Using Geographic Models in the Simulation of Mobile Communication

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Illya Stepanov
aus Aluschta (Ukraine)

Hauptberichter:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
1. Mitberichter:	Prof. Dr.-Ing. Volodymyr Svjatnyj
2. Mitberichter:	Prof. Dr.-Ing. Dr. h. c. Peter Göhner

Tag der mündlichen Prüfung: 6. Oktober 2008

Institute für Parallele und Verteilte Systeme (IPVS)
der Universität Stuttgart
2008

Acknowledgements

This research would not have been possible without the help and support of many kind people around me.

I would like to thank Professor Kurt Rothermel (Universität Stuttgart) and Professor Volodymyr Svjatnyj (Donetsk National Technical University) for the cooperation among the universities. Professor Rothermel was my supervisor over the past years. Through his guidance I learned a lot about performing scientific research, expressing the results in written form, and presenting them. I would also like to thank Professor Göhner, Professor Frühauf, and Professor Roth-Stielow for kindly accepting to be a part of my PhD committee.

I would like to thank all my colleagues for their valuable suggestions and comments on my work. A special thanks goes out to Olga Saukh for reading the initial version of this thesis and for her valuable feedback. Thanks to Sean Cummins for correcting the grammar in many parts of the thesis. I also appreciate the help of Philipp Wertz on the radio propagation tool.

I am grateful to the Land Baden-Württemberg for supporting me through the Landesgraduiertenförderung scholarship.

A separate thanks to my friends for correcting different parts of this thesis and for just being around me all this time. Thanks to my parents and to my wife Natalia for their patience and support.

Abstract

Network simulation tools are frequently used for the performance analysis of mobile networks. Their common shortcoming lies within the approaches they use for the modeling of user mobility and radio wave propagation. The provided mobility models describe random movements within the area, which is similar to the motion of molecular particles. For the modeling of a radio channel, the tools assume a line of sight between communicating nodes, and thus, a simple dependency of the signal loss to the distance from the transmitter. These models poorly reflect real scenarios, in which the characteristics of the spatial environment have a significant impact on the network performance.

In this thesis more realistic mobility and radio propagation models are described and integrated into a network simulation. These models are based on the solutions from related research areas like physics, transportation planning, traffic modeling, and electrical engineering, which have been validated against real-world data. They consider digital maps of the simulation area, which are taken from a geographic information system (GIS). This thesis analyzes common geospatial data standards to provide input to the used mobility and radio propagation models.

The proposed user-oriented mobility meta-model considers three key factors, which determine user mobility in the area: a spatial environment, user travel decisions, and a user movement dynamics. Different methods for reflecting various aspects of user movement are described along with their integration into the proposed mobility model. They allow for the modeling of network scenarios under diverse assumptions of a modeler. The selection of model parameters for concrete scenarios is explained.

For the better modeling of radio wave propagation, a network simulator is extended with the intelligent ray tracing model. This model relies on a map of the simulation area. It uses the methods similar to ones from computer graphics for determining possible signal paths between the transmitter and the receiver. The

model is combined with the small-scale fading and bit-error modeling approaches for obtaining more realistic simulation results.

The evaluations show significant differences between the simulation results obtained with simpler and more realistic models. It is caused by the changes in the distribution of network users due to their mobility in the area and the obstacles of the propagation environment, which simple models cannot reflect.

Kurzfassung

Verwendung von Geographischen Modellen in der Simulation von mobiler Kommunikation

1. Einleitung

Die neuesten Entwicklungen im Bereich der portablen Rechnerplattformen und der Mobilkommunikation führen zu einer stetigen Steigerung der Popularität von drahtlosen Netzen. In Netzen mit Infrastrukturunterstützung werden die Klienten zu einem Access-Point (Basisstation) verbunden, der die Geräte mit einem größeren Netz oder mit dem Internet verknüpft. In Ad-hoc-Netzen müssen die Netzgeräte ohne Infrastrukturunterstützung auskommen und daher kommunizieren sie nur miteinander, wenn sie sich in Kommunikationsreichweite befinden. Die Geräte dienen nicht nur als Sender oder Empfänger, sondern auch als Router für die anderen Netzknoten. Darunter gibt es ein mobiles Ad-hoc-Netz (MANET), das von mobilen Geräten, wie tragbare Minicomputer, Laptops oder Mobiltelefone aufgebaut wird. Aufgrund der Mobilität ändert sich die Netztopologie dabei ständig, daher werden die Routingentscheidungen dynamisch getroffen, basierend auf der aktuellen Konnektivität. Viele Anwendungsszenarien existieren für solche Netze, sowohl im zivilen als auch in militärischen Bereich. Beispiele dafür sind der Informationsaustausch zwischen mobilen Benutzern in stark frequentierten Gebieten wie Stadtzentren [JHP+03], die Verbreitung von Meldungen und Nachrichten [BBH02], Fahrer-Assistenzsysteme, die auf Kommunikation zwischen Fahrzeugen basieren [RMM+00] oder der Nachrichtenaustausch zwischen Einsatzkräften für Rettungseinsätze [JLH+99].

Netzsimulatoren ([BEF+00], [ZBG98], [Ril03]) werden häufig benutzt, um die Leistung von Anwendungen und Netzprotokollen in bestimmten Szenarien zu analysieren oder zu vergleichen. Die Werkzeuge bieten ein Simulationsmodell, das die Netzknoten abstrahiert und die folgenden Komponenten enthält: mobile

Anwendungen, Protokollstapel, Kommunikationskanal und Benutzermobilität. Da sich die Forschung bisher auf Netzprotokolle und Anwendungen konzentrierte, können diese als der am weitesten entwickelte Bestandteil von Simulationswerkzeugen betrachtet werden. Für Kommunikationskanal und die Beschreibung der Nutzermobilität kommen dagegen sehr einfache Modelle zum Einsatz, so dass sie realistischere Szenarien schlecht nachbilden können. Um Mobilität zu simulieren, bieten die meisten Werkzeuge nur einfache Modelle an, die eine zufällige Bewegung in einem Gebiet annehmen (z.B. [BMJ+98], [CJB+01], [DCYS98], [KV98]), ähnlich zur Brownschen Molekularbewegung [Nel67]. Um die Kanaleigenschaften bzw. Empfangsleistung zu berechnen, werden Wellenausbreitungsmodelle (z.B. das „Free Space“ Modell [RMK97]) eingesetzt, die ungestörte Wellenausbreitung voraussetzen und keine geographische Information über die Umgebung berücksichtigen.

Solche Modelle sind für realistische Szenarien, in denen die Umgebung die Netzleistung stark beeinflusst, schlecht geeignet. Die Wellenausbreitung wird von Hindernissen, wie Gebäuden, gestört und von diesen einfacheren Modellen nicht berücksichtigt. Die Umgebung schränkt die Benutzermobilität ein, so dass die Menschen z.B. Straßenverläufen folgen und nicht durch Mauern laufen. Die Routen sind außerdem in der Realität nicht zufällig gewählt. So ordnen sich z.B. Straßenverkehrsteilnehmer dem gesamten Verkehrsverlauf unter.

Diese Dissertation beschäftigt sich mit der Entwicklung von realistischeren Modellen und deren Integration in eine Netzsimulation. Die Modelle berücksichtigen die geographische Umgebung und das Benutzerverhalten. Diese Modelle basieren auf den Methoden aus verwandten Forschungsgebieten, wie Physik, Verkehrsplanung, Verkehrsdynamik, Nachrichtentechnik und ermöglichen damit realistischere Simulationen von mobilen Netzen. Evaluationen zeigen dabei signifikante Änderungen in den Simulationsergebnissen durch die Anwendung der realistischeren Modelle.

Diese Arbeit leistet folgende wissenschaftliche Beiträge:

- Analyse von standardisierten Beschreibungsformen für räumliche Gebiete, um ein generisches räumliches Modell zu entwickeln

- Integration vom entwickelten räumlichen Modell in eine Netzsimulationsumgebung
- Zusammenfassung der Verfahren zur Benutzermobilität- und Kommunikationskanalmodellierung aus verwandten Forschungsgebieten
- Entwicklung eines realistischeren Mobilitätsmodells („User-Oriented Mobility Meta-Model“), das die Hauptfaktoren, wie die Umgebung und das Benutzerverhalten, in Stadtszenarien berücksichtigt
- Integration eines realistischeren Wellenausbreitungsmodells („Intelligent Ray Tracing Model“), das auf der Strahlenverfolgung basiert und die Stadtumgebung berücksichtigt, in ein Simulationswerkzeug
- Analyse von Auswirkungen der realistischeren Modelle auf Simulationsergebnisse
- Entwicklung eines frei verfügbaren Programmpakets, das die Parser für räumliche Daten in verschiedenen Formaten, ein Framework für Mobilitätsmodellierung und die Unterstützung von realistischeren Wellenausbreitungsmodellen für Netzsimulatoren anbietet

Die beschriebene Arbeit wurde ursprünglich für die Outdoor Szenarien gemacht. Aus diesem Grund enthält die Dissertation hauptsächlich die Outdoor Beispiele und verwendet ein Outdoor Szenario für die Evaluierungen. Diese Arbeit kann aber auch nach einer kleinen Definitionsänderung in Indoor Szenarien eingesetzt werden, d.h. statt Straßen und Gebäuden sollten Flure und Räume verwendet werden.

Die eigentliche Dissertation ist in englischer Sprache verfasst und besteht aus sechs Kapiteln. Neben einer allgemeinen Motivation in Kapitel 1 werden in Kapitel 2 gebräuchliche standardisierte Beschreibungsformen für Umgebungsdaten analysiert. Diese Daten werden von den beschriebenen Mobilitäts- und Wellenausbreitungsmodellen verwendet. Kapitel 3 befasst sich mit der Mobilitätsmodellierung. In diesem Kapitel werden die verwandten Arbeiten aus dem Bereich von mobilen Netzen und ihre Nachteile beschrieben und das eigene Verfahren untersucht, das auf den Methoden der speziellen Forschungsbereiche basiert. Kapitel 4 behandelt die Modellierung des physikalischen Kanals. Das Kapitel beschreibt die bereits existierenden Ansätze, die von Netzsimulatoren unterstützt

werden. Danach wird ein realistischeres Verfahren eingeführt („Intelligent Ray Tracing Model“), das am Institut für Hochfrequenztechnik der Universität Stuttgart, entwickelt wurde. Dieses wird in eine Netzsimulationsumgebung zusammen mit Modellen der Signalabschwächung und der Bitfehlerhäufigkeit integriert. Kapitel 5 vergleicht die Simulationsergebnisse, die aufgrund von einfacheren und realistischeren Modellen entstanden sind. Hierfür wird ein mobiles Ad-hoc-Netz anhand eines Innenstadtszenarios in Stuttgart simuliert; die Auswirkungen auf die Netzkonnektivität, Leistung von Routing Protokollen und mobilen Anwendungen werden dabei analysiert. Kapitel 6 schließt diese Abhandlung mit einer Zusammenfassung des wissenschaftlichen Beitrags und der Ergebnisse sowie einem Ausblick auf zukünftige Forschungsthemen ab.

2. Darstellung von Umgebungsdaten

Für viele Anwendungen, wie Lokations- und Navigationsdienste, sind digitale Karten des Ortes unabdingbar. Dies motiviert die Entwicklung von standardisierten Beschreibungsformen für räumliche Daten. Eine Gruppe davon basiert auf der „Geography Markup Language“ (GML) [ISO04], welche die Basisregeln zur Beschreibung der Objekte und mögliche Erweiterungsmechanismen definiert. Die andere Gruppe stellt die Formate dar, die nicht auf der GML beruhen und typischerweise binäre Darstellung verwenden, z.B. „Geographic Data Files“ (GDF) [CEN95]. Da GDF und die GML-basierte Formate die gebräuchlichsten sind, werden diese zwecks Dateneingaben an die Mobilitäts- und Wellenausbreitungsmodellen genauer untersucht.

GML basiert auf der „Extensible Markup Language“ (XML) [W3C04] und bietet XML-Schemata für die Beschreibung von Umgebungsdaten an. Das Gebiet wird durch eine Menge von räumlichen Objekten (z.B. Straßen, Kreuzungen, Gebäude für Außenobjekte, Flure und Räume für Innenobjekte) dargestellt. Die Objektsbeschreibung besteht aus der Typinformation, Geometrie und den Attributen. Für die Geometriebeschreibung bietet GML die notwendigen geometrischen Primitiven an, wie Punkt, Linie, Kurve, Polygon oder Fläche. Die Koordinaten

werden innerhalb eines räumlichen Bezugssystems definiert. GML unterstützt eine Vielfalt von möglichen Koordinaten- und Bezugssystemen, z.B. kartesische, ellipsoidale, polare usw. Eigene Systeme können ebenfalls definiert werden. Die Objektattribute sind als {Name, Wert}-Paare gespeichert. Der Standard definiert nur die Basisattribute, wie Objektidentifikatoren, Objektnamen oder die freie textuelle Bezeichnungen. Eine detaillierte Spezifikation von möglichen Attributen und Objekttypen ist jeweils in Schemata gespeichert, die sich zwischen den Anwendungen unterscheiden können. GML unterstützt auch Relationen zwischen Objekten. Diese werden benutzt, um z.B. zu zeigen, dass ein Gebäude neben einer Straße liegt oder ein Manöver zwischen zwei Straßenabschnitten verboten ist. Ein räumliches Objekt kann ebenfalls andere (kleinere) Objekte kapseln.

GDF wurde ursprünglich für Fahrzeugsnavigationssysteme entwickelt, wird aber inzwischen auch in anderen Bereichen benutzt. Der Standard nutzt ein eigenes Format für die Datenspeicherung und spezifiziert mögliche Objekttypen und deren Attribute. Dies maximiert die Interoperabilität zwischen Anwendungen. Die Umgebungsbeschreibung wird in drei so genannte Kataloge unterteilt: Objekte („Feature Catalogue“), deren Attribute („Attribute Catalogue“) und Relationen („Relationship Catalogue“). Objekte werden auf drei Präsentationsebenen beschrieben: Geometrie (Ebene 0) mit den kartographischen Primitiven, wie Punkt, Linie, Polygon, einfache Elemente (Ebene 1), wie Kreuzungen, Straßenabschnitte und Gebäude, und komplexe Elemente (Ebene 2), wie Straßen und große Kreuzungsbereiche, die sich aus einfachen Elementen zusammensetzen. Zusätzlich bietet GDF die Beschreibung von so genannten „Points of Interest“ (Supermärkte, Museen, Restaurants usw.), die für die Benutzer von besonderem Interesse sind. Ähnlich zu GML werden die Objektattribute und Relationen als {Name, Wert}-Paare gespeichert.

Aus dem Vergleich der zwei gebräuchlichsten standardisierten Beschreibungsformen ergeben sich deren strukturelle Ähnlichkeiten, die die Arbeit mit verschiedenen Umgebungsdatenformaten erleichtern und eine Erstellung einer formatunabhängigen Abstraktion ermöglichen:

- Ein Gebiet wird durch eine Menge von räumlichen Objekten dargestellt

- Die Objekte können ebenfalls in einer Hierarchie repräsentiert werden (dies entspricht der Kapselung von Objekten in GML oder der mehrstufigen Repräsentation in GDF)
- Die Geometrie wird mit typischen Primitiven beschrieben
- Die Koordinaten werden innerhalb eines Bezugssystems definiert
- Die Objektattribute und Relationen werden als {Name, Wert}-Paare gespeichert

Die Studien in [VGH+02] zeigen, dass andere standardisierte Beschreibungsformen für Umgebungsdaten ebenfalls diesem Paradigma folgen. Daraus ergibt sich folgende Repräsentation von Straßennetzen und Gebäudedaten, die wir für die Spezifikation und die Entwicklung unseres Generischen Geographischen Datenmodells benutzen. Dieses Modell wird als Eingabe für die Mobilitäts- und Wellenausbreitungsmodelle verwendet.

Das Straßennetz besteht aus Kreuzungen und Straßenabschnitten, die die Kreuzungen verbinden. Diese Repräsentation ist einem Graphen gleich, in dem die Knoten die Kreuzungen und die Straßenendpunkte darstellen und die Kanten die Straßenabschnitte. Abhängig von der Entscheidung des Datenanbieters kann ein Straßenabschnitt eine oder mehrere Spuren beschreiben. Diese Repräsentationen sind austauschbar: im ersten Fall wird durch zusätzliche Informationen (z.B. andere Repräsentationsstufen in GDF) darauf hingewiesen, dass die bestimmten Spuren zu denselben Straßenabschnitten gehören. Im zweiten Fall wird die Anzahl der Spuren und deren Fahrtrichtungen mit der Hilfe von Attributen erläutert. Im Graphen werden die Bewegungsflächen sowohl für Fußgänger als auch für Fahrzeuge gespeichert. Die Objektattribute werden benutzt, um zwischen diesen Bereichen zu unterscheiden.

Die Geometrie von Straßennetzobjekten wird mit Vektoren beschrieben. Dabei werden die Kreuzungen durch Punkte und die Straßenabschnitte durch Linien definiert. Die Objektkoordinaten beziehen sich auf ihren Mittelpunkt. Koordinaten werden ins kartesische System konvertiert, um die Datenmanipulation zu erleichtern. Diese Darstellung wird mit der Hilfe von Objektattributen (z.B. Straßenbreite, Anzahl der Spuren usw.) erweitert.

Die Attribute enthalten zusätzliche objektspezifische Eigenschaften. Diese können beschreibenden (wie Straßenname, Straßenbreite), restriktiven (erlaubte Geschwindigkeit, erlaubte Fahrzeugtypen) oder systemspezifischen (Genauigkeit der Positionsdaten) Charakter haben. Die Eigenschaften können auch eine gewisse Gültigkeit haben, z.B. eine Fahrtrichtung, zwischen bestimmten Positionen u.Ä. Die Relationen beschreiben assoziative („ein Teil von“) oder restriktive („ein verbotenes Manöver“) Verbindungen zwischen Straßenobjekten. Da die Beschreibungsformen für Umgebungsdaten verschiedene Bezeichnungen und Werte für die Attribut- und Relationsnamen verwenden, muss auf diese Informationen abhängig vom bestimmten Format zugegriffen werden. Im Gegensatz zu GML, das nur die Basisattribute und Relationen spezifiziert, enthält GDF die vollständige Beschreibung und daher wird GDF als die Schnittstelle zum Generischen Geographischen Datenmodell verwendet. Die GML-Datenquellen werden beim Lesen in GDF konvertiert.

Ähnlich zur Straßennetzbeschreibung werden die Gebäude als räumliche Objekte mit ihrer Geometrie, ihren Attributen und Relationen dargestellt. Die Geometrie ist durch Vektoren definiert. Aufgrund der komplexeren Datenverarbeitung wird die 3-D Repräsentation selten verwendet. Gebäude werden dagegen durch 2-dimensionale Polygone ihrer Grundfläche und gleichmäßiger Höhe angenähert (2,5-D Darstellung). Die Spezifikation von Attributen und Relationen entspricht deren Beschreibung innerhalb von Straßennetzen.

3. Modellierung von Benutzermobilität

Mobile Netze unterstützen nomadische Benutzer durch spezielle Algorithmen, Kommunikationsprotokolle und Anwendungen. Um deren Leistungsfähigkeit zu evaluieren und zu vergleichen, müssen Simulationswerkzeuge auch die Benutzermobilität modellieren. Viele Studien (z.B. [CBD02], [NG03], [THB+02]) zeigen, dass die verwendeten Mobilitätsmodelle Simulationsergebnisse stark beeinflussen können. Für aussagekräftige Simulationsergebnisse ist daher entscheidend, dass ein realistisches Mobilitätsmodell verwendet wird.

In realen Szenarien wird das Verhalten der Menschen stark von der räumlichen Umgebung beeinflusst. So müssen sie beispielsweise Hindernissen ausweichen und Straßen folgen. Die Benutzer bewegen sich, um gewisse Aktivitäten, wie Einkäufe, Besichtigung von Sehenswürdigkeiten u.Ä., ausführen zu können und verfügen über eine bestimmte Dynamik. Fußgänger haben typischerweise eine relativ niedrige Bewegungsgeschwindigkeit und machen viele kurze Pausen, dagegen fahren die Fahrzeuge mit höherer Geschwindigkeit und ordnen sich dem gesamten Verkehrsverlauf unter. Diese Faktoren bestimmen die Benutzermobilität in Stadtszenarien wesentlich und müssen entsprechend in einer Netzsimulation berücksichtigt werden.

Die zurzeit benutzten Verfahren können nach ihren Eigenschaften in verschiedene Gruppen unterteilt werden. Analytische Modelle ermöglichen es Mobilitätsparameter in geschlossener Form (mit Formeln) zu beschreiben. Wegen der Komplexität dieser Formeln werden normalerweise lediglich einfache Annahmen zum Benutzerverhalten getroffen. Die Modelle betrachten nur zufällige Geschwindigkeits- und Richtungsänderungen, was rigide und abrupte Bewegungen verursacht [TJ99]. Die meisten analytischen Modelle betrachten nur die Mobilität zwischen Basisstationen. Dafür wird das Simulationsgebiet gleichmäßig in Zellen unterteilt und die Modelle beschreiben die Mobilität zwischen diesen Zellen. Im „Random Walk“ Modell [AHL96], [ZD97] wählt der Benutzer zufällig eine der Zellen in der Nachbarschaft als den nächsten Zielpunkt aus. Da sich solche Bewegungen mit Markow-Ketten [HA95], [JJ00] beschreiben lassen, wird dieses Modell häufig für die mathematischen Analysen verwendet. Dennoch bildet die zufällige Bewegung das reale Bewegungsverhalten von Menschen schlecht ab, deshalb wird das „Random Walk“ Modell in weiteren Studien erweitert [LB98], [LM95]. Trotzdem sind solche Modelle nicht für alle Netzstudien geeignet, da sie nicht die vollständigen Bewegungspfade sondern nur die Zellen betrachten. Die räumliche Umgebung wird von den Modellen ebenfalls nicht berücksichtigt.

Einige Mobilitätsmodelle betrachten komplette Bewegungstrajektorien. Dennoch nehmen die meisten Ansätze, wie das Brownsche Modell [DCYS98] und das „Random Waypoint“ Modell [BMJ+98], [JM96], wieder die zufällige Mobilität der

Benutzer im Zielgebiet an. Diese Modelle sind relativ einfach und erfordern wenige Eingabeparameter. Dadurch werden sie häufig für Netzsimulationen verwendet. Dennoch berücksichtigen diese Modelle die Umgebung nicht. Im graphbasierten Modell [THB+02] wird das Gebiet mit der Hilfe eines Graphen dargestellt, in dem die Knoten sowohl Ziel- als auch Zwischenpunkte sind; die Kanten kennzeichnen Straßenabschnitte. Da die Benutzer sich nur auf den Kanten bewegen, wird dabei die räumliche Umgebung berücksichtigt. Dennoch werden die Zielpunkte rein zufällig ausgewählt, was das Menschenverhalten zu sehr vereinfacht. Ebenfalls wird eine Benutzerdynamik mit unveränderlicher Geschwindigkeit angenommen, was für manche Situationen, wie z.B. im Fahrzeugverkehr, schlecht geeignet ist.

Noch eine Gruppe der Mobilitätsmodelle wird durch reale Bewegungstrajektorien gebildet, z.B. [JHP+03] und [SK99]. Diese werden mit der Hilfe von verschiedenen Technologien, wie Positionierungssysteme oder Benutzerbefragungen, ermittelt. Offensichtlich kann die Mobilität durch die Verwendung solcher Trajektorien realistisch nachgebildet werden. Dennoch enthalten die Daten nur die Bewegungen einer begrenzten Anzahl von Benutzern und können dadurch nicht in einem größeren Szenario verwendet werden. Die Bewegungsparameter, wie Geschwindigkeit oder Routenverlauf, sind ebenfalls schwer zu variieren. Außerdem entstehen durch das Sammeln von solchen Spuren hohe Kosten und weitere Probleme, beispielsweise der Schutz persönlicher Daten.

Das in dieser Dissertation vorgeschlagene Verfahren („User-Oriented Mobility Meta-Model“ [SHB+03], [KBST04], [SMR05]) berücksichtigt die drei Hauptfaktoren, die die Mobilität der Benutzer in realen Szenarien beeinflussen: die räumliche Umgebung, das Benutzerverhalten und die Bewegungsdynamik. Daher integriert es die folgenden Teilmodelle:

- Modell der Umgebung („Spatial Model“)
- Modell des Benutzerverhaltens („User Trip Model“)
- Modell der Benutzerdynamik („Movement Dynamics Model“)

Das Umgebungsmodell enthält eine digitale Karte des Simulationsgebietes mit Informationen zu Straßenverläufen und ausgezeichneten Punkten („Points of Interest“). Das Modell wird automatisch nach der Beschreibung des Generischen

Geographischen Modells aus Kapitel 2 generiert. Demnach werden die Objekte nach GDF mit ihren Typinformationen, ihrer Geometrie, ihrer Attributen und Relationen definiert. Zusätzlich wird ein Graph des Straßennetzes angeboten, in dem die Knoten die wichtigen Punkte und Kreuzungen repräsentieren und die Kanten die Straßenabschnitte. Diese Daten werden aus einem geographischen Informationssystem eingelesen. Die innerhalb dieser Dissertation entstandene Implementierung stellt die dafür benötigten Parser zur Verfügung.

Das Modell des Benutzerverhaltens beschreibt Benutzerentscheidungen, wie die Ziel- und Pfadauswahl, und basiert auf Verfahren aus der Verkehrsplanung und „Discrete Choice“ Methoden. Information über die möglichen Ziele und Pfade wird aus dem Umgebungsmodell entnommen. Die Zielwahl wird nach dem „Activity-based Travel Demand Modeling Approach“ [Kit96], [Pas96] modelliert. Den Kern dieses Ansatzes bilden Aktivitäten, die die Benutzer während ihres Tagesablaufs ausführen. Dadurch entsteht ein realitätsnahes Modell des menschlichen Verhaltens. So genannte Ketten beschreiben die Abfolge möglicher Aktivitäten, wie Einkaufen oder die Besichtigung von Sehenswürdigkeiten. Diese Ketten können sowohl für einen als auch für mehrere Benutzer definiert werden. Die einzelnen Aktivitäten können an mehren Punkten ausgeführt werden; verschiedene Übergänge zwischen den Aktivitäten sind ebenso möglich. Jeweils eine Aktivität und ein Zielpunkt werden zufällig nach definierten Wahrscheinlichkeiten ausgewählt. Nachdem der Benutzer an seinem Zielpunkt angekommen ist, bleibt er für die Zeit des Aktivitätsausführens stehen. Danach wird die nächste Aktivität mit einem neuen Zielpunkt ausgewählt, und der Benutzer setzt seine Bewegung fort. Innerhalb des Modells des Benutzerverhaltens wird das Schalten zwischen den Aktivitäten mit Hilfe von nichtdeterministischen endlichen Automaten nachgebildet.

In realen Szenarien sind auch mehrere Pfade zwischen der bisherigen Position und dem nächsten Zielpunkt möglich. Um die Pfadauswahl zu modellieren, bietet das vorgeschlagene Mobilitätsmodell verschiedene Verfahren an, die für diverse Situationen geeignet sind. Falls der Modellentwickler keine Information zum Benutzerverhalten bzw. zu den Faktoren, die die Pfadauswahl beeinflussen, hat, wird der kürzeste Pfad ausgewählt, z.B. mit der Hilfe des Algorithmus nach Dijkstra

[Dij59]. Alternativ können auch „Discrete Choice“ Methoden verwendet werden [BAB99], [Tra03]. Diese liefern realistischere Ergebnisse, allerdings benötigen sie zusätzliche Information zum Benutzerverhalten.

Schließlich beschreibt das Modell der Benutzerdynamik die Bewegung zur Zielposition entlang des ausgewählten Pfades. Die dafür geeigneten Modelle, wie das Modell von Weidmann für Fußgänger [Wei93] und das „Intelligent Driver“ Modell für Fahrzeuge [THH00], stammen aus dem Straßenbau und der Verkehrsplanung und sind in der eigentlichen Dissertation ausführlich beschrieben.

Das „User-Oriented Mobility Meta-Model“ wird in einem Rahmenwerk für die Mobilitätsmodellierung implementiert. Dieses unterstützt verschiedene Netzsimulationswerkzeuge, wie ns-2 [BEF+00] und GlomoSim [ZBG98]. Um die Nutzung zu vereinfachen, bietet das Rahmenwerk verschiedene vordefinierte Szenarien an, beispielweise ein Einkaufsszenario oder die Besichtigung von Sehenswürdigkeiten. Zusätzlich können Mobilitätsparameter, sowie Aktivitäten und Übergänge, aus Bewegungstrajektorien automatisch abgeleitet werden. Der dafür benötige Algorithmus wird ebenso detailliert in der Dissertation beschrieben.

4. Modellierung der Bitübertragungsschicht

In realer Welt wird die Wellenausbreitung stark von Hindernissen gestört, daher muss die räumliche Umgebung auch bei der Simulation von Funkkanälen berücksichtigt werden. Die Netzsimulationswerkzeuge modellieren eine Übertragung des Netzrahmens in drei Schritten:

- Berechnung der Empfangsleistung für jeden Empfänger
- Berechnung der Interferenzen mit anderen Signalen
- Entscheidung über den Empfang des Rahmens

Wenn ein Netzknoten einen Rahmen sendet, berechnet der Simulator anhand des Wellenausbreitungsmodells die angekommene Signalleistung für jeden potenziellen Empfänger. Die Signale, deren Leistung unterhalb des Schwellwerts der Trägerprüfung („Carrier Sense Threshold“, CSThresh) liegt, sind für den Empfang zu schwach und werden deshalb ignoriert. Danach wird die Interferenz mit anderen

Signalen berechnet. Einige Simulatoren, wie ns-2 [BEF+00], vergleichen die Leistungen der Signale. Eine Kollision tritt auf, wenn das Verhältnis zwischen zwei Signalen unterhalb des Kollisionsschwellwerts („Collision Threshold“, CPThresh) liegt; was eine gute Abstraktion für manche Fälle darstellt. Andere Simulatoren, wie GlomoSim [ZBG98], betrachten die kumulative Rahmeninterferenz. Sie berechnen das Signal-zu-Interferenz-plus-Rausch-Verhältnis („Signal-to-Interference-and-Noise Ratio“, SINR), das während des nächsten Schrittes für den Empfang des Rahmens berücksichtigt wird. Der finale Schritt ist ebenso von den Simulatoren abhängig. Manche davon, wie ns-2, vergleichen die empfangene Signalleistung mit dem Empfangsschwellwert der Netzkarte („Receive Threshold“, RXThresh). Falls die Signalstärke über RXThresh liegt, wird der Rahmen ohne Transmissionsfehler empfangen. Andere Simulatoren, wie GlomoSim, führen einen ähnlichen Vergleich durch, aber mit dem SINR. Optional wird von einigen Simulatoren auch der Bitfehlerrate-basierte Rahmenempfang unterstützt. Anstatt die Werte mit dem Schwellwert zu vergleichen, wird die entsprechende Bitfehlerrate verwendet, die z.B. anhand von Messungen ermittelt wurde. Die Werkzeuge entscheiden dann probabilistisch über den Empfang des Rahmens; dadurch wird der Prozess realistischer nachgebildet.

Die Netzsimulationswerkzeuge bieten für die Berechnung der Empfangsleistung nur die einfachen empirischen Modelle an, wie z.B. die Kombination aus „Free Space“ [Fri46] und „Two-Ray Ground“ [RMK97] Modellen, oder das „Log-Distance Path-Loss” Modell [RMK97]. Diese Modelle setzen eine direkte Sichtverbindung („Line of Sight“) und somit eine ungehinderte Wellenausbreitung voraus. Der Abstand zwischen den Knoten ist der einzige dynamische Parameter solcher Modelle; die räumliche Umgebung wird dabei nicht berücksichtigt. Der niedrige Berechnungsaufwand ist der Hauptvorteil solcher Modelle. Dennoch sind sie schlecht anwendbar in realen Szenarien, in denen Hindernisse, wie Gebäude, die Mobilkommunikation beeinflussen. Optische Modelle [SDR92] basieren auf der Strahlenverfolgung und berücksichtigen die Umgebungsinformation. Sie liefern realistischere Ergebnisse, benötigen jedoch einen höheren Berechnungsaufwand.

In dieser Dissertation wird das „Intelligent Ray Tracing“ Modell [WHL99] verwendet. Das ist eine erweiterte Version des klassischen Strahlenverfolgungsverfahrens. Um die Berechnungen zu beschleunigen, werden die Umgebungsdaten vorverarbeitet mit dem Ziel, die Sichtverhältnisse zwischen den Wänden zu berechnen. Dadurch wird die Ausführung um ein Tausendfaches beschleunigt. Die Genauigkeit des Modells wurde durch Messungen in europäischen Städten nachgewiesen. Für Helsinki (Finnland) ist die mittlere Abweichung nahezu 0 dB und die Standardabweichung etwa 8 dB [RWH02]. Für München und Nancy (Frankreich) ist die mittlere Abweichung 0 dB und die Standardabweichung weniger als 7 dB [WGL97], [WHL99]. Für die Stadt Stuttgart, die für die Evaluation des Verfahrens in Kapitel 5 benutzt wird, ist die mittlere Abweichung 0,3 dB und die Standardabweichung 5,8 dB [HWL03].

