131] and average path availability [131].

These metrics are mainly designed for non-delay-tolerant MANET routing protocols, where route information with higher mobility becomes stale more quickly [129]. Unfortunately, these metrics are not suitable for quantifying the mixture of nodes of the MANET, since they do not explicitly consider the identity of nodes.

6.2.2 Definition of Contact-Based Mobility Metrics

Now, we present our methodology to define metrics for mobility on a large time scale. Then, we reflect on the various aspects of contacts that could be helpful for mobility-assisted ad hoc networking an define our set of metrics.

Definition of Contacts

In the following we define a contact between two given nodes. For this, we represent an *encounter* by the IDs of involved nodes, the *time of incidence* and the *duration*. We denote by e_{nm} an encounter of node n with node m. We define e_{nm} as follows:

$$e_{nm} = \{n, m, t, \triangle t\} \tag{6.1}$$

where t is the time of incidence of the encounter and $\triangle t$ the duration of the encounter.

We define a *contact* between two nodes as the list of all encounters between them. A contact between two nodes begins with the first encounter between them, and ends with the last one. A contact is considered to be lost if there is no encounter between the nodes. We denote the contact between node n and node m by c_{nm}. We represent c_{nm} as a set of e_{nm}:

$$c_{nm} = \{e_{nm}\} \tag{6.2}$$

We assume that each node manages its contacts in a so-called *contact table*. An example of such a contact table is shown in Fig. 6.3. The contact of node 1 with node 6 until time t consists of two encounters: $c_{1,6} = \{\{1, 6, 7.5, 7.5\}, \{1, 6, 22.5, 12.5\}\}$.

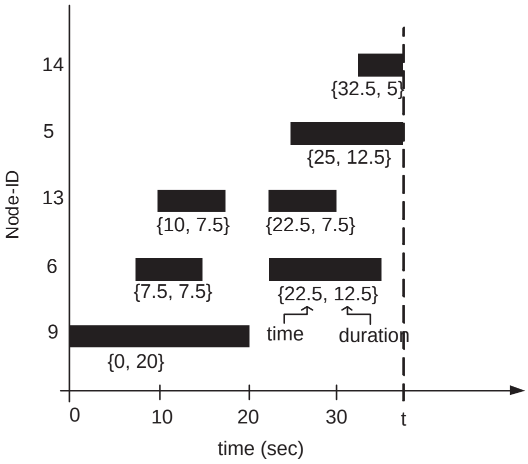


Figure 6.1: Contact table of node 1 at time t

Methodology

In epidemiology, contacts are important for the analysis and prediction of the spreading of infectious diseases. There, contacts have a great impact on the quarantine and vaccination decisions.

We are convinced that contacts are useful for the analysis of the performance of delay-tolerant ad hoc applications and of the underlying mobility-assisted protocols. Furthermore, contact information can help adapt these applications and protocols to the mobility of nodes.

In the following we list some information that is very helpful for delay-tolerant and mobility-assisted networking.

- How many new contacts does a node acquire per unit of time?
- How frequent does a node encounter the same node?
- How long does an encounter last? Or how long does a contact remain lost?

This information quantifies the contact process between nodes on a large time scale, making the information interesting for delay-tolerant ad hoc applications and protocols that act on a large time-scale.

For mobility-assisted routing protocols, we would select the node with the highest probability of next encountering the destination, as relay node. This decision might increase the probability of delivering the message and decrease the delivery delay. For mobility-assisted broadcasting protocols, we would select the nodes that contact the most nodes or that provide a larger encounter rate to buffer more broadcast messages and to rebroadcast them later, e.g. on partition join. This might increase the number of nodes reached, and decrease delivery delay as well as the overall buffer overhead. For delay-tolerant content distribution applications, nodes that make contact with other nodes more frequent may play the role of the content providers (publishers) more intensively.

We can define contact-based mobility metrics either at the network, or at the node level. At the network level, this information helps us to understand the mixture of the population. If the information is node-centric it describes the relative mobility of that node, compared to other nodes. Network-wide metrics describe the contact process on a macroscopic scale, i.e. how the population is mixed.

Definition of Metrics

We now define contact-based mobility (cbm) metrics by first defining them at the node level (node-centric) and then by taking the average over all nodes (network-wide). We denote the set of contacts of node n within T_{obs} (our observation period) as C_n :

$$C_n = \{c_{nm}\} \tag{6.3}$$

We denote the set of all encounters experienced by node n within T as E_n:

$$E_n = \{e_{nm}\} \tag{6.4}$$

- Contact Rate:

  We denote the number of new contacts experienced by node n per unit of time (e.g. 3 contacts/min) as ACR_n. Where ACR_n is defined as follows:

  $$ACR_n = \frac{|C_n|}{T_{obs}} \tag{6.5a}$$

  Therefore, the network-wide Average Contact Rate (ACR) is the average of ACR_n over all N nodes:

  $$ACR = \frac{1}{N} * \sum_{n=0}^{N-1} ACR_n \tag{6.5b}$$

- Encounter Frequency:

  We denote the number of encounters experienced by node n within T_{obs} divided by the number of contacts experienced by node n within T_{obs} as AEF_n.

  $$AEF_n = \frac{|E_n|}{|C_n|} \tag{6.6a}$$

  We define the Average Encounter Frequency (AEF) as the Average number of encounters per contact. Therefore, AEF is given by the average of AEF_n over all N nodes:

  $$AEF = \frac{1}{N} * \sum_{n=0}^{N-1} AEF_n \tag{6.6b}$$

- Encounter Rate:

  We denote the number of new encounters experienced by node n per unit of time (e.g. 9 encounters/min) as AER_n. We compute AER_n as follows:

  $$AER_n = \frac{|E_n|}{T_{obs}} \tag{6.7a}$$

  We define the Average Encounter Rate (AER) as the average number of new encounters experienced by a node per unit of time. Therefore, AER is given by the average of AER_n over all N nodes:

  $$AER = \frac{1}{N} * \sum_{n=0}^{N-1} AER_n \tag{6.7b}$$

- Contact Duration:

  We define the Average Contact Duration of node n over all contacted nodes as:

$$ACD_n = \frac{\sum_{e_{nm} \in E_n} e_{nm}.\triangle t}{|C_n|} \tag{6.8a}$$

  Therefore, the Average Contact Duration (ACD) in the network is the average of ACD_n over all nodes:

$$ACD = \frac{1}{N} * \sum_{n=0}^{N-1} ACD_n \tag{6.8b}$$

- Contact Loss Duration:

  We define $ACLD_n$ as the average contact loss duration of node n over all nodes it has established a contact with during T.

$$ACLD_n = T_{obs} - \frac{\sum_{e_{nm} \in E_n} e_{nm}.\triangle t}{|C_n|} = T_{obs} - ACD_n \tag{6.9a}$$

  The network-wide Average Contact Loss Duration ($ACLD$) is therefore the average of $ACLD_n$ over all nodes:

$$ACLD = \frac{1}{N} * \sum_{n=0}^{N-1} ACLD_n = T_{obs} - ACD \tag{6.9b}$$

- Encounter Duration:

$$AED_n = \frac{\sum_{e_{nm} \in E_n} e_{nm}.\triangle t}{|E_n|} \tag{6.10a}$$

  We define the network-wide Average Encounter Duration AED as the average of AED_n over all nodes:

$$AED = \frac{1}{N} * \sum_{n=0}^{N-1} AED_n \tag{6.10b}$$

We notice here that some contact-based metrics are identical to the existing link-based metrics. For example the Average Encounter Duration is equivalent to the Average Link Duration [131], which has been already investigated in [130,131].

In [29] we statistically and analytically investigated these metrics for the random waypoint model. However, we do not present these results here.

6.2.3 Providing cbm Metrics for ns-2 Users

In this subsection, we present our framework that helps ns-2 users to easily generate and use cbm metrics. For ns-2 we provide a tool, `CBM`, that annotates a given mobility scenario (in ns-2 format) with cbm information such as the contact table of nodes and some network-wide metrics. GOD can then load this information from the annotated file and supply it to the ns-2 protocol and application developers.

This approach is similar to our approach to provide network partitioning information for ns-2 users (Section 5.5.7). Fig. 6.2 illustrates our approach.

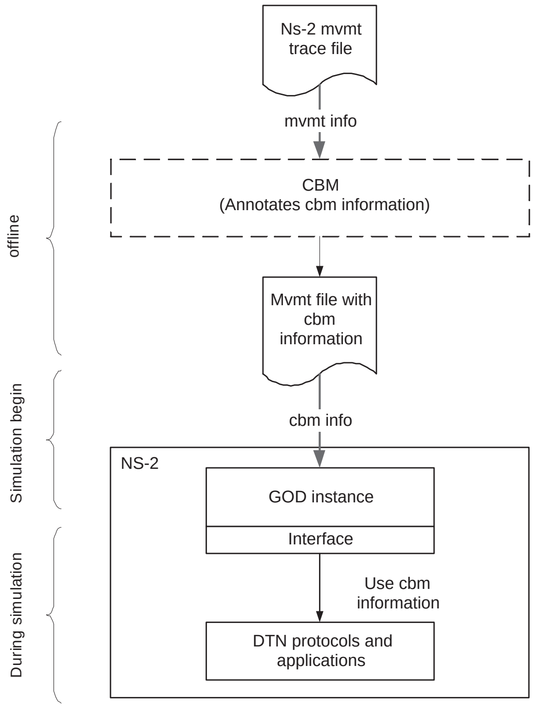


Figure 6.2: Preparation and use of cbm metrics

The tool `CBM` is able to annotate already existing ns-2 movement trace files, independently from the mobility model and from the generation tool. The tool generates cbm information according to the ns-2 network model, i.e. an undirected graph, where an edge between two nodes is added if their distance is below R.