Für die Integration des „Intelligent Ray Tracing“ Modells in eine Netzsimulation wird das kommerzielle Produkt „WinProp“ von AWE Communications verwendet, welches das Verfahren implementiert. Das Programm benötigt 2,5-dimensionale Umgebungsdaten in einem eigenen Format. Diese können aus einer digitalen Karte extrahiert werden. In dieser Dissertation werden die Daten der Stuttgarter Innenstadt (die Gebietsfläche umfasst 2,4 km × 1,9 km) verwendet. Für eine angegebene Senderposition und andere statische Parameter, wie die Senderhöhe, Sendeleistung, Wellenlänge u.Ä., berechnet WinProp eine Tabelle der Empfangsleistungen für ein Raster, das mögliche Empfängerpositionen darstellt. In unseren Simulationen wird ein 5 m × 5 m Raster verwendet. Ein noch feineres Raster können wir nicht verwenden, da kleinere Größen deutlich längere Berechnungen und höheren Speicherplatzbedarf benötigen.

Da WinProp ca. 30 Sekunden für die Berechnungen der Empfangsleistungen für eine Senderposition braucht, ist es wenig sinnvoll, das Tool jedes Mal auszuführen, wenn der Simulator einen Empfangswert braucht; pro Simulation werden in der Regel mehrere Millionen Aufrufe gemacht. Die Kommunikationsparameter bleiben während einer Simulation konstant, daher liefert WinProp stets die gleichen Ergebnisse für dieselbe Senderposition. Aus diesem Grund wurden die Empfangswerte für alle möglichen Sender- und Empfängerpositionen im 5 m × 5 m

Raster vorberechnet und in einer Datenbank gespeichert. Für die Stuttgarter Innenstadt ergeben sich etwa 32 Milliarden Positionspaare. Wegen des Vorberechnens ist die gesamte Simulationsdauer mit der von einfachen empirischen Modellen vergleichbar.

Um noch realistischere Simulationen durchführen zu können, wird das beschriebene Verfahren mit den Modellen der Signalabschwächung und der Bitfehlerhäufigkeit benutzt. Diese wurden von anderen Forschern entwickelt und zur Vollständigkeit auch in diesem Kapitel beschrieben.

5. Evaluierung

Bei der Evaluierung wird der Einfluss der dargestellten Mobilitäts- und Wellenausbreitungsmodellierungsverfahren auf die Simulationsergebnisse gezeigt. Dafür werden die Resultate, die jeweils mit den einfacheren und realistischeren Modellen entstanden sind, systematisch verglichen und die Auswirkungen auf die Netzkonnektivität, die Leistung von Routing Protokollen und auf mobile Anwendungen werden analysiert.

Ein MANET wird anhand eines Innenstadtszenarios von Stuttgart simuliert. Die Fußgänger bewegen sich in einem 1,5 km × 1,5 km großen Gebiet zwischen verschiedenen wichtigen Punkten. Da wir keine vollständige Information über das Benutzerverhalten in diesem Szenario haben, wurden die Benutzertrips zufällig nachgebildet; die Pausezeiten nach dem Erreichen des Zielpunktes (oder die Dauer der einzigen Aktivitätsausführung) wurden auf einem Bereich zwischen 10 und 15 Minuten festgelegt. Die Benutzergeschwindigkeiten liegen zwischen 0,56 und 1,74 m/s [HM95]. Bei dem „Random Waypoint" Modell werden die Geschwindigkeiten aus diesen Rahmen gleichverteilt ausgewählt. Mit dem „User-Oriented" Modell wird ein realistischeres Verfahren verwendet, in dem die Geschwindigkeiten der Gaußschen Verteilung mit dem Mittelwert 1,34 m/s und Standardabweichung 0,26 m/s folgen [Wei93].

Die Benutzer tragen mobile Geräte, die nach IEEE 802.11 im Ad-hoc-Modus vernetzt werden können. Die Hardwareparameter werden den Datenblättern des Herstellers entnommen [Pro03].

Für den Vergleich von Netztopologien werden die MANET-Konnektivitätsgraphen betrachtet, in denen die Kanten mögliche Verbindungen zwischen den Netzknoten darstellen, d.h. ein Knoten kann die Transmissionen von einem anderen mit einer Signalleistung über RXThresh empfangen. Die Metriken, wie Anzahl der Kanten, Hamming-Abstände zwischen Adjazenzmatrizen, Anzahl der Partitionen u.Ä., werden dabei berücksichtigt. Die Ergebnisse zeigen signifikante Unterschiede zwischen den einfachen und realistischeren Modellen.

Da die Benutzerverteilung mit dem „Random Waypoint" Mobilitätsmodell nah an der Gleichverteilung liegt [BRS03], sind in solchen Simulationen nur wenige Netzverbindungen möglich. Dafür gibt es aber auch weniger Partitionen. Dagegen hängt die Benutzerverteilung mit einem realistischeren Mobilitätsmodell stark von der Umgebung ab (durch das Straßennetz und die Lage von wichtigen Punkten). Außerdem zeigen die Simulationen, dass die Netztopologien ebenso von solchen Hindernissen, wie Gebäuden, stark beeinflusst werden. So unterscheiden sich z.B. die Graphen von „Two-Ray Ground" und „Intelligent Ray Tracing" Modellen in 40-60% der Kanten. Die Analyse zeigt, dass diese Kanten durch Gebäude verhindert werden und deshalb im realistischeren Wellenausbreitungsmodell fehlen. Das andere empirische Modell („Log-Distance Path-Loss Model") verkürzt die Kommunikationsreichweite durch den Pfadverlustexponent, um die Wellenausbreitung in Städten besser nachbilden zu können, unterschätzt aber dann die Ausbreitung in freien Bereichen. Eine interessante Eigenschaft des realistischen Modells sind die vielen kleinen Netzpartitionen, die einzelne Benutzer oder kleine Benutzergruppen umfassen. Solche Partitionen treten wegen den Hindernissen auf, die die einfachen empirischen Modelle nicht berücksichtigen.

Um den Einfluss auf die Leistung von Routing Protokollen zu analysieren, werden Metriken, wie der Anteil von empfangenen Paketen oder der Anteil an Routing Paketen, die Paketverzögerung und die Anzahl von Hops in Netzpfaden, verwendet. Die Simulationsergebnisse zeigen dabei, dass die beobachteten Änderungen in

Netztopologien auch die Leistung von Protokollen beeinflussen. Die Hindernisse erschweren die Suche und die Verwaltung von Netzpfaden. Dadurch werden unter dem „Intelligent Ray Tracing“ Modell mehr Routing Pakete gesendet und höhere Paketverzögerungen verursacht. Das am meistens verwendete „Two-Ray Ground“ Modell zeigt auf relativ niedrigen Transmissionsgeschwindigkeiten (1 und 2 Mbps) dicht vernetzte Topologiegraphen, in denen häufig Paketkollisionen auftreten. Deshalb funktioniert das Routingverfahren in solchen Simulationen sogar schlechter als mit dem realistischeren Modell.

Schließlich wird der Einfluss von realistischeren Modellen auf die Simulation von mobilen Anwendungen analysiert. Eine unveränderte Implementierung des Usenet Systems [Hor83] für Ad-hoc-Netze („Usenet-on-the-Fly“) [BBH02] wird dafür in virtuellen „User-Mode Linux“ Maschinen [Dik00] gestartet und der Netzverkehr wird durch den Simulator geführt. Dabei zeigten die realistischeren Simulationen eine 2-4 fach langsamere Nachrichtenverbreitung, dafür aber einen niedrigeren Energiekonsum.

Zusammengefasst, zeigen die Simulationsergebnisse aus diesem Kapitel, dass der Einfluss von Hindernissen der räumlichen Umgebung und die veränderte Verteilung der mobilen Knoten durch einfache Modelle nicht nachgebaut werden kann. Realistischere Modelle werden benötigt, um realistischere Simulationsergebnisse zu bekommen. Diese Dissertation beschreibt solche Modelle, ihre Integration in einen Netzsimulator und ihre praktische Verwendung für die Simulationen von mobilen Netzen.

6. Zusammenfassung und Ausblick

Netzsimulationswerkzeuge bieten zurzeit nur einfache Mobilitäts- und Wellenausbreitungsmodelle an, die die räumliche Umgebung nicht berücksichtigen und daher realistische Szenarien schlecht nachbilden können. In dieser Dissertation wurden realistischere Modelle beschrieben und in die Netzsimulation integriert. Diese Modelle basieren auf Methoden diverser Forschungsgebiete, wie der Physik, Verkehrsplanung, Verkehrsdynamik und Nachrichtentechnik.

Zunächst wurden standardisierte Beschreibungsformen für geographische Daten analysiert, die die Umgebungsinformation für die beschriebenen Modelle anbieten. Als nächstes wurden die zurzeit in Simulationswerkzeugen verwendeten Mobilitätsmodelle diskutiert. Ein eigenes Verfahren („User-Oriented Mobility Meta-Model“) wurde beschrieben, das die Hauptfaktoren der Benutzermobilität, wie Straßenumgebung, Benutzerverhalten und Benutzerdynamik, berücksichtigt. Die Methoden zur Modellierung dieser Faktoren wurden zusammen mit deren Integration in das Mobilitätsmodell dargestellt. Weitere Untersuchungen konzentrierten sich auf die Modellierung des physikalischen Kanals. Die von Simulatoren unterstützten Wellenausbreitungsmodelle wurden beschrieben. Diese Modelle setzen eine direkte Sichtverbindung zwischen Netzgeräten und somit eine ungehinderte Wellenausbreitung voraus, was in realen Szenarien selten möglich ist. Ein realistischeres Verfahren („Intelligent Ray Tracing“) basiert auf der Strahlenverfolgung und berücksichtigt dabei die Umgebungsinformation. Es wurde in einer Kombination mit den Modellen der Signalabschwächung und der Bitfehlerhäufigkeit eingesetzt, um noch realistischere Simulationen durchführen zu können. Hierdurch wird die resultierende Netztopologie wesentlich beeinflusst. Es wurde gezeigt, dass die Verwendung der neuentwickelten, realistischeren Modelle in der Simulation zu unterschiedlichen Ergebnissen bei der Leistungsbewertung von Vermittlungsprotokollen und mobilen Anwendungen führt. Der erhöhte Benutzungs- und Rechenaufwand, der aus dem Einsatz der entwickelten Modelle resultiert, wird somit durch einen erhöhten Grad an Realitätstreue gerechtfertigt. Diese Arbeit zeigte, dass es möglich ist, solche komplexere Modelle in Netzsimulatoren zu integrieren. Die Implementierungen, die in dieser Dissertation entstanden sind, stehen auf der Homepage des Autors frei zur Verfügung und wurden bereits in verschiedenen Forschungsprojekten der Universität Stuttgart und anderer Universitäten eingesetzt.

Ein interessantes Thema für die zukünftige Forschung ist die weitere Analyse und die Integration von realen Bewegungsdaten und Profilen in eine Netzsimulation. Dies wird bereits von dem beschriebenen Rahmenwerk durch den automatischen Parameterableitungsmodul unterstützt. Da wir zurzeit keine realen Daten haben, müssten wir in unseren Simulationen zufälliges Benutzerverhalten zwischen

bestimmten Punkten der Umgebung annehmen. Das Verwenden von realen Daten würde noch realistischere Mobilität erzeugen. Durch das Verbreiten von solchen Simulationsszenarien würden wir in der Lage sein einige gemeinsame Szenarien zu erschaffen, die für den Leistungsvergleich von Algorithmen, Protokollen und mobilen Anwendungen unter realistischeren Bedienungen benutzt werden könnten.

Ein anderes wichtiges Thema ist die Optimierung des Speicherns der vorberechneten Wellenausbreitungsdaten für das „Intelligent Ray Tracing“ Modell (Empfangswerten). Momentan speichern wir die Daten für alle möglichen Sender- und Empfängerpositionen im 5 m × 5 m Raster. Dies entspricht ca. 120 GB Gesamtdatengröße für das in dieser Dissertation verwendeten Stuttgart-Innenstadt Szenario. Dennoch liegen einige Positionen weit von einander entfernt, wodurch die Signale zwischen solchen Positionen sehr schwach sind und haben fast keinen Einfluss auf Simulationsergebnisse. Durch das Entfernen von solchen Einträgen kann die Datenmenge erheblich reduziert werden und kleinere Rastergröße ermöglichen. Diese kleinen Raster werden für die Simulationen von kleineren mobilen Geräten benötigt.

Contents

1 Introduction

1.1 Motivation

Recent advances in portable computing platforms and wireless communication have led to a growing popularity of wireless networks. Ease of deployment, installation flexibility, support of roaming users, and “anytime, anywhere” information access are the main advantages of such networks.

Two types of wireless networks exist: infrastructure networks and ad-hoc networks. In an infrastructure network, clients are connected via an access point (base station). The access point can also act as a bridge between the wireless network and a wired local network or the Internet. In ad-hoc networks, the devices communicate directly with each other without any preinstalled infrastructure. Since communication is only possible between the devices that are in each other’s transmission range, some devices have to act as intermediate packet forwarders for other devices. Routing decisions are made dynamically based on the current network connectivity.

A mobile ad-hoc network (MANET) is a wireless ad-hoc network that is formed by mobile devices like user carried handhelds, notebooks, or devices mounted on a moving vehicle (Figure 1.1). Due to the mobility of devices, network topology can rapidly change. Many usage scenarios for such networks exist both for military and civilian applications. Examples are information exchange among mobile users in a city center [JHP+03], dissemination of news or announcements [BBH02], a cooperative driving system [RMM+00], or rescue operations [JLH+99].

The network simulation ([BEF+00], [ZBG98], [Ril03]) is a widely used method for evaluating and comparing the performance of mobile protocols and applications in their target scenarios. The tools offer a simulation model abstracting a mobile network. This model commonly includes the following main parts:

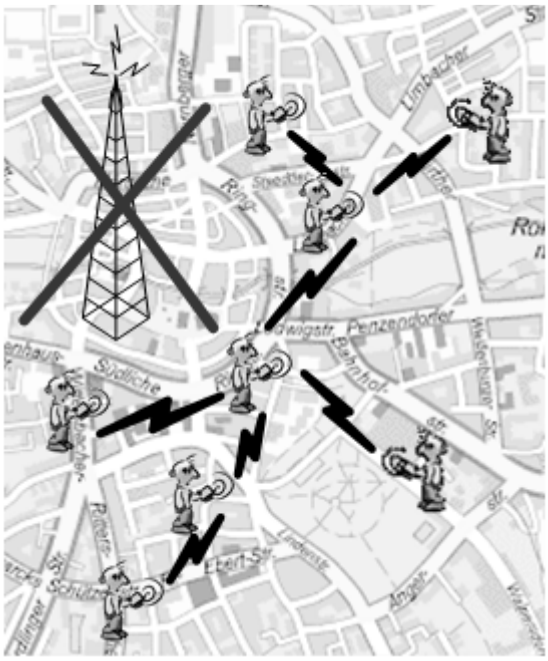


Figure 1.1: MANET example

- Mobile application (frequently abstracted by the network traffic sources and sinks)
- Wireless network protocol stack
- Properties of the communication channel (e.g., signal loss)
- Mobility of network users

Since most of researchers focus on communication protocols and applications, these parts of the simulation software are the most advanced ones. However, the parts responsible for the modeling of user mobility and wireless channel are often oversimplified with straightforward approaches. To model user mobility, most tools only offer the mobility models that produce randomly user movements within a rectangular area (e.g., [BMJ+98], [CJB+01], [DCYS98], [KV98]). The resulting movements are similar to the Brownian motion of molecular particles [Nel67]. For modeling of the wireless channel, usually the free space path loss model is used [RMK97]. It assumes a line of sight between communicating nodes, and thus, a simple dependency of the signal loss to the distance from the transmitter.

These models poorly reflect real scenarios, in which the characteristics of the spatial environment have a significant impact on mobile communication and pose

constraints on the user movements. For example, car traffic only moves along roads and pedestrians do not go through walls. Obstacles (e.g., buildings) obstruct radio wave propagation, which is not reflected by the free space model. Moreover, people do not travel completely randomly in an area; their movement underlies certain regularity. The movements of some users also depend on each other. For instance, when a car in front brakes, the succeeding vehicles also slow down. The mobility and the physical channel models that are currently used in simulation tools do not take these factors into account.

In this thesis, more realistic models are developed that reflect the spatial environment and the user travel behavior. These models are based on solutions from related research areas, like physics, transportation planning, traffic modeling, and electrical engineering. They consider digital maps of the simulation area, which are taken from a geographic information system (GIS). The models are integrated into a network simulation, which allows for more realistic studies of mobile networks in target scenarios. As the evaluation shows, they change simulation results significantly.

This work focuses on mobile ad-hoc networks that use the IEEE 802.11 (Wi-Fi) standard [IEE99] for communication. However, the described models can also be applied to infrastructure networks and cellular networks. Due to the evaluation scenarios of connected research projects, this work was originally performed for outdoor environments. Therefore, this thesis mainly contains the outdoor examples and uses the outdoor scenario in the evaluation section. However, this work can be also applied in indoor environments after a simple terminology change, i.e. by considering rooms and corridors instead of buildings and roads.

1.2 Contributions

Here are the main contributions of this work.

We analyze existing geospatial data standards to create a generic geographic data model. We integrate this model into a network simulation, thus providing different

simulator components, like mobility or radio propagation models, with access to geospatial data.

We develop the user-oriented mobility model, which reflects the more realistic movement. The model considers such factors as: constraints of the spatial environment, user trips, and the user movement behavior. The user-oriented mobility model relies on approaches from related research fields, like physics, transportation planning, and traffic modeling. We summarize those approaches and solve different problems that arise upon putting them together into a single model. We integrate the user-oriented mobility model into a network simulation environment.

Another important contribution of this work is the integration of the intelligent ray tracing model into a network simulation. The model is based on geometric optics and considers a map of the spatial environment. Therefore, the intelligent ray tracing model delivers more realistic results than simple approaches. The accuracy of the model is proven by measurements in European cities. However, the model requires much longer computation time. We present an approach for integrating this model, so the simulation time stays comparable to the simple models.

We compare the simulation results that we get with simple and the more realistic models. We discuss the changes and analyze the reasons for those changes. The implementations of the models (parsers for geographic data in different formats, mobility modeling framework, and radio propagation model for the network simulator ns-2) are freely available from the author's homepage.

To analyze the performance of a mobile application, we also introduce a small improvement to ns-2. This improvement allows using virtual machines in a network simulation, thus increasing scalability of emulation.

1.3 Overview

This thesis is structured as follows: in Chapter 2, we analyze geospatial data standards. They are used for describing spatial environments in digital form. More realistic approaches to user mobility and radio propagation modeling that are developed in this work rely on these geospatial data.

Chapter 3 deals with the modeling of mobility. We discuss related work in the field to show the shortcomings of existing models. A new approach is presented, which is based on research in related disciplines. We describe the methods for modeling various aspects of user movement along with their integration into the proposed mobility model. The setting of model parameters for concrete scenarios and implementation details are also explained.

Chapter 4 focuses on the physical layer modeling. After basic definitions, we discuss related work in the area, which includes the common physical layer abstractions used in network simulation tools and radio propagation models. Then a more realistic intelligent ray tracing model is presented, which is based on geometric optic and relies on a map of the simulation area. For obtaining even more realistic results, we use the described intelligent ray tracing model in a combination with small-scale signal fading and bit-error modeling approaches. They are also introduced in this chapter.

Chapter 5 compares the simulation results obtained with simple and the more realistic models. A mobile ad-hoc network is simulated in the city center of Stuttgart. We compare the impact of the models on network connectivity, routing protocols, and mobile application performance. The chapter describes the simulation setups and discusses the obtained results.

Chapter 6 concludes this thesis with a summary of the main contributions of this work and the achieved results. An outlook to future work finalizes this thesis.

2 Geographic Data Model

Obviously, a spatial environment has impact on the performance of mobile networks. The points of interest and the movement area constraints impact user mobility in the area and change network topologies. The points of interest serve as anchor points of user movements. The movement area constraints, such as corridors in indoor and roads in outdoor scenarios, predefine possible movement paths between those anchor points. Besides, materials and internal structure of buildings strongly influence radio wave propagation. So, transmitted signals weaken faster in the presence of obstacles. This makes successful signal reception and demodulation more difficult.

Therefore, in order to obtain more realistic results, spatial environments must be reflected in network simulations. To achieve it, the corresponding tools must rely on geographic data models, which are used for representing the spatial environments in digital form.

This chapter is a basic block of this work. It analyzes existing standards of geospatial data in order to create a generic, standard-independent geographic model. This model provides the information about the movement area and buildings to the network simulation. It is used as input to the developed mobility and radio propagation modeling approaches.

This chapter is structured as follows: Section 2.1 gives an overview of common geographic data standards. Section 2.2 introduces a generic model, which is used for integrating diverse geospatial data sources into network simulation. Section 2.3 describes the road network representation, which is required for the modeling of user mobility in the area. The specification of building data for the modeling of wireless communication is described in Section 2.4. Section 2.5 gives model implementation details. Section 2.6 concludes this chapter.

2.1 Overview of Standards for Geographic Data

Many applications, like location-based services and navigation, rely on digital maps of spatial environments. This motivates the development of geographic data formats[1] (Figure 2.1).

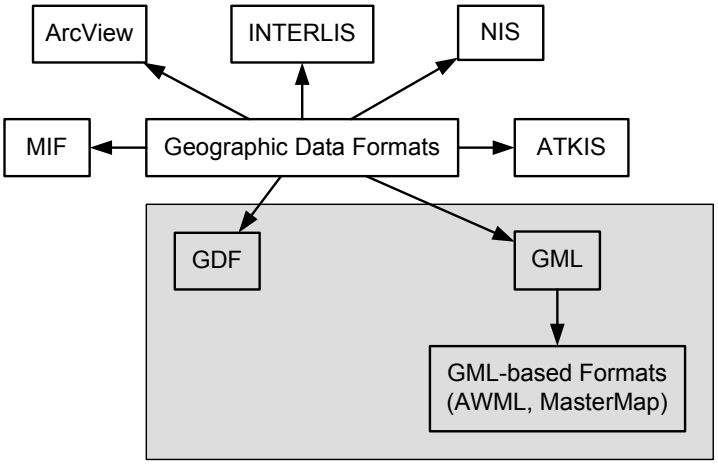


Figure 2.1: Geographic data formats

Among this multitude of standards, two are the mostly used, and hence, are considered in this thesis: Geography Markup Language (GML) [ISO04] and Geographic Data Files (GDF) [CEN95]. GML is an extensible text format. It defines only basic rules for describing spatial objects and possible extension mechanisms. A detailed object specification is then done in custom applications and derived geospatial standards. In opposite to GML, GDF (as well as the most of other standards) is a final binary format. It defines the specification of all possible spatial object types, their attributes, and relationships with other objects.

Next sections describe these two standards in more detail.

---

[1] In [Mic04], the word "format" is defined as "the structure or organization of digital data for storing, printing, or displaying." "Standard" is "something established by authority, custom, or general consent as a model" [MW02]. Since the examined formats of geographic data are standardized and are widely accepted, these terms are used interchangeable here.

2.1.1 Geography Markup Language

GML originates from the Extensible Markup Language (XML) [W3C04], which is a general purpose markup language for describing tree-structured information. GML provides the XML schemes that are specially designed for encoding geospatial data.

A spatial environment is represented as a collection of abstract objects (geographic *features*). For outdoor environments, these can be, for example, roads, street crossings, buildings, and vegetation areas. Among the indoor objects, are, for example, rooms and corridors.

Object description (Figure 2.2) includes the type information, geometry, and additional attributes. The attributes are specified using {name, value} pairs. GML defines only the basic attributes, such as an object's identifier, name, and textual description. A detailed specification of object types and attributes is done by custom application schemes.

To encode object geometry, GML provides the necessary geometric primitives, like point, line, curve, polygon, or surface. The coordinates are defined within a spatial reference system (SRS). GML supports a variety of SRS types, e.g.,

```
<Road gml:id="c371">
      <gml:name>Long Road</gml:name>
      <gml:extentOf>
            <gml:LineString>
                  <gml:posList count="3" srsDimension="2"
                        srsName="urn:epsg:crs:62836405">
                        -30.440357 136.889312
                        -30.5 136.7 -30.6 136.5
                  </gml:posList>
            </gml:LineString>
      </gml:extentOf>
      <numberLanes>2</numberLanes>
      <laneWidth>3.5</laneWidth>
      <maximumSpeed>120.0</maximumSpeed>
</Road>
```

Figure 2.2: GML example

Cartesian, ellipsoidal, spherical, polar, etc. It also offers schemes for defining custom coordinate systems.

GML supports *collections* of features, i.e. containment of smaller objects within larger ones. Other associations between features can also be specified, e.g., to indicate that a building is located near a particular road or that a maneuver is prohibited between given road elements. GML allows for the construction of such relationships.

GML serves as a basis for many geospatial data encodings, such as the MasterMap of national mapping agency of Great Britain [OS01] or the Augmented World Modeling Language of Nexus project (AWML) [NM01]. Since GML originates from XML, it is possible to use XSLT [W3C99] for transforming geospatial data between different GML-based representations.

2.1.2 Geographic Data Files

GDF was originally designed for car navigation systems. However, it is now also used in other outdoor transport and traffic applications. Although GDF focuses on encoding road network data, it also allows for the specification of *services*. The services represent the different places (e.g., shops, restaurants, museums) that are of interest for mobile users.

GDF uses own format to store data (Figure 2.3). It is a binary format, which is encoded with text (ASCII) symbols for easier readability. GDF standardizes possible types of spatial objects and properties, thus maximizing interoperability across applications.

The standard splits the description of spatial area in three parts (*catalogues*): the Feature Catalogue, the Attribute Catalogue, and the Relationship Catalogue.

The Feature Catalogue stores real-world objects. They are distinguished by their type (“Road,” “Intersection,” “Public Transport Stop,” “Traffic Sign,” “Restaurant,” “Shopping Center,” etc.) and are described in three presentation levels (Figure 2.4).

```
01NAVTECH             GDF       3.0 010817                          1         1
00ISO-8859-1       10000000006   1NAVIGATION TECHNOLOGIES]  SV03.31,FV03.60,DV21
00.4.3,CT06:41,DVNGERMANY_G6_22.0.0,LNG6G014                                    0
23  58521873            1    916905   4876882         0                        0
23  58521877            2    917040   4876886         0    916976   4876971    1
00         0                                                                    0
23  58521887            1    916942   4876982         0                        0
24  52908271           58523802  58523805                    2                 0
24  52908274  58523586 58523805  58523807                    2                 0
24  52908277           58523516  58523518                    2                 0
51   1875046     4120    1  58523776    0                                       0
51   1875071     4120    1  58523701    0                                       0
52     21221     41100    1  52852290 1    3     49909     50490     50491     1
00   1850851   1854136                                                          0
52     21222     41100    1  52852291 1    3     49909     50492     50493     1
00   1847396   1850851                                                          0
52     21223     41100    1  52852292 1    3     49909     50494     50495     1
00   1845351   1847396                                                          0
50     559921019      2 3     13182 2     21705    0                          0
50     559931011      2 3      4094 2     21706    0                          0
44     51784                5ON           9145!B           9146!T     GERFUSTR  1
00HS             1@A     B                                                      0
44     51785                5&R          41842@7     3         !B          418421
00HS             1@A     B                                                      0
44     51786                8FW              3AS             40FC             11
00@I             6@N              117              127              129       1
00         1                                                                    0
```

Figure 2.3: GDF example

Level 0 provides the geometry of real-world objects in terms of cartographic primitives, such as points, lines, and polygons. The points are used for describing the geometry of street crossings, the lines are used for roads, and the polygons are used for buildings and vegetation areas. The object coordinates are defined either within a national (local) geodetic system or within the World Geodetic System (WGS) [NIMA00].

In Level 1, the geometric elements of Level 0 get their real-world semantic. The level contains relatively simple feature types, such as junctions (street crossings), road elements (single road sections), buildings, etc. In Level 2, these simple features are combined into more complex objects. For example, road elements form a road, junctions (possibly in a combination with road elements) form an intersection.

Describing the spatial area in several levels simplifies working with the data. For instance, the features of Level 2 are inspected during a shallow search, like to find highways between places A and B. The features of Level 1 are examined to get a list of road segments.

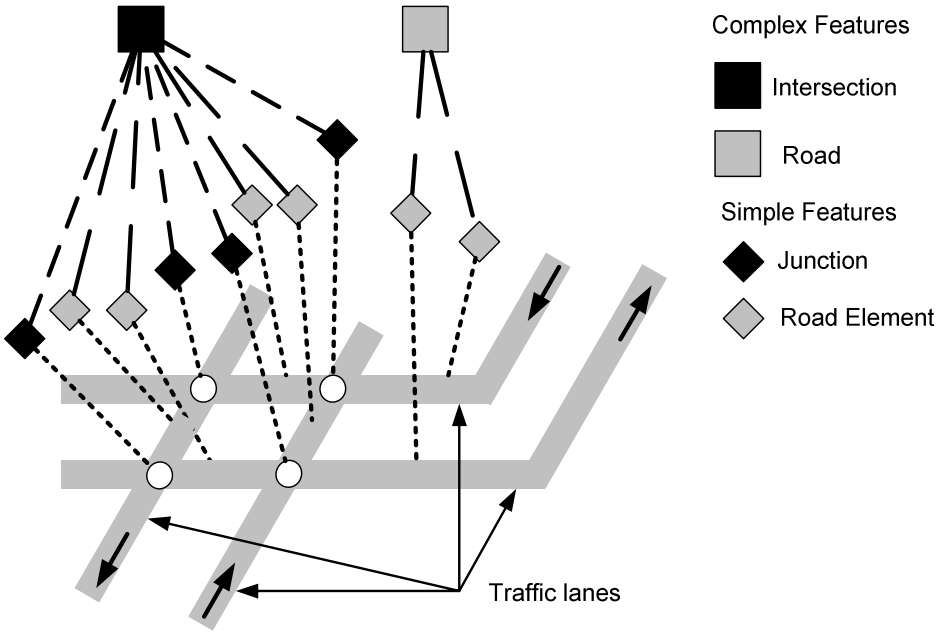


Figure 2.4: Example of multilevel road representation (based on [Wal96])

The Attribute Catalogue contains characteristics of objects. Similar to GML, they are stored as string pairs {name, value}. In difference to GML, GDF standardizes all possible attribute names and values for each object type.

The Relationship Catalogue stores associations between two or more objects, for example, “Prohibited Maneuver,” “Priority Maneuver,” or “Service along Road”. They are also defined using {name, value} pairs. The relationships might have own attributes, e.g., validity period.

For extensibility, GDF also supports custom (non-standard) object types, attributes, and relationships.

2.1.3 Discussion

Many geographic standards exist. But which of them should be chosen for integration into network simulation? The overview of two mostly used geospatial standards shows their following structural similarities:

- A geographic area is specified as a collection of real-world objects providing the information about movement area and buildings

- Object description includes their type, geometry, attributes (additional characteristics), and associations (relationships) between the objects
- The objects can be represented in a hierarchy (e.g., encapsulation of features by other features in GML, multilevel representation in GDF)
- Geometry of objects is described using common cartographic primitives, such as points, lines, or polygons
- Coordinates are specified within one of the commonly used geodetic systems
- Attributes of objects and their relationships are stored as {name, value} pairs

The study in [VGH+02] shows that other geographic data formats also share this paradigm. This allows creating a common model supporting diverse geographic formats.

2.2 Generic Geographic Model

The proposed architecture is shown in Figure 2.5. Geographic data sources in different formats, such as GDF and GML, are processed by dedicated parsers. Those parsers initialize the generic geographic data model, which contains the movement area objects, the movement area graph (which is described later in this chapter), and the building objects. The objects are described with their type information, geometry, attributes, and relationships, as shown in the previous section.

Different standards use their own name (code) and value domains for specifying the object types, attributes, and relationships. In order to support a standard-independent access to the spatial objects, they must be converted to a common format. Because GDF standardizes all possible object types, their attributes and relationships, this specification is used as the primary source for encoding the geospatial information in the described model. The parsers for other geospatial formats convert the object types, attributes, and relationships to GDF upon processing the data source.

Encoding of object geometry is similar among the standards. The generic model supports the common cartographic types, such as point, line, and polygon.

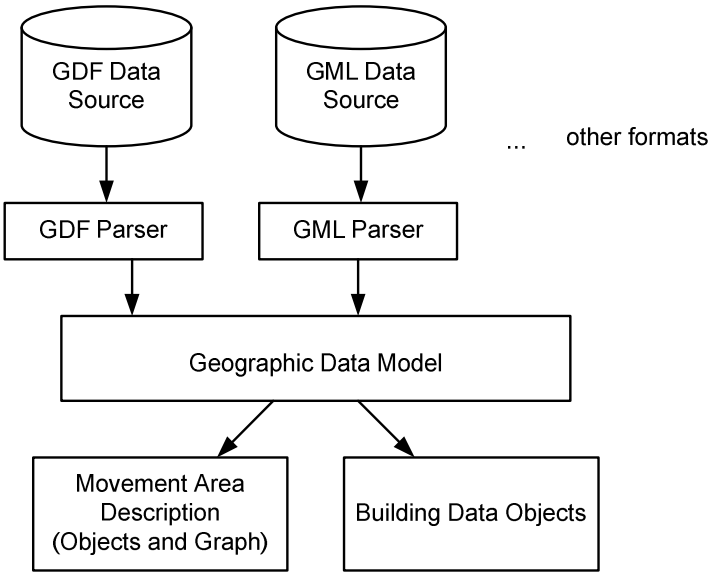


Figure 2.5: Generic geographic model

Next sections describe the representation of movement area and building data in more detail.