The annotation is a set of OTcl commands for ns-2. The annotation is done offline, in order to increase reusability of trace files and to reduce simulation run time. One disadvantage of the offline generation is that cbm information is dependent on the fixed communication range. We extend the GOD object with procedures to query network-wide as well as node-centric cbm metrics.

6.2.4 Uses of Contact-Based Mobility Metrics

Now we discuss how the contact-based metrics can be used and present one project that is investigating node encounters for protocol design.

Contact-based metrics may be used at design-time to evaluate delay-tolerant protocols and applications, but also at run-time to adapt these protocols and applications. In [129], the authors mention the requirements for mobility metrics with regards to adaptation of protocols.

Network-wide metrics are metrics that are not easy to compute at run-time, since they require a high communication overhead. Network-wide metrics are appropriate at the design stage and should be used by developers, in order to suitably design their protocols for a wide range of mobilities.

Node-centric metrics are easy to acquire at run-time. Nodes have to manage a history of their encounters in a local contact-table. Encounters can be perceived using a simple neighbor discovery protocol, such as HELLO beaconing. These metrics can be then used to adapt protocols and applications on-the-fly. The maintained history should have low memory requirements.

In [133,134], the authors use the so-called last encounter age or the time elapsed since last encounter to efficiently route unicast messages to destinations. To this end, nodes maintain a history of the closest last encounters. If the source wants to send a message to the destination, it forwards the message to the neighbor that encountered the destination more recently than the source and other neighbors. The rational behind their approach is that "a node that was my neighbor 5 seconds ago is probably closer to me than a node that was my neighbor 5 minutes ago". This work shows the utility of encounter history or contacts even for non-delay-tolerant networking.

In Section 6.3, we use the defined contact-based mobility metrics to adapt the mobility-assisted broadcast repetition of hypergossiping to node mobility. We use the contact-based mobility metrics to detect the key mobility patterns on a sufficient large time-scale.

6.3 Mobility-Aware Probabilistic Buffering for Hypergossiping

In this section, we briefly show the importance of the detection of relevant mobility patterns for successful design and adaptation of store-and-forward mechanisms. For hypergossiping, we present an approach that heuristically detects nodes moving in a group and enables them to cooperate to buffer a given message.

6.3.1 Motivation

In the following, we motivate our approach by investigating some real-world scenarios, from which certain mobility patterns become apparent, which are relevant to mobility-assisted broadcasting.

Observations and Ideas

Users in the real world are likely to show certain movement patterns such as correlation between users or repeating behavioral patterns. Some users such as soldiers and rescuers tend to move in groups. Others such as busses and trains tend to repeat their movements. However, there are also singular nodes that behave unpredictably. Without loss of generality, we can consider a MANET as a set of node groups that meet and leave over time.

We are convinced that mobility characteristics of nodes play a major role in the design of store-and-forward mechanisms in mobility-assisted networking. Using the example of mobility-assisted delay-tolerant broadcasting, such as hypergossiping, mobility patterns are crucial for developing efficient buffering strategies. Since we are dealing with highly diverse mobility patterns in MANETs that may change over scenarios or over time within the same scenario, we also believe that the perception of relevant mobility characteristics is a major factor for the adaptation of store-and-forward mechanisms at run-time.

Two examples of mobility patterns, from which buffering strategies could benefit, are examined in the following. Firstly, nodes moving in a group can cooperate concerning message buffering. The group can 'select' one of its members to buffer the message. This should reduce the total number of nodes buffering a certain message. Secondly, nodes that constantly encounter new nodes (e.g. racer cars on the highway) should buffer more messages than other nodes, that rarely encounter new nodes.

It is challenging for mobile nodes to detect such patterns at run-time. Since these patterns have a large time-scale, we are convinced that contact-based metrics are valuable for recognizing such patterns and for detecting them at run-time.

Motivation Scenarios

In the following we motivate, using the example of a campus scenario and a highway scenario, the idea of mobility-aware buffering and the major role that contact-based mobility metrics can play for the perception of certain relevant mobility patterns.

Campus Scenario: Consider the MANET formed by mobile devices carried by students and staff members on a campus during working hours. Firstly, we

	staff members	students
encounter rate	low	high
encounter duration	high	low
encounter frequency	≈ 1	≈ 1
contact rate	low	high
contact duration	high	low

Table 6.1: Campus scenario

qualitatively analyze the properties of the mobility of nodes on a large time-scale using our contact-based metrics. Secondly, we show the usefulness of these metrics for mobility-assisted broadcasting in the considered MANET.

Staff members are generally grouped into departments, which in their turn are grouped in faculties. Offices for one department or one faculty are normally grouped geographically. Staff members work most of time in their offices, and sometimes meet each other. We assume that nodes located in different campus buildings can not communicate directly with each other. Also we assume that not all nodes located in the same building can communicate directly. From these observations, we can conclude that mobile devices carried by staff members form a relatively stable network topology. Thus they show low encounter and contact rates but high encounter and contact durations.

Students commute frequently between departments, faculties, classrooms, libraries and cafeterias. Therefore, their encounter and contact rates are higher, but their contact and encounter durations are lower than that of staff members. Table 6.1 shows a qualitative analysis of the contact-based metrics for both groups.

We suggest the following two simple heuristics to reduce buffer overhead. Firstly, students have to buffer messages, since students commute more probably between different network partitions. Therefore, students are more suitable as a transport mechanism between partitioned groups. From Table 6.1 hypergossiping can easily approximate, whether a node is suitable for buffering the message. Nodes with higher contact rate should be chosen. Secondly, not all nodes moving in a group intuitively have to buffer the same message. Therefore, group members should cooperate to buffer messages, in order to reduce total buffer overhead.

Highway Scenario: Since VANETs are more vulnerable to network partitioning than other MANETs, store-and-forward is commonly used, which increases the demands on storage. Furthermore, the highly dynamic VANET topology leads to increasing requirements for higher sensor updates, which in its turn increases the requirements for storage. Although vehicles' buffering limitations are less severe than notebooks or PDAs, a conservative use of storage is required.

In reality, VANETs show complex mobility behavior. Some nodes move in groups, while others move individually and independently. Moreover the group affiliation is not permanent, since groups can dynamically reconfigure themselves triggering group splits and merging.

We now consider the following highway scenario. Trucks commonly drive in groups, since drivers follow each other or it is forbidden to overtake. Cars in the overtaking lane of the highway (e.g. car A) however, show a lower group movement pattern, since they hop from one group to the next. The overtaking cars are therefore likely to encounter more new cars driving in the same direction than trucks driving in the right lane.

It is obvious that trucks and speeding cars should behave differently concerning message buffering for the purpose of broadcasting. Trucks may cooperate with respect to buffering. Racer cars however have to buffer more messages.

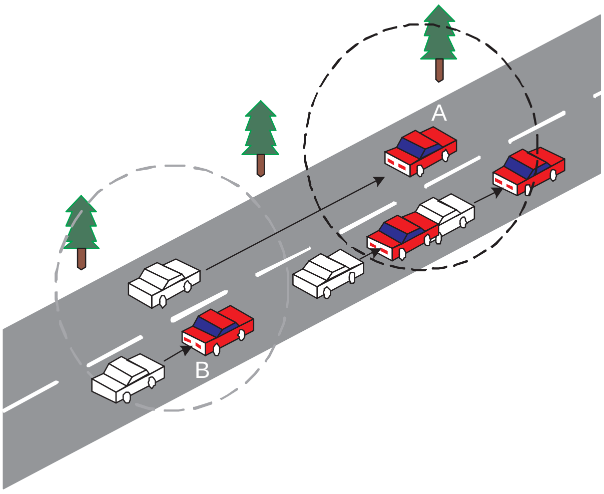


Figure 6.3: Cars on a highway

The challenge is to detect the relevant mobility patterns at run-time. In the following, we use contact-based mobility metrics to characterize racer cars and trucks qualitatively. We expect that racer cars show higher average encounter rates and lower average encounter durations, because overtaking normally takes only a few seconds. Racers should have low average encounter frequency (≈ 1), since most of the cars are encountered only once. Therefore, the contact duration is lower and contact rate is higher (Table 6.2).

6.3.2 Overview of Approach

In order to exploit the previous observations on mobility characteristics of real users with the purpose of reducing the hypergossiping buffering overhead, we

	racer	truck
encounter rate	high	low
encounter duration	low	high
encounter frequency	≈ 1	≈ 1
contact rate	high	low
contact duration	low	high

Table 6.2: Highway scenario

propose a new buffering strategy for hypergossiping. Our approach uses contact-based mobility metrics to compute the utility of buffering a certain broadcast message for future rebroadcasting.

We only consider the nodes' mobility pattern for the selection of buffering nodes. We do not select the buffering nodes according to their storage capabilities. In MANETs that are composed of nodes of heterogeneous capabilities we propose to also use other information such as available memory, CPU power, and available battery energy to differentiate nodes, while defining the utility of a certain node to buffer a certain message.

Our approach attempts to allow the node, which is most likely to deliver the message and with lowest cost, to buffer the messages. We use utility metrics to determine the node most appropriate to buffer the message and to deliver it to the destination. Since we differentiate nodes only with regard to their movement patterns, we should analyze the patterns most relevant for buffering and design efficient concepts to detect these patterns at run-time. Afterwards, we should define metrics for the selection of the appropriate nodes with regards to their movement patterns.

Our approach is based on two components. The first component efficiently detects movement patterns relevant for the broadcast repetition of hypergossiping. We realize this component based on particular contact-based mobility metrics. The second component is used by nodes to compute the buffering utility for each newly received message. This component provides a method for computing the buffering utility, depending on the mobility pattern perceived by the first component.

6.3.3 Relevant Mobility Patterns

In the following we discuss two examples of mobility patterns that can be used to reduce the buffering overhead of hypergossiping.

As stated before, we consider a MANET as set of node groups that meet and leave over time. Some real-world scenarios, such as the campus scenario, confirm this view of the MANET. If we consider nodes moving in isolation as one-node group, we can transform every mobility model, where nodes move independently, to a group mobility model. Thus, by the appropriate choice of parameters, existing mobility models, such as random waypoint and graph-based, can be considered as group mobility models.

While investigating the campus scenario, we showed that nodes moving in a group or nodes roaming very frequently between different groups may play a particular role for mobility-assisted broadcasting. Therefore, we are looking to exploit these movement patterns, to reduce the buffering overhead of hypergossiping.

The *group movement pattern* helps the group members to share the buffering task. Our approach is to let nodes moving in a group *cooperate* in order to select nodes that buffer a certain message.

Nodes showing a *roaming pattern* are good candidates to buffer messages and transfer the broadcast from one partition to the next. Roaming nodes in our campus scenario are mainly the students.

The store-and-forward feature of hypergossiping takes place on a large time-scale (given by the lifetime of data). Subsequently, the mobility patterns that are relevant for hypergossiping should be of a similar time-scale. Since our contact-based metrics model the mobility on a large time-scale, we are convinced that these metrics are valuable for capturing the group movement and roaming movement patterns.

6.3.4 Detection of Relevant Mobility Patterns

In the following we present our approach, which uses the cbm metrics defined in previous section, in order to easily detect the mobility patterns mentioned above.

Group Movement Detection

We say a set of nodes move in a group over a time interval $[t_1, t_2]$, if they show very correlated movements during this time interval. At the macroscopic level, group members show a certain physical proximity.

The motion group members are identified based on a time window w, i.e. the length of the time interval $[t_1, t_2]$ (e.g. 5 minutes). The value of w is application dependent.

We assume that the groups are not known in advance and that they can form dynamically. Therefore, we need mechanisms to discover nodes moving in groups at run-time.

We assume that the geographical proximity of group members is within the scale of the communication range or lower. Therefore, we expect that group members are within the communication range of each other, most of time. We also tolerate group members leaving each others communication range for a short time period and then encountering each other again. Fig. 6.4 outlines a contact between node A and node B that consists of three encounters. It is obvious on the macroscopic level that nodes A and B move in a group. Using contact-based mobility metrics it is possible to recognize that both nodes A and B are moving in a group. Observing the contact-based metrics of nodes moving in a group we can easily expect them to have long encounter duration (subsequently also higher contact duration), or short encounter durations but a long *contact duration* with the other group members. Thus, we define two nodes as moving in a group, if they show a contact duration comparable to the time window w.

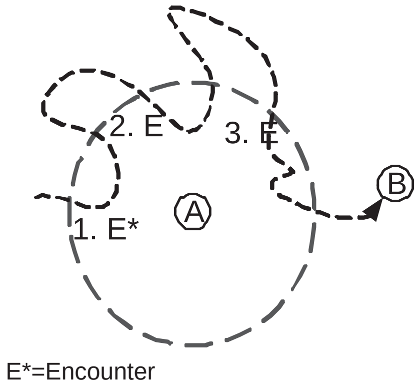


Figure 6.4: Detection of nodes moving in a group

Accordingly, one node is able to detect neighbors that move with it in a group. Again, the length of the contact history should be set appropriately (depending on the application): If two nodes encounter each other for some minutes and then leave, should we consider this encounter is a group movement or not? The decision depends on the time-scale of the broadcasting process, which in turn depends on the delay-tolerance of the broadcast application. We modeled the delay tolerated by the application using the lifetime of broadcast data. Considering broadcasting by means of hypergossiping, we conclude from the observations above that the contact history time depends on the lifetime of broadcast data. In the evaluation section, we calibrate the length of the contact history.

The higher the current encounter duration of a certain node with a certain neighbor, the higher the probability that both nodes are moving in the same group. In this work, we simply assume that, if two nodes have a contact duration higher than 80% of contact history time T_{obs}, i.e. $w = 0.8 * T_{obs}$, then both nodes are moving in a group.

Related Approaches

In [135, 136], the authors pointed out that the fundamental characteristics of group mobility is the similarity of the velocity. According to this, the authors presented a localized method to detect group members by sharing velocity information with neighbors. Since this approach relies on velocity information, the approach has strong limitations regarding its use in application scenarios, where it can be used. As we are designing a generalized solution for MANETs, we will not consider this approach further.

In [137], the authors proposed a scheme to detect the presence of groups among the nodes of a network by performing a correlation index test on the mobility traces. The method assumes a global view (position of all nodes over the time-interval considered) and is, therefore, unapplicable for our purposes.

In [138] the authors presented a group discovery method that assumes the availability of an up-to-date routing table, which in turn assumes the existence of a proactive routing protocol. Proactive routing protocols are, however, only appropriate for low mobile MANETs. Therefore, the deployment of the discovery strategy is limited to low mobile MANET scenarios. Hence, the strategy is not applicable for our generalized broadcasting technique.

Roaming Movement Detection

Nodes that frequently move between groups constantly encounter new nodes. Considering the contact-based metrics for these nodes, we expect that these nodes are characterized by high contact rates.

6.3.5 Utility-Based Probabilistic Buffering

Our approach for buffering is to define a utility for each node. The utility presents a metric for the relevance of buffering the received broadcast message. The higher the utility, the more useful is the buffering.

To this end, we define a utility metric that we call a *buffering utility*, $u(n, m_i) \in [0, 1]$, at every node n for each broadcast message m_i received by that node. This indicates how useful it is for the node n to buffer the message m_i for broadcast repetition. If a node receives a broadcast message for the first time, it computes the buffering utility of that message and buffers it with a probability equal to the utility value $p(n, m_i) = u(n, m_i)$ and for a time period equal to the message's residual lifetime.

This strategy offers the following two advantages. Firstly, the buffering of the messages will be shared equally between nodes. This leads to a fair buffering. Secondly, the strategy is completely decentralized, since there is no need for coordination between nodes, in order to determine which node buffers which message.

The buffering utility has to be updated by a node according to the mobility patterns detected by that node. Since we aim to consider two mobility patterns, we superimpose two components to calculate the buffering utility. The first part is to update the utility depending on the group movement pattern. The second is to update the utility with respect to the roaming degree of the nodes.

Group-Based Buffering

We propose the cooperation of nodes moving in a group and suggest the following approach. We assume that nodes currently moving in a group are also likely to remain in the same group. Nodes belonging to the same group should cooperate in order to share the buffering of messages. The subsequent step is to fix which node has to buffer the messages. Clustering and centralized coordination, in order to distribute the buffering task is one approach that we should avoid, since it produces a high message overhead, especially in highly mobile networks, where the clustering algorithms have to be run more frequently.