2.3 Movement Area Description

According to the discussion above, the movement area is specified as a collection of spatial objects, such as roads and street crossings. The object description includes their geometry, attributes, and relationships.

2.3.1 General Representation

A general representation of the movement area that is used in geographic standards is a graph $G:=(V,E)$ with a set of vertices V and a set of pairs of distinct vertices E (Figure 2.6). The graph vertices correspond to the road junctions and street end-points. The edges represent the road elements, which connect those junctions. The road junctions and road elements are stored as spatial objects in the element collection.

Depending on a data provider, a road element can characterize either one or multiple traffic lanes. These representations are interchangeable. In the first case, additional information (for instance, other GDF representation levels) gives a hint

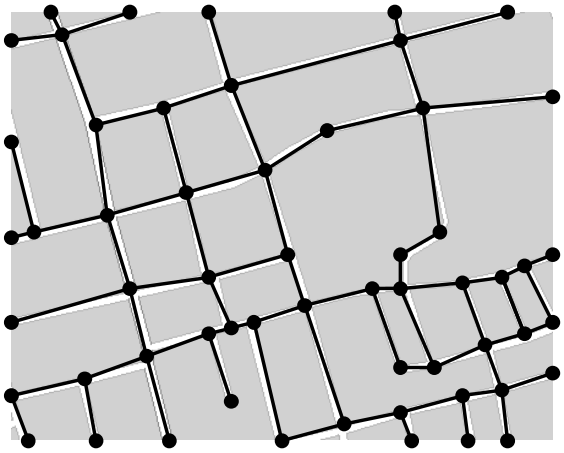


Figure 2.6: Movement area graph

that particular lanes belong to the same road. In the second case, attributes of road elements specify the number of traffic lanes and their directions.

In outdoor environments, the graph contains the movement areas for vehicular traffic and pedestrians. Movement *restrictions* (which are also stored in object attributes) help to distinguish themselves from each other.

2.3.2 Object Geometry

The geometry of movement area objects is in vector format. Therefore, the junctions are described with points; the road elements are described with lines. The coordinates refer to object center points.

The coordinates are within one of the geodetic reference systems. A geodetic system is a combination of an ellipsoid, which defines the size and shape of the earth, and a base point from which the other points are referenced. Calculating a distance between such two geodetic coordinates is not easy and requires the use of trigonometric formulas. To simplify data operations, the generic geographic model uses a 2-dimensional Cartesian system for road objects, so the distance can be simply computed with the help of Pythagorean Theorem. The parsers perform the necessary coordinate transformations upon reading the source data and initializing the model.

In addition to coordinates, the object attributes, like the road's width, the number of traffic lanes, etc., help specifying the shape of movement area.

2.3.3 Object Attributes

The attributes hold object-specific information. They are stored as {name, value} pairs and are encoded as in GDF for the sake of interoperability.

As of their category and validity, the object attributes can be classified as in Figure 2.7.

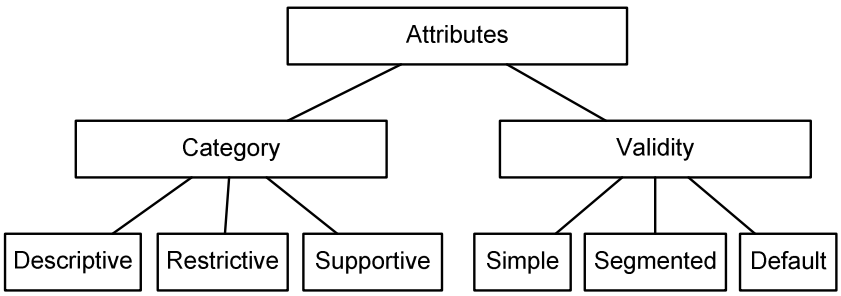


Figure 2.7: Classification of attributes

As of category, the attributes might be:

- Descriptive attributes: specify informational characteristics, like road name, road class, number of lanes, direction of traffic flow, road width, traffic sign information, etc.
- Restrictive attributes: specify restraining properties, like maximum speed allowed, maximum height allowed, vehicle types allowed, traffic sign information, etc.
- Supportive attributes: specify additional (service) information for data processing, like data validity period, positional accuracy, used geospatial system, etc.

As of validity, the attributes might be:

- Simple attributes: the attribute and its value are applicable to the given object.
- Segmented attributes: The attribute and its value are applicable to the given object only between the given curvimetric positions and in the specific direction.

- Default attributes: the attribute and its value are applicable to all geospatial objects, unless redefined.

2.3.4 Object Relationships

Relationships express associations between two or more movement area objects. The relationships are stored as {name, value} pairs. Similar to attributes, their encoding follows GDF.

The objects participating in a relationship are referenced by their unique identifiers. Those identifiers are either stored in the data sources or are assigned dynamically by the data parsers.

The object relationships can be classified into:

- Associative relationships: describe associations between given objects, e.g., "part-of" and "has-part" relationships specify objects in different representation levels.
- Restrictive relationships: specify the restrictions between objects such as driving maneuvers, e.g., "prohibited maneuver," "priority maneuver," etc.

The relationships might also define own attributes.

2.4 Building Data Description

Similar to the movement area, the building data are specified as a collection of geospatial objects. The object description includes their geometry, attributes, and relationships.

2.4.1 General Representation

In outdoor environments, each geospatial object represents individual building (Figure 2.8). For some of the objects, their service specifications (e.g., shopping center, supermarket, museum, restaurant, cinema) are provided, which give the information about typical activities that people usually perform there. In GML, the service specification is either a part of the object's type or is an additional attribute. In GDF, it is specified with an extra point feature which is located inside the corresponding object.

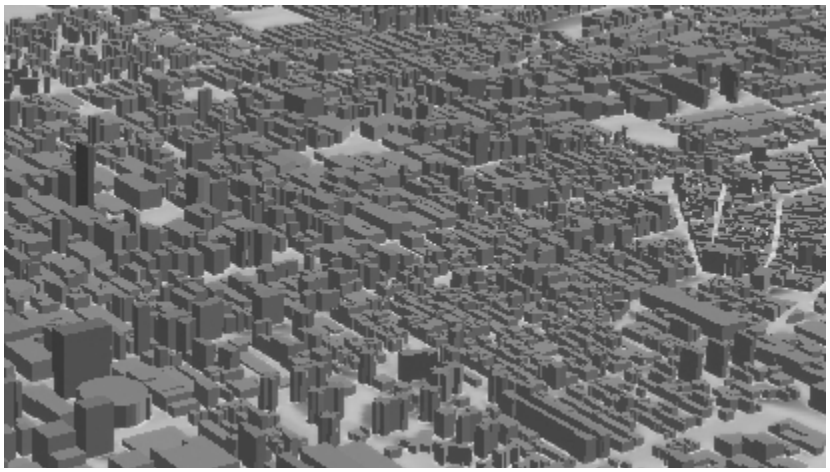


Figure 2.8: Example of outdoor building data (from www.awe-communications.com)

The representation of indoor environment is similar to outdoor, except that instead of buildings, their interior like rooms and furniture is specified. Some GML-based formats (like MasterMap) also support the description of combined indoor and outdoor environment by using object containments or relationships. However, in practice, the indoor and outdoor environments are normally specified in separate data sources.

2.4.2 Object Geometry

The geometry of building data is in vector format. Due to the complexity of data processing, 3-dimensional representation is rarely used. Instead, the objects are approximated with 2-dimensional ground area, which is described by a polygon, and uniform height (the so-called 2.5-dimensional representation). This 2.5-represenation is also used by the generic geographic data model in this thesis. Similar to road objects, the geodetic coordinates of buildings are transformed to Cartesian upon reading the source data.

2.4.3 Object Attributes

The building objects might also have their own characteristics which are specified using attributes. Those attributes are stored as {name, value} pairs. Possible name and value domains are as in GDF.

Similar to the movement area objects, the building attributes can be classified as in Figure 2.7. Here are some attribute examples:

- Descriptive attributes: “service specification,” “postal number,” “wall material properties.”
- Restrictive attributes: “opening period.”
- Supportive attributes: “positional accuracy,” “geospatial system.”

2.4.4 Object Relationships

The relationships describe associations of buildings with other spatial objects, e.g., “building along the road” or “room along the corridor.” They are similar to the relationships of movement area objects described in Section 2.3.4.

2.5 Implementation

The described generic geographic data model is implemented in Java. It is a part of CANUMobiSim framework[1], which is described later in this thesis. The implementation provides parsers for geographic data in GDF, GML, and some GML-based formats, for example, AWML [NM01]. The implementation can be extended to support additional formats.

The geospatial objects are read from data sources and are stored in the object collection. The object’s description (types, attributes, and relationships) is converted to GDF. For example, for AWML, the type “road” is converted to “4110,” which is GDF type for road sections, and the type “building” is converted to “7110.” The “name” property, which describes object’s official name, is stored under “ON.” Geographic data parsers are aware of the necessary mappings to be performed.

In addition to the object collection, a graph of the movement area is constructed. The points of interest and the buildings (or more precisely, the coordinates of the buildings’ center points) are also added to the graph to ease the construction of user trips during the mobility modeling. To support the evaluation of attributes of

[1] http://canu.informatik.uni-stuttgart.de/mobisim/index.html

geographic objects upon trip construction, graph vertices and edges contain references to the corresponding objects.

To ease distance calculations, object geodetic coordinates are transformed to Cartesian. The transformations take into account the ellipsoid that is used by the original geodetic system.

2.6 Summary

Spatial environment impacts the user mobility in the area and the propagation of radio waves. Therefore, it must be reflected in network simulations. Geospatial data standards define its representation in digital form. The overview of commonly used standards shows their conceptual similarities. The spatial environment is represented as a collection of real-world objects with their geometry, attributes, and relationships. This allows creation of a generic data model, which supports diverse geographic data formats and provides spatial information to the network simulation.

Each geographic data standard uses own notation for defining geospatial objects, their attributes and relationships. Since GDF standardizes all possible object types, attributes, and relationships, this specification is used as the primary source for encoding geospatial objects in the described generic data model. Parsers for other geographic data formats convert object type information, attributes, and relationships to GDF upon processing a data source.

The road network is used for modeling of user mobility. It consists of junctions and road elements. From these objects, a movement area graph is constructed, which eases user trip modeling. Object attributes and relationships enrich the description. They can be used, for instance, for simulating traffic in accordance with the travel restrictions, or for differentiating among pedestrian and non-pedestrian movement areas.

Buildings are important for the modeling of radio wave propagation. They are described with additional spatial objects. Some of their attributes, like “wall material,” help computing wave propagation paths. Other attributes can also be used

for mobility modeling, e.g., for reproducing typical people activities like shopping, sightseeing, and for reflecting service operating hours.

3 Modeling Mobility of Users

This chapter presents a more realistic approach to mobility modeling. Section 3.1 motivates the development of new approach. Section 3.2 describes the models used in related work and their shortcomings. Section 3.3 introduces the user-oriented mobility model. Sections 3.4−3.7 describe the parts of the proposed model in more detail. Section 3.8 discusses defining the model parameters for concrete scenarios. Implementation details are given in Section 3.9. Section 3.10 concludes this chapter.

3.1 Motivation

Mobile networks support nomadic users through special algorithms, communication protocols, and applications. To evaluate and to compare the performance of proposed solutions, network simulation tools must also include the modeling of user mobility. Many studies (e.g., [CBD02], [NG03], [THB+02]) show that networks perform differently under diverse mobility patterns. Hence, it is important that user movements in a simulation follow the scenario under evaluation.

In many scenarios, the user movements are constrained by a spatial environment, so they move along certain paths, like streets or corridors, and avoid obstructions of the area. In addition, the users travel to execute certain activities, such as shopping, sightseeing, visiting cinemas, etc. The users move with a certain dynamics. For example, pedestrians tend to move at low speeds with frequent interruptions, while vehicles move at higher speeds and influence dynamics of neighboring vehicles. Obviously, these factors have impact on user movements and must be reflected in simulations.

As shown later, many network simulation studies rely on rather simple mobility models. Such models either assume totally random movements within an area, which is similar to a chaotic motion of particles, or only focus on particular movement factors, thus underestimating other factors. To get more realistic simulation results, more complex mobility models are required. This chapter introduces such a mobility

model, which reflects the spatial environment, the user travel behavior, and the user movement dynamics, and is implemented in an easy-to-use framework.

3.2 Related Work

Let us discuss the approaches to mobility modeling that are currently used in network simulations.

3.2.1 Classification

According to the aspects of movement they focus on, existing mobility models can be classified as follows (Figure 3.1).

As of their *usage*, there are analytical and simulation models. Analytical models (e.g., [Gué87], [HR86], [Sch95]) are expressed by using mathematical equations. They allow describing in closed form different mobility-induced network parameters, for example, cell residence times, handover, and location update events. However, as more mobility factors are taken into consideration, the equations become more complex and it gets simply impossible to express them in closed form. In this case, simulation models are used (e.g., [Chi96], [LB98]), which imitate the behavior of a system with the help of a computer program. In general, simulation models allow for more detailed description of mobility process, however, they cannot be expressed analytically.

Mobility models use different *spatial granularity*. Detailed movement paths are not that important for cellular and base station networks, as the knowledge of which cells (areas covered by base stations) are visited by the users during their movement [JJ00], [Sch95]. This is different for ad-hoc networks, in which network topologies depend on exact positions of users [CBD02]. Therefore, mobility is either described at the level of network cells or is based on precise user coordinates depending on the network type.

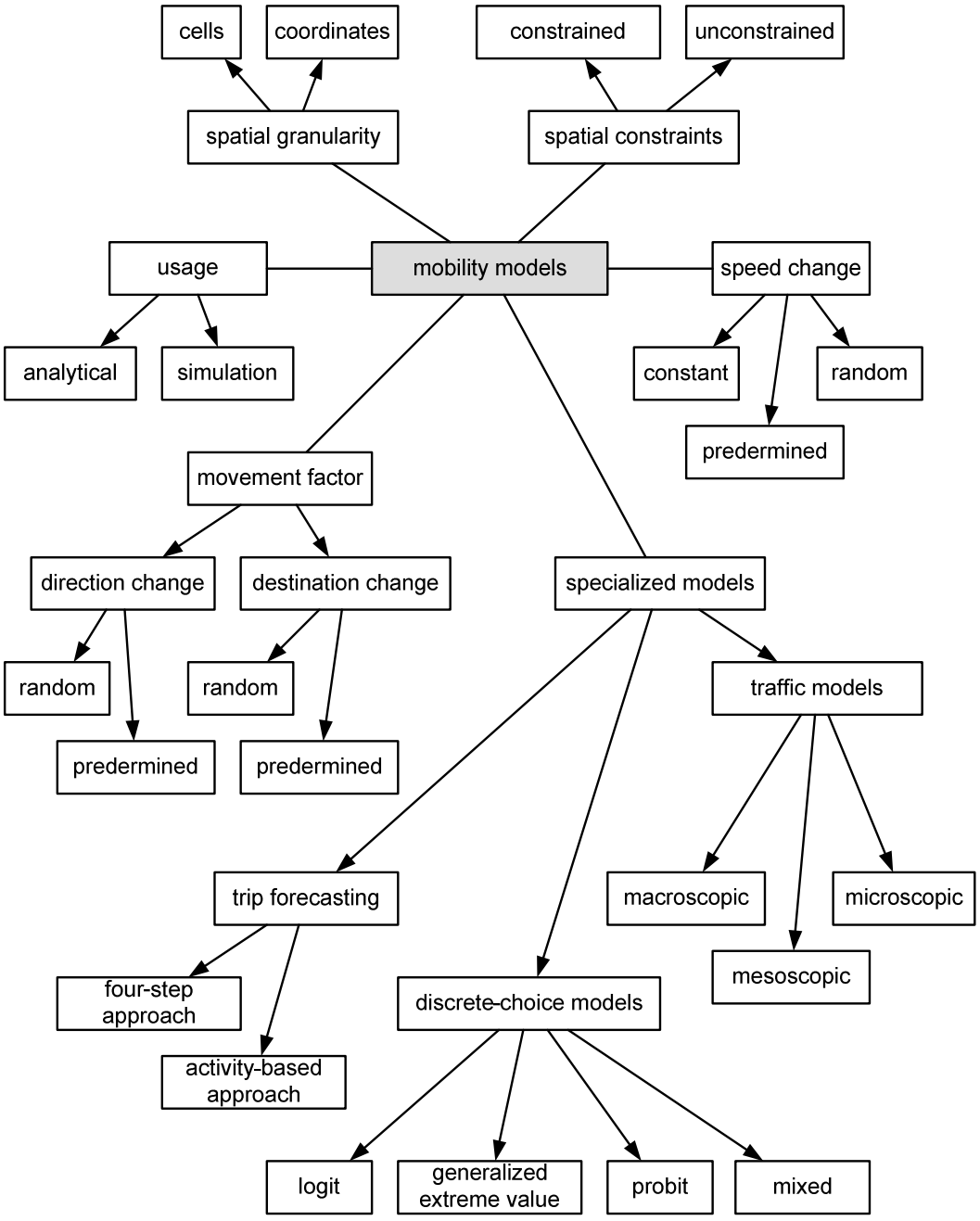


Figure 3.1: Classification of mobility models

Mobility is modeled under consideration of two major *movement factors*. One group of models represents the movement as a sequence of direction changes [KV98], [DCYS98]. In such models, a user moves in some direction for a certain amount of time (or makes a certain number of steps, or travels a certain distance, etc.), then a new direction is chosen and the movement continues. Other models, like

[JM96], consider that users move between different destination points, so after arriving to the current destination, new destination is chosen and the movement continues. The selections of new direction or movement destination are either random, or predetermined. The latter is performed, for example, as in the previously recorded user mobility trace [JHP+03].

Mobility models also differ in their reflection of *spatial constraints*. Some models assume an obstacle-free area, like a field [JM96]. Consequently, they do not take spatial constraints into account. Another group of models (for example, [THB+02]) takes spatial constraints into consideration, which broadens their usage to more scenarios.

As of *speed change* behavior, there are three groups of models. The models in the first group (e.g., [Gué87], [HR86]) assume that mobile objects move with constant speeds. As speed values, typical speeds are used, like for pedestrians or vehicular traffic. The models belonging to the second group, like [Bet01] and [KV98], allow for speed changes during the movement. New movement speeds are randomly chosen after certain time intervals or as the users travel certain distances. For the models in the third group [JHP+03], the speed change behavior is predetermined. It is performed, for instance, as in the previously recorded mobility trace.

There are also *specialized models* that are used in dedicated subjects for reflecting particular aspects of user movement. Trip forecasting models [Opp95] are used in transportation planning to predict user trips in the area, e.g., upon evaluating efficiency of a transportation system. Among these models, there is the four-step approach (see, for instance, [OW01]), which is a traditional method. The activity-based approach [Kit96] is a more recent method. It provides a deeper understanding of user travel behavior.

Discrete choice models [Tra03] predict the user's choice between the given alternatives. They are widely used in many areas, e.g., marketing, transportation, energy, housing, etc. In mobility modeling, the discrete choice methods are used for simulating a selection of movement path or a movement destination [BAB99]. The decision modeling is based on the factors that influence travel decisions, and dependencies between them. The methods also consider uncertainty of decision

making and unobserved factors. Several classes of discrete-choice models exist, depending on their assumptions (logit, generalized extreme value, probit, and mixed models).

The models describing dynamics of pedestrian and vehicular movement are developed in traffic modeling. They can be classified into three groups depending on the modeling granularity. Macroscopic models ([Khi98], [New93]) do not consider the dynamics of individual objects. They describe the movement of aggregated traffic and contain the cumulative parameters, such as traffic density or mean velocity. Microscopic models ([Kra97], [LR99]) describe the movement of individual objects. These models are more precise; however, require higher computational overhead. Mesoscopic models ([KBH98]) combine microscopic and macroscopic approaches.

Let us now look at these model classes in more detail.

3.2.2 Analytical and Simulation Models

Analytical models express the mobility-related parameters in closed form. Due to the complexity of obtained equations, these models typically rely on rather simple assumptions regarding the user behavior.

Hong and Rappaport [HR86] analyze the probability density function (pdf) of times that mobile users spend in network cells. The authors assume uniform user distribution over a cell. Movement speeds v and directions φ are uniformly distributed in the intervals $(0, v)$ and $[0, 2\pi)$, respectively. They are kept constant until a user crosses a cell boundary. Guérin [Gué87] proposes a more generic model, in which directions can be changed within a cell at certain moments of time. Del Re *et al.* [DFG95] assume that mobile users before crossing a cell travel the distance that is uniformly distributed in $(0, 2R]$, where R is the size of hexagonal cell. Similar to the models above, the movement speeds are also uniformly distributed. Zonoozi and Dassanayake [ZD97] introduce incremental speed and direction changes. The increments are within ±10% of current velocity and $\pm\Delta\varphi_{max}$ of current direction and follow uniform distribution.

Ho and Akyildiz [HA95] use discrete-time random walk model:

- At discrete times *t*, a user either moves to one of the neighboring cells with probability *q* or stays in the current cell with probability *1-q*.
- If the user decides to move to a neighboring cell, one of them is chosen with equal probability.

The cells are of equal size. The advantage of the model is that it can be described with Markov chains. The original paper presents this model in one and two dimensions (Figure 3.2). Mesh configuration with rectangular cells is added in [AHL96]. Jeon and Jeong generalize random walk further [JJ00].

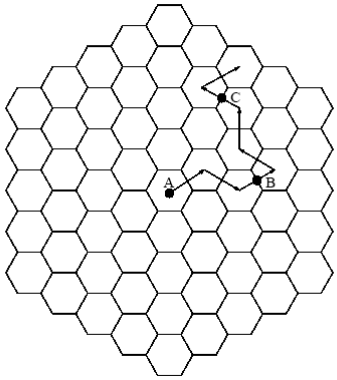


Figure 3.2: Random walk model in two dimensions (hexagonal cells, from [AHL96])

Tsai and Jan [TJ99] argue that random walk curve is rigid and abrupt. To make a user mobility trace smoother, they instead of choosing one of the neighboring cells adjust the movement direction φ by $\Delta\varphi$, which is chosen from normal distribution with zero mean. The new position is then rounded to the nearest cell. Normal distribution with zero mean is used due to following reasons. The majority of user trips have the same forward and backward routes, and consequently, the same number of left and right turns. This conforms to the symmetric property of normal distribution. Most trips follow shortest paths (i.e. straight lines). This corresponds to the probability density function of normal distribution, which increases as the

argument approaches zero, i.e. $\Delta\varphi$ values that are close to zero are chosen more often.

Schopp [Sch95] takes geographic distribution of user mobility into account. Five user mobility types are considered according to the distance they typically travel from their home location. The users of type 1 always stay at their initial position. The users of type 2 normally visit only the locations in the proximity of the home location. Types 3 and 4 perform larger trips. The users of type 5 visit the mostly remote locations. The model also considers sojourn times within cells, which are assumed to be negative-exponentially distributed.

From the overview above, it can be seen that analytical models oversimplify the modeling of user behavior. They do not explicitly model user travel decisions, such as the selection of movement paths and movement destinations. They do not take spatial constraints into account. These simplifications were specially introduced in order to reduce the model complexity for expressing the mobility parameters in closed form. Simulation models imitate the behavior of mobile users and can take more factors into account. Hence, they are able to describe the movement in a more detailed manner.

The model used by Chiang [Chi96] is an example of simulation model. The movement is described with a 3x3 probability matrix:

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix}$$

Matrix element p_{ij} contains the probability of switching from state S_i to S_j for X- and Y- movement directions, which are modeled independently. In the state S_0, a user stays at the current location, so $Coord_{t+1} = Coord_t$. In S_1, the user moves in negative direction, so $Coord_{t+1} = Coord_t - 1$. In S_2, the user moves in positive direction, so $Coord_{t+1} = Coord_t + 1$. Initially, both X- and Y-directions are set to S_0. Each move is one pixel, which represents one meter. The process is implemented in a computer program. A move is performed every triggering period Δt, which is individually set for each user according to his/her movement velocity. The movement speeds are kept constant.

This model is still simple and shares the shortcomings of the above-presented analytical models. It is described here only to show a conceptual difference between analytical and simulation models (i.e. closed-form expression vs. computer program). More simulation models are presented in further sections.

3.2.3 Spatial Granularity

Mobility models use different levels of spatial granularity. The models for scenarios with base stations normally consider only the movement between the cells that represent the areas covered by those stations. Many of such models are described analytically and are presented in the previous section, for instance, [AHL96], [HA95], [JJ00], [Sch95], and [TJ99].

For some types of networks, like a mobile ad-hoc network, the topology heavily depends on the positions of mobile users. Evaluations of such networks require the mobility model that relies on a more detailed modeling process and considers exact positions of users.

Liu *et al.* [LB98] describe two-level hierarchical model. On top, there is the *global mobility model*, which operates in terms of network cells. It is a deterministic model, which is characterized by a movement profile. This profile contains the sequence of cells that are to be visited by a user. Below the global model is the *local mobility model*, which describes a movement within a cell. It is based on the equation that combines user speed, direction, and position. The local model is a random model with dynamically changing state variables, which allows reflecting acceleration, speed, and direction changes. This combined model is still far from reality: random speed and direction changes poorly reflect the behavior of real network user. The model does not take spatial constraints into account.

3.2.4 Movement Factor

The mobility models consider two major movement factors. One group of the models focuses on user direction changes. Ko and Vaidya [KV98] assume that mobile users continuously move in rectangular area. The initial positions of users are chosen randomly. The movement speeds v and movement directions θ are selected from two

uniform distributions: [v_{min}, v_{max}] and [0, 2π) respectively. These parameters are kept until the user travels a distance d, which is randomly chosen from the exponential distribution with mean d_{mean}. Then new v, θ, and d are selected and the process repeats. If a user comes to the border of the area, he/she "bounces" and continues moving for the remaining portion of d. An example of the mobility trace that is generated for one user is in Figure 3.3.

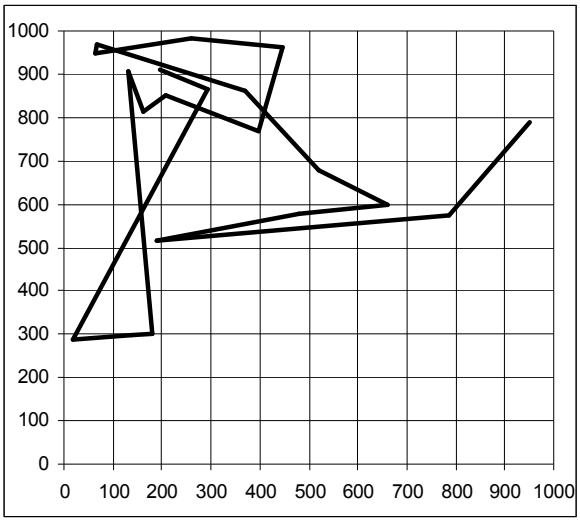


Figure 3.3: Sample mobility trace with d_{mean}=500

Camp *et al.* [CBD02] describe a similar model, but instead of distance d, time interval t is considered. To make direction changes smoother, Das *et al.* [DCYS98] limit $\Delta\varphi$ to [φ_{min}, φ_{max}]. The direction increment is then randomly selected and is added to the previous movement direction.

Bettstetter [Bet01] enhances the model further by adding correlations in state variables. In the *smooth mobility model*, new movement speed and direction are set not at once but incrementally in several time intervals. They change independently of each other. The moments of change are selected randomly from two exponential distributions. The model also considers typical movement speed and acceleration values for a given scenario, e.g., downtown. The approach is primarily designed to reflect the movement of vehicles.

Clearly, the mobility models that rely on random direction changes oversimplify the modeling of user travel decisions. In real life, users rather move between certain locations, like points of interest. Moreover, the model parameters must be carefully chosen. For instance, when d_{mean} (mean distance to travel) or t_{mean} (time interval till the next parameter change) are set to rather small values, the obtained mobility pattern is restricted to a small portion of simulation area. None of these models considers spatial constraints.

Another group of models focuses on the movement between the points of area. The *random waypoint mobility model*, which is proposed by Johnson and Maltz [JM96], is the most frequently used model in the simulations of mobile ad-hoc networks. It is supported by nearly all simulation tools. Initially, mobile users are placed at random locations within the rectangular simulation area. Then each user randomly chooses a destination point and moves there along a straight line with a constant speed v between $[v_{min}, v_{max}]$. Upon arriving to the destination, a user stays there for a pause time t_p between $[t_{pmin}, t_{pmax}]$. Next, new destination and speed are chosen and the process repeats (Figure 3.4). It is interesting to note that a majority of movement paths pass the central part of the area [BRS03]. This causes non-uniform distribution of mobile users in simulation.

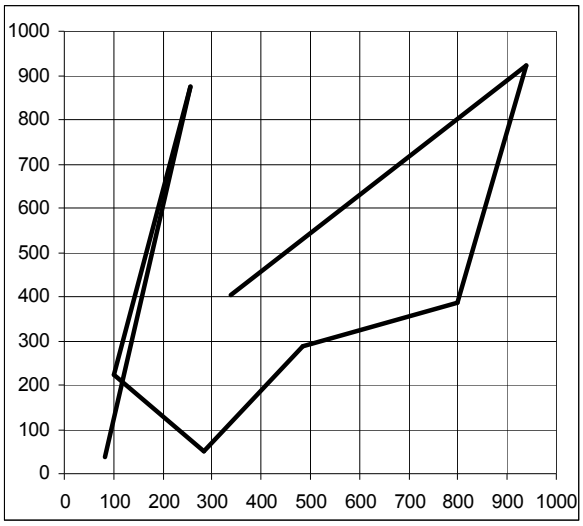


Figure 3.4: Sample mobility trace of random waypoint model

Realism of the above models can be improved by choosing the movement directions and destinations not randomly but as in real-world user mobility traces. Jetcheva *et al.* [JHP+03] analyze a performance of MANET routing protocol in a city by using bus movement traces. These traces were obtained with a help of odometry (i.e. by measuring rotations of wheels) and signpost transmitters.

Scourias and Kunz [SK99] reconstruct user trips from a survey. The survey contains information about the trip such as start and end times, their purposes, origins, and destinations. The authors define movement paths between those locations based on everyday experiences. Liu and Maguire [LM95] use a similar approach for the modeling of cellular networks. They create movement profiles that store the sequences of cells visited by users.

Obviously, the application of real traces allows obtaining more realistic simulation results. However, the traces are recorded for a given number of users only, and therefore, are not suitable for larger scenarios. The traces are difficult to obtain, e.g., due to financial expenses and privacy. Collecting traces is a long process and requires high post-processing overhead for extracting the necessary information from raw trace data. The traces hardly allow varying the movement parameters like speed or sequence of visited places.

3.2.5 Reflection of Spatial Constraints

The above presented mobility models do not take spatial constraints into account. However, the usage of geographical data is important, especially for the modeling of city scenarios. Tian *et al.* [THB+02] propose *graph-based mobility model*. The movement area is described with a graph (Figure 3.5). Graph vertices represent the points of interest, which are visited by mobile users (e.g., university, central station, castle, theater, etc.). Edges represent street elements connecting those locations. Initially, every user is placed at a random graph vertex. Then each user randomly chooses a destination vertex and starts moving there on the shortest path in the graph. Due to the movement along graph edges, the spatial constraints of the area are considered. The movement speeds are randomly chosen from the interval $[v_{min}, v_{max}]$ and remain constant during the movement. Upon arriving to the destination, the user

makes a random pause between [t_{pmin}, t_{pmax}]. Then a new destination is chosen and the process repeats. The authors argue that this model is realistic balance between completely deterministic and completely random models. Although this model makes rather simple assumptions concerning the user behavior (it basically assumes that all points of interest receive similar attention from a user), it is a first step towards the development of more realistic mobility models.