In the following, we consider a set of nodes moving in a group. We denote this set of nodes as G. We denote $|G|$ as the number of members of the group G, i.e. $|G| = card(G)$. The utility should be defined inversely proportional to the number of group members. As an example, we deploy the simple function $1/x$. We propose to compute the group-based buffering utility as shown in Eq. (6.11).

$$u_{group}(n, m_i) = \frac{1}{|G|} \tag{6.11}$$

This approach ensures that that $p_{group}(n, m_i) \in [0, 1]$ since $|G| \geq 1$. Please note that we use the same utility for all messages, since we are only considering the mobility characteristics. We then set the probability for probabilistic buffering equal to the utility as shown in Eq. (6.12).

$$p_{group}(n, m_i) = u_{group}(n, m_i) = \frac{1}{|G|} \tag{6.12}$$

The probability that the group of nodes G fails to buffer the message is shown in Eq. (6.13) and Eq. (6.14).

$$p_e = \prod_{n \in G}(1 - p_{group}(n, m_i)) = \prod_{n \in G}(1 - \frac{1}{|G|}) = (1 - \frac{1}{|G|})^{|G|}) \tag{6.13}$$

For large groups, p_e is on average equal to:

$$p_e \approx 1/e \approx 1/2.72 \approx 0.37 \approx 37\% \tag{6.14}$$

The probability of successful message buffering (p_s) is given in Eq. (6.15).

$$p_s = 1 - p_e \approx 1 - 1/e \approx 0.63 \approx 63\% \tag{6.15}$$

Using this simple calculation, we recommend increasing the buffering utility and probability by introducing an efficiency parameter k as follows:

$$u_{group_2}(n, m_i) = p_{group_2}(n, m_i) = min(\frac{k}{|G|}, 1.0) \tag{6.16}$$

The probability of buffering failure is then $p_e \approx \exp(-k)$. If $k = |G|$ all nodes buffer and we get the reachability of the original algorithm. In the evaluation subsection we vary the value of k and calibrate it.

Roaming-Based Buffering

Cross-moving nodes that roam between different groups are expected to show very low encounter frequencies (≈ 1). These nodes should buffer more messages than other nodes. Therefore, the utility should be defined inversely proportional to the encounter frequency. As an example, we deploy the simple function $1/x$.

$$u_{roaming}(n, m_i) = \frac{1}{AEF_n} = \frac{ACR_n}{AER_n} \tag{6.17}$$

Where AEF_n, ACR_n and AER_n are the node's Average Encounter Frequency, Contact Rate and Encounter Rate respectively.

Similar to the group-based buffering, we set the roaming probability as depicted in Eq. (6.18).

$$p_{roaming}(n, m_i) = u_{roaming}(n, m_i) = \frac{1}{EF_n} = \frac{CR_n}{ER_n} \tag{6.18}$$

Please note that $p_{roaming}(n, m_i) \in [0, 1]$ since $EF_n \geq 1$.

Integrated Probabilistic Buffering

In general, different factors such as node capabilities, MANET characteristics and message properties may impact the buffering decision of nodes. To consider different factors, we propose that nodes define different utilities for different factors and to superimpose these utilities depending on their relevance for buffering.

In this work, we investigate only mobility patterns and exactly two mobility patterns: Group motion and roaming patterns. Therefore, we consider both utilities for a buffering decision, i.e. group-based and roaming-based utilities. Hence, we should weight each probability and use the average for buffering decision (Eq. 6.19).

$$p(n, m_i) = \lambda * p_{group}(n, m_i) + (1 - \lambda) * p_{roam}(n, m_i) \tag{6.19}$$

The appropriate value of λ depends on the decision, which mobility pattern roaming or group motion is more appropriate for reducing buffer overhead. In this work, we mainly aim to prove the concept and therefore focus on the group motion pattern, i.e. we set $\lambda = 1$.

6.4 Performance Evaluation

In this section, we first introduce the simulation model and the performance metrics. Then, we calibrate the length of the contact history and the efficiency parameter of buffering k. Finally, we present the performance of hypergossiping with mobility-aware buffering and compare it to the performance of the original algorithm.

6.4.1 Simulation Settings

If not otherwise stated, our simulation settings are identical to the settings summarized in Table 4.3 and Table 5.1. Each sender sends one single packet. We fix the lifetime value of broadcast data at 600s.

Before initiating hypergossiping, we run a warm-up phase for a period of time equal to the T_{obs}. This allows nodes to have a complete history of contacts, before starting hypergossiping. According to this, we set the simulation time to a value equal to $T_{obs} + lifetime + 20s$.

6.4.2 Performance Metrics

For the evaluation of the buffering strategy, we define the following evaluation metrics:

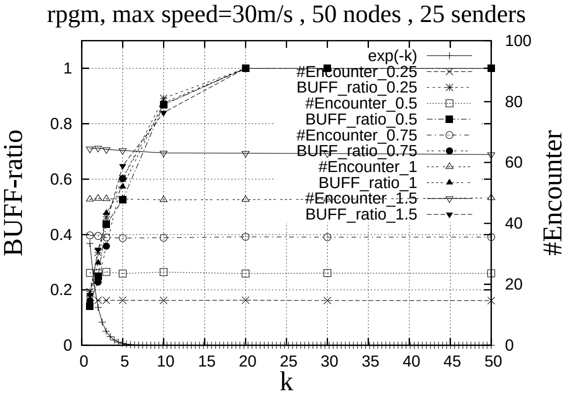
- *BUFF-ratio*: The ratio of nodes buffering a given message to the total number of nodes that have received this message. We note here that BUFF-ratio $\in [0, 1]$.
- *Average number of encounters*: The major additional overhead to perceive contact-based mobility metrics is the storage overhead for the contact table. This is simply given by the number of encounters forming the table. For each encounter we need 4 bytes to store the encounter ID (e.g. MAC address), 4 bytes for its time of incidence and 4 bytes for its duration. Therefore, we need 12 bytes in total, for each encounter.

6.4.3 Calibration of Mobility-Aware Probabilistic Buffering

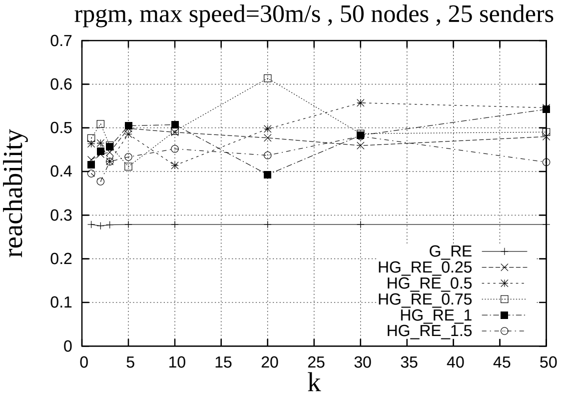
Contact-based group detection, as well as the probabilistic buffering strategy have some parameters that still have to be calibrated, i.e. the observation time period (T_{obs}) and the efficiency parameter of buffering (k).

As stated before, the contact history depends on how the application defines a set of nodes to be moving in a group. Since our protocol acts on a time-scale given by the lifetime of broadcast data, we fix the obs-interval depending on the lifetime. We set $T_{obs} = obs - scale * lifetime$ and vary obs-scale for calibration. The obs-scale parameter impacts the size of the contact table and the groups detected. Hence the goal for calibration should be, we have to minimize the obs-scale value, while keeping the groups detected correctly from the point of view of the application.

The efficiency parameter k impacts the ratio of group members that decide to buffer the message. k also determines the error that no group member would buffer the message. For calibration, we should minimize k, to reduce as much buffering overhead as possible, but we also have to minimize the buffering error, so that the reachability of hypergossiping will not degrade much. According to the error function (Fig. 6.5 a)), if we tolerate buffering errors of 5% or lower, we have to choose $k \geq 3$.



(a) #Encounter and BUFF-ratio



(b) Impact of k and obs-scale on reachability

Figure 6.5: Calibration of the probabilistic buffering strategy

We use the RPGM mobility model for the calibration process. So far, we assumed that the group's geographic proximity is within the range of the communication range. For our settings for the RPGM mobility model, we expect a maximum distance of 20m between two group members. Since we set the communication range to 100m, the assumption above holds. However, if the communication range is lower than 20m, our group detection strategy may not detect some group members. This leads to a higher buffering utility, which increases the number of nodes buffering a certain broadcast message.

From Fig. 6.5 a), we conclude that for $k = 3$ the buffering overhead of hypergossiping can be reduced by approximately 50%-64%. We therefore select $k = 3$ for our buffering strategy.

The investigation of the reachability of hypergossiping (Fig. 6.5 b)) shows that probabilistic buffering introduces some oscillations. Due to the probabilistic nature of buffering, the nodes that buffer the message change from one simulation run to another. This may lead to the group members that detect a partition join not being those that buffer the appropriate messages. However, the impact of k and the obs-scale on the reachability of hypergossiping is not clear. Therefore, the calibration of obs-scale can not be done using this simulation set.

For the calibration of obs-scale we proceed as follows. Intuitively, a message that is received by a node has a residual lifetime equal, on average to half of the lifetime value, which is set by the source of the message. If the node decides to buffer the message, it will buffer it, on average, for the residual half lifetime value. Subsequently, a node has to detect the nodes, with which it has been moving in last half lifetime period and assume that this group will hold for the next half lifetime period. From this observation we propose to use $obs - scale = 0.5$ for the buffering strategy.

6.4.4 Simulation Results

In this study, we vary the mobility models, and arbitrarily set the maximum speed of nodes to 3m/s. We set the lifetime of broadcast data to 600s. Nodes maintain contact tables for 300s. In Fig. 6.6, we observe that for the random waypoint, our strategy does not detect any motion groups and therefore does not reduce the number of nodes buffering a certain broadcast message. Using the graph-based mobility model, our strategy detects a few groups for a higher number of nodes and therefore saves some bufferings (about 5% for N=300). Since almost all receivers buffer the message for both random waypoint and graph-based models, the performance of hypergossiping (reachability, MNFR and delay) using probabilistic buffering is very close to the performance of hypergossiping without probabilistic buffering (Fig. 6.7 a)b)).

The RPGM model shows inherently more motion groups. Our strategy detects many of the groups and prohibits between 63% (N=50 nodes) and 70% (N=300 nodes) of receivers from buffering the broadcast message. The reachability of hypergossiping with probabilistic buffering (Fig. 6.7 a)), decreases compared to the case without probabilistic buffering. This is due to the massive reduction of the number of nodes that buffer the broadcast messages. In order to avoid a degradation in reachability, nodes can sacrifice some saved bufferings by using higher values for the efficiency parameter k than the current value ($k = 3$).

For the overhead caused by the management of contact tables (Fig. 6.6), we

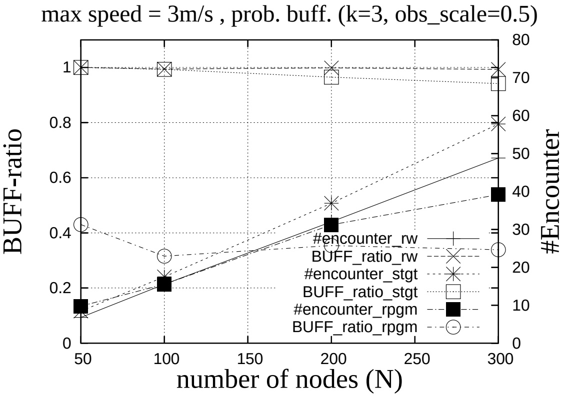
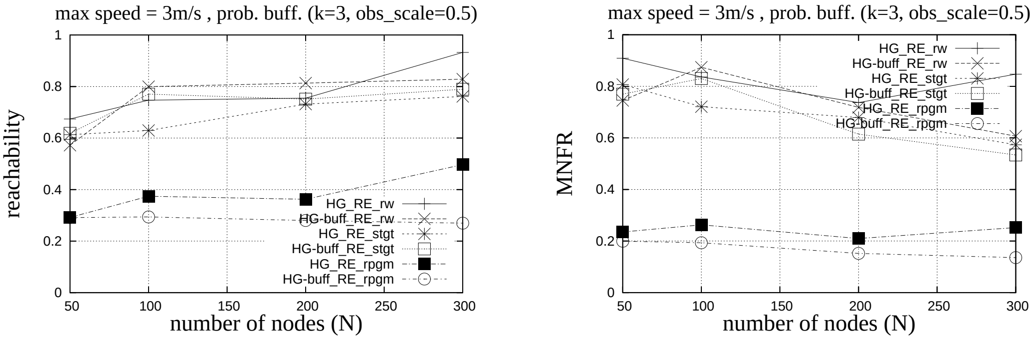
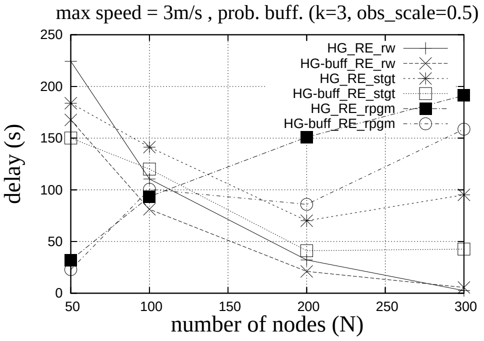


Figure 6.6: BUFF ratio and #Encounter



(a) Reachability (b) MNFR



(c) Delay

Figure 6.7: Performance of HG with probabilistic buffering

note that HELLO beaconing is needed by the hypergossiping protocol anyway. The main additional overhead is therefore the storage overhead for the contact tables. The maximum table size needed, would be for N=300 nodes and for the graph-based mobility model, i.e. 58 encounters. This implies a storage overhead of 58*12 byte, which is equivalent to the buffering overhead of 2.5 messages. This demonstrates the limitation of the additional overhead induced by the maintenance of contact-based information. Although, this overhead will be higher for higher T_{obs} value and higher node speeds, we are convinced that this overhead remains tolerable, compared to the buffer space gained. We note also that this overhead remains constant for higher broadcast traffic, where the saved buffer overhead increases. Furthermore, contact-based mobility information can be used for further protocols and applications.

6.5 Summary

In this chapter, we proposed a set of novel mobility metrics for a deeper understanding of the mobility on a large time-scale. Following the methods of epidemiology, our metrics are contact-based, since they quantify the spatial encounters between nodes. For ns-2 users we provide the required support to easily use our novel metrics. Recently, some groups are using our metrics such as [139].

We also demonstrated how contact-based mobility metrics can help developers in designing and adapting mobility-assisted ad hoc protocols using the example of hypergossiping. Using these metrics nodes can detect, in an efficient and localized way, other nodes that move with them in a group. Nodes that move in a group can then cooperate to share the buffering of broadcast messages. Simulation results show that our mobility-aware probabilistic buffering strategy can significantly reduce the hypergossiping buffering overhead without much degrading the overall performance of the protocol.

Chapter 7

Conclusions and Future Work

In this thesis we have investigated broadcasting in MANETs. The existing broadcasting techniques are appropriate for many scenarios, however they have significant shortcomings in others. Since a MANET may show a wide range of evolving operating conditions, a generalized broadcasting technique is needed. We developed a solution that suits for a wide range of MANET scenarios with respect to node spatial distribution, node mobility and network load. Our strategy can be simply deployed in a wide range of potential application scenarios, without any pre-configuration. In the following, we draw our main results and discuss some future research directions.

A Generic Analytical Methodology for Modeling Broadcasting: In order to establish a *common theoretical platform* for modeling and analysis of broadcasting in MANETs, we were the first to exploit the strong similarity between the spreading of broadcast messages and the spreading of infectious diseases. For this purpose, we first explained how protocol developers should proceed to select the appropriate epidemic model for the broadcast protocol of interest. Then, step by step we detailed how to adopt the selected mathematical model from the epidemiology to describe the message spreading. The epidemic models usually depend on few parameters, which should be determined from the MANET properties and the broadcast protocol parameters.

In this work, we focussed on the SI model and showed that it is appropriate for the SPIN-based broadcast protocol and for gossiping. Generally, the SI model is suitable for the broadcasting strategies, where nodes upon reception of a message remain ready to forward this message until it is received by all nodes. This is valid for most of broadcasting schemes reviewed in the related work section. We presented a hierarchical and modular methodology that transforms the computation of the single model parameter, i.e. the infection rate, into the computation of two subparameters, i.e the rate of encounters among nodes and the probability of message transmission upon an adequate encounter. We showed how to proceed to determine the subparameters either by performing further analytical work or by using simulations. However, for simplicity we determined the infection rate using few simulations.

The SI model we adopted provides analytical expressions to compute the message spreading in time, which presents an elegant way to describe and obtain insight in the broadcasting process as well as to discover important trends. Consequently, we used the results for a better understanding of the impact of relevant MANET or protocol properties on the performance of the modeled protocols. In particular, we showed how the SI epidemic model simplifies the adaptation of broadcasting strategies to relevant MANET properties.

A Generalized Broadcasting Technique for MANETs: The main contribution of this thesis is a generalized broadcasting technique for MANETs, which we refer to as *hypergossiping*. Hypergossiping acts on a MANET as a set of network partitions. Our technique efficiently distributes messages within one network partition using *adaptive gossiping*, and efficiently reiterates the message gossiping upon a partition join using our novel *broadcast repetition strategy*.

Adaptive Gossiping: In gossiping a node forwards a received message to all neighbors with a fixed probability. The determination of the optimal gossiping probability is a major concern. If it is fixed too high, many redundant forwards occur. These redundant transmissions waste energy and bandwidth. If the probability is too low the flood dies out before reaching all nodes. Node density is the primary criteria for selecting the appropriate gossiping probability.

Instead of fighting with a large amount of simulation data and deciding visually which is the appropriate forwarding probability for a given node density, we exploited the elegant way of describing the spreading process by the infection rate. We measured the infection rate for different node densities and forwarding probabilities. For each given node density, we selected the probability that maximizes the infection rate. The result of the adaptation is a simple function that nodes can easily use to calculate the appropriate gossiping probability for the current number of neighbors (n): $p = min\ (1.0\ ,\ 0.175 + \frac{6.05}{n}\ )$. Our methodology to use the SI model for adaptation of key protocol parameters to relevant network properties can be easily repeated for further adaptation needs.

Simulations that we conducted for a *wide* range of node spatial distributions, node mobilities, number of nodes and communication ranges showed that adaptive gossiping saves many redundant forwards while maintaining the reachability high, emphasizing the reliability, efficiency and scalability of adaptive gossiping. This performance is achieved for the wide range of operating conditions, provided that the network is not partitioned.