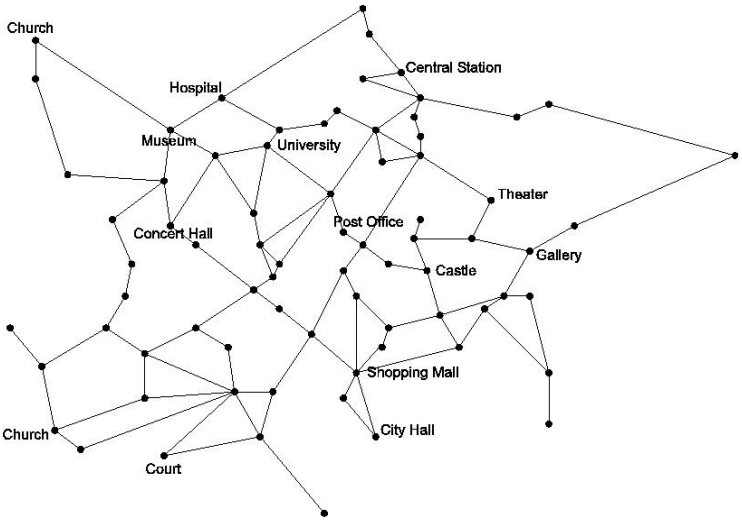


Figure 3.5: Graph of Stuttgart downtown (from [THB+02])

Jardosh *et al.* [JBAS03] describe the *obstacle mobility model*, in which the movement area graph is dynamically generated based on the locations of buildings. The buildings are defined with the help of polygonal shapes. Authors use Voronoi diagrams to compute the pathways that interconnect, lead into, and lead out of buildings (Figure 3.6). The vertices of the Voronoi graph and the intersections of the graph with outer boundary of the simulation region act as movement source and destination points. The intersections between the Voronoi graph and the obstacle boundaries act as doorways. The movement is modeled similar to the graph-based mobility model.

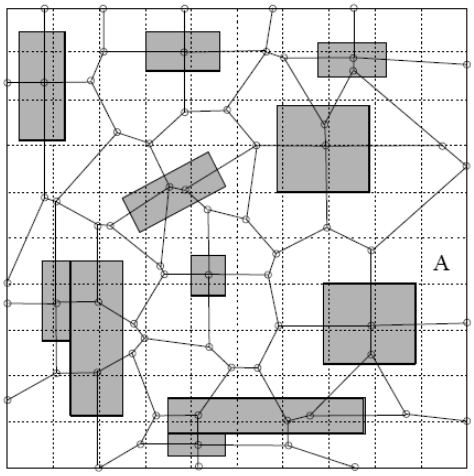


Figure 3.6: Simulation area and Voronoi-diagram based pathways (from [JBAS03])

3.2.6 Speed Change Behavior

As of speed change, mobility models assume constant, random, and predetermined behavior. The models, like [Gué87], [HR86], and [JM96], which are described above, assume that mobile users always move with constant speed. It is chosen at the beginning of movement and changes only after major modification in mobility parameters (e.g., movement direction, destination) occurs. This behavior is typically used to reflect the movement of low-speed objects, like pedestrians.

The models, like [Bet01], [KV98], [LB98], [THB+02], and [ZD97], allow changes in speed during the movement. This reflects a real-world behavior better. New movement speed is randomly chosen as the user travels certain distance or after a certain time interval. The new speed value becomes effective immediately or the current speed is changed incrementally until reaching the target value.

The third group of models, like [JHP+03], relies on predefined speed changes, for example, as in real-world mobility traces. Obviously, this results in more realistic behavior, however, shares the shortcomings of other models that are based on mobility traces. For instance, they have limited scalability and high collecting and post-processing overhead.

3.2.7 Specialized Models

There are also models that are used in dedicated subjects to reflect particular aspects of user movement. Thus, *trip forecasting models* [Opp95] are used in transportation planning to predict user trips in the area, e.g., for evaluating efficiency of a transportation system. Traditionally, the *four-step approach* is used (see, for instance, [OW01]). It consists of:

1) Generation of trips between large zones of the area
2) Distribution of trips across individual origin and destination locations
3) Splitting the trips into different transportation modes, such as public bus, private auto, walk, etc.
4) Assignment of trips to individual routes

The main disadvantage of this approach is that it does not reflect the underlying decision mechanisms of travel. In real life, people planning a day do not focus on the number of trips they are going to perform. This motivates the development of *activity-based travel demand modeling approach* [Kit96], [Pas96]. This approach represents the travel as a demand of participating in activities and provides a deeper understanding of travel behavior. According to the approach, users move to perform actions at certain places, like shopping in particular shops or visiting predefined sights. Such sequences of activities describe a user movement behavior.

Discrete-choice methods [Tra03], [BAB99] are used to predict a user's selection among several alternatives. The methods are based on the behavioral process function $y=h(x,\varepsilon)$, which describes the user's choice. x labels the choice factors that are observed by a modeler. For path choice modeling, these could be, for example, a path length or traffic jam probability. ε defines the choice factors that are not known to the modeler, and hence, are treated as random.

Depending on the type of random distribution and other assumptions, like interdependency of alternatives, several classes of discrete-choice models are derived. *Logit* model assumes that hidden factors follow Gumbel distribution and that the alternatives do not depend on each other. Although the latter can be unrealistic in particular situations, the advantage of this approach is the easiness of its mathematical description comparing to other approaches. This also results in

shorter computation times. Similar to the logit model, *generalized extreme value* model relies on Gumbel distribution but allow for correlation over alternatives. *Probit* model uses normal distribution. *Mixed models* combine different assumptions.

The models describing dynamics of pedestrian and vehicular movement are developed in the field of *traffic modeling*. According to the degree of granularity, the models can be classified into *macroscopic*, *microscopic*, and *mesoscopic* models. The macroscopic models (e.g., [Khi98], [New93], and [Pap83]) represent the traffic as a fluid. They do not consider the dynamics of individual objects. Instead, the models operate over aggregated parameters, such as traffic density or mean velocity. The microscopic models (e.g., [Kra97], [LR99], and [Wei93]) focus on the movement of individual objects. The models are more precise; however, require higher computational overhead. They reflect such aspects as car-following with traffic lane changing, vehicle acceleration and deceleration behavior, collision avoidance, traffic jams, car driver's perception, etc. Mesoscopic models ([KBH98]) combine the precision of microscopic and computational efficiency of macroscopic approaches.

It can be seen that the described specialized models reflect only particular aspects of user movement. Thus, the trip forecasting models allow predicting user trips in the area but they do not consider actual movement dynamics. The discrete choice models focus on estimating the user choices, while the traffic models only reflect the user movement along the given path.

3.2.8 Discussion

Existing mobility models can be classified according to different factors. There are analytical models that allow expressing mobility-related parameters in closed form. Due to the complexity of obtained equations, these models typically rely on rather simple assumptions regarding the user behavior. They do not explicitly reflect user travel decisions, such as the selection of movement paths and movement destinations. They normally consider random user speed and direction changes, which result in rigid and abrupt movement [TJ99]. Many of the analytical models only reflect the movement between network cells and neglect intracell movement.

These models do not take spatial constraints of the area into account. In contrast to the analytical models, simulation models imitate the behavior of mobile users, and hence, are able to describe their movement in a more detailed manner.

As of spatial granularity, there are two groups of models. The models in the first group only consider user movement between network cells. These models are specially designed for base station scenarios. The cells are applied to reduce the complexity of the models and to ease their analytical expression. The models in the second group rely on a more detailed modeling process and consider exact positions of mobile users. They are used in the modeling of networks having the topologies that are highly dependent on positions of mobile users, such as mobile ad-hoc networks. The models in the second group have a broader usage, since they can also be applied in the modeling of cellular networks, however, at cost of higher modeling overhead.

Mobility models consider two major movement factors. One group of models reflects user movements as a sequence of random speed and direction changes. Clearly, such models poorly reflect user travel decisions, since in real life, mobile users rather move between certain locations, such as points of interest. Also the simulation parameters must be carefully chosen. For example, if the interval between the direction changes is set to rather short values, the result mobility pattern is restricted to a small portion of simulation area. Another group of models assumes that the users move between the randomly chosen points of simulation area. These models reflect better the behavior of users.

However, the random movement is still far from reality. To improve the realism, several authors perform the selection of movement directions and destinations based on previously recorded mobility traces from real world. Obviously, the application of real data allows obtaining more realistic simulation results. However, the traces are recorded for a given number of users only, and therefore, are not suitable for larger scenarios. The traces are difficult to obtain, e.g., due to financial expenses and privacy. Collecting traces is a long process and requires high post-processing overhead for extracting the necessary information from raw trace data. The traces

hardly allow varying the movement parameters like speed or sequence of visited places.

Regarding the reflection of spatial constraints, most mobility models do not take them into account. They assume an obstacle-free area, like a field, which is a clear limitation of these models. For example, they can be hardly used in city scenarios. There are mobility models, like the graph-based mobility model [THB+02], which consider the spatial constraints of the area. However, they make rather simple assumptions regarding the user behavior. For example, they assume that all points of interest receive similar attention from the user. No movement profiles or typical user trips are considered by these models.

To reflect speed changes, mobility models assume that user speeds are constant, randomly chosen, or predetermined. Movement with a constant speed is typically used for reflecting the movement of low-speed users, like pedestrians. The speed is chosen at the beginning of movement and is kept constant until a major change in mobility parameters, like a new movement direction or destination. Random and predetermined models allow user speed changes during the movement, which better reflects a real behavior. However, the random speed changes poorly characterize the movement of certain types of mobile objects, like vehicles. Predetermined speed changes correspond to those in real-world mobility traces. Obviously, they allow for more realistic behavior, however, they share the shortcomings of other models that are based on mobility traces, like limited scalability, collecting and post-processing overhead.

We also see that there are specialized models, which are used in dedicated subjects, such as trip forecasting, discrete-choice modeling, and traffic modeling. They focus on particular aspects of user movement. These models are precise and are calibrated against real-world observations. Hence, they have a great potential for application in mobility modeling to better reflect the movement behavior of mobile user and their travel decisions. Analysis of related work shows that it has not been done so far, and thus, there is room for improvements.

3.3 Concept of the User-Oriented Mobility Model

Which parts must a mobility model include? From the review of related work, it can be seen that a spatial environment must be definitely considered. The environment constrains movements of users. So, they move along certain movement paths like streets and do not go through obstructions of the movement area. In addition, the spatial environment contains the points of interest (e.g., supermarkets or museums), which normally serve as movement destination points. The generic geographic model described in Chapter 2 can provide the spatial information to the mobility model.

The movement in a random direction or towards the purely random destination, which is frequently assumed by the approaches in related work, poorly reflects the behavior of real users. Obviously, people do not move completely randomly in the area. According to the activity-based travel demand approach [Kit96], [Pas96], people move to perform an action in certain places, for example, shopping in particular shops or visiting predefined sights. A sequence of such actions (trip sequence) defines user movements in the area. Hence, movement profiles and certain regular trips must be introduced. Besides, in order to get to a place where an activity can be executed, a user has a choice among several movement paths. Since this choice affects a trajectory of movement, it must be also considered in a simulation.

We see that mobile clients have different movement dynamics. For instance, pedestrians tend to move at low speeds with frequent interruptions, while vehicles move at higher speeds and influence dynamics of neighboring vehicles. Since the dynamics of client movement impacts stability of network topology, it must be reflected in a simulation as well. More realistic models for pedestrians and vehicles (i.e. not only based on constant and completely random speeds) must be integrated into mobility model. The specialized models mentioned in related work are a good basis for improving the realism of mobility models. However, since they only focus on individual aspects of user movement, they must be adapted and combined in order to be used together and with the generic geographic model.

Clearly, the new mobility model must be capable of reflecting diverse mobility scenarios and must be implemented in easy-to-use framework. Besides, as mentioned

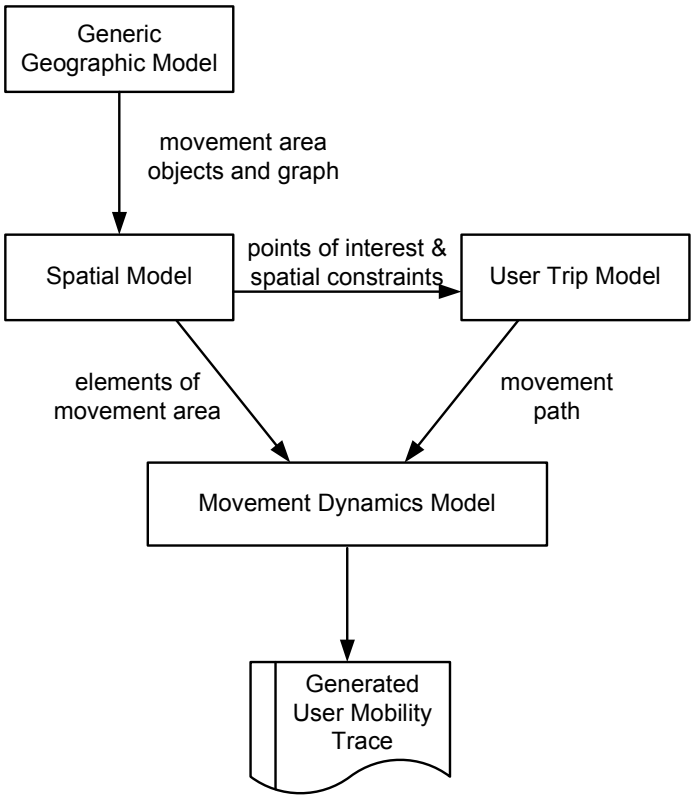


Figure 3.7: Design of user-oriented mobility model

in related work, there are cases when trip survey or mobility trace data are available, which describe the user movement behavior in the target area. If such data are available, the framework must make use of it to improve the realism of the modeling.

As a result, the proposed *user-oriented mobility meta-model* [SHB+03], [KBST04], [SMR05] is based on three sub-models (Figure 3.7): spatial model, user trip model, and movement dynamics model.

The *spatial model* contains a description of the movement area (spatial environment) and provides necessary information about area constraints and points of interest. The spatial model is automatically generated from the generic geographic model described in Chapter 2. For the mobility modeling, the movement area objects and the spatial model graph are considered.

The *user trip model* reflects user travel decisions. They consist of user trip sequences and movement path selections. This sub-model describes the movement at the level of user trips, e.g., "Move from point A to point B along the path P_{AB1}, and

then to C along P_{BC5}." The paths follow the spatial model, i.e. contain points of interest as the destination points and take constraints of movement area into account. This model relies on the approaches transportation planning and discrete choice theory. To obtain more realistic simulation results, the model parameters can be correlated with real-world traces, when such data are available.

The *movement dynamics model* describes the dynamics of user movement (position changes) along the chosen movement path. It is based on the approaches from traffic modeling. Different models are available here to support the modeling of pedestrians and vehicles in diverse scenarios. User position changes in time form the mobility traces, which serve as an input for MANET simulation tools.

Various types of mobile users can be represented in the same simulation by initializing individual model instances with corresponding parameters. Since this model is generic and is *oriented* towards a particular mobile user or a group of users, we call it user-oriented mobility meta-model.

The next sections describe these three sub-models in detail and their integration.

3.4 Spatial Model

The spatial model is based on the generic geographic model described in Chapter 2. It contains the movement area objects, such as road elements and junctions, and points of interest. The points of interest symbolize different places, like museums, hotels, bars, restaurants, cinemas, that are of interest for mobile users. The movement area objects are stored in a collection. The object's description includes their type information, geometry, attributes, and relationships with other objects. The type identifies the object's category, e.g., "road element," "junction," "restaurant," "museum," etc. The attributes hold additional object's properties, like "road name," "road class," "traffic direction," "number of traffic lanes", "opening hours". The relationships specify associations between two and more movement area objects. The encoding of object types, attributes, and relationships follows GDF specification.

Among the attributes, the "service description" is provided for points of interest. It contains the typical activities which people usually perform at those places, e.g.,

shopping, sightseeing, free-time, etc. As it is shown later, this information can be used by the trip model for reproducing user trips in the area.

The geometry of movement area objects is in vector format. So, the junctions and the points of interest are described with points; the road elements are described with lines. The coordinates refer to the object center points. The coordinates are specified in a 2-dimensional Cartesian system, which simplifies data operations. For example, a distance between two points can be computed with the help of Pythagorean Theorem instead of more complex computations on geodetic data.

The movement area objects are used for constructing a spatial area graph. The graph vertices correspond to the road junctions and street end-points. The edges represent the road elements, which connect those junctions. The points of interest are also added to the graph vertices. This eases construction of user trips during the mobility modeling. In particular, this allows using a graph path-searching algorithm to obtain a user movement path between the given source and destination points. The found paths then reflect the spatial area constraints.

3.5 User Trip Model: Trip Sequences

The necessity of the user trip model is motivated by the fact that, in real life, people with a goal neither move to make a certain number of steps in different directions nor move between randomly chosen points of the area. The movement of users often has also a distinct repeatability, e.g., commuting to work on weekdays. Such meaningful trips must be considered inside the mobility model.

The activity-based travel demand modeling approach [Kit96], [Pas96] focuses on reflecting the user trips. The approach defines travel as the demand to participate in activities. “No one would think about how many trips to make when developing a plan for a day; rather, one would think about what she wants to or needs to do, where the activities can or need be engaged, and, only then, would think about how to visit these places. Importantly, how many trips will be made depends on how the visits to different places are sequenced and combined into trip chains” [Kit96].

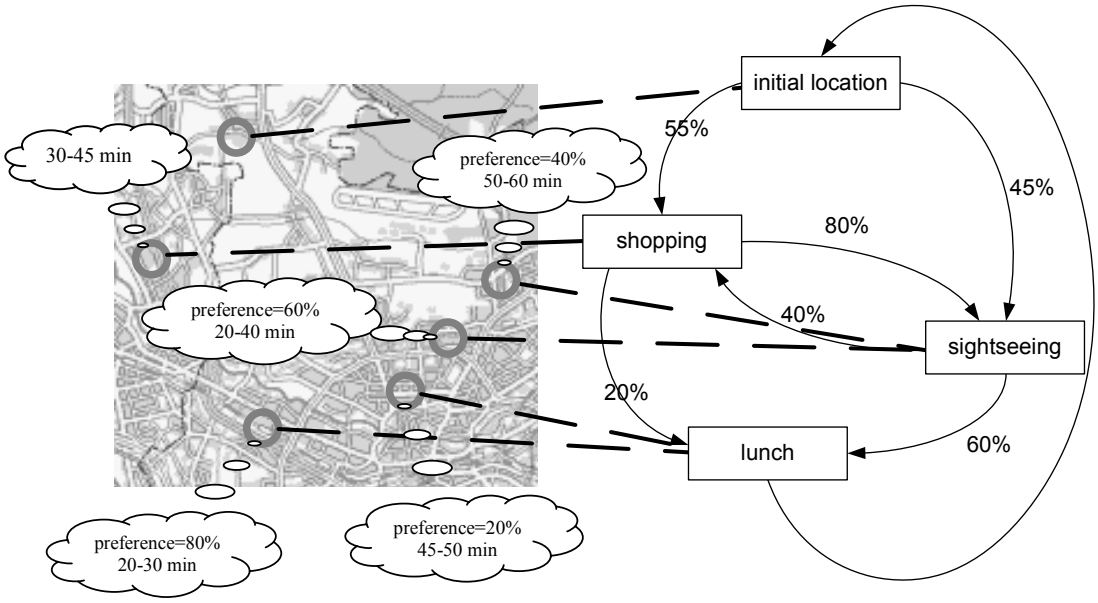


Figure 3.8: Example of a trip chain and its integration with the spatial model

As a consequence, our trip chain data include single actions (activities) performed by a user (Figure 3.8), such as shopping or sightseeing. The chain itself can be either defined individually for every user or aggregated over a group of users. The latter combines many individual trip chains, thus building a single group profile.

The activities can be performed at many locations; different transitions between the activities are possible. Each of the locations and transitions gets a certain "level of popularity" (probability of being selected), which reflects how many users choose the particular transition or location in real life. The spatial model provides these locations as points of interest. The trip chain references them by using the spatial coordinates (e.g., [1100762.20; 4934457.40]).

To integrate the trip chains into a simulation, the user-oriented mobility model uses a concept of *non-deterministic finite automata of activity sequences*. A non-deterministic finite automaton [HU79] (Figure 3.9) is composed of a finite set of states Q and transitions between the states. The transitions occur on input symbols α_i from a finite input alphabet $\sum$ according to the transition function δ. The automaton is non-deterministic, therefore, it has to perform zero, one or more transitions out of the state on the same input symbol. The automaton starts in the initial state q_0 and stops in one of the final states $q_i \in F$ ($q_i \in F$, $F \subseteq Q$).

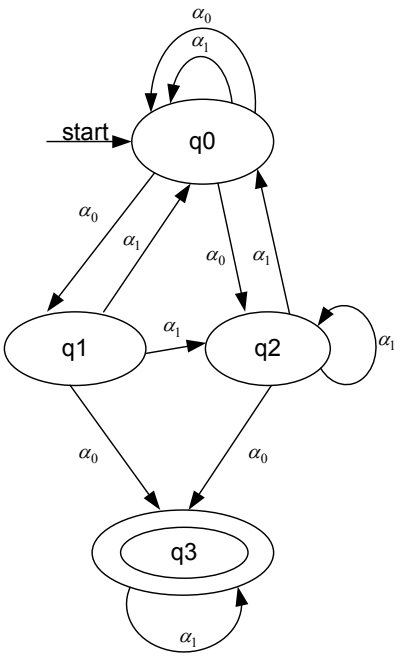


Figure 3.9: Non-deterministic finite automaton

Such automaton is used for a trip chain description. That is, the set of activities to be executed matches one-to-one with the automaton states. Each state denoting one activity contains a set of places where this action may be performed, its probability of being selected, and duration of the activity execution at this location (Figure 3.10).

$$q_i \rightarrow \begin{Bmatrix} loc_1 & p(loc_1) & d(loc_1) \\ loc_2 & p(loc_2) & d(loc_2) \\ \dots & & \\ loc_n & p(loc_n) & d(loc_n) \end{Bmatrix} \qquad \sum_{k=1}^{n} p(loc_k) = 1$$

Figure 3.10: Data associated with an activity (locations, selection probabilities, and execution durations)

For example, the “lunch” activity might contain the following data, which include two possible locations, their probabilities and durations:

$$\text{lunch} \rightarrow \begin{Bmatrix} 12\ \text{AlbSt.} & 80\% & 20-30\ \text{min} \\ 7\ \text{MaySt.} & 20\% & 45-50\ \text{min} \end{Bmatrix}$$

The activity from which the user starts the movement is denoted as the automaton's initial state, the final activities correspond to final states. The initial activity is associated with the user's initial position.

The automaton switches non-deterministically between the states, which corresponds to random transitions between the activities in a trip chain. Internally, the switching is performed by the transition function in accordance with transition probabilities. The input value serves as a signal to the automaton to perform a transition; the input alphabet contains a single value $\sum=\{\alpha\}$. After the next activity is selected, a location for its execution is chosen among the possible locations. Upon arriving to this point, the user stays there for the amount of time, which is needed to accomplish that activity. Next, a new transition is performed and the movement continues. For users in the end state, it is assumed that their final destination is reached.

3.6 User Trip Model: Path Choice

After the next destination point is chosen, the trip model must decide on a path there. The path starts from the current user position. In city areas, there is a variety of paths possible between the movement origin and the destination. Such diversity in movement path selections impact user mobility in the area, and hence, must be also reflected in a simulation. Depending on a simulation scenario and the modeler's assumptions, possible solutions to this problem fall into the following groups:

- Shortest-path choice: users choose a geographically shortest path for the movement
- Probabilistic path choice: users randomly choose one of the movement paths with a certain probability

To reflect diverse scenarios, the user-oriented mobility model supports both kinds of approaches. They are described in the following sections along with their integration into the proposed mobility model.

3.6.1 Shortest-Path Choice

Often a modeler does not have detailed information regarding the user behavior in the area and has only the movement paths. Then it is assumed that the user always chooses the shortest path. This path can be found by applying a shortest-path searching algorithm from the trip origin to the trip destination. The path searching is performed in the graph, which is provided by the spatial model. For the path searching, the weights of the graph edges are set according to the lengths of the corresponding road segments.

The Dijkstra's algorithm [Dij59] and the A^* algorithm [HNR68] are frequently used as path-searching approaches. The former is a traditional greedy algorithm. The latter applies heuristics to improve the greedy search.

Instead of searching for one shortest path, it is also possible to find multiple shortest paths, as described in [Epp94] and [Rup00]. Among these paths, one is randomly chosen with equal probability. Li [Li99] suggests considering up to 10 movement paths, which is sufficient for small and medium urban areas.

3.6.2 Probabilistic Path Choice

Simplicity is the main advantage of the shortest-path choice. However, according to the research in transportation planning, mobile users do not always select the shortest path. This motivates the development of probabilistic path choice models [BAB99]. In such models, users can use various paths and one of these paths is chosen with a certain probability. The probabilities are assigned based on different factors, for example, depending on estimated path travel times. The application of these models requires knowledge about the user behavior in the scenario.

The Dial's model and the corresponding STOCH algorithm [Dia71] perform such a probabilistic traffic assignment to multiple paths. Since approach is based on the logit discrete choice model [Tra03] (Section 3.2.7 of related work), the

corresponding methods can be used to correlate the algorithm parameters with real-world observations [Abr00], [Li99]. The computational complexity of the STOCH algorithm is comparable to the complexity of shortest-path searching algorithms.

The Dial's model and the STOCH algorithm are described below in more detail.

Notation

In the description of the model, the following notation is used (Figure 3.11). It is assumed that a mobile user is currently located at vertex *s* (trip source location) of the movement area graph. The user select the activity to be executed next and the trip destination point, which corresponds to graph vertex *d* (trip destination location). *e*=(*i, j*) denotes a single graph edge directed from vertex *i* to vertex *j*. The edge represents a transportation link[1], which is basically a road element of the spatial model. For pedestrians, the edges of the spatial model graph are assumed to be bidirectional and their attributes are valid in both directions.

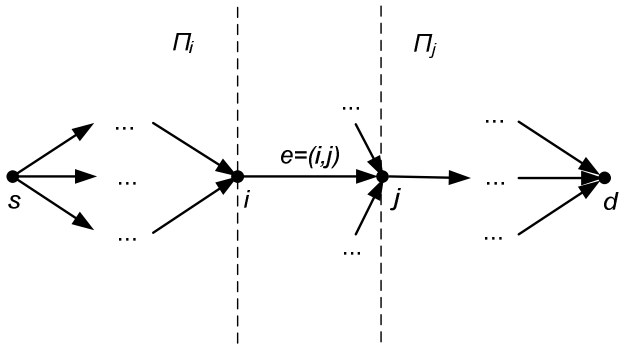


Figure 3.11: System model for probabilistic path assignment

Every edge has the associated cost *t*(*i, j*). This cost is the estimated travel time from *i* to *j* along the corresponding transportation link. It can be calculated from the

[1] In urban travel modeling, the term "link" stands for a road element used for transportation. The terms "graph edge," "road element," and "link" are interchangeable throughout this chapter.

geometric length of the link and the user typical movement speed. $t^*(s, i)$ denotes the shortest cost (the shortest time) to get from s to i.

Dial's Multipath Traffic Assignment Model

In the model, it is assumed that the selection probability a for transportation link $e=(i, j)$ is exponentially dependant on the difference between the cost of the shortest path $t^*(s, j)$ from s to j and the cost of the shortest path from s to j that contains the link e (a combination between the shortest path from s to i and the link itself, i.e. $t^*(s, i)+t(i, j)$):

$$a(e) = e^{\theta[t^*(s,j)-t^*(s,i)-t(i,j)]} \qquad (3.1)$$

This exponential expression is analogous to the expression of choice probability from the multinomial logit discrete choice model. The parameter θ ($\theta \geq 0$) is set by the modeler. It reflects the importance of the path length for the user choice and user knowledge about the movement area. As θ increases (i.e., the path length becomes more important and the user knows that the particular paths are shorter), the selection probabilities for shorter paths also increase. When θ is zero (i.e., the path length is not important or the user cannot estimate it), all efficient paths are considered equally. Similar as in the logit discrete choice model, this parameter is correlated with real-world observations. For the correlation, the methods of numerical maximization are used, like maximizing a log-likelihood function, as described in [Tra03].

Similar to user behavior in real life, the Dial's model considers only "reasonable" paths with *efficient* links. A link $e=(i, j)$ is considered to be efficient if the shortest path cost $t^*(s, i)$ from the trip origin s to the link start vertex i is less than the shortest path cost $t^*(s, j)$ from s to the link end vertex j (thus, users would come farther from the origin with every move). This reduces the number of links under consideration.

The rest of the model is defined as follows. The probability that the user choose a particular path P is proportional to the product of likelihood of all links in the path:

$$p(P) = k \prod_{e \text{ in } P} a(e) \qquad (3.2)$$

Obviously, the probability of using a link *e=(i, j)* is the sum of probabilities of all paths that contain this link:

$$p(e) = \sum_{\{P:\, e \text{ in } P\}} p(P) \tag{3.3}$$

Let Π_i denote all efficient paths from *s* to *i* and Π_j denote all efficient paths from *j* to *d*. We can also say that Π_i contains all efficient paths that topologically precede *e=(i, j)* and Π_j contains all efficient paths that topologically follow *e=(i, j)*. As shown in Figure 3.11, all efficient paths from *s* to *d* that contain *e=(i, j)* can be constructed by taking a path from Π_i, adding the link *e=(i, j)*, and appending one path from Π_j. Thus, (3.3) can be continued as:

$$p(e) = \sum_{\{P:\, e \text{ in } P\}} p(P) = \left[\sum_{P' \text{ in } \Pi_i} \prod_{e' \text{ in } P'} a(e')\right] ka(e) \left[\sum_{P' \text{ in } \Pi_j} \prod_{e' \text{ in } P'} a(e')\right] \tag{3.4}$$

A conditional link selection probability given its end vertex is visited *p(e | j)* is obtained by dividing its "absolute" selection probability *p(e)* by the sum of probabilities of all edges that have *j* as their end vertex:

$$p(e \mid j) = \frac{p(e)}{\sum_l p(l, j)} \tag{3.5}$$

By putting (3.4) into (3.5), we get:

$$p(e \mid j) = \frac{\left[\sum_{P' \text{ in } \Pi_i} \prod_{e' \text{ in } P'} a(e')\right] ka(e) \left[\sum_{P' \text{ in } \Pi_j} \prod_{e' \text{ in } P'} a(e')\right]}{\sum_l \left\{\left[\sum_{P' \text{ in } \Pi_l} \prod_{e' \text{ in } P'} a(e')\right] ka(l, j) \left[\sum_{P' \text{ in } \Pi_j} \prod_{e' \text{ in } P'} a(e')\right]\right\}}$$

$$= \frac{ka(e) \left[\sum_{P' \text{ in } \Pi_i} \prod_{e' \text{ in } P'} a(e')\right] \left[\sum_{P' \text{ in } \Pi_j} \prod_{e' \text{ in } P'} a(e')\right]}{k \left[\sum_{P' \text{ in } \Pi_j} \prod_{e' \text{ in } P'} a(e')\right] \left\{\sum_l \left[a(l, j) \sum_{P' \text{ in } \Pi_l} \prod_{e' \text{ in } P'} a(e')\right]\right\}}$$

$$= \frac{a(e)\left[\sum_{P' \text{ in } \Pi_i} \prod_{e' \text{ in } P'} a(e')\right]}{\sum_{l}\left[a(l,j)\sum_{P' \text{ in } \Pi_l} \prod_{e' \text{ in } P'} a(e')\right]} \qquad (3.6)$$

By introducing:

$$w(e) = a(e)\left[\sum_{P' \text{ in } \Pi_i} \prod_{e' \text{ in } P'} a(e')\right] \qquad (3.7)$$

and by denoting as F_j the set of edges that have j as end vertex, the expression (3.6) can be rewritten as:

$$p(e \mid j) = \frac{w(e)}{\sum_{e' \text{ in } F_j} w(e')} \qquad (3.8)$$

These conditional selection probabilities correspond to the ratio of mobile users that use a particular link for movement.

STOCH Algorithm and Movement Path Selection

STOCH algorithm [Dia71] (Figure 3.12) computes the conditional link selection probabilities as in the Dial's multipath traffic assignment model. For efficiency reasons, this algorithm obviates path enumeration (so it does not search complete movement paths between *s* and *d*) but determines selection probabilities only for individual graph edges. These probabilities are later used to successively select movement edges, thus obtaining a resulting movement path.

The algorithm is implemented in three steps. During the first step, "Initialization" is performed. STOCH calculates the shortest path costs from *s* to all other vertices, for example, by using the Dijkstra's algorithm [Dij59]. Then the link selection likelihoods $a(e)$ are computed as in Equation 3.1. The non-efficient links are assigned a likelihood of zero, and thus are excluded from further consideration.

1. Initialization

a) Calculate shortest path costs t^* from the trip origin s to all other graph vertices using a shortest path algorithm (e.g., Dijkstra [Dij59]).

b) For each edge $e=(i, j)$, determine its likelihood $a(e)$:

$$a(e) = \begin{cases} e^{\theta[t^*(s,j)-t^*(s,i)-t(i,j)]} & \text{if } t^*(s,i) < t^*(s,j) \\ 0 & \text{otherwise} \end{cases}$$

2. Forward Step

Starting with the origin s, for each edge $e=(i,j)$ determine its weight (likelihood with respect to likelihoods of the edges having i as end vertex (denoted as F_i)):

$$w(e) = \begin{cases} a(e) & \text{if } F_i = \varnothing \\ a(e)\sum_{e' \text{ in } F_i} w(e') & \text{otherwise} \end{cases}$$

Stop when the destination d is reached.

3. Backward Step

Starting from the destination d, for each edge $e=(i, j)$ determine its conditional selection probability $p(e|j)$ with respect to the edges having j as end vertex (denoted as F_j):

$$p(e|j) = \frac{w(e)}{\sum_{e' \text{ in } F_j} w(e')}$$

Stop when the origin s is reached.