Efficient Broadcast Repetition: The broadcast repetition strategy consists of a heuristic for detecting relevant partition joins and a rebroadcasting protocol to rebroadcast the appropriate messages upon the detection of a partition join. Our *partition join detection heuristic* is fully decentralized. It adapts to the number of neighbors and shows high efficiency and accuracy. In addition to detection of partition joins, this heuristic is also able to detect the situations, where the adaptive gossiping does not reach all nodes for a given partition due to broadcast storms. The *rebroadcasting protocol* repeats the gossiping of the appropriate messages and incorporates a suppression mechanism to minimize the number of redundant repetitions. The rebroadcasting protocol assumes that nodes buffer the messages that they receive, as long as the messages are relevant.

In order to increase buffering efficiency for hypergossiping, we presented our first steps towards the reduction of the buffer overhead of the broadcast repetition strategy. Our approach focuses on adapting the buffering strategy to the mobility of nodes. The basic idea consists in that, nodes moving in a group should cooperate to buffer broadcast message. We designed a *mobility-aware probabilistic buffering* solution for hypergossiping. Our localized solution efficiently detects nodes moving in a group using our mobility metrics that we have defined for this purpose. The approach is to allow group members to cooperate in a decentralized and probabilistic manner, in order to reduce the number of nodes buffering a certain broadcast message. Simulation results showed that this mobility-aware buffering strategy can significantly reduce the hypergossiping buffering overhead without degrading the overall performance of hypergossiping.

Relevance of Hypergossiping for Real MANET Deployments: Simulations in ns-2 showed that hypergossiping outperforms all the existing broadcast strategies. Hypergossiping provides a high reachability for a wide range of node spatial distributions, node mobilities and network loads.

Hypergossiping is very resource efficient, i.e. *frugal*. It provides energy and message efficiency by adapting gossiping and the broadcast repetition to node density and thus reducing the number of redundant forwards in dense regions and the number of broadcast repetitions in sparse ones. The protocol also reduces the buffering overhead by adapting the buffering strategy to node mobility. The frugality of hypergossiping allows for an easy deployment of the protocol on a wide range of nodes, and in particular on resource-limited devices.

Hypergossiping does *not* require any special node capability such as the knowledge of position or speed. This broadens the application scenarios, where our generalized technique can be deployed.

Hypergossiping takes into account both network partitioning and density changes *independent* from their causes. The broadcast repetition does not only consider partitioning caused by the node spatial distribution and limited communication range, but also network partitioning caused by broadcast storms, communication obstacles, transient disconnections of mobile nodes due to on-off usage, hardware or software failures. Adaptive gossiping also adapts to local density changes independent from their origins. A change in node density can be triggered by the on-off usage, node failures or by the switching between different energy modes. These changes are automatically captured by gossiping, so that active nodes always install the appropriate gossiping probability. In particular, we note that hypergossiping supports different stages of deployment of open applications. These stages are characterized by different node penetration rates and subsequently by different node densities.

From the above observations, we conclude that hypergossiping simplifies the deployment of broadcasting in real-world MANETs, since it supports a wide range of network connectivities and dynamics levels, runs on a wide range of devices, tolerates a large number of network and node failures, and suits for different deployment stages.

Our Contributions for Ns-2: The network simulator ns-2 is certainly the most used simulator in the MANET community. Its popularity originates from the wired network community. During this thesis, we developed two frameworks for the ns-2 environment. The first framework simplifies the evaluation of MANET protocols concerning network partitioning. The second one defines novel mobility metrics and provides an easy way to access them in ns-2. We showed the applicability of both frameworks on the example of hypergossiping.

Network partitioning is the norm in MANETs and therefore crucial for the design of MANET protocols. We designed a framework that simplifies the access to valuable *network partitioning information* during simulations with ns-2. We provide the following utilities for ns-2 community: The `calcpartition` tool that annotates arbitrary movement trace files with partitioning information, and a generic interface to query partitioning information and statistics or to subscribe for partitioning events.

Since mobility plays a major role for mobility-assisted networking, we defined a novel set of metrics that quantify node mobility on a large time-scale. Our approach is based on the pair-wise encounters and contacts between nodes and we refer to our metrics as *contact-based metrics*. We use some of these metrics to efficiently detect mobility patterns on a large time-scale, such as nodes moving in a group. For ns-2 community, we also provide a generic framework that simplifies the access to our metrics. We provide the `CBM` utility to offline annotate arbitrary movement trace files with node contact information, and an intuitive interface for accessing the contact-based metrics at arbitrary times during simulations.

Future Research Directions: The overall performance of hypergossiping proves its generality for deployment in arbitrary MANET scenarios. However, we propose some future research directions.

- Hypergossiping exploits the delay tolerated by the application to maximize the delivery ratio of broadcast messages. Ideally, applications tolerate delays that allow for the connectivity-in-time of all nodes. However, this requires the knowledge of the current and future MANET topology, which is a global view and hard to acquire in MANETs. Therefore, developing efficient and localized methods that can predict the required delay value to distribute broadcast messages to all destinations remains an open research issue. We propose to investigate the use of our contact-based mobility metrics to design such methods.

- The evaluation of hypergossiping with respect to network partitioning using our developed framework showed that there is some possible improvements for the broadcast repetition strategy. These improvements could increase the reachability and reduce the message overhead of hypergossiping. One example of improvements can be to fine-tune the adaptation of the partition join detection heuristic to the number of nodes.

- The results achieved through the mobility-aware buffering by using contact-based metrics, showed the feasibility of our approach and the utility of our mobility metrics. Hence, we propose to detect further relevant mobility patterns using the contact-based mobility metrics and incorporate them into a more sophisticated buffering strategy. We also suggest to consider relevant node capabilities, e.g. available storage, and message properties, e.g. residual lifetime, for further improvements of the buffering strategy.

- Another important issue that is gaining importance in MANETs is security. In this thesis, we assumed all entities to be trusted. Eventually, the design of broadcast protocols will also have to incorporate concepts that protect against malicious nodes. Like in epidemiology, we propose to use our methodology for analytical modeling to design counter measures and control policies to prohibit the spreading of malicious code such as worms and viruses in MANETs.

- Finally, the implementation of hypergossiping on different platforms for mobile devices would be a significant contribution for the existing MANET demonstrators [140] and for future real-world MANET deployments.

Bibliography

[1] P. Bellavista, A. Corradi, and E. Magistretti. REDMAN: An optimistic replication middleware for read-only resources in dense manets. *Pervasive and Mobile Computing*, 1(3):279–310, September 2005.

[2] J. Liu, F. Sailhan, D. Sacchetti, and V. Issarny. Group management for mobile ad hoc networks: Design, implementation and experiment. In *Proceedings of the 6th IEEE International Conference on Mobile Data Management (MDM)*, May 2005.

[3] E. Vollset and P.D. Ezhilchelvan. Design and performance-study of crash-tolerant protocols for broadcasting and reaching consensus in manets. In *Proceedings of the 24th IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 166–178, October 2005.

[4] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing (Kluwer Academic Publishers)*, 353, 1996.

[5] C. Perkins, E. Royer, and S. Das. Ad hoc on demand distance vector (aodv) routing. In *http://www.ietf.org/internet-drafts/draft-ieft-manet-aodv-03.txt*, 1999.

[6] K. Obraczka, G. Tsudik, and K. Viswanath. Pushing the limits of multicast in ad hoc networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2001.

[7] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 151–162, August 1999.

[8] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, August 2002.

[9] K. Viswanath and K. Obraczka. An adaptive approach to group communications in multi-hop ad hoc networks. In *Proceedings of the International Conference on Networking (ICN)*, 2002.

[10] K. Viswanath and K. Obraczka. An adaptive approach to group communications in multi hop ad hoc networks. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, 2002.

[11] S. McCanne and S. Floyd. Ns network simulator. In *http://www.isi.edu/nsnam/ns/*.

[12] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. Glomosim: A scalable network simulation environment. In *TR 990027, UCLA Computer Science Department*, May 1999.

[13] Qualnet. In *http://www.scalable-networks.com/*.

[14] Opnet modeler. In *http://www.opnet.com/products/modeler/home.html*.

[15] K. Viswanath and K. Obraczka. Modeling the performance of flooding in wireless multi-hop ad hoc networks. In *Proceedings of the 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, 2004.

[16] H. Zhang and Z.P. Jiang. Theoretic analysis of broadcasting schemes for two-dimensional manets. In *Proceedings of the Workshop on Performance of Wireless Networks and Communication Systems (WiNCS)*, July 2005.

[17] B. Williams, D.P. Mehta, T. Camp, and W. Navidi. Predictive models to rebroadcast in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 3(3):295–303, July-September 2004.

[18] M. Papadopouli and H. Schulzrinne. Seven degrees of separation in mobile ad hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Nov. 2000.

[19] O. Diekmann and J.A.P. Heesterbeek. *Mathematical Epidemiology of Infectious Diseases.* 2000. series in mathematical and computational biology.

[20] F. Brauer and C. Castillo-Chavez. *Mathematical models in population biology and epidemiology.* 2000.

[21] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM)*, pages 174–185, 1999.

[22] A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in MANETs. In *Proceedings of the ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, pages 54–60, September 2002.

[23] Y.C. Tsenga, S.Y. Ni, , and E.Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc networks. *IEEE Transactions on Computers*, 52(5):545–557, May 2003.

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

[25] A. Khelil, P.J. Marrón, C. Becker, and K. Rothermel. Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. In *Proceedings of The 2005 Conference on Communication in Distributed Systems (KiVS)*, February 2005.

[26] A. Khelil, P.J. Marrón, C. Becker, and K. Rothermel. Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. *Elsevier Ad Hoc Networks Journal*, 5(5):531–546, July 2007.

[27] O. Wolfson and B. Xu. Data-on-the-road in intelligent transportation systems. In *Proceedings of the IEEE International Conference on Networking, Sensing, and Control (ICNSC)*, March 2004.

[28] A. Khelil, P.J. Marrón, R. Dietrich, and K. Rothermel. Evaluation of partition-aware MANET protocols and applications with ns-2. In *Proceedings of the 2005 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, July 2005.

[29] A. Khelil, P.J. Marrón, and K. Rothermel. Contact-based mobility metrics for delay-tolerant ad hoc networking. In *Proceedings of The 13th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 435–444, September 2005.

[30] A. Khelil. Mobility-aware buffering for delay-tolerant broadcasting in MANETs. In *Proceedings of the 2006 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, July 2006.

[31] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.

[32] Y. Yi, M. Gerla, and T.J. Kwon. Efficient flooding in ad hoc networks: A comparative performance study. In *Proceedings of the IEEE International Conference on Communications (ICC)*, 2003.

[33] F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):1027–1040, November 2004.

[34] I. Kang and R. Poovendran. A comparison of power-efficient broadcast routing algorithms. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, December 2003.

[35] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. Mddv: A mobility-centric data dissemination algorithm for vehicular networks. In *Proceedings of the First ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, October 2004.

[36] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[37] C. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. On the scalability of ad hoc routing protocols. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, June 2002.

[38] O. Arpacioglu and Z. Haas. On the scalability and capacity of planar wireless networks with omnidirectional antennas. *Wireless Communications and Mobile Computing, Special Issue on Scalability Issues in Wireless Networks*, 4(3):263–279, May 2004.

[39] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1998.

[40] A. Khelil. Contact-based buffering for delay-tolerant ad hoc broadcasting. *to appear in the Elsevier Computer Communications Journal: Special issue on Advances in Communication Networking*, 2007.

[41] C. Siva Ram Murthy and B.S. Manoj. *Ad Hoc Wireless Networks, Architectures and Protocols.* 2004.

[42] T. Korsch. Den horizont der fahrassistenz erweitern: Vorrausschauende systeme durch ad-hoc vernetzung. In *BMW Group Forschung und Technik, www.ftm.mw.tum.de/zubehoer/pdf/Tagung_AS/18_kosch.pdf.*

[43] The cartalk2000 project. In *www.cartalk2000.net.*

[44] J. Tian, P.J. Marrón, and K. Rothermel. Location-based hierarchical data aggregation for vehicular ad hoc networks. In *Proceedings of The 2005 Conference on Communication in Distributed Systems (KiVS)*, February 2005.

[45] The network on wheels (now) project. In *www.network-on-wheels.de.*

[46] The PReVENT project. In *www.prevent-ip.org.*

[47] The fleetnet project. In *www.et2.tu-harburg.de/fleetnet/.*

[48] The carTel project. In *http://cartel.csail.mit.edu/.*

[49] The Car2Car Communication Consortium. In *www.car-to-car.org.*

[50] The DSRC project. In *www.leearmstrong.com/DSRC/DSRCHomeset.htm.*

[51] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, 2002.

[52] I.F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: Research challenges. *Ad Hoc Networks Journal (Elsevier)*, 2:351–367, October 2004.

[53] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Sytems (ASPLOS)*, 2002.

[54] B. Thorstensen, T. Syversen, T. Walseth, and T.A. Bjornvold. Electronic shepherd - a low-cost, low-bandwidth, wireless network system. In *Proceedings of the International Conference of Mobile Systems, Applications, and Services (MOBISYS)*, 2004.

[55] T. Small and Z.J. Haas. The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way). In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, June 2003.

[56] A. Lindgren, A. Doria, and O. Schelén. Poster: Probabilistic routing in intermittently connected networks. In *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, June 2003.

[57] R.C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pages 30–41, May 2003.

[58] L.M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, April 2001.

[59] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM)*, pages 85–96, 2001.

[60] H. Zhou, L.M. Ni, and M.W. Mutka. Prophet address allocation for large scale manets. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, pages 1304–1311, April 2003.

[61] G. Karumanchi, S. Muralidharan, and R. Prakash. Information dissemination in partitionable mobile ad hoc networks. In *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 4–13, October 1999.

[62] C. Becker, M. Bauer, and J. Hähner. Usenet-on-the-fly: supporting locality of information in spontaneous networking environments. In *Proceedings of the Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments (CSCW)*, 2002.

[63] J. Su, A. Chin, A. Goel, and E.D. Lara. User mobility for opportunistic ad-hoc networking. In *Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications*, 2004.

[64] C. Lindemann and O. Waldhorst. Epidemic dissemination of presence information in mobile instant messaging systems. In *Proceedings of The 2005 Conference on Communication in Distributed Systems (KiVS)*, Februar 2005.

[65] J. Hähner, K. Rothermel, and C. Becker. Update-linearizability: A consistency concept for the chronological ordering of events in manets. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[66] A. Ouksel, O. Wolfson, and B. Xu. Opportunistic resource exchange in inter-vehicle ad hoc networks. In *Proceedings of the IEEE International Conference on Mobile Data Management (MDM)*, January 2004.

[67] DTNRG: The Delay Tolerant Networking Research Group. In *www.dtnrg.org*.

[68] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proceedings of the annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, Aug/Sep 2004.

[69] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, September 2003.

[70] V. Cerf et al. Delay-tolerant network architecture. In *Internet draft, draft-irtf-dtnrg-arch-08.txt*, December 2006.

[71] M. Sun, W. Feng, and T.H. Lai. Location aided broadcast in wireless ad hoc networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 2842–2846, November 2001.

[72] W. Peng and X.C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, November 2000.

[73] M. Heissenbüttel, T. Braun, M. Waelchli, and T. Bernoulli. Optimized stateless broadcasting in wireless multi-hop networks. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, April 2006.

[74] Z.J. Haas, J.Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, June 2002.

[75] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. In *Proc. of the International Conference on Distributed Computing Systems (ICDCS)*, pages 460–467, 2003.

[76] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, pages 61–68, August 2000.

[77] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. In *CAIP Technical Report 248 - Rutgers University*, September 2000.

[78] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, August 1999.

[79] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, January 2002.

[80] M. Seddigh, J.S. Gonzalez, and I. Stojmenovic. Rng and internal node based broadcasting algorithms for wireless one-to-one networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(2):37–44, April 2001.

[81] A. Laouiti, A. Qayyum, and L. Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, 2002.

[82] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2), April-June 2002.

[83] W. Peng and X. Lu. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Computer Science Technology*, 16(2):114–125, 2001.

[84] M. Gerla and T.J. Kwon. Efficient flooding with passive clustering (pc) in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 32:44–56, January 2002.

[85] Y. Yi, M. Gerla, and T.J. Kwon. Efficient flooding in ad-hoc networks using on-demand (passive) cluster formation. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, June 2003.

[86] W. Lou and J. Wu. Double-covered broadcast (dcb): A simple reliable broadcast algorithm in manets. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, March 2004.

[87] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, pages 585–594, March 2000.

[88] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, pages 2210–2217, March 2003.

[89] I. Kang and R. Poovendran. Maximizing static network lifetime of wireless broadcast adhoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2003-1.

[90] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proc. of IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, pages 1388–1397, April 2001.

[91] Y. Wang and J.J. Garcia-Luna-Aceves. Broadcast traffic in ad hoc networks with directional antennas. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 210–215, December 2003.

[92] C. Hu, Y. Hong, and J. Hou. On mitigating the broadcast storm problem with directional antennas. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 104–110, May 2003.

[93] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized energy efficient broadcast for wireless networks with directional antennas. In *Proceedings of the Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, September 2002.

[94] I. Kang, R. Poovendran, and R. Ladner. Power-efficient broadcast routing in adhoc networks using directional antennas: technology dependence and convergence issues. In *Tech. Rep. UWEETR-2003-0015, University of Washington*, July 2003.

[95] J. Hähner, C. Becker, and K. Rothermel. A protocol for data dissemination in frequently partitioned ad-hoc networks. In *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC)*, 2003.

[96] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, pages 64–71, August 1999.

[97] W. Goffman and V.A. Newill. Generation of epidemic theory, an application to the transmission of ideas. *Nature*, 204:225–228, 1964.

[98] A. Demers, D. Greene, C. Hauser, J. Larson W. Irish, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. *ACM SIGOPS Operating Systems Review*, 22:8–32, 1988.

[99] J.O. Kephart and S.R. White. Directed-graph epidemiological models of computer viruses. In *Proceedings of the 1991 IEEE Computer Society Symposium on research in security and privacy*, 1991.

[100] Wolfram Research Inc. Mathematica, version 4.0, 1999.

[101] L. Kleinrock and J. Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *Proceedings of the IEEE National Telecommunications Conference (NTC)*, pages 431–435, 1978.