Figure 3.12: Algorithm to estimate conditional edge selection probabilities (STOCH algorithm)

An example of computations during the first step is shown in Figure 3.13-Figure 3.15. For simplicity, only the results for efficient graph edges are presented. Initial edge costs (estimated travel times) are shown in Figure 3.13. Figure 3.14 presents the shortest path costs from the source vertex to other vertices. Figure 3.15 contains the computed link selection likelihoods for θ=0.1. We see that the likelihood $a(e)$ equals 1.0 if the link belongs to the shortest path $t^*(s, j)$ to its end vertex. This agrees to the

Equation 3.1. The larger is the difference between $t^*(s, j)$ and $t^*(s, i)+t(i, j)$, the smaller is the value of the likelihood for that link.

During the second step ("Forward Step" in Figure 3.12), the algorithm computes link's weights $w(e)$ depending on the weights of topologically preceding links (Equation 3.7). The algorithm starts from the edges that originate in s and continues with successive edges, until d is reached. The results of computations for the example above are shown in Figure 3.16. Clearly, since $w(e)$ depends on the weights of preceding edges, its value increases as we get farther from the source vertex.

In the third step ("Backward Step" in Figure 3.12), STOCH estimates conditional link selection probabilities $p(e \mid j)$. The computations are based on Equation 3.8. The algorithm traverses the graph from the trip destination vertex d back to the source vertex s. For each efficient edge, its conditional selection probability is calculated. Then the links that precede this edge are considered. Obviously, the sum of resulting conditional selection probabilities for the edges that have the same end vertex equals 1.0, so $\sum_{e' \text{ in } F_j} p(e' \mid j) = 1.0$. Figure 3.17 presents the computed selection probabilities for the given example.

It is clear that the STOCH algorithm terminates after the "forward" and "backward" steps. Since it excludes the non-efficient links from consideration, it traverses the graph edges successively in the topological order, until it reaches the trip destination vertex d.

Having the conditional selection probabilities calculated, it is possible to model a movement path selection for a user. The graph is then traversed in the reverse direction from the destination vertex to the source vertex. The resultant path is successively constructed by adding edges. The edges are chosen stochastically at each intermediate vertex from the set of efficient incoming edges in accordance with the computed selection probabilities (i.e. the edge with higher selection probability has a better chance of being selected).

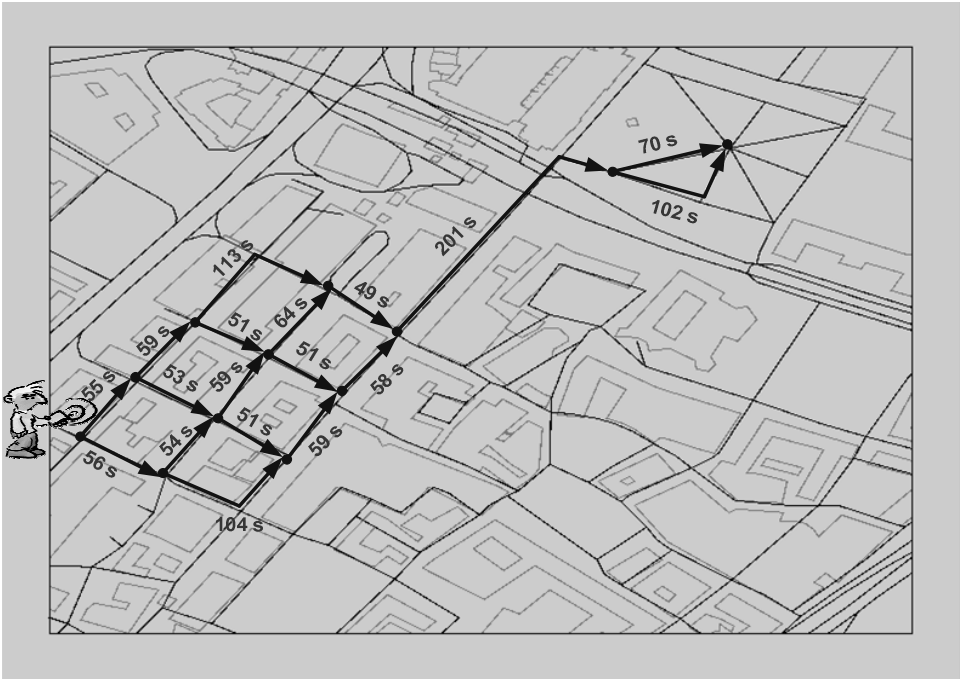


Figure 3.13: Estimated travel times *t(i, j)* for efficient graph edges

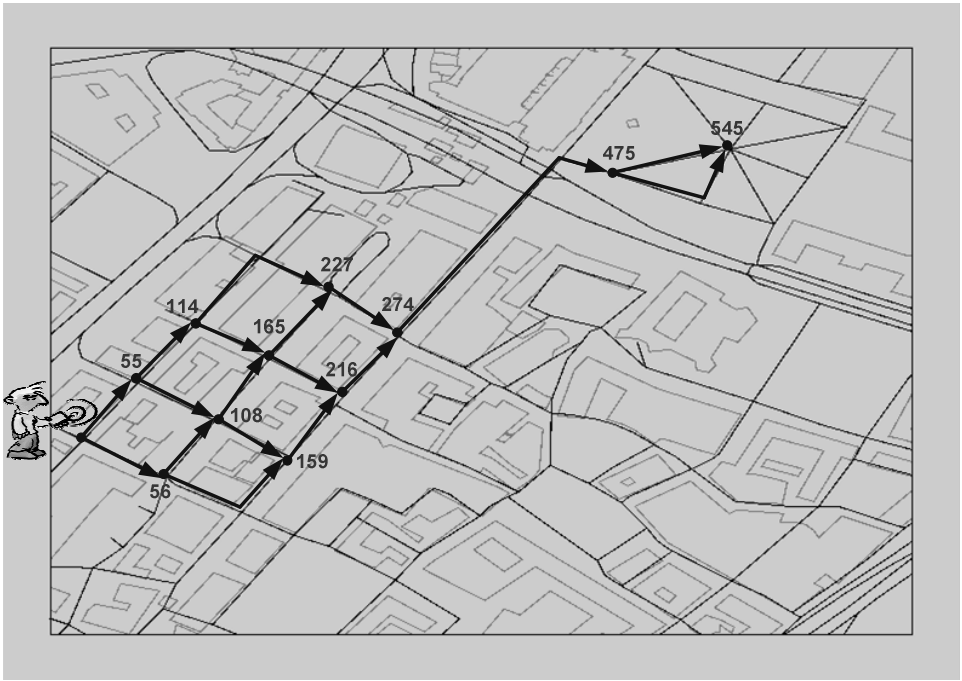


Figure 3.14: Shortest path costs t^* from the trip origin (Step 1a)

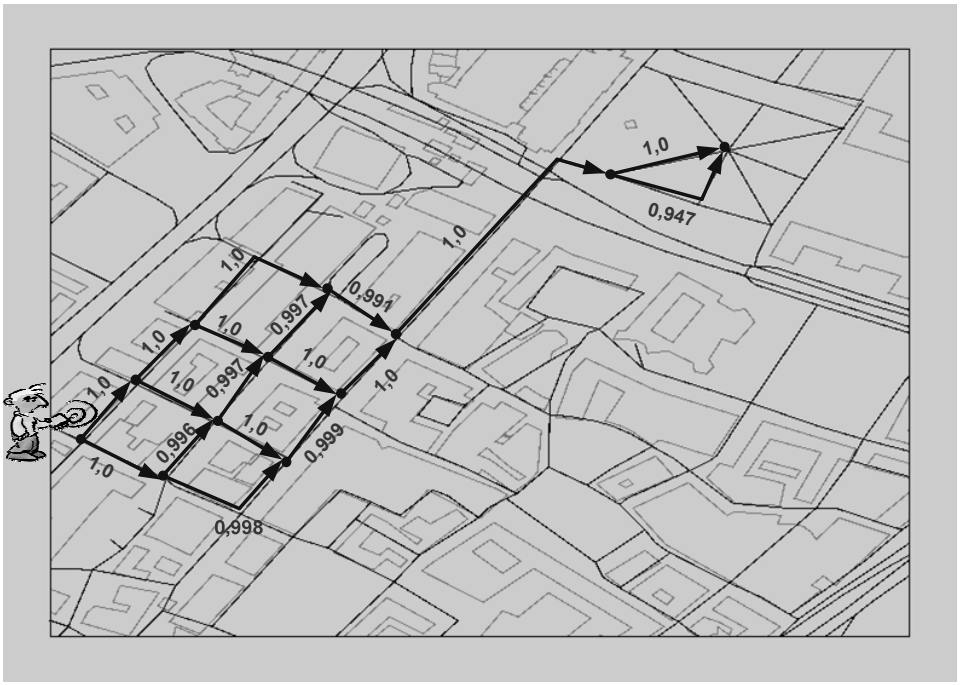


Figure 3.15: Link selection likelihoods *a(e)* for θ=0.1 (Step 1b)

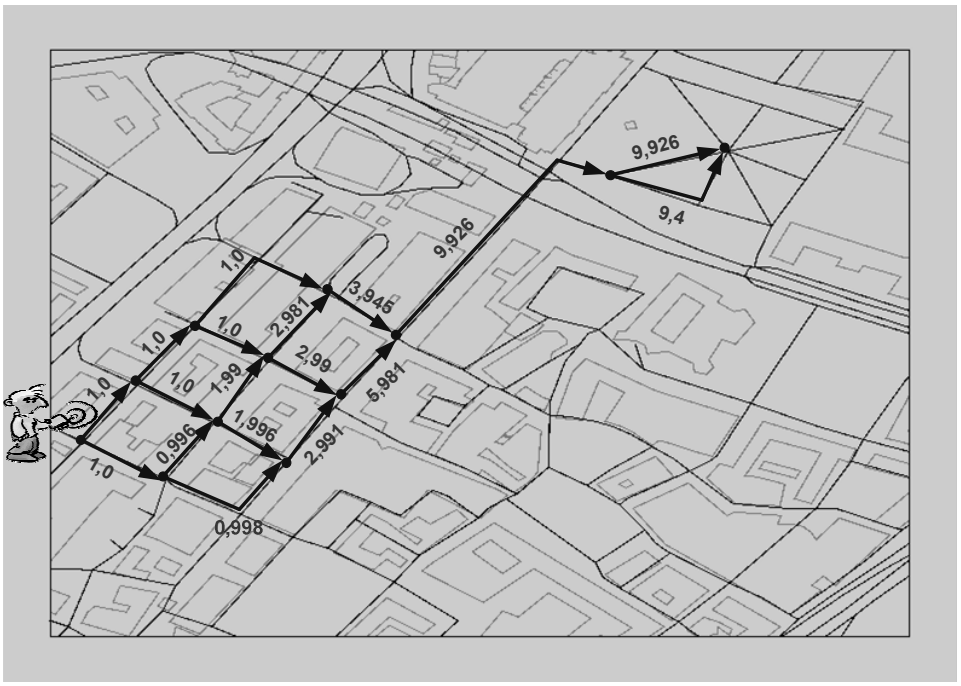


Figure 3.16: Link weights *w(e)* (Step 2)

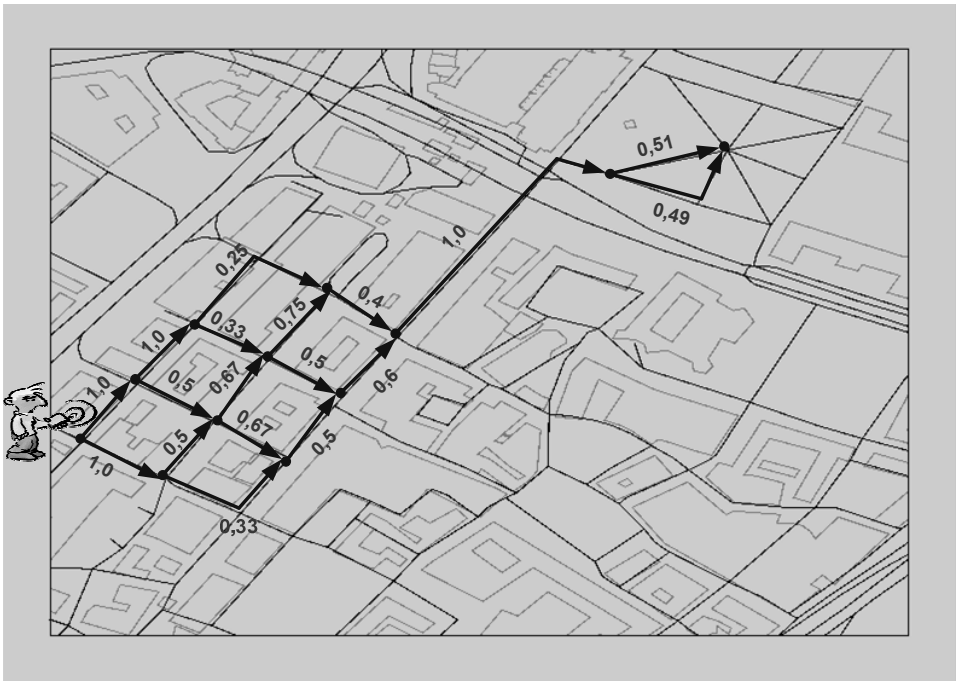


Figure 3.17: Conditional link selection probabilities *p(e|j)* (Step 3)

Since the link travel times only depend on path distances and typical user movement speeds, the corresponding edge costs do not change for the same user over time. Therefore, the STOCH algorithm needs to be applied only once for this user and for the given source and destination vertices. Once computed, the probabilities can be reused by future computations.

Extensions to the Probabilistic Path Choice

The described probabilistic path choice approach can be extended further. Car drivers try to avoid congested roads, since their travel speed highly depends on current traffic volumes. In transportation engineering, the so-called *volume-delay* functions model the impact of traffic volumes on link travel times, e.g., Bureau of Public Roads (BPR) function [BPR64], Overgaard function [Ove67], and Spiess function [Spi90]. Among these functions, the BPR is the mostly used. Its parameters are well documented for different road types based on real-world studies (see, for instance, [Hor91] for details). By applying this function before the STOCH

algorithm, it is also possible to model probabilistic route assignments for the vehicular traffic.

The BPR function defines the following dependency between the link travel time t and the link traffic volume Q:

$$t(Q) = t_0\left[1+\alpha\left(\frac{Q}{Q_{\max}}\right)^{\beta}\right] \qquad (3.9)$$

where:

t_0 is link travel time at free traffic flow. It is estimated from typical vehicle travel speed V_0 and the link length.

α and β are empirical coefficients. The BPR suggested values are α=0.15 and β=4 [BPR64], [Hor91].

Q is the current link traffic volume. It is expressed in vehicle/h. The traffic volume is calculated from the current density of vehicles (vehicle/m), traveling on the graph edge in the direction from the start vertex to the end vertex, multiplied by the mean traffic speed.

Q_{max} is the maximum link capacity, which depends on a road type. The spatial model provides the necessary road class identifiers in the attributes of road objects. Typical values for different road classes can be obtained from [Akç91] and [Khi98] (Table 3.1).

Road Class	Description	V_0 (km/h)	Q_{max} (vehicle/h)
1	Freeway	120	2000
2	Arterial (uninterrupted)	100	1800
3	Arterial (interrupted)	80	1200
4	Secondary (interrupted)	60	900
5	Secondary (high friction)	40	600

Table 3.1: BPR function parameters for various road classes (based on [Akç91])

This BPR function is used for estimating edge travel times. Then the STOCH algorithm is applied for calculating the edge selection probabilities. The algorithm also needs to consider the attributes of spatial objects. Thus, if a movement along a particular road element or in a particular direction is prohibited, this edge should be assigned the probability $p(e|j)=0$ and excluded from further consideration. As opposed to the pedestrian case, link travel times change dynamically due to changing traffic volumes. Therefore, the selection probabilities need to be recomputed each time when a path selection is performed.

An interesting option would be to apply the STOCH algorithm in the opposite direction (from the trip destination to the trip source point). As the result, the probabilities are obtained that the link and its *start* vertex are visited. This allows deciding on a movement path by traversing the graph from the source vertex. According to [Tob77], it reflects user behavior better in some cases.

It is also possible to model user path choices under consideration of factors other than the estimated travel time, e.g., a number of sights to be passed or a number of junctions. In this case, a modeler defines a *behavioral process* function [Tra03]. This function is a linear combination of attributes impacting the user choice. A modeler defines this function for a specific group of users in the target scenario. The user-oriented mobility model supports the behavioral process depending on the attributes of spatial model objects.

3.7 Movement Dynamics Model

The movement dynamics model describes a user movement along a chosen path. The model defines patterns in speed and direction changes based on the methods from traffic modeling. To support diverse mobility scenarios and different kinds of mobile users, the user-oriented mobility model provides the following movement dynamics models for pedestrian and vehicular traffic.

3.7.1 Pedestrian Models

The following models reflect the dynamics of pedestrian movement. The first group of the models considers traffic as a fluid. They operate over aggregated parameters,

such as density and mean velocity. Since the models do not focus on the movement of individual users, they are less precise but are computationally faster.

The simplest model assumes that the users always move with a constant speed. This speed is either fixed in the scenario or is randomly chosen between [v_{min}, v_{max}], as done by many mobility models in the related work.

A more realistic model is based on the dependency between pedestrian speed and density. Figure 3.18 shows the relationships for different types of users upon the studies of Older, Fruin, Oedin, Navin, and Wheeler, which were combined by Pushkarev and Zupan [PZ75]. By using these curves, the user movement speeds can be estimated from the total number of users that are currently located within the same transportation link. In a simulation, these speeds need to be periodically recomputed to reflect the density changes caused by the mobility of users.

Unlike the first group, the second group of approaches focuses on the dynamics of individual users. These models deliver more realistic results but also require more computations. The approach of Helbing and Molnár [HM95] relies on the study of

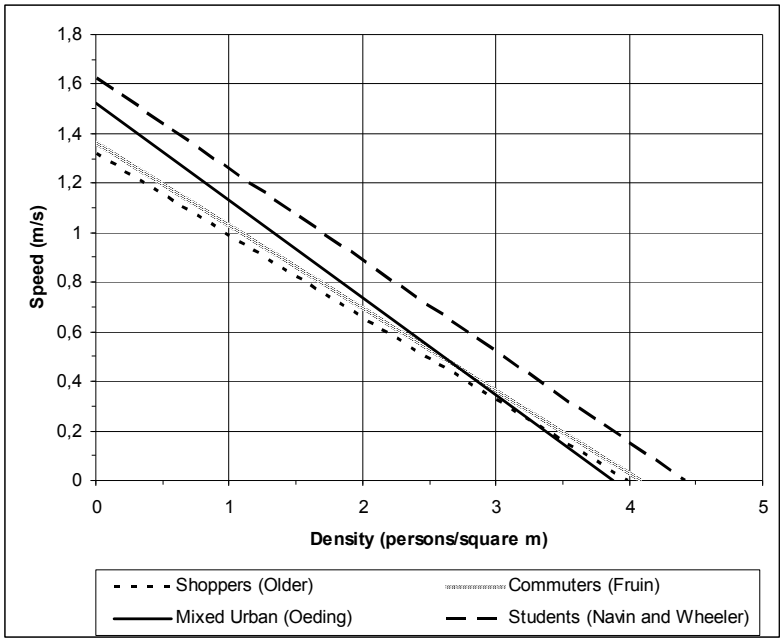


Figure 3.18: Relationships between pedestrian speed and density (based on [Khi98], [PZ75])

pedestrian behavior from [Wei93]. The speeds follow the Gaussian distribution between 0 and 1.742 m/s with mean v_0=1.34 m/s and standard deviation σ=0.26 m/s. In this model, the speeds are chosen randomly for each user and are kept constant throughout the movement.

3.7.2 Vehicular Models

The following models reflect dynamics of vehicular traffic. Similar to the pedestrian case, the simplest model assumes that the vehicles move with a constant speed along the chosen path. The speeds are fixed for the scenario. As an option, the speeds can be also selected randomly between [v_{min}, v_{max}], as done by many mobility models from the related work.

A more realistic model relies on a "volume-delay" function, e.g., BPR function [BPR64], Overgaard function [Ove67], or Spiess function [Spi90]. For the BPR function, the speeds will be periodically recalculated for each road segment as in Equation 3.9.

The intelligent driver model [THH00] is the most realistic model for vehicular traffic. It relies on the fact that car drivers keep certain distance between their vehicles for safety reasons. According to the model, acceleration of vehicle *i* at time *t* depends on its current velocity $v_i(t)$, the distance $s_{i,i-1}(t)$ to the vehicle *i-1* in front, and the safety distance $s^*_{i,i-1}(t)$ between the vehicles:

$$\dot{v}_i = a_i \left(1 - \left(\frac{v_i}{v_i^0} \right)^{\sigma} - \left(\frac{s^*_{i,i-1}}{s_{i,i-1}} \right)^2 \right) \qquad (3.10)$$

where:

- a_i is the maximum acceleration of the vehicle *i*. The typical value is a_i =0.6 m/s^2.
- v_i^0 is the desired speed of vehicle *i*, which depends on the driver. In simulations, this speed is usually chosen randomly from a typical interval, e.g., between 40 km/h and 50 km/h in city scenarios.
- σ is the exponent controlling the acceleration behavior of vehicles until they reach their desired speed. The typical value is σ =4.

Then the safety distance $s^*_{i,i-1}(t)$ between vehicles i and $i+1$ is expressed as:

$$s^*_{i,i-1} = s_0 + \max\left(v_i T_i + \frac{v_i \Delta v_{i,i-1}}{2\sqrt{a_i b_i}}, \quad 0 \right) \qquad (3.11)$$

where:

- s_0 is the minimum distance between vehicles, like in a traffic jam. The typical value is s_0=2 m.
- T_i is the safe time headway in congested traffic. It represents the maximum reaction time of a driver, e.g., to apply breaks. The typical value is T_i=1.5 s.
- $\Delta v_{i,i-1}$ is the difference in velocities between the vehicles i and $i-1$: $\Delta v_{i,i-1}=v_i-v_{i-1}$. This parameter is dynamically calculated.
- b_i is the driver's typical deceleration in regular (e.g., non-critical) situations. The typical value is b_i=0.9 m/s^2.

According to the model, on distances $s_{i,i-1} >> s^*_{i,i-1}$ the vehicle i accelerates until it reaches its desired speed. As the gap $s_{i,i-1}$ approaches the safety distance, the vehicle decreases its acceleration. When $s_{i,i-1} \leq s^*_{i,i-1}$, the vehicle breaks harder to avoid a collision.

The model is also capable of handling traffic jam situations, e.g., as the leading vehicles slow down, the succeeding vehicles will also decrease their speeds.

3.8 Defining Model Parameters

In order to use the user-oriented mobility model in simulations, its parameters need to be defined. They include the elements of the spatial model, the parameters of the user trip model (trip chains and path selections) and the parameters of the movement dynamics model. This section discusses how to set them in a given scenario.

3.8.1 General Considerations

The spatial model is basically a map of the simulation area. It contains the elements of the spatial environment, such as roads and buildings. The implementation provides the parsers for common geospatial data standards, like GDF, GML, and some GML-based formats. This enables the initialization of the spatial model from a geographic

information system. Converters from other data formats to GDF and GML are available, for instance, inside the Feature Manipulation Engine[1]. Otherwise, parsers for additional geospatial data formats need to be implemented.

The parameters of the movement dynamics model are specific to the model in use. The choice of the particular movement dynamics model depends on the simulation scenario, i.e. whether the movement of pedestrian or vehicular users is considered, or either aggregated traffic or individual objects are modeled. The implementation provides the necessary models for pedestrians and vehicles. For some models, like the models of Pushkarev and Zupan [PZ75], Helbing and Molnár [HM95], and the intelligent driver model [THH00], their authors specify typical parameter values. The user-oriented mobility model relies by default on these values. The parameters for other movement dynamics models can be estimated based upon daily experiences.

Defining the trip model parameters, such as user activities and trip chains, requires more effort. If the information about user trips in the area is not available, the model implementation can generate random trips between certain locations, such as bus stops, supermarkets, and sightseeing places, etc. The resulting mobility modeling approach would rely on a digital map of the area, real activity locations, and use a more realistic movement dynamics model, which is still closer to reality than the purely random motion of users.

The real-world data about user trips are available in the form of user position traces or travel surveys. To help defining the trip model parameters, the framework can automatically extract them from user position traces, as described below. The data from travel surveys can be directly incorporated into the model.

3.8.2 Travel Surveys

Travel surveys collect information about the user behavior in the area by the means of interviews. The corresponding questionnaires cover trip parameters (movement origin, destination, time, purpose, etc.) and person characteristics (gender, age,

[1] Feature Manipulation Engine is a product of Safe Software Inc., Suite 2017, 7445 132nd Street, Surrey, BC, Canada, http://www.safe.com/products/fme/index.php

education, employment, etc.). The survey results are then integrated into profiles, which define the user trip behavior.

Scourias [SK99] uses the Waterloo travel survey data [Tra89] in a mobility framework. The data are stored in activity transition (Figure 3.19) and activity duration matrices (Figure 3.20). The former contains the transition probabilities between the activities, which are performed at the predefined locations. The latter stores durations of activity executions. The structure of these two matrices is similar to the definition of the activity sequence automata in the user trip model. Thus, the first matrix describes the switching between the automaton states, as in Figure 3.9. The second matrix contains some information for an activity execution, as in Figure 3.10.

Previous Activity	Next Activity	Probability
8	1	0.121
8	2	0.041
…	…	…
8	8	0.062
8	9	0.037
9	1	0.056

Figure 3.19: Sample data from the activity transition matrix (based on [SK99])

Activity	Duration	Probability
6	400	0.021
6	460	0.013
…	…	…
7	0	0.126
7	150	0.110
7	100	0.178

Figure 3.20: Sample data from the activity duration matrix (based on [SK99])

3.8.3 Deriving Trip Model Parameters from Position Traces

Another option is to obtain the trip model parameters from position traces. A trace is a sequence of user location changes in time. It is automatically recorded from positioning devices, like General Positioning System (GPS) [DOD01] in outdoor scenarios or position sensors [MMLR05] in indoor. The implementation of the user-oriented mobility model offers the functionality for deriving certain trip model parameters from such traces.

Trace entries are of the following form (Figure 3.21):

- User ID
- Time
- User position (coordinate)

These entries are stored at constant intervals. Wolf *et al.* [WSS+03] state that in order to reconstruct movement paths for vehicles, the granularity should be less than 10 s. For pedestrians, a larger time interval might be used.

The provided method for deriving trip model parameters is based on associating the locations in the area with typical activities (Figure 3.22) [SMR05]. These locations are provided by the spatial model as "points of interest." Their "service specification" attribute, for example, "shopping center," "supermarket," "museum," "restaurant," "cinema" gives a hint on which typical activities the users usually perform there. We use this information for associating activities with the

User ID	Time (ms since 1/1/1970)	User Coordinate (WGS84)
1	1189457250704	47°15.376' -121°14.830'
1	1189457250800	47°15.512' -121°14.813'
…	…	…
1	1189457251486	47°14.939' -121°13.528'
2	1189457260745	47°15.050' -121°13.706'
2	1189457270812	47°14.800' -121°12.996'

Figure 3.21: Sample data from a position trace

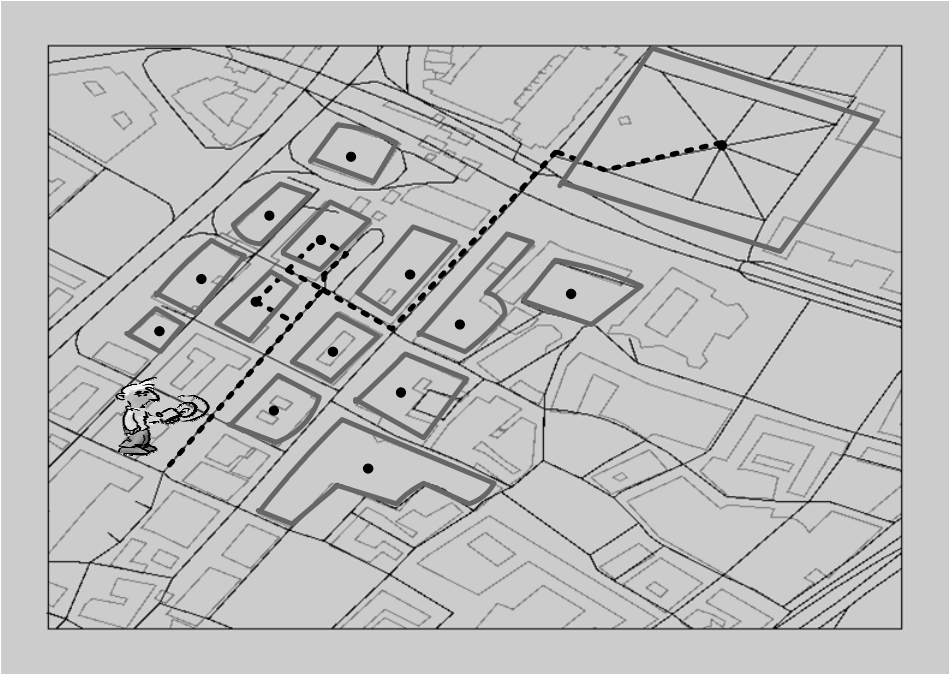


Figure 3.22: Example of parameter derivation

corresponding locations in the area. This allows the "reverse" mapping from a location to the performed activity. Thus, if we detect a user stays at a particular location, we can also determine which activity was performed there. The association is unambiguous, i.e. although multiple activities can be performed at a point, we consider only the primary activity (e.g., a shopping center is used only for shopping).

Our current implementation automatically distinguishes the following activities from GDF sources: "business", "cultural", "educational", "meal", "parking", "recreation", "shopping", and "sightseeing". Besides, we introduce two special activities: "initial" and "unclassified". The initial positions of mobile users in the trace sample are associated with the "initial" activity. All the points of interest that do not belong to one of the standard activities are associated with the "unclassified" activity. Moreover, if upon a trace analysis we detect a user stays at a location, which is not associated with an activity, it is also added to the "unclassified" activity. Obviously, the described splitting into activities is neither obligatory nor complete. It just reflects our current simulation scenarios. In our implementation, it is possible to change the standard associations and / or introduce new ones.

```
process_traces(traces: array of Traces)
      foreach user u in traces
            // process user trace entries in chronological order
            trace_entries = sort_by_time_ascending(traces[u])
            foreach entry e in trace_entries
                  process_trace_entry(e)
            end for
      end for

process_trace_entry(e: TraceEntry)
      if e.location == last_location
            if abs(e.time – last_time) >= min_activity_time
                  // an activity execution is detected
                  activity = get_activity(e.location)
                  increase_transition_probability(
                                    previous_activity, activity)
                  increase_location_probability(activity,
                                                e.location)
                  previous_activity = activity
            fi
      else
            if activity != null
                  // the activity execution is finished
                  update_activity_duration(activity,
                                           e.time – last_time)
                  activity = null
            fi
            last_location = e.location
            last_time = e.time
      fi
```

Figure 3.23: Pseudo-code of trace processing algorithm

After deciding on possible places and activities, the process of trip chain derivation is straightforward. The following parameters are to be found: probabilities of transitions between activities, visiting probabilities of locations, and the durations of

the activity executions at the locations. These are obtained by comparing user positions with the coordinates of points of interest.

The pseudo-code of the algorithm is presented in Figure 3.23. It processes the trace entries for each user in chronological order and checks the following. If a user stays within a point of interest longer than a minimal activity execution time (e.g., 10 minutes as described in [AS03]), an activity is detected. We increase the transition probability between the previous and the current activities (by incrementing a counter) and increase the probability of selecting the location for executing the corresponding activity. If a user departs from a point of interest (i.e. the user position is no longer within the previous point of interest), the execution of the previous activity is finished. We then update the durations of activity execution at the location (the minimum and maximum values; as an option, the arithmetic mean or the weighted arithmetic mean can be applied).

After the trace data is processed completely, we calculate the selection probabilities for locations within every activity (location's preference) by dividing the counter for the activity at the location by the counter for the activity. We compute the probabilities of transitions between two activities by dividing the count of transitions between the activities by the total count of transitions from the source activity.

3.9 Implementation

The described user-oriented mobility model is implemented with all sub-models in a framework for user mobility modeling CANUMobiSim (Figure 3.24). The framework is written in Java, which allows it to run on different hardware platforms. It is a standalone application, thus it does not directly influence performance of the original network simulation tool.

To use geospatial data in different formats, the framework includes the generic geographic model from Chapter 2 and parsers for GDF and some GML-based formats. The generic geographic model contains the movement area and building objects. The objects are described with their type information, geometry, attributes,

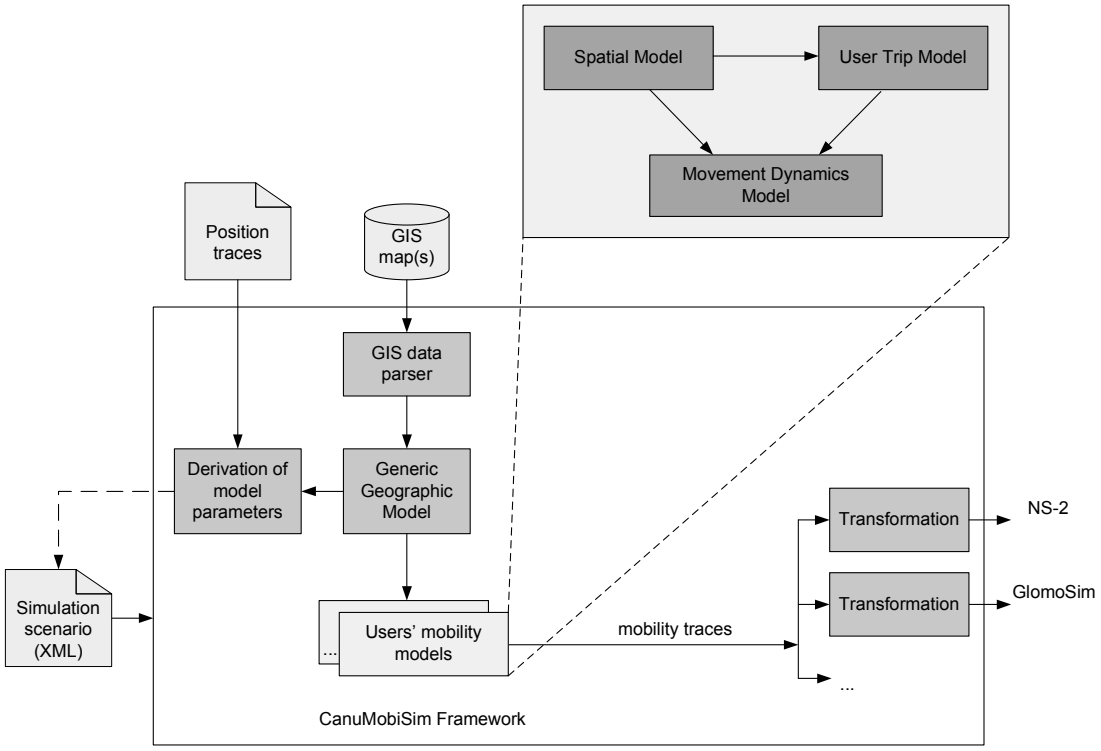


Figure 3.24: Architecture of the CANUMobiSim framework

and relationships. To access these data in common way, they are encoded as in GDF. The parsers for non-GDF formats convert object notation to GDF. In addition, the movement area graph is constructed, which includes points of interest and street information.

Movements of every user are simulated with a separate instance of the user-oriented mobility model. This model includes the spatial model, the user trip model, and the movement dynamics model. This allows having different kinds of users (i.e. having dissimilar travel behavior, or dynamics, or moving within different parts of area) in the same simulation. An instance of the spatial model might be also shared among multiple instances of the user-oriented model, if the users are moving in the same region.

The spatial model is basically a subset of the generic geographic model, which is used for mobility modeling. It contains the movement area objects and the movement area graph. The user trip model reflects the user trip sequences and movement path selections. The trip sequences are defined by using the non-deterministic finite state

automata, which combine activities, transitions between them, possible locations and execution durations. If there is no information about the user trips in the area available, the framework can also generate random trips between certain locations, such as bus stops, supermarkets, and sightseeing places, etc. These locations can be automatically extracted from the spatial model. Alternatively, the module of parameter derivation helps to obtain some trip model parameters from user position traces.

To model the movement path selection, the framework offers the Dijkstra's shortest-path searching algorithm [Dij59], the STOCH probabilistic path selection [Dia71], and also the described STOCH extension for vehicular traffic, in which the path costs are dependent on current traffic volumes. For the modeling of user movement dynamics, the framework provides simple random models, as well as more sophisticated approaches for pedestrian and vehicular traffic, such as the models of Pushkarev and Zupan [PZ75], Helbing and Molnár [HM95], and intelligent driver model [THH00]. The framework also includes default parameter values for these models.

The obtained user position changes are saved in mobility traces, which are later used by network simulation tools, like ns-2 [BEF+00], GlomoSim [ZBG98], etc. Since the tools use different trace format, we need a transformation module to convert the position information and to arrange the data properly. CANUMobiSim provides several transformation modules supporting different network simulation tools.

The scenario file is defined in XML. It specifies the simulation parameters. Two scenario examples are given in Figure 3.25 and Figure 3.26. In the first example, pedestrians are modeled in a city center. The mobility trace is saved in ns-2 format. Simulation time is 14400 seconds. The spatial environment is read by the GDF reading module from file "Boston.gdf." The module is configured to process only the central part of the area (between the *(min_x, min_y)* and *(max_x, max_y)* coordinates). User trips are modeled according to the automaton of activity sequences. The locations were previously extracted to text files "initial.txt" and "shopping.txt" (the framework can perform this operation). The movement dynamics

```
<?xml version="1.0"?>
<!-- Pedestrians in a City Center -->
<universe>
  <extension class="canumobisim.extensions.NSOutput"/>
  <extension class="canumobisim.simulations.TimeSimulation" param="14400.0"/>
  <extension class="spatialmodel.core.SpatialModel"/>
  <extension class="gdfreader.GDFReader" source="Boston.gdf" min_x="4000"
max_x="6000" min_y="3000" max_y="4000"/>
  <extension name="Gen" class="tripmodel.generators.ActivityBasedTripGenerator">
    <activity id="initial">
      <points>initial.txt</points>
      <minstay>0.0</minstay> <maxstay>0.0</maxstay>
    </activity>
    <activity id="shopping">
      <points>shopping.txt</points>
      <minstay>900.0</minstay> <maxstay>1800.0</maxstay>
    </activity>
    <!—More activities …  -->
    <transition>
      <src>initial</src> <dest>shopping</dest> <p>0.55</p>
    </transition>
    <transition>
      <src>initial</src> <dest>museum</dest> <p>0.45</p>
    </transition>
    <!—More transitions …  -->
  </extension>
 <nodegroup n="100">
    <extension class="uomm.PedestrianMotion" initposgenerator="Gen"
tripgenerator="Gen"/>
  </nodegroup>
</universe>
```

Figure 3.25: Scenario 1 – pedestrians in a city center

is set to the model of Helbing and Molnár [HM95], which is implemented by the "PedestrianMotion" module.

The second example shows the modeling of vehicular traffic. The mobility trace is saved in GlomoSim format. The trips are randomly modeled between the locations in file "points.txt." The movement dynamics is set to the intelligent driver model [THH00], which is implemented by the "IntelligentDriverMotion" module.

```
<?xml version="1.0"?>
<!-- Car Traffic in a City Center -->
<universe>
  <extension class="sim.extensions.GlomosimOutput"/>
  <extension class="sim.simulations.TimeSimulation" param="3600.0"/>
  <extension class="spatialmodel.core.SpatialModel"/>
  <extension class="gdfreader.GDFReader" source="Boston.gdf" min_x="4000"
max_x="6000" min_y="3000" max_y="4000"/>
  <extension name="PosGen"
class="tripmodel.generators.RandomInitialPositionGenerator">
    <points>points.txt</points>
  </extension>
  <extension name="TripGen" class="tripmodel.generators.RandomTripGenerator">
    <points>points.txt</points>
    <minstay>120.0</minstay> <maxstay>600.0</maxstay>
  </extension>
  <nodegroup n="100">
    <extension class="uomm.IntelligentDriverMotion" initposgenerator="PosGen"
tripgenerator="TripGen"/>
  </nodegroup>
</universe>
```

Figure 3.26: Scenario 2 – car traffic in a city

The framework is extensible. It is based on the concept of plug-ins (extension modules), so the new modules may be easily added, e.g., parsers for new geographical data standards, new user trip or movement dynamics models, or simulation tools support. The framework's source and binary files are freely available for download[1] and may be used by other researchers for modeling the user mobility in custom scenarios.

3.10 Summary

Most network simulation studies currently rely on rather simple mobility models. They neglect spatial environments, assume totally random trips in the area, or oversimplify a user movement dynamics. In this chapter, a more realistic user-oriented mobility model was presented. It reflects the main factors that impact a user behavior in real life, such as a spatial environment, user travel behavior, and user

[1] http://canu.informatik.uni-stuttgart.de/mobisim/index.html

movement dynamics. Hence, the model consists of three sub-models: the spatial model, the user trip model, and the movement dynamics model.

The spatial model contains a description of the movement area (spatial environment) and provides necessary information about area constraints and points of interest. The spatial model is automatically generated from the generic geographic model that is described in Chapter 2. The user trip model reflects user travel decisions. It performs the modeling of user trips and movement path selections. The user trips are modeled with the help of activity-based travel demand modeling approach [Kit96], which defines travel as the demand to participate in activities. The trip chain includes the activities performed by a user, transitions between them, possible locations and durations of activity executions. The switching between the activities is modeled by using non-deterministic finite state automata. The movement path selections are modeled with the help of shortest-path choice or probabilistic path choice approaches. The corresponding algorithms were described in this chapter along with their integration into the proposed mobility model. The movement dynamics model reflects the user position changes along the chosen path. It is based on the approaches from traffic modeling. The user-oriented mobility model includes different movement dynamics models to support the modeling of pedestrians and vehicles in diverse scenarios.

The model is implemented with all sub-models in a framework for user mobility modeling CANUMobiSim. It also includes the implementations of geographical data parsers for GDF and GML-based formats. The module of parameter derivation helps to obtain some trip model parameters from user position traces. The framework is freely available for download and may be used by other researchers to model the user mobility in different scenarios.

4 Physical Layer Modeling

Besides user mobility, the spatial environment also influences wireless communication between mobile devices. This chapter describes the integration of a more realistic radio propagation model into a simulation tool for mobile networks. This model is based on ray tracing and considers geographic data. It delivers the most accurate results in comparison to other models, which is also proven by measurements in indoor and outdoor environments.

This chapter is structured as follows: Section 4.1 introduces the subject and describes the basic terminology. Section 4.2 discusses related work in the area, which includes the common physical layer abstractions used in network simulation tools and radio propagation models. Section 4.3 describes the intelligent ray tracing model, which has major advantages over other approaches. In Section 4.4, the integration of the intelligent ray tracing model into a network simulator is presented. Section 4.5 gives implementation details. Section 4.6 describes further physical layer modeling improvements. Section 4.7 concludes this chapter.

4.1 Introduction

Let us start with motivation and then have a look at basic terminology.

4.1.1 Motivation

Obstacles of the spatial environment change signal propagation conditions from a clear line-of-sight to a non-line-of-sight. This causes additional signal attenuations and hardens frame reception. The obstacles also create multiple radio propagation paths, thus producing rapid fluctuations of the received signal. Due to these factors, the spatial environment must be definitely considered upon the modeling of mobile communication. However, reflecting the spatial environment is not a straightforward task. Indeed, due to its complexity, the modeling of a wireless channel has historically been one of the challenging parts of radio system design [RMK97].

Nevertheless the radio propagation models that are used in common simulators for mobile networks assume an obstacle-free area, and hence, a line-of-sight between all communicating partners. As a consequence, the distance between the transmitter and the receiver is the only dynamic parameter of these models. This poorly reflects radio wave propagation in a typical scenario, like in a city center, in which buildings significantly affect communication between nodes [SHR05], [SR06a]. Also, detailed statistics about the publications at top conferences in [KNG+04] proves that the papers with simple physical layer models outnumber others significantly.

This chapter describes the integration of a more realistic radio propagation model into a simulation tool for mobile networks. The model considers geographic data. In a combination with a fading model and the BER-based frame reception (which are already supported by some tools), it allows for more realistic simulations of mobile networks in indoor and outdoor scenarios.

4.1.2 Basic Terminology

In electrical engineering, *radio propagation models* describe signal spreading. Normally, they focus on predicting the received signal strength at a given distance from the transmitter, since the signal strength decreases with a distance. The propagation models that characterize signal strength over relatively long distances (several hundreds or thousands of wavelengths) are called *large-scale models*.

The main factors impacting the radio wave propagation are: reflection, diffraction, and scattering [RMK97]. *Reflections* of electromagnetic waves occur from objects whose dimensions are very large in comparison to wavelength, such as the earth surface, buildings, or walls. The reflected waves may interfere at the receiver. *Diffractions* of waves occur when the radio path is obstructed by an object with sharp edges. The waves then bend around the obstacle. *Scattering* takes place in the medium with many objects whose dimensions are comparable to the wavelength, such as street lamps, foliage, or dust.

These factors give rise to multiple radio propagation paths (*multipath propagation*). Wave interference between multipath components and the relative motion between the transmitter and the receiver cause rapid change in the received

signal strength. *Small-scale fading models* describe such signal fluctuations over the short distances (a few wavelengths) or the short time intervals (order of seconds).

The received signal is demodulated to obtain the frame bit data. Each frame contains a cyclic redundancy code (CRC) for error detection. If the CRC computed for the newly received frame does not match the original CRC in this frame, the receiver assumes that the frame is corrupted and asks the sender to resend it. The ratio of incorrectly received bits (which is obtained by comparing the CRCs) to the total number of bits is defined as a *bit-error ratio* (BER). It is closely connected to the *frame-error ratio* (FER), which corresponds to the ratio of incorrectly received frames:

$$FER = (1 - (1 - BER)^{l}) \qquad (4.1)$$

where l is a frame length. For Wireless LANs (WLANs), l=1024, since the FER is calculated for 1024 byte frames [IEE99].

4.2 Related Work

Related work in the area of wireless channel modeling includes several topics. First of all, it is the physical layer modeling in different simulation tools for mobile networks. Then these are namely the radio propagation models. For ultra high frequency (UHF) communication technologies, which are the main focus of this thesis, the radio propagation models can be classified into two major groups [WHBL00]: *empirical models* and *ray optical models*. This classification is also used in this thesis.

4.2.1 Physical Layer Modeling in Network Simulation Tools

The following three steps are commonly performed by network simulation tools upon the modeling of a packet transmission [MMB01]:

1. Estimation of the signal strength at each receiver
2. Computing interference with other signals
3. Deciding on the frame reception

Each time a mobile node transmits a frame, the simulators use a radio propagation model (either a large-scale model alone or in combination with fading) to compute the received signal power for every potential receiver. The result depends on attenuation that the signal experiences during propagation, e.g., due to distance or environment. The signals with a power below the carrier sense threshold (*CSThresh*) of the network card are considered too weak and are ignored by the receiver.

Next, interference with other signals is computed. Some tools like ns-2 [BEF+00] only compare the power of the received signal with the strengths of other signals. They assume that two simultaneously arriving frames collide if the ratio of their signals is below the collision threshold (*CPThresh*), which is a good approximation for many cases. Other tools like GlomoSim [ZBG98] take cumulative frame interference and the receiver background noise into account. They compute the so-called *signal-to-interference-and-noise ratio* (SINR):

$$SINR = \frac{P_r}{P_0 + \sum_{i \neq r} P_i} \qquad (4.2)$$

where P_r is the power of the received signal, $\sum_{i \neq r} P_i$ is the sum of powers of other signals at the receiver, and P_0 is the constant receiver noise floor. The SINR value is used at the next step to decide on a packet reception.

The implementation of the final step differs among simulators. Some tools like ns-2 compare the received signal power with the receive threshold of the network card (*RXThresh*). They assume that a frame is received successfully if the corresponding signal strength is above the threshold. The threshold is normally set to the "receiver input level minimum sensitivity," at which the ratio of corrupted frames is less than 8% (defined in the Section 15.4.8.1 of IEEE802.11 specification [IEE99]). Other tools, such as GlomoSim, do a similar comparison using the SINR value.

Some simulators (e.g., GlomoSim and OPNET[1]) also support a BER-based frame reception, which is more realistic than the simple threshold-based models. They use a

[1] OPNET is a product of OPNET Technologies, Inc., 7255 Woodmont Avenue, Bethesda, MD 20814, USA, http://www.opnet.com

ratio between SINR and BER, which is, for instance, based on measured statistics. Then the tools decide probabilistically on the reception of this frame.

4.2.2 Empirical Radio Propagation Models

Empirical models are described with formulas that provide estimations for the receive power based on the distance between the communication partners. The following empirical models are commonly used in simulations of mobile networks. They are also supported by nearly all simulation tools.

The *free space model* is the oldest model. It was proposed by Friis [Fri46]. The model assumes exactly one path between the transmitter and the receiver that is clear from obstacles. In the model, the signal power received by a receiver antenna is obtained with the formula:

$$P_r = P_t \frac{G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \qquad (4.3)$$

where:

P_r is the received signal power (in W).

P_t is the transmitted signal power (in W).

G_r and G_t are the gains of the receiving and the transmitting antennas, respectively. These are commonly assumed to be 1.0.

λ is the wave length (in m). It is a simulation parameter.

L is the system loss ($L \geq 1.0$). It describes the propagation environment. Usually, L=1.0.

d is the distance between the transmitter and the receiver.

This model does not consider the spatial environment. By assuming that mobile nodes receive the frame successfully if the corresponding signal strength is above the receive threshold of the network card ($P_r > RXThresh$), it is clear that the distance between the transmitter and the receiver is the only dynamic parameter of this model (P_t is normally fixed in a simulation). Hence, the simulation tools that follow this assumption only need to determine the maximum value of d in Equation 4.3, such that $P_r > RXThresh$ holds. Then they can decide on a frame reception by simply comparing d and d_{max}.

Figure 4.1 shows an example of communication area for a WLAN node. We see that although the transmitter is partially blocked by buildings, its maximum communication range is the same in all directions, which is described with a circle.

Figure 4.1: Communication area of a WLAN node computed with the free space model

According to [RMK97], only the direct path between the communicating partners exists rarely at longer distances. The *two-ray ground model* considers both the clear path and the ground reflected path (Figure 4.2). The received electric field E_r is the sum of the direct line-of-sight component E_{LOS} and a ground reflected component E_g. This results in the following equation for the received signal power:

$$P_r = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4 L} \qquad (4.4)$$

In addition to the parameters of the free space model, the equation contains h_r and h_t, which are the heights of receiving and the transmitting antennas (in m), respectively. Similar to the free space model, the model neglects obstacles of the propagation environment. The communication area is also a circle around the transmitter.

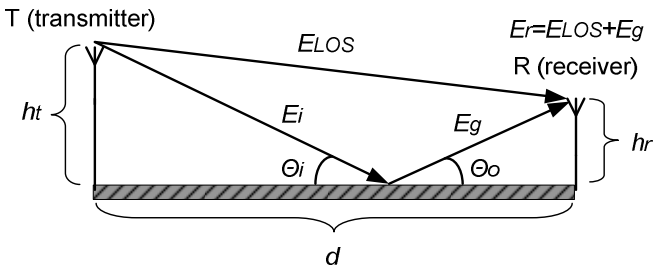


Figure 4.2: Two-ray ground radio propagation model (based on [RMK97])

However, the two-ray ground model is too optimistic for the short transmitter-receiver separation distances. Hence, in most applications, the two models are combined. The free space model is used at small distances, while the two-ray ground model is used at longer distances (Figure 4.3). The distance at which both models give identical results ($d_c=4\pi h_t h_r/\lambda$) is used as cross-over distance.

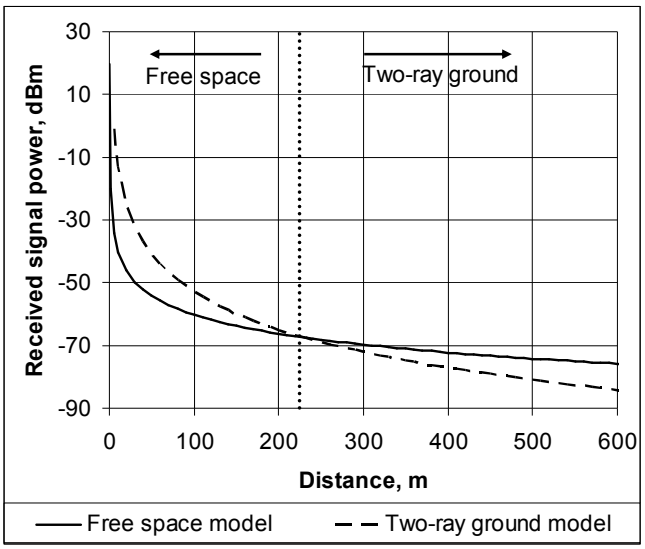


Figure 4.3: Combining free space and two-ray ground models

The described *combination of free space and two-ray ground models* is the most frequently used model in the MANET research community.

The *log-distance path-loss model* [RMK97] expresses the decrease of the received power with distance raised to some exponent:

$$P_r = P_{r0}\left(\frac{d_0}{d}\right)^{\beta} \qquad (4.5)$$

where: P_{r0} is the free space receive power at the close-in reference distance d_0, β is the path-loss exponent. Usually $d_0=1\ m$.

The exponent β depends on the propagation environment. For the open-space area like in the free space model, $\beta=2$. For urban areas with obstacles, β is between 2.5 and 3.5. Manufactures of wireless cards normally use $\beta=2.7$ to specify the maximum transmission range for a typical outdoor environment (cp. [Pro03]). In indoor environments, higher values up to 6 are used. With the help of this exponent β, the model shortens the maximum transmission range in order to approximate the radio propagation in obstructed areas. However, then the propagation in open areas is underestimated.

Empirical *COST-Walfisch-Ikegami model* [COST91] is an interesting alternative to the free space and two-ray ground models. It is a combination of Walfisch [WB88] and Ikegami [IYU84] models, which is proposed by the European COST-231 project[1]. The model considers certain characteristics of the spatial environment, namely (Figure 4.4):

- mean height of buildings h_{roof}
- mean width of streets w
- mean building separation distance b

The model distinguishes between line-of-sight and non-line-of-sight situations. If the path between the transmitter and the receiver is clear from obstacles, like in a street canyon, the following formula is applied:

$$L = 42.6 + 26\log\left(\frac{d}{\text{km}}\right) + 20\log\left(\frac{f}{\text{MHz}}\right) \qquad (4.6)$$

where:

[1] http://www.lx.it.pt/cost231/

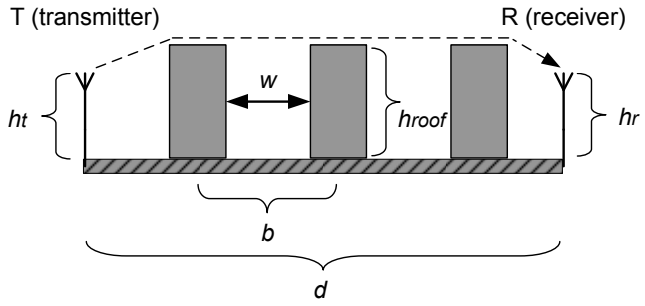


Figure 4.4: COST-Walfisch-Ikegami radio propagation model

L is the propagation loss in dB. It is basically a ratio in dB between the transmitted (P_t) and the received (P_r) signal powers, which are expressed in W: $L = 10\log\left(\frac{P_t}{P_r}\right)$.

d is the distance between the transmitter and the receiver (in km).

f is the signal frequency (in MHz).

The numeric constants are based on the measurements in Stockholm.

In the non-line-of-sight situation, the model assumes only the propagation over roofs with diffractions at the first and the last building (Figure 4.4). So the propagation loss is composed of the free-space path loss L_0, the roof-top-to-street diffraction and scatter loss L_{rts}, and the multiple screen diffraction loss L_{msd}:

$$L = \begin{cases} L_0 + L_{rts} + L_{msd} & \text{if } L_{rts} + L_{msd} > 0 \\ L_0 & \text{if } L_{rts} + L_{msd} \le 0 \end{cases} \qquad (4.7)$$

The free-space path loss is computed similar to the Equation 4.3 but is converted to dB domain:

$$L_0 = 32.4 + 20\log\left(\frac{d}{\text{km}}\right) + 20\log\left(\frac{f}{\text{MHz}}\right) \qquad (4.8)$$

The root-top-to-street diffraction and scatter loss L_{rts} (in dB) describes the coupling of the wave after its propagation over roofs into the street, where the receiver is located. Besides other terms, it takes into account the mean width of the street and its orientation (Figure 4.5):

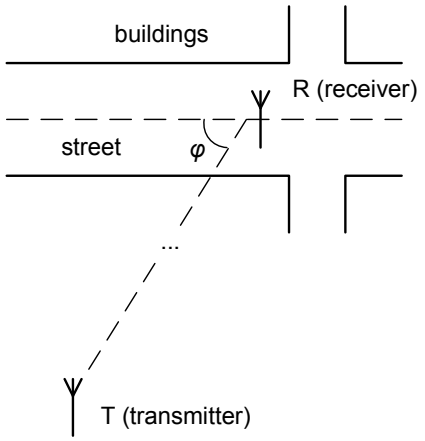


Figure 4.5: Definition of the street orientation angle φ

$$L_{rts} = -16.9 - 10\log\left(\frac{w}{\text{m}}\right) + 10\log\left(\frac{f}{\text{MHz}}\right) + 20\log\left(\frac{h_{roof} - h_r}{\text{m}}\right) + L_{ori} \qquad (4.9)$$

where:

w is the mean width of streets (in m).

h_r is the height of the receiver above the ground (in m).

h_{roof} is the mean height of buildings (in m).

L_{ori} is the street orientation loss (in dB). It is an empirical correlation factor depending on the street orientation angle φ. The street orientation loss is gained from measurements:

$$L_{ori} = \begin{cases} -10 + 0.354\dfrac{\varphi}{\text{deg}} & \text{for } 0° \le \varphi < 35° \\ 2.5 + 0.075\left(\dfrac{\varphi}{\text{deg}} - 35\right) & \text{for } 35° \le \varphi < 55° \\ 4.0 - 0.114\left(\dfrac{\varphi}{\text{deg}} - 55\right) & \text{for } 55° \le \varphi < 90° \end{cases} \qquad (4.10)$$

Multiple screen diffraction loss L_{msd} (in dB) mainly depends on the transmitter height h_t above the rooftops of the adjacent buildings h_{roof}. It is computed by using the following formula:

$$L_{msd} = L_{bsh} + k_a + k_d \log\left(\frac{d}{\text{km}}\right) + k_f \log\left(\frac{f}{\text{MHz}}\right) - 9\log\left(\frac{b}{\text{m}}\right) \qquad (4.11)$$

where:

d is the distance between the transmitter and the receiver (in km).

b is the mean building separation distance (in m).

L_{bsh} and k_a describe the increase in path loss for the transmitters that are located below the rooftops of buildings:

$$L_{bsh} = \begin{cases} -18\left(1 + \dfrac{h_t - h_{roof}}{\text{m}}\right) & \text{for } h_t > h_{roof} \\ 0 & \text{for } h_t \le h_{roof} \end{cases} \tag{4.12}$$

$$k_a = \begin{cases} 54 & \text{for } h_t > h_{roof} \\ 54 - 0.8\left(\dfrac{h_t - h_{roof}}{\text{m}}\right) & \text{for } d \ge 0.5 \text{ km and } h_t \le h_{roof} \\ 54 - 0.8\left(\dfrac{h_t - h_{roof}}{\text{m}}\right)\left(\dfrac{d/\text{km}}{0.5}\right) & \text{for } d < 0.5 \text{ km and } h_t \le h_{roof} \end{cases} \tag{4.13}$$

k_d and k_f control the dependence of the multiple screen diffraction loss versus the distance and frequency, respectively:

$$k_d = \begin{cases} 18 & \text{for } h_t > h_{roof} \\ 18 - 15\left(\dfrac{h_t - h_{roof}}{h_{roof}}\right) & \text{for } h_t \le h_{roof} \end{cases} \tag{4.14}$$

$$k_f = -4 + \begin{cases} 0.7\left(\dfrac{f/\text{MHz}}{925} - 1\right) & \text{for medium - sized cities and sub - urban centers} \\ 1.5\left(\dfrac{f/\text{MHz}}{925} - 1\right) & \text{for metropolitan centers} \end{cases} \tag{4.15}$$

Similar to the previous equations, the constants were obtained from measurements in European cities. This model is still a statistical model, so it mainly relies on characteristic values for the area and not on exact geographical map. The map is only required for distinguishing between the line-of-sight and non-line-of-sight propagations and for computing the street orientation angle between the transmitted wave and the street. The rest of the parameters, such as the street width, the building

height, and the distance between buildings, are assumed to be the same in the area. This introduces certain inaccuracies in prediction results. The COST-Walfisch-Ikegami model is not a standard model for existing network simulation tools and is available only as a separate extension. Gruber *et al.* [GKL04] integrate it into ns-2 and use it for simulating mobile ad-hoc networks in urban environments.

4.2.3 Ray Optical Models

The main advantage of the empirical models lies in their simplicity, and therefore, in their low computational complexity. However, these models do not take a detailed description of the spatial environment into account. Ray optical models [SDR92] rely on ray tracing or similar techniques from geometric optics. They consider a map of the propagation area to determine all possible signal paths between the transmitter and the receiver. The components of individual paths are summed to obtain the received signal power. Ray optical models provide more accurate results than empirical models, but require much longer computation time.

Two major groups of optical approaches exist: ray tracing and ray launching [GWBL95]. In ray launching, the rays are launched from the transmitter in all directions with a constant angular increment $\Delta\varphi$ (Figure 4.6). Each ray is traced individually. The prediction is done for the given transmitter position and all points of the area.

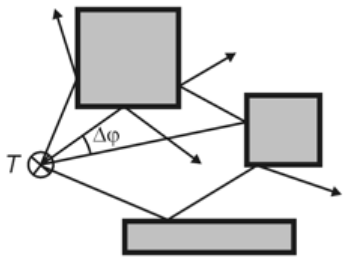


Figure 4.6: Schematic figure for ray launching (from www.awe-communications.com)

Ray tracing approaches use the image theory [DPR97], [For98] to find all possible paths between the given transmitter and receiver positions. The algorithms assume equal energy for the rays that have similar interactions (i.e. transmission, reflection, diffraction) with the same sequence of walls. Therefore, only one ray from the group needs to be traced (Figure 4.7). This makes the ray tracing more efficient than the ray launching, since the overall number of rays is much less. Another disadvantage of the ray launching is the constant angular increment, so the rays might miss some obstacles.

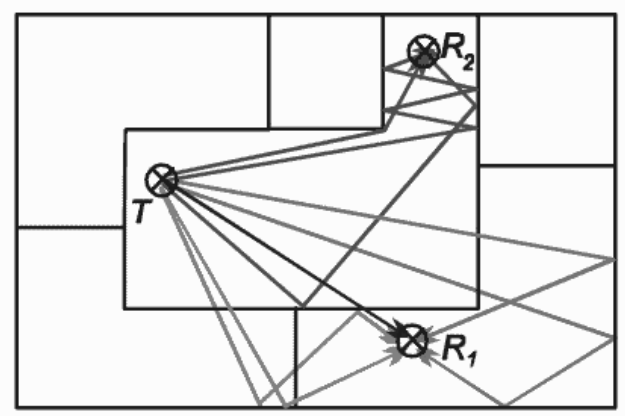


Figure 4.7: Major ray propagation paths between a transmitter and two receivers (from [HWL03])

Upon tracing a ray, its interactions with walls are detected. Ray transmissions (attenuated propagation through walls), reflections, and diffractions are considered (Figure 4.8). The maximum number of interactions for a single ray is a parameter of optical models, since there is an exponential dependency between the computational effort and the number of interactions. Good prediction results are achieved with the maximum of six interactions (transmission, reflections, and diffractions in different combinations with a maximum of two diffractions in each ray) [WHL99]. After more interactions, the ray power falls off rapidly. Another optimization used by the algorithms is to stop tracing a ray if its energy becomes below a certain minimum.

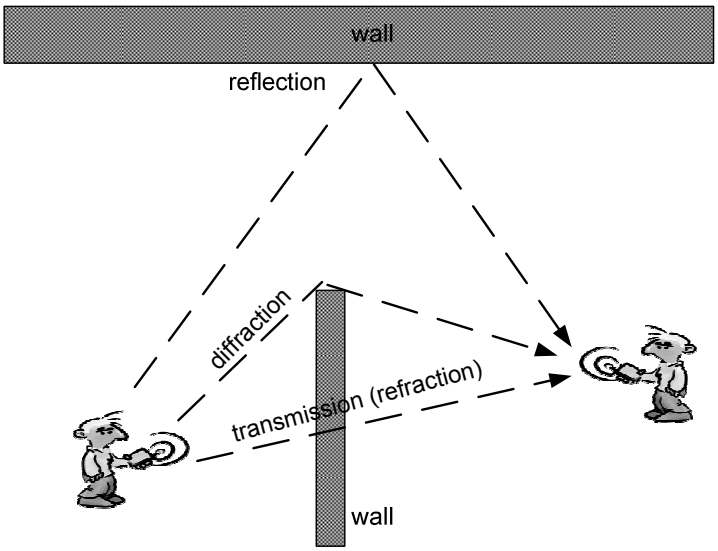


Figure 4.8: Ray interactions during the propagation

The power of a ray k after i transmissions, j reflections, and l diffractions is given by [For98]:

$$P_k = P_t \frac{G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \prod_i \tau_i \prod_j \rho_j \prod_l a_l \qquad (4.16)$$

where:

P_k is the ray power (in W).

P_t is the transmitted signal power (in W).

G_r and G_t are the gains of the receiving and the transmitting antennas, respectively.