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

[103] M. Sanchez, P. Manzoni, and Z.J. Haas. Determination of critical transmission range in ad-hoc networks. In *Multiaccess Mobility and Teletraffic for Wireless Communications Workshop (MMT)*, 1999.

[104] R.G. Cole, N. Phamdo, M. Abu Rajab, and A. Terzis. Requirement on worm mitigation technologies in manets. In *Proceedings of the 19th Workshop on Parallel and Distributed Simulation (PADS)*, June 2005.

[105] S.A. Khayam and H. Radha. Poster, analyzing the spread of active worms over vanet. In *Proceedings of the First ACM workshop on Vehicular ad hoc networks (VANET)*, pages 86 – 87, October 2004.

[106] H. Wu, R. Fujimoto, and G. Riley. Analytical models for information propagation in vehicle-to-vehicle networks. In *Proceedings of the International Conference on Vehicular Technologies (VTC)*, 2004.

[107] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[108] F. Bai and A. Helmy. A survey of mobility models in wireless adhoc networks. In *Book chapter in the book on "Wireless Ad Hoc and Sensor Networks", Kulwer Academic Publishers*, June 2004.

[109] F. Bai, N. Sadagopan, and A. Helmy. Important: A framework to systematically analyze the impact of mobility on performance of routing protocols in adhoc networks. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, April 2003.

[110] C. Bettstetter and C. Wagner. The spatial node distribution of the random waypoint mobility model. In *Proceedings of the 1st German Workshop on Mobile Ad-Hoc Networks (WMAN)*, March 2002.

[111] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, April 2003.

[112] J. Yoon, M. Liu, and B. Noble. Sound mobility models. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM)*, pages 205–216, September 2003.

[113] W. Navid and T. Camp. Stationary distributions for the random waypoint model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, 2004.

[114] X. Hong, M. Gerla, R. Bagrodia, and G. Pei. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, pages 53–60, August 1999.

[115] J. Tian, J. Hähner, C. Becker, I. Stepanov, and K. Rothermel. Graph-based mobility model for mobile ad hoc network simulation. In *Proceedings of the 35th Annual Simulation Symposium (ANSS)*, April 2002.

[116] bonnmotion. In *www.informatik.uni-bonn.de/IV/BonnMotion/*.

[117] I. Stepanov, J. Hähner, C. Becker, J. Tian, and K. Rothermel. A meta-model and framework for user mobility in mobile networks. In *Proceedings of the 11th International Conference on Networking 2003 (ICON)*, October 2003.

[118] J. Mickens and B. Noble. Modeling epidemic spreading in mobile environments. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, pages 77–96, September 2005.

[119] M. Musolesi and C. Mascolo. Epidemic-style communication primitives exploiting network topology information. In *Technical Report RN/05/05, UCL department of computer science, UK*, 2005.

[120] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Proceedings of the IEEE 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM)*, June 2005.

[121] P. Tennent, M. Hall, B. Brown, M. Chalmers, and S. Sherwood. Three applications for mobile epidemic algorithms. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, 2005.

[122] J. Kurhinen and J. Vuori. Information diffusion in a single-hop mobile peer-to-peer network. In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC)*, pages 137–142, 2005.

[123] M. Östergren and O. Juhlin. Road talk: A roadside location-dependent audio message system for car drivers. *Journal of Mobile Multimedia, Rinton Press, Princeton, New Jersey*, 1(1):47–61, 2005.

[124] A. Datta, S. Quarteroni, and K. Aberer. Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in mobile ad-hoc networks. In *Proceedings of the International Conference on Semantics of a Networked World (ICSN)*, 2004.

[125] P. Han, K. Hosanagar, and Y. Tan. Diffusion of digital products in peer-to-peer networks. In *Proceedings of the International Conference on Information Systems (ICIS)*, pages 505–516, 2004.

[126] P. Downey and R. Cardell-Oliver. Evaluating the impact of limited resource on the performance of flooding in wireless sensor networks. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, June 2004.

[127] R. Cardell-Oliver. Why flooding is unreliable in multi-hop, wireless networks. In *Technical Report UWA-CSSE-04-001*, February 2004.

[128] J. Hähner, D. Dudkowski, P.J. Marrón, and K. Rothermel. A quantitative analysis of partitioning in mobile ad hoc networks. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (Sigmetrics-Performance) (extended abstract)*, pages 12–16, June 2004.

[129] J. Boleng, W. Navidi, and T. Camp. Metrics to enable adaptive protocols for mobile ad hoc networks. In *Proceedings of the International Conference on Wireless Networks (ICWN)*, pages 293–298, 2002.

[130] M. Gerharz, C.D. Waal, M. Frank, and P. Martini. Link stability in mobile wireless ad hoc networks. In *Prodeecings of the 27th Annual IEEE Conference on Local Computer Networks (LCN)*, pages 30–39, November 2002.

[131] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. Paths: Analysis of path duration distributions in manet and their impact on routing protocols. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 245–256, June 2003.

[132] Y.C. Hu and D.B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM)*, pages 231–242, 2000.

[133] M. Grossglauser and M. Vetterli. Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, 2003.

[134] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 257–266, June 2003.

[135] K. Wang and B. Li. Group mobility and partition prediction in wireless ad-hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1017–1021, April 2002.

[136] K. Wang and B. Li. Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks. In *Proceedings of the IEEE Joint Conference of Computer and Communication Societies (INFOCOM)*, pages 1089–1098, June 2002.

[137] Z.R. Zaidi, B.L. Mark, and R.K. Thomas. A two-tier representation of node mobility in ad hoc networks. In *Proceedings of the IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, October 2004.

[138] X. Hong and M. Gerla. Dynamic group discovery and routing in ad hoc networks. In *Proceedings of the First Annual Mediterranean Ad Hoc Networking Workshop (Med-hoc-Net)*, September 2002.

[139] G. Karlsson, V. Lenders, and M. May. Delay-tolerant broadcasting. *IEEE Transactions on Broadcasting*, April 2007.

[140] W. Kiess and M. Mauve. A Survey on Real-World Implementations of Mobile Ad-Hoc Networks. *Elsevier Ad Hoc Networks Journal*, 5(3):324–339, April 2007.

Index

Abdelmajid Khelil

Alte Hochheimer Str. 37
D-65439 Flörsheim
adkhelil@web.de

EDUCATION

Ph.D. in Computer Science, April 2007
Universität Stuttgart, Germany

Diploma in Electrical Engineering, September 2000
Universität Stuttgart, Germany

RESEARCH

Post-Doc: Technische Universität Darmstadt, since 03/2006
Dependable, Embedded Systems & Software Group,
Cooperative, Adaptive and Responsive Monitoring
in Mixed Mode Environments.

Research Assistant: Universität Stuttgart, 10/2000-9/2005
Distributed Systems Group,
Mobile Ad Hoc Networks & Wireless Sensor Networks.

TEACHING

Teaching Assistant: 10/2000-9/2005 & 03/2006-now
Wrote and graded assignments and tutored students for
- Lecture ”Computer Networks”
- Lecture ”Introduction to Distributed Systems”

Provided own courses
- Short course ”Network Programming in Java”
- Several seminars and labs

PUBLICATIONS

- A. Khelil, C. Becker, J. Tian, and K. Rothermel ”An Epidemic Model for Information Diffusion in MANETs”, in Proc. of the International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems at MobiCom (MSWiM), Atlanta, USA, 2002.
- A. Khelil, P.J. Marron, C. Becker, and K. Rothermel ”Hypergossiping: A Generalized Broadcast Strategy for Mobile Ad Hoc Networks”, in

Proc. of Communication in Distributed Systems 2005 (KiVS), Kaiserslautern, Germany, 2005.

- A. Khelil, P.J. Marron, R. Dietrich, and K. Rothermel ”Evaluation of Partition-Aware MANET Protocols and Applications with ns-2”, in Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Philadelphia, USA, 2005.
- A. Khelil, P.J. Marron, and K. Rothermel ”Contact-Based Mobility Metrics for Delay-Tolerant Ad Hoc Networking”, in Proc. of the 13th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Atlanta, USA, 2005.
- A. Khelil, P.J. Marron, C. Becker, and K. Rothermel ”Hypergossiping: A Generalized Broadcast Strategy for MANETs”, the Elsevier Ad Hoc Networks Journal, 5(5):531-546, July 2007.
- A. Khelil ”Mobility-Aware Buffering for Delay-Tolerant Broadcasting in MANETs”, in Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Calgary, Canada, 2006.
- A. Khelil ”Contact-Based Buffering for Delay-Tolerant Ad Hoc Broadcasting”, to appear in the Elsevier Computer Communications Journal: SI on Advances in Communication Networking, 2007.

HONORS & AWARDS

- Scholarship granted by Tunisian Government for pursuing studies in Germany
- DaimlerChrysler Scholarship Program for best students of top German universities

MEMBERSHIPS

- IEEE: The Institute of Electrical and Electronics Engineers
- SCS: The Society for Modeling and Simulation International
- ACS: Arab Computer Society

June 16, 2007.