λ is the wave length (in m).

L is the loss of propagation environment.

d is the distance traversed by the ray (in m).

τ_i is the i-th transmission loss.

ρ_j is the j-th reflection loss.

a_l is the l-th diffraction loss.

It is basically the Equation 4.3 of the free space model extended with the transmission, reflection, and diffraction loss coefficients. The values of these

coefficients are calculated from the material parameters by using either empirical model [LFR96], or Fresnel Equations [GJ99] together with the Geometrical Theory of Diffraction and the Uniform Theory of Diffraction (GTD/UTD) [LFR96]. In order to reflect the contribution of the ground reflected ray, the two-ray ground model is applied at long distances instead of the free space model:

$$P_k = P_t \frac{G_t G_r h_t^2 h_r^2}{d^4 L} \prod_i \tau_i \prod_j \rho_j \prod_l a_l \qquad (4.17)$$

where:

h_r is the height of the receiving antenna (in m).

h_t is the height of the transmitting antenna (in m).

The distance at which the free space and the two-ray ground models give identical results ($d_c = 4\pi h_t h_r / \lambda$) is used as cross-over distance for both formulas.

The contributions of individual rays are summed at a site to obtain the total received signal power:

$$P_r = \sum_k P_k \qquad (4.18)$$

Figure 4.9 shows a communication area for a WLAN node computed with the help of empirical two-ray ground model and a ray optical model. The transmitter is partially blocked by buildings. We see how the obstacles change a communication

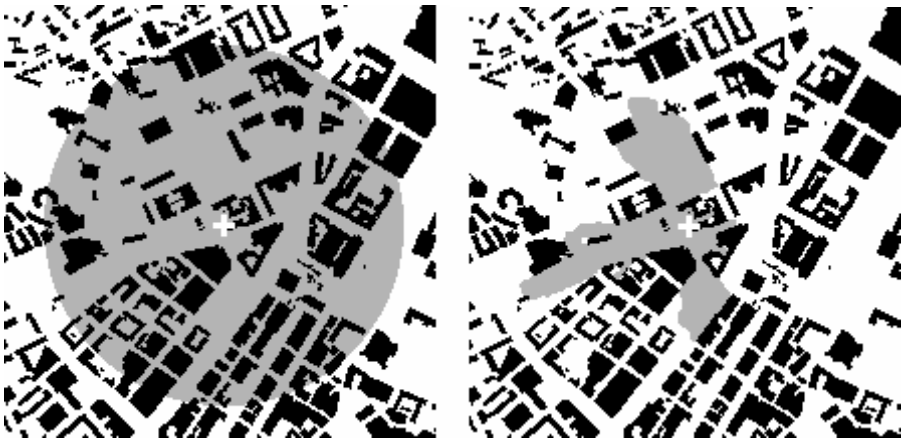


Figure 4.9: Communication area of a WLAN node computed with the two-ray ground model (on the left) and with the ray optical model (on the right)

area in the ray optical model. The result of the prediction done by the two-ray ground model is a circle of certain radius around the transmitter. In the line-of-sight situations, when a path is clear from obstacles, both models deliver similar results.

Although the ray optical models are precise, the accuracy of their predictions is heavily impacted by the accuracy of spatial data. Due to ray tracing, the optical models require much longer computation time than the empirical models. The execution time of optical models is also dependant on the number of obstacles in the area.

4.2.4 Discussion

Network simulation tools use radio propagation models for estimating the received signal power for every potential receiver. The obtained values are used for computing interference with other signals and for deciding on a frame reception.

Empirical models are described with mathematical expressions that predict the received signal power at certain distance from the transmitter. The free space model assumes a clear path between the transmitter and the receiver. It considers only a direct ray, which holds only for short separation distances between the communicating nodes. Two-ray ground model also considers a ground-reflected ray, thus making more accurate predictions at longer distances. In network simulations, these two models are normally combined. The transition point between the models is the cross-over distance, at which both the models deliver equal results. The distance between the transmitter and the receiver is the only dynamic parameter of these models. By assuming the threshold-based signal reception, the communication range can be represented by a perfect disc around the transmitter. All the nodes residing within this disc receive the signals from that transmitter. Due to their simplicity, these models are supported by nearly all simulation tools and are commonly used in network studies [KNG+04]. The models deliver excellent results in open areas, but become unrealistic in many scenarios, like a city center, in which buildings significantly affect mobile communication, or in indoor scenarios with walls.

The log-distance model uses a path-loss exponent to shorten the transmission range in obstructed areas, but then the propagation in open space is underestimated.

The communication range of a node in the log-distance model is also a circle, but just of a smaller radius.

Empirical COST model considers the over-rooftops propagation with diffractions at the first and the last buildings. It relies on certain properties of the spatial environment, such as the mean building height, the mean street width, and the mean building separation distance. However, the average values of these properties in the area are considered, which introduces certain inaccuracies in prediction results. A detailed geographical map is only used for distinguishing between the line-of-sight and non-line-of-sight propagations and for computing the street orientation angle between the transmitted wave and the street.

Ray optical models rely on ray tracing or similar techniques from geometric optics. They consider a map of the propagation area to determine all possible signal paths between the transmitter and the receiver. These models deliver very accurate results, however, require much longer computation time. Typical network simulation performs several millions of calls to a radio propagation model. Even if the prediction would take several seconds, which is still far away from the typical ray tracing performance [For98], [GWBL95], it makes the total simulation time simply unacceptable. In order to use a ray optical model in a network simulation, optimizations in the model and in the integration approach need to be performed.

4.3 Intelligent Ray Tracing Model

In this thesis, we use the *intelligent ray tracing model* [WGL97], [WHL99], which is developed by our colleagues at the Institute of Radio Frequency Technology, Universität Stuttgart. It is an improved version of the standard ray tracing approach. The model contains numerous optimizations and performs much faster than the standard approaches. To achieve such computation speed, the following optimizations are introduced.

4.3.1 Preprocessing Spatial Data

Complexity of the spatial data has significant impact on ray tracing performance, since ray interactions with all walls need to be analyzed. Therefore, the database is preprocessed to decrease the number of subsequent computations.

First of all, the complexity of spatial objects is reduced by simplifying their geometrical contours. An example is shown in Figure 4.10. The outer section of building A originally consists of 27 corner points. After simplification, it is reduced to 5 points, which cuts down the computation time significantly. The authors claim that although such simplifications might cause certain inaccuracies in predictions, in general, spatial databases for ray tracing are already in lower resolution. If a high-resolution spatial database is used, computation time increases dramatically due to the total number of walls and corners [WGL97].

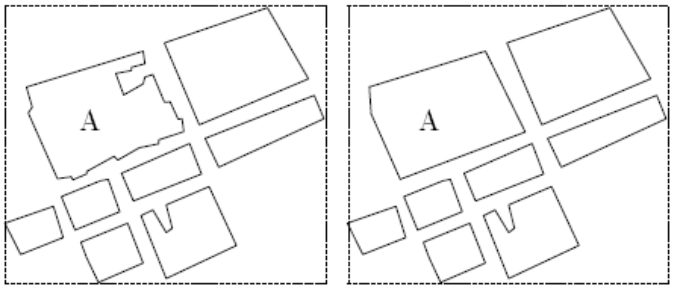


Figure 4.10: Reducing database complexity (from [WGL97])

One common operation in the ray tracing is computing the visibility relations between walls. They are independent of the positions of the transmitter and the receiver and stay constant (at least until the next substantial change in the spatial environment). Therefore, it is possible to accelerate the algorithm performance by computing the visibility relations between every two walls in advance and storing them in additional database [WHL99]. If there is a line-of-sight between the walls or their parts, both walls will be considered for searching a potential reflected ray. Precomputing the visibility relations takes additional time (a few minutes up to 1 or 2

hours depending on the complexity of spatial data), however, leads to the acceleration factor in the range of 2-10 [Woe00].

4.3.2 Ray Tracing Acceleration

The following optimizations are introduced in the ray tracing algorithm. There are rays that leave the transmitter or the receiver in the opposite direction of the transmitter-receiver line. Such rays have a very small contribution due to multiple reflections and diffractions and the longer paths. Thus, they can be neglected. Important rays are inside the Fresnel hexagon, which is shown in Figure 4.11. The hexagon is defined with three parameters: the opening angle α, the back region a, and the width w.

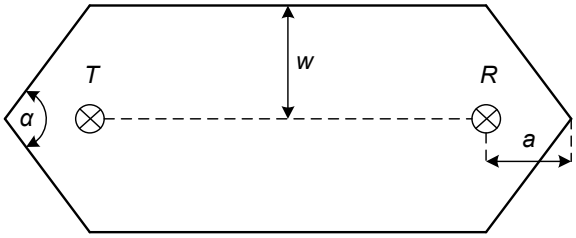


Figure 4.11: Definition of the Fresnel hexagon (based on [WGL97])

The smaller are the hexagon parameters, the higher is the acceleration factor. However, the prediction error also increases. The intelligent ray tracing model uses four predefined hexagons (Table 4.1). It switches between them automatically depending on the transmitter and the receiver positions. This leads to a high

Fresnel hexagon type	Back region a	Opening angle α	Width w
0	50 m	90°	50 m
1	100 m	90°	100 m
2	150 m	90°	150 m
3	90 m	180°	200 m

Table 4.1: Predefined hexagon parameters (based on [AWE03])

acceleration factor (≈10-15) and reduces error to less than 4 dB [WGL97].

The rays have different strength depending on the interactions they experience. Therefore, the algorithm classifies the propagation paths according to their strength (Table 4.2) and computes the rays in this order. If the contribution of the following class is very small comparing to the already accumulated field strength, the computation of further ray classes can be stopped. In practice, best results in terms of accuracy and acceleration are achieved by taking the contributions of the last three classes into account. Therefore, the intelligent ray tracing model stops determining further ray classes if the sum of the last three classes is smaller than 20% of the already accumulated field strength. This leads to an acceleration factor of 2-3 and very small errors (less than 2 dB) [WGL97].

Path class	Description
1	Direct path
2	Single reflection
3	Double reflection
4	Single diffraction
5	Triple reflection
6	One reflection, one diffraction
7	Double diffraction
8	Two reflections, one diffraction
9	Four arbitrary interactions
10	Five arbitrary interactions
11	Six arbitrary interactions

Table 4.2: Path classes (based on [AWE03])

4.3.3 Accuracy and Performance

The accuracy of the model is proven by measurements in European cities. The results are summarized in Table 4.3. Thus, for Helsinki (Finland), the mean error of predicted signal-loss results is close to 0 dB and the standard deviation is around 8 dB [RWH02]. For Munich (Germany) and Nancy (France), the mean error is 0 dB

Area	Mean error μ, dB	Standard deviation σ, dB
Helsinki	0	8
Munich	0	7
Nancy	0	4.9
Stuttgart	0.3	5.8
Office building	0.2	6.5

Table 4.3: Accuracy of path-loss predictions

and the standard deviation is less than 7 dB [WGL97], [WHL99]. For Stuttgart city center (Germany), which is used in the evaluation section of this thesis, the mean error is 0.3 dB and the standard deviation is 5.8 dB [HWL03]. For an office building of the Universität Stuttgart, the mean error is 0.2 dB and the standard deviation is 6.5 dB [WHL99].

The entire computation time for predicting path-loss at many points of the area are shown in Table 4.4 [HGL99]. They were obtained on an Intel© Pentium II, 266 MHz processor system. Depending on the area and its size, it takes between 2700 and 33500 seconds to obtain the results with the standard ray tracing approach. Because of the described optimizations, the intelligent ray tracing model reduces computation times up to 2-40 seconds. This corresponds to 0.06%-0.17% of the original model performance, or to the total acceleration factor of 600-1600.

Area		400 m x 400 m	600 m x 600 m	800 m x 800 m	1000 m x 1000 m
Nancy	standard ray tracing	2743 s (100%)	4607 s (100%)	11232 s (100%)	29548 s (100%)
	intelligent ray tracing	2 s (0.07%)	8 s (0.17%)	12 s (0.11%)	47 s (0.16%)
Stuttgart	standard ray tracing	3127 s (100%)	5134 s (100%)	13428 s (100%)	33541 s (100%)
	intelligent ray tracing	2 s (0.06%)	6 s (0.12%)	11 s (0.08%)	21 s (0.06%)

Table 4.4: Computation times (based on [HGL99])

Clearly, database preprocessing in the intelligent ray tracing model takes additional time. The results for some areas are shown in Table 4.5 [HGL99] (Intel© Pentium II, 266 MHz). Although the preprocessing requires considerable time, the overall overhead will be amortized upon doing multiple predictions in the area, for example, with different transmission power. The combined database preprocessing and prediction time for the intelligent ray tracing model are still 11%-37% of the time required by the standard ray tracing approach (Table 4.6).

Area	400 m x 400 m	600 m x 600 m	800 m x 800 m	1000 m x 1000 m
Nancy	360 s	1080 s	2520 s	6120 s
Stuttgart	360 s	1440 s	5040 s	11160 s

Table 4.5: Database preprocessing times for the intelligent ray tracing model (based on [HGL99])

Area		400 m x 400 m	600 m x 600 m	800 m x 800 m	1000 m x 1000 m
Nancy	standard ray tracing	2743 s (100%)	4607 s (100%)	11232 s (100%)	29548 s (100%)
	intelligent ray tracing + preprocessing	362 s (13.2%)	1088 s (23.62%)	2532 s (22.54%)	6167 s (20.87%)
Stuttgart	standard ray tracing	3127 s (100%)	5134 s (100%)	13428 s (100%)	33541 s (100%)
	intelligent ray tracing + preprocessing	362 s (11.58%)	1446 s (28.17%)	5051 s (37.62%)	11181 s (33.34%)

Table 4.6: Combined computation times

4.4 Integration of the Intelligent Ray Model into a Network Simulation

However, even with several seconds of computation time, the overhead introduced by the intelligent ray tracing model is still too high in order to use it directly in a network simulation. A typical simulation study, like the one in this thesis, contains

several millions requests for computing the received signal power between communicating nodes. This would result in weeks of computations with the intelligent ray tracing model.

The intelligent ray tracing model is implemented in commercial product WinProp [RWH02] from AWE Communications[1]. The tool supports area predictions, in which a map of receive power values is calculated for the given transmitter position (and other static parameters like sender height, transmission power, wavelength etc.) and a grid, representing all possible receiver positions. The algorithm used in WinProp does not allow obtaining the receive power value for just one receiver position efficiently. For computations, the area is divided into grid cells of equal size. Each cell represents possible transmitter or receiver position; the same receive power value is assumed over a cell. Cell size is an important parameter for such area predictions. Too large sizes cause high prediction inaccuracies. Too small cell sizes drastically increase the computation time and the amount of output data, since the result for each cell needs to be returned.

We use this advanced area prediction functionality of WinProp for integrating the intelligent ray tracing model into a network simulation. The basic idea is to precompute the receive power values for all possible transmitter-receiver pairs in a grid and to save them in a database. These values will be returned by the radio propagation model whenever a received signal power is requested for the given transmitter and receiver positions. This eliminates the need of ray tracing during the simulation and makes the simulation time comparable to the time with an empirical model. The architecture of the approach is shown in Figure 4.12.

As an input, WinProp requires building data (indoor and outdoor) in custom format. We get these data from our generic geographic model described in Chapter 2. Similar to the generic representation, WinProp requires individual indoor and outdoor objects. Their geometry is defined with 2-dimensional ground area, which is described by a polygon, and uniform height (the so-called 2.5-dimensional

[1] AWE Communications GmbH, Moltkestr. 28, Gaertringen, 71116 Germany, http://www.awe-communications.com

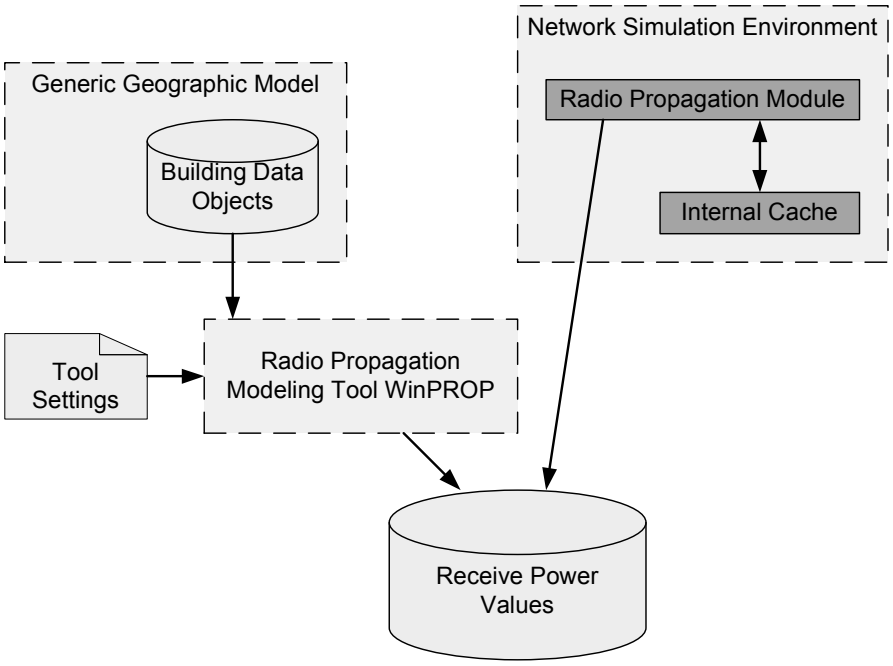


Figure 4.12: Integration of the intelligent ray tracing model into a network simulation environment

representation). To use own geographic data with the tool, we convert the object geometry and attributes (in particular, wall material properties) to WinProp format.

The building data used in this thesis was extracted from a digital map of Stuttgart city center. The whole area size is about 2.4 km × 1.9 km. The used geospatial data source does not provide building material properties, which are required for computing the rays during the ray tracing. Hence, we specify them externally based on typical Stuttgart values.

For computations, we use a 5 m × 5 m grid, which is the smallest grid size we could manage. Smaller grid sizes would require much longer computation time and more disk space for storing the receive power values, as shown in Table 4.7. For instance, the 5 m × 5 m grid translates to about 32 billion position pairs ($\left(\frac{2400}{5}\times\frac{1900}{5}\right)^2$). Since we store each power value as a 4-byte float, the total data size is about 120 GB ($\frac{32\times10^9\times4}{1024\times1024\times1024}$). By using the size 4 m × 4 m, we get

Cell size *d*, m	Number of position pairs	Database size, GB
1	20.793.600.000.000,00	77.462,20
2	1.299.600.000.000,00	4.841,39
3	256.711.111.111,11	956,32
4	81.225.000.000,00	302,59
5	33.269.760.000,00	123,94
6	16.044.444.444,44	59,77
7	8.660.391.503,54	32,26
8	5.076.562.500,00	18,91
9	3.169.272.976,68	11,81
10	2.079.360.000,00	7,75
11	1.420.230.858,55	5,29
12	1.002.777.777,78	3,74
13	728.041.735,23	2,71
14	541.274.468,97	2,02
15	410.737.777,78	1,53

Table 4.7: Database size for different grids

about 80 billions of position pairs and about 300 GB of output data, which was too much for storing it on our RAID.

Besides, WinProp requires the settings file, which contains the computation parameters, such as the sender position, transmitted signal power, communication frequency, transmitting and receiving antenna heights, and further antenna parameters. Most of them (except for the transmitted signal power) must be fixed. Changing one of them in simulation requires recomputing all receive power values for the new setting. The received signal power is proportional to the transmitted power, so we can easily adjust the received power result in our radio propagation module for a new transmission power value. Therefore, no database recomputation is required in this case.

The precalculation step took three days on a 50-node PC cluster. Once computed, the data are used till the next substantial change in the spatial environment (and in

the corresponding digital map) and a fundamental change in simulation parameters. We assume that those changes happen rather rarely. So the overhead for the precalculation will be typically amortized over many simulation runs. Currently, we need to recalculate the data from scratch in the case of any change in a spatial environment. In the future, it should be possible to perform a "smart update" by limiting the update area to the vicinity of change (similar as done by ns-2, which considers only the neighbors within *CSThresh* as potential signal receivers).

To decide on a frame reception, network simulation tools issue a call to the module that implements the required radio propagation model. The module computes the received signal power for the given parameters and the transmitter/receiver positions. The module for the intelligent ray tracing model simply gets the proper value from the database and returns it to the simulation.

Since wireless communication parameters are normally kept constant over a simulation, the tool would deliver identical results for the same transmitter and receiver positions. Hence, we also add a caching module, which stores the previously obtained results for repeating queries, thus accelerating the execution.

4.5 Implementation

The implementation of the described approach is straightforward. Conversion of the spatial data to urban database ASCII (UDA) format of WinProp is performed by using the "UDAWriter" extension module, which is a part of CANUMobiSim framework. This module gets building objects from the generic geographic model (these objects belong to GDF type "7110") and saves them in UDA.

Next, the UDA spatial data is preprocessed for use by the intelligent ray tracing model. The preprocessing includes simplification of geometrical contours of buildings and computation of visibility relations between every two walls for ray tracing acceleration. This is done by the WallMan tool, which is a part of WinProp software package. AWE Communications kindly performed this spatial database preprocessing for us.

We start WinProp on 50 cluster nodes with a script. This script logs-in automatically on each node and executes the required program. The computations are done for all transmitter positions in the grid. The map used in this thesis contains 182.400 transmitter and receiver positions in total, which is determined by the 2.4 km × 1.9 km area and 5 m grid $\left(\frac{2400}{5} \times \frac{1900}{5}\right)$. On one cluster node, the computations for 182.400/50=3648 positions are performed.

WinProp calculates the values for the given transmitter position and all area cells at once. The resulting received power values of type float are saved to a binary file. In our implementation, the files are named according to the transmitter's coordinate in the grid, e.g., *x0y0.raw*, *x10y25.raw*, etc. The files are placed in directories based on the transmitter's y-coordinate: *y0/x0y0.raw*, *y25/x10y25.raw*, and so on.

The radio propagation module "PrecomputedPropagationModule"[1] that we implement for the network simulator ns-2 uses this data for obtaining the received signal power based on the given transmitter and receiver positions. The coordinates are internally converted to cell indices. The total area size and a grid cell size are supplied as the module's parameters. Besides, the transmission power used in WinProp computations is provided, which is required for adjusting the receive power in the case that transmission power in a simulation does not match that value. The final module parameter is the maximum size of internal data cache. The cache stores the previously obtained results for the given transmitter and receiver positions, thus minimizing accesses to external data. The cache keeps track of the most- and the least recently used items for updating its contents.

4.6 Further Physical Layer Modeling Improvements

We use the network simulator ns-2 in the evaluation part of this thesis, which is the mostly used tool in the community. Unlike some other simulators, it does not model the small-scale fading (signal fluctuations over the short distances) and BER-based

[1] Available for download at: http://www.ipvs.uni-stuttgart.de?id=illya.stepanov&lang=en

frame reception. These are available as separate 3rd-party extensions, which we also use in our implementation. They are described in this section for the sake of completeness.

4.6.1 Small-Scale Fading Modeling

Small-scale fading stands for short-term fluctuations of the received signal strength. It is caused by the interference between multipath components that arrive at the receiver at slightly different times or while the user travels a short distance. These waves combine at the antenna to produce the signal, which can vary widely in amplitude and phase. When a direct path between the transmitter and the receiver is blocked by obstacles (the so-called *non-light-of-sight condition*), the envelope of the received signal typically has a Rayleigh distribution. The Ricean distribution is used to describe the line-of-sight situations [RMK97].

An approach to compute the small-scale fading is proposed in [PNS00]. It is also used in this thesis. Unlike other implementations, it considers the correlation of a signal envelope in time. This allows for more realistic simulations of burst errors. The efficient implementation is based on a table lookup, thus minimizing the computational requirements.

4.6.2 BER-based Frame Reception

To model a BER-based packet reception in MANET, we use the statistics of a WLAN chip manufacturer [Int00]. The data contain the bit-error ratios estimated for different signal-to-interference-and-noise ratios of the received signal and modulation schemes. WLANs use DBPSK (differential binary phase-shift keying) at the transmission speed of 1 Mbps, DQPSK (differential quadrature phase-shift keying) at 2 Mbps, CCK5.5 (complementary code keying) at 5.5 Mbps, and CCK11 at 11 Mbps [IEE99]. The data are available for both theoretical and practical cases (Figure 4.13).

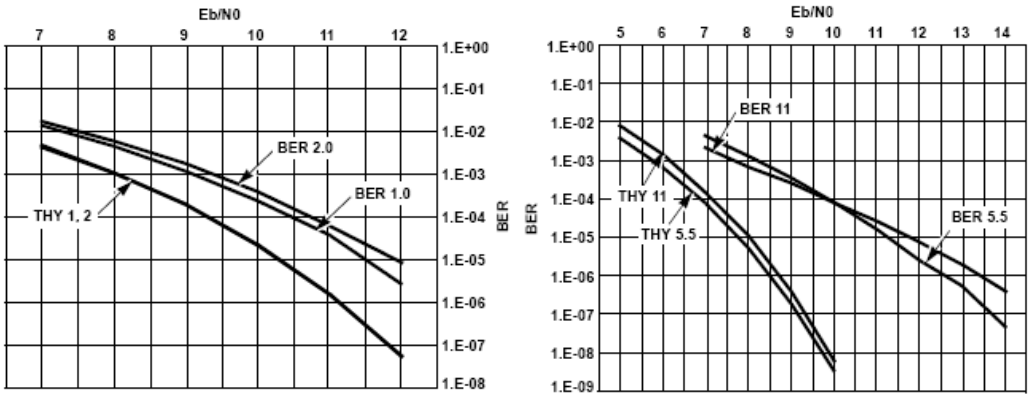


Figure 4.13: Dependency between signal-to-interference-and-noise ratio of the received signal (Eb/N0) and bit-error ratio (BER) at transmission speeds of 1, 2, 5.5, and 11 Mbps (from [Int00]) (THY denotes theoretical performance)

An implementation of this approach for ns-2 is described in [WA04]. It models errors upon the transmissions of IEEE 802.11 data and control frames. The algorithm determines a BER from the table using the transmission speed and the frame's signal-to-interference-and-noise ratio as indexes. Then it calculates a FER and decides probabilistically on the frame reception.

4.7 Summary

Network simulation tools use radio propagation models for modeling signal spreading across a distance from the transmitter. The models can be classified into two major groups: empirical models and ray optical models. Empirical models are commonly used in network simulation tools. They are described with mathematical expressions that predict the received signal power at certain distance from the transmitter. Such models, like the free space model or the two-ray ground model, normally assume a clear path between the transmitter and the receiver. Therefore, the distance between the transmitter and the receiver is the only dynamic parameter of these models. The communication range can be represented by a perfect disc around the transmitter. All the nodes residing within this disc receive the signals from that transmitter. These models deliver good results in open areas, but become unrealistic

in many scenarios, like a city center, in which buildings significantly affect mobile communication, or in indoor scenarios with walls.

Two empirical models take the spatial environment into account. The log-distance model uses a path-loss exponent to shorten the transmission range like in obstructed areas. However, this underestimates the open space propagation. The communication range of a node in the log-distance model is also a circle, but just of a smaller radius. Empirical COST model considers the over-rooftops propagation with diffractions at the first and the last buildings. It relies on certain properties of the spatial environment, such as the mean building height, the mean street width, and the mean building separation distance. However, the average values of these properties in the area are considered, which introduces certain inaccuracies in prediction results. A detailed geographical map is only used for distinguishing between the line-of-sight and non-line-of-sight propagations and for computing the street orientation angle between the transmitted wave and the street.

Ray optical models rely on ray tracing or similar techniques from geometric optics. They consider a map of the propagation area to determine all possible signal paths between the transmitter and the receiver. These models deliver very accurate results, however, require much longer computation time. In this chapter, we integrate the intelligent ray tracing model into network simulation. To speed-up the computations, the model contains many optimizations, like simplification of geometrical contours of spatial objects, precomputing visibility relations between all wall pairs, Fresnel hexagons, and optimal ray path selection. They make this model perform 600-1600 times faster than the standard ray tracing approach. The accuracy of the model is proven by measurements in European cities. For Stuttgart city center (Germany), which is used in the evaluation section of this thesis, the mean error is 0.3 dB and the standard deviation is 5.8 dB.

To integrate the intelligent ray tracing model into a simulation, we precompute the receive power values for all possible transmitter-receiver pairs in a grid and save them in a database. These values will be returned by the radio propagation model whenever a received signal power is requested for the given transmitter and receiver positions. This eliminates the need of ray tracing during the simulation and makes

the simulation time comparable to the time with an empirical model. Together with the small-scale fading model and a BER-based frame reception, which were also described in this chapter, it allows for more realistic simulation studies of mobile networks in indoor and outdoor scenarios.

5 Evaluation

In previous chapters, more realistic mobility and physical layer models were described. They reflect a spatial environment, user travel decisions, and the dynamics of user movement. In this chapter, we compare the simulation results obtained with those models and with the simpler models, which are default models in network simulation tools. The goal of this evaluation is to see, whether and how the application of more realistic models changes simulation results. We also analyze the factors that cause those changes. This chapter also demonstrates the practical use of the described models for simulating mobile networks under more realistic conditions.

This chapter is structured as follows: Section 5.1 describes a simulation scenario, which is used in our evaluations. Section 5.2 compares the network topologies. In Section 5.3, a routing protocol performance is evaluated. Section 5.4 compares the impact on the performance of mobile application. Section 5.5 discusses the obtained results and concludes this chapter.

5.1 Simulation Scenario

Our simulations are performed by using ns-2 [BEF+00]. It is the most frequently used simulator in MANET community. In evaluation, we use a typical scenario for mobile ad-hoc networks (Table 5.1). The network is formed by pedestrians that move in a certain area. We consider the Stuttgart downtown of 1.5 km × 1.5 km. Pedestrians travel between different “points of interest” like shops, restaurants, and museums. We compare this scenario with two user mobility models: the random waypoint mobility model (RW) and the user-oriented mobility model (UO). The RW model was chosen since it is commonly used in network simulations and is the default model in ns-2.

In RW simulations, mobile users move between the randomly chosen points of the area. The UO model relies on a digital map of Stuttgart (Figure 5.1), which is read from a GDF data source. This map also provides the points of interest that are used in

UO simulations. User trips between those points are randomly generated by CANUMobiSim, since we currently do not have access to the data describing real human behavior in this area. However, the generated paths reflect the spatial constraints of the area. We assume that mobile users always choose the shortest paths on the street network graph. In both mobility models, upon arriving to the destination, the users stay in a point of interest for a time between 10 and 15 minutes. Then the movement to the next trip point is initiated.

Figure 5.1: Map of the simulation area

The movement speeds correspond to typical pedestrian speeds in city environments, which are between v_{min}=0.56 and v_{max}=1.74 m/s [HM95]. The random waypoint model chooses the speeds randomly, so the speeds are uniformly distributed within the given range. The user-oriented model relies on a more realistic approach of Helbing and Molnár [HM95], which is described in Section 3.7.1. The pedestrian speeds follow the Gaussian distribution with the mean μ=1.34 m/s and standard deviation σ=0.26 m/s [Wei93].

The users are equipped with mobile devices, like user-carried PDAs. The devices communicate wirelessly by using IEEE 802.11. ns-2 only models the communication at a fixed transmission speed, so the auto-fallback between speeds based on the

receive signal strength is not supported. Therefore, the four transmission speeds used in IEEE 802.11 (1, 2, 5.5, and 11 Mbps) are considered in separate evaluations. The basic hardware parameters (transmission power and signal receive threshold) are taken from the manufacturer's datasheets [Pro03]. The carrier-sense threshold and the collision threshold, which are introduced in Section 4.2.1, are not directly specified by hardware manufacturers. They are set to the values suggested in [WA04].

We compare the following physical layer models: a combination of free space and two-ray ground models (TRG), the log-distance path-loss model with β=2.7 (LG2.7), and the described intelligent ray tracing model with the small-scale fading and the bit-error ratio-based packet reception (IRT). The TRG model is the most frequently used model in the MANET research community. The LG2.7 model is used by the manufacturers of network equipment for specifying a typical transmission range in outdoor environments. These two models are directly supported by ns-2 without installing additional extensions or applying software patches. In IRT simulations, the receive threshold is set to the carrier-sense threshold, so the frame reception is only determined by the bit-error ratio.

In comparisons, we analyze the impact of the mentioned mobility and physical layer models on network topologies, routing protocol performance, and on a performance of mobile applications. As a routing protocol, we use Ad-hoc On-demand Distance Vector protocol (AODV) [PR99], which combines the strengths of two other popular routing protocols for mobile ad-hoc networks, namely: classic table-driven routing scheme of Destination-Sequenced Distance-Vector Routing (DSDV) [PB94] and on-demand route discovery of Dynamic Source Routing (DSR) [JM96].

For comparing a routing protocol performance, we generate constant-bit rate UDP traffic among some network users (5 connections between 8 users). Packet transmissions start at different times (between 5 and 150 s); variable number of packets (between 1 and 10) are sent per second. The presented results are average over 20 simulation runs with different mobility patterns, which are independently generated for each mobility model.

Parameter	**Value**
Simulation time	900 s
Movement area size	1500 m × 1500 m
Number of mobile users	50, 75, 100, and 150
Movement speed	0.56 - 1.74 m/s
Mobility models	Random waypoint mobility model (RW) User-oriented mobility model (UO)
Number of generated mobility patterns (simulation runs)	20
Radio propagation models	Combination of free space and two-ray ground models (TRG) Log-distance path-loss model with $\beta=2.7$ (LG2.7) Intelligent ray tracing model with small-scale fading and BER packet reception (IRT)
Transmission power (P_t)	15 dBm
Radio frequency	2.442 GHz
Transmission speed (bit rate)	1, 2, 5.5, and 11 Mbps
Receive threshold (*RXThresh*)	-94, -91, -87, -82 dBm (depending on a transmission speed) -104 dBm in the simulations with the IRT model
Carrier-sense threshold (*CSThresh*)	-104 dBm
Collision threshold (*CPThresh*)	10 dB
System loss (L)	1
Antenna type	Omnidirectional
Antenna gain (G)	1
Antenna height (h)	1.5 m
Routing protocol	AODV
Data traffic	Constant-bit rate UDP traffic
Packet size	512 bytes
Data traffic intensity	Variable (between 1 and 10 packets/s)

Table 5.1: Simulation parameters

5.2 Comparison of Network Topologies

First, we analyze the impact of more realistic models on MANET connectivity. To monitor topology changes in time, we use own event-based topology simulation tool. The tool determines for every pair of mobile nodes times when they enter or leave each other's transmission range (the small scale fading and transmission errors are not considered in this section). Thereby MANET topology graphs are constructed at different time steps.

5.2.1 Metrics

We define the following metrics for comparing the topologies. They are applied to a directed MANET graph $G:=(V,E)$, in which V represents a set of mobile nodes and E contains possible connections between them. An edge e_{ij} from node i to node j exists if j can receive transmissions from i with the power above the signal receive threshold. It is assumed that the edge e_{ii} exists always, so a node can hear own transmissions (i.e. over loopback network interface).

Edge density β in the MANET topology graph is a fraction of graph edges $|E|$ to the number of edges in a complete (fully connected) graph $|E_0|$:

$$\beta = \frac{|E|}{|E_0|} \tag{5.1}$$

In the *complete* graph, all the nodes are interconnected, which basically means that they reside in the communication range of each other. Consequently, this metric reflects how close the network graph to a complete graph is (or how densely the nodes are interconnected). Like other topology comparison metrics, it also helps explaining routing protocol and application performance results in the next sections. Since e_{ii} exists always, this metric takes its values in:

$$\beta \in \left[\frac{1}{|V|}, 1 \right] \tag{5.2}$$

where $|V|$ is the total number of mobile nodes in the scenario.

The lowest value ($|V|^{-1}$) indicates that there are no interconnections between mobile nodes in the graph. The value of β increases as more nodes reside in the

communication range of each other. The value of 1 indicates a fully connected (complete) graph.

Degree of dissimilarity σ expresses a difference between two topology graphs $G_1:=(V_1,E_1)$ and $G_2:=(V_2,E_2)$ that have the same set of vertices ($V_1=V_2$):

$$\sigma = \frac{|E_1 - E_2| + |E_2 - E_1|}{|E_1| + |E_2|} \tag{5.3}$$

It is basically a fraction of the edges differing in both graphs to the sum of edges in both graphs. This metric describes how different two topologies are and how many distinct edges do they have. Since e_{ii} exists always, this metric takes its values in:

$$\sigma \in \left[0, \frac{(|V|-1)|V|}{|V|^2 + |V|}\right] \tag{5.4}$$

The value of 0 indicates that both topology graphs have the same edges. The value of σ increases as more distinct edges exist in both topologies. This metric takes its highest value in the case when two graphs have only e_{ii} edges in common.

Ratio of unique edges ρ in a topology is also related to the difference between two topology graphs $G_1:=(V_1,E_1)$ and $G_2:=(V_2,E_2)$ with the same set of vertices ($V_1=V_2$):

$$\rho = \frac{|E_1 - E_2|}{|E_1|} \tag{5.5}$$

It stands for a fraction of edges existing only in this topology to the total number of edges in the topology. It basically describes the ratio of unique edges in this topology. The metric takes its values in:

$$\rho \in \left[0, \frac{(|V|-1)|V|}{|V|^2}\right] \tag{5.6}$$

The value of 0 indicates that $E_1 \subseteq E_2$. The value of ρ increases as more unique edges exist in E_1. This metric takes its highest value in the case when G_1 is a fully connected graph and G_2 only contains e_{ii}.

Using the degree of dissimilarity and the ratio of unique edges in topology comparisons only makes sense under the same mobility model. We cannot use them for comparing UO and RW. Due to the random generation, mobile nodes are located in completely different places. This makes the topologies be completely different

upon comparing them on edge-by-edge basis. Therefore, we use these metrics for analyzing the impact of physical layer models under the same mobility pattern (UO).

The next two metrics are related to network partitioning. *Number of partitions n* corresponds to the number of isolated groups of nodes in the topology. Since the groups are isolated, packet forwarding is only possible among the nodes belonging to the same group. Thus, this metric helps explaining bad network performance results. The metric takes its values in:

$$n \in \left[1, |V|\right] \tag{5.7}$$

The value of 1 corresponds to the case when all nodes reside in the same partition. $|V|$ indicates that all mobile nodes are isolated, so none of them resides in the communication range of each other (and only edges e_{ii} are present).

Ratio of nodes in the largest partition m is a fraction of nodes residing in the largest partition to the total number of mobile nodes in the scenario. This metric also helps analyzing the network partitioning and explaining network performance results. The metric takes its values in:

$$m \in \left[\frac{1}{|V|}, 1\right] \tag{5.8}$$

The lowest value indicates that all mobile nodes are isolated and none of them resides in the communication range of each other. The value of 1 corresponds to the case when all nodes reside in the same partition.

In the charts, the *weighted arithmetic mean* of results is presented. So, the individual results are aggregated by taking a relevance of each component into account, which is the duration of time that the network connectivity graph was in the particular state.

5.2.2 Impact of Mobility Models

First, we analyze the impact of the more realistic mobility model. We compare RW and UO mobility models with the same empirical radio propagation model (TRG or LG2.7, which are standard models in ns-2).

From the results of edge density β (Figure 5.2), we see that β(TRG+UO) > β(TRG+RW) and β(LG2.7+UO) > β(LG2.7+RW) for all transmission

speeds. This means that mobile nodes are closer located to each other in UO than in RW. This increases the number of possible network connections in UO. The reason for this phenomenon is that mobile nodes are almost evenly distributed in the area in RW model. In contrast, the street network graph limits possible locations of users in UO, thus changing the node distribution.

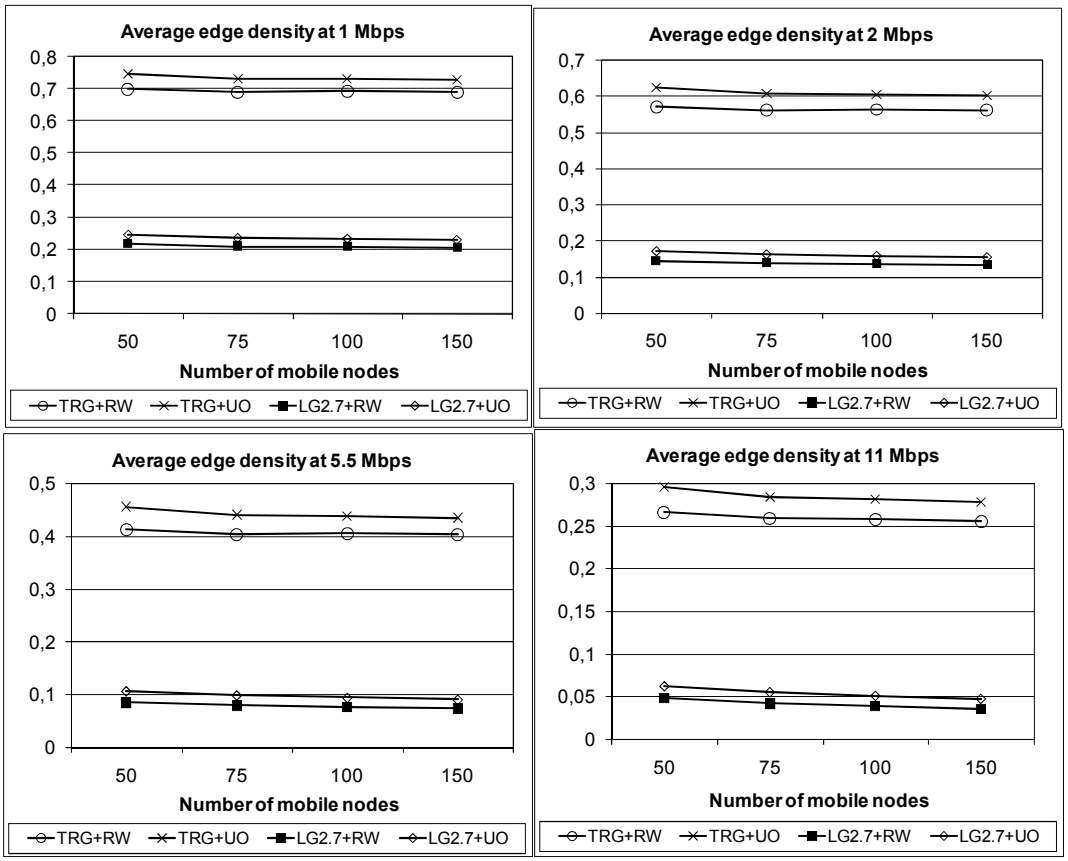


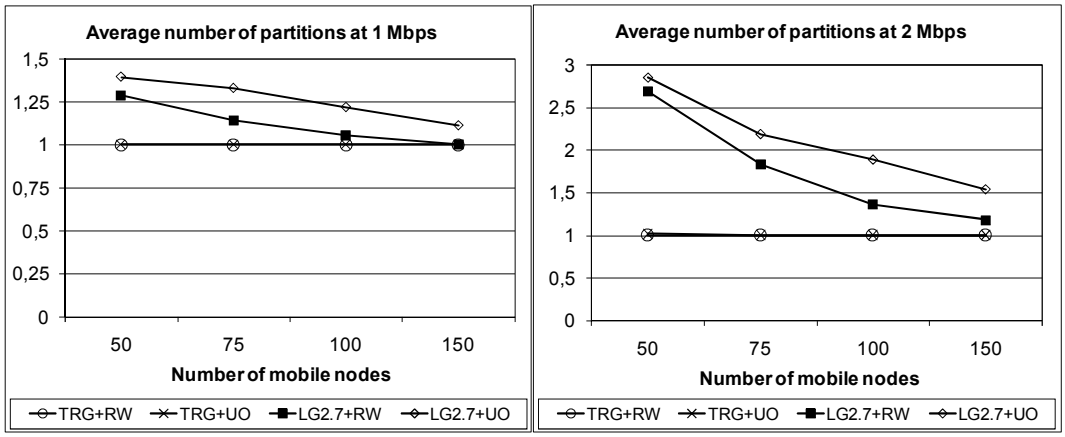
Figure 5.2: Average edge density

Higher IEEE802.11 transmission speeds are more sensible to the power of the received signal. At higher speeds, network devices use higher signal receive thresholds, and thus, have a shorter maximum transmission range (Table 5.2). Consequently, the edge density decreases, and therefore, we get a smaller number of network connections for the same combination of mobility and radio propagation models.

Radio propagation model	Transmission speed, Mbps	Maximum transmission range, m
TRG	1	796 m
TRG	2	670 m
TRG	5.5	532 m
TRG	11	400 m
LG2.7	1	353 m
LG2.7	2	274 m
LG2.7	5.5	195 m
LG2.7	11	127 m

Table 5.2: Maximum transmission range for empirical models at different IEEE802.11 transmission speeds

The results for the number of partitions and for the ratio of nodes in the largest partition are presented in Figure 5.3 and in Figure 5.4 respectively. Transmission range of TRG is that long that mobile nodes reside in the same partition nearly all the time disregards the mobility model. There is no network partitioning in TRG, except for some rare cases lasting just a few seconds (because of them, particular values in charts are not exactly 1.0 but rather ≈0.998). Consequently, the number of partitions and the ratio of nodes in the largest partition are close to 1.0 for both TRG+RW and TRG+UO at all transmission speeds.



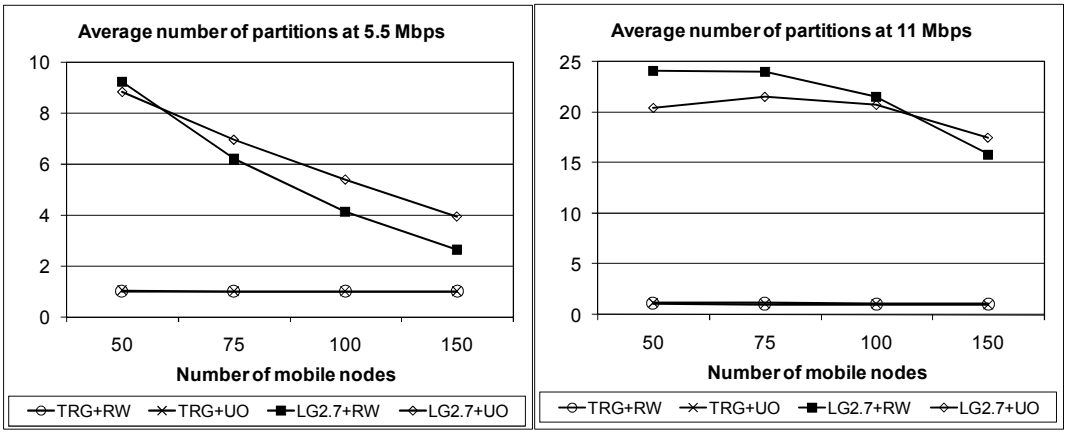


Figure 5.3: Average number of partitions

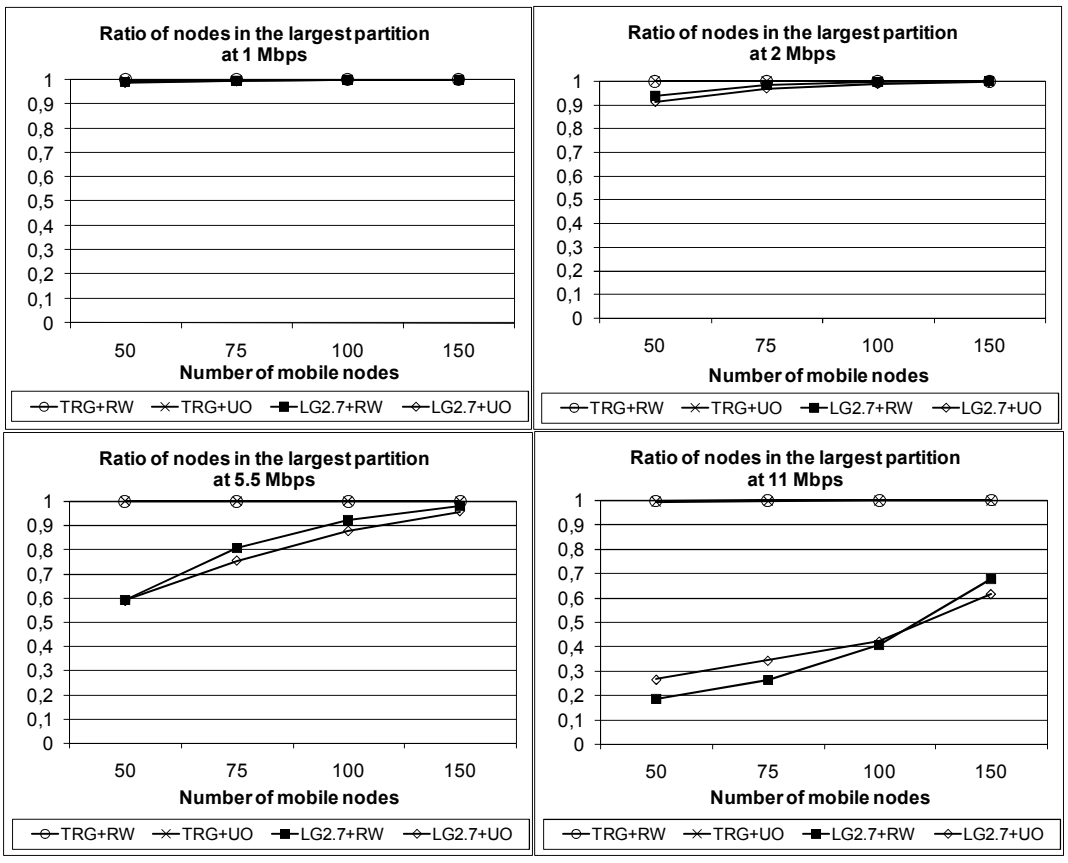


Figure 5.4: Ratio of mobile nodes in the largest partition

The LG2.7 has a shorter transmission range, so we can observe some differences in partitioning between RW and UO models. The nodes are more evenly distributed

in the area in the RW model. Therefore, we have fewer network partitions and more nodes in the largest partition with LG2.7+RW at 1, 2, 5.5 (75, 100, and 150 nodes), and 11 (150 nodes) Mbps. Higher transmission speeds (5.5 and 11 Mbps) have shorter transmission ranges. Since the nodes are evenly distributed in the area in RW model, more nodes are required to connect the parts of the network. Therefore, we have more partitions and fewer nodes in the largest partition in LG2.7+RW with 50 nodes at 5.5 Mbps and 50, 75, 100 nodes at 11 Mbps. There are basically not enough nodes to connect the groups that reside in different parts of the area. The LG2.7+UO shows better results for those speeds and node densities because the street network graph makes mobile nodes be located closer to each other. As the number of nodes increases, UO tends to be more partitioned than RW, since the node distribution in this model depends on the locations of points of interest. In RW, the nodes are more evenly distributed in the area, therefore, network connectivity gains more from the increase in the number of users. For both mobility models holds that with the increase of mobile nodes in a simulation, we get fewer network partitions and more nodes in the largest partition.

To sum up, the application of a more realistic mobility model changes network topologies. The spatial environment (street network graph) limits possible locations of nodes. Therefore, mobile nodes are closer located to each other in UO. The values of edge density in UO are higher, which basically means there are more networks connections in the topology graph. In RW, mobile nodes are more evenly distributed in the area. Therefore, RW topology graphs have fewer network connections. However, the more even node distribution in RW reduces a chance of network partitioning, except for high transmission speeds (5.5 and 11 Mbps) with smaller numbers of mobile nodes (50 nodes at 5.5 Mbps; 50, 75, 100 nodes at 11 Mbps.). These cases profit more from the constrained movement area in UO.

5.2.3 Impact of Physical Layer Models

For analyzing the impact of the more realistic physical layer model on network topologies, we compare the results of TRG, LG2.7, and IRT models with the same mobility model (UO).

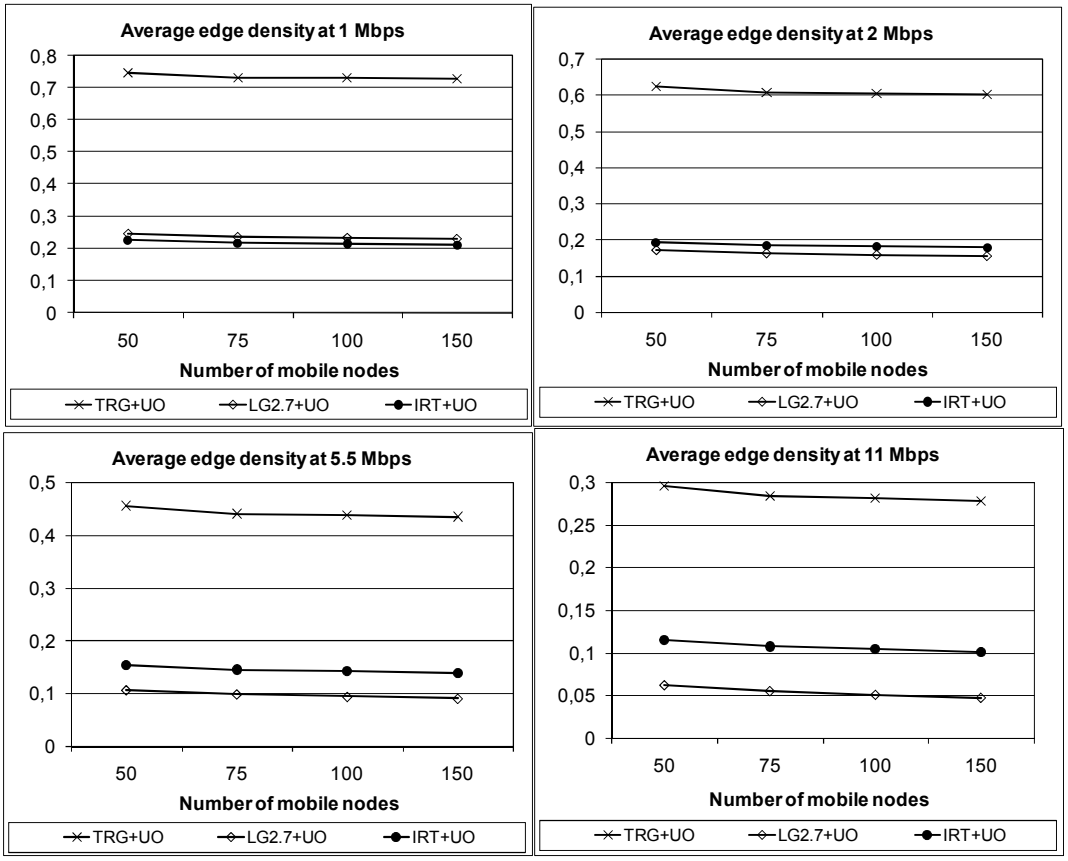


Figure 5.5: Average edge density

The results for edge density β are shown in Figure 5.5. We see that at 1Mbps, β(TRG+UO) > β(LG2.7+UO) > β(IRT+UO). At higher transmission speeds, β(TRG+UO) > β(IRT+UO) > β(LG2.7+UO). The difference between TRG and LG2.7 is due to the fact that LG2.7 reduces the maximum transmission range by the factor of 2.7. This allows the model approximating radio propagation in obstructed areas. However, this obviously decreases the number of possible network connections. To explain the results for IRT, let us look at two additional metrics: degree of dissimilarity and the ratio of unique edges.

The results for degree of dissimilarity σ are shown in Figure 5.6. In charts, the IRT topology is compared against TRG and LG2.7. We see that σ(IRT, TRG) > σ(IRT, LG2.7) at all transmission speeds. For TRG, the values of σ(IRT, TRG) decrease with a speed from ≈0.55 at 1 Mbps to ≈0.45 at 11 Mbps. LG2.7 demonstrates different behavior, so σ(IRT, LG2.7) increases from ≈0.3 at 1 Mbps to

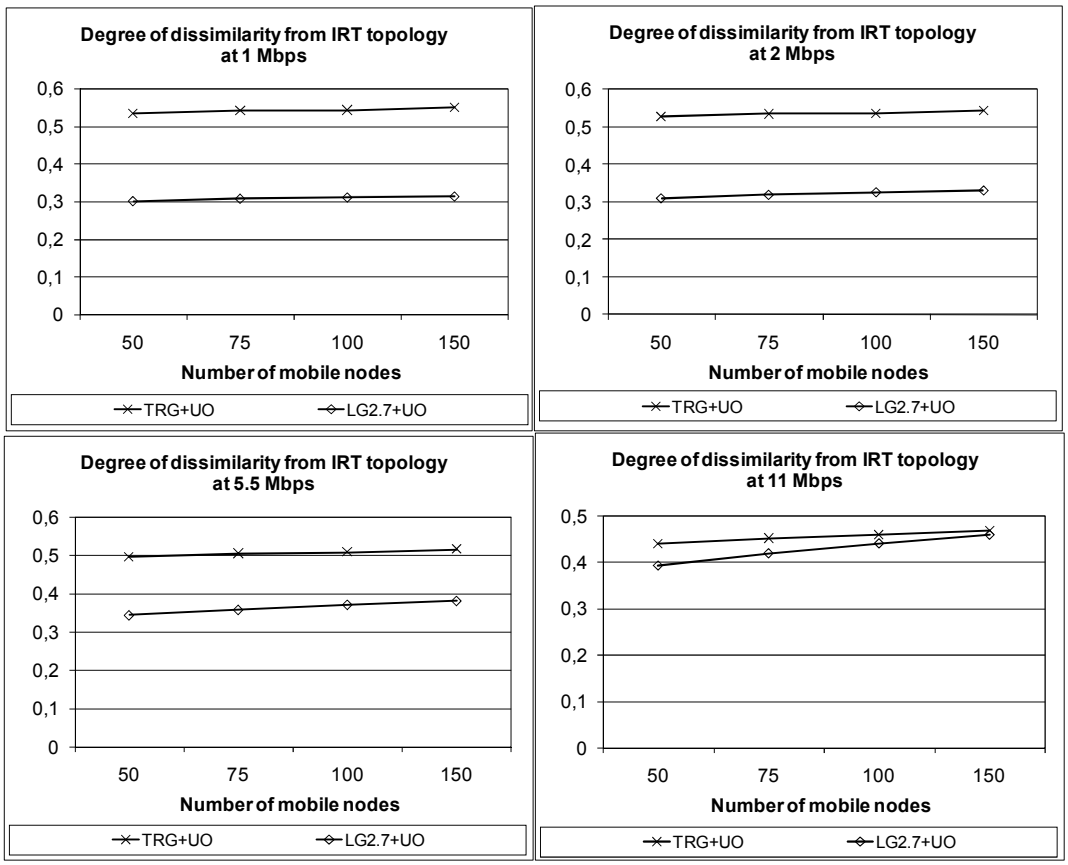


Figure 5.6: Degree of dissimilarity between the IRT, TRG, and LG2.7 topologies

≈0.43 at 11 Mbps. The ratios of unique edges, which are shown in Figure 5.7 and Figure 5.8, help us clarifying the following reasons for that.

From the comparison with TRG (Figure 5.7), we see that the ratio of unique edges in IRT is 0 for all transmission speeds. It basically means that all the differences between IRT and TRG topologies are caused by the edges that exist only in TRG (Figure 5.8). IRT and TRG have the same transmission range in open areas, and consequently, the same edges. Unlike TRG, IRT also considers obstacles of the spatial environment. The obstacles block certain network connections in IRT, thus changing the network topologies. At the same time, those edges exist in TRG because this model does not consider the spatial environment.

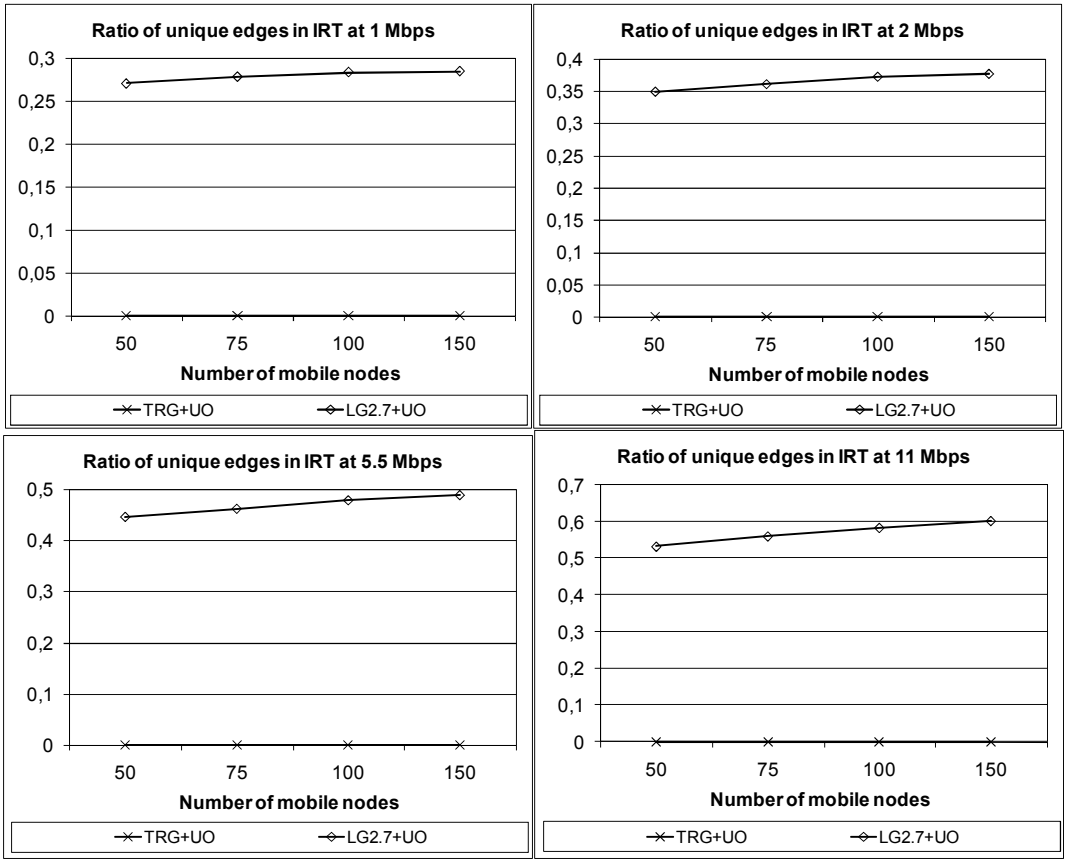


Figure 5.7: Ratio of unique edges in the IRT compared to TRG and to LG2.7 topologies

The comparison between IRT and LG2.7 show nearly the same amount of unique edges in both models at 1 Mbps (Figure 5.7 and Figure 5.8). Therefore, we can conclude that the differences between IRT and LG2.7 come from both models. On one hand, LG2.7 shortens the maximum transmission range in order to approximate the propagation in obstructed areas. However, then the propagation in open areas is underestimated. Consequently, unique edges appear in IRT, which is because of the shorter transmission range of LG2.7 in open areas. On the other hand, LG2.7 does not consider a detailed map of the propagation environment. The model only reduces the transmission range, which is still a circle around the sender. Therefore, certain edges that are blocked in IRT by the obstacles still exist in LG2.7. These unique edges of LG2.7 are also counted in the topology difference results.

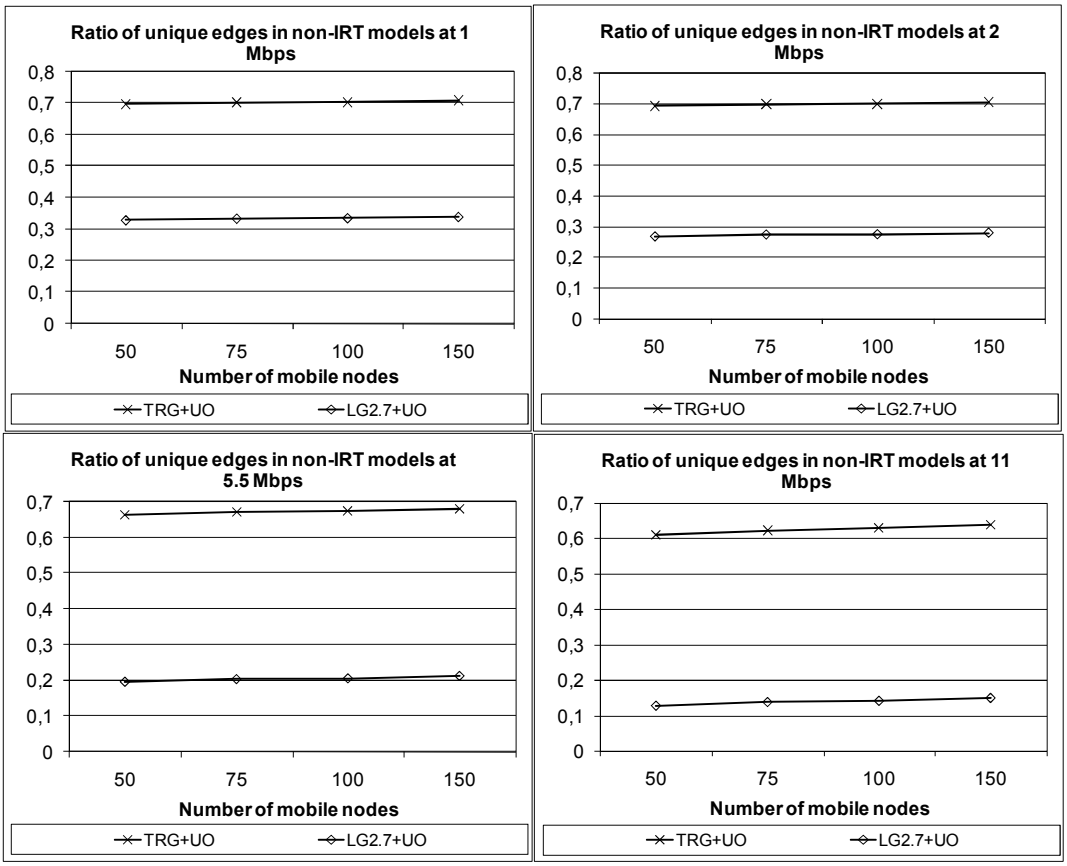


Figure 5.8: Ratio of unique edges in non-IRT models (TRG, LG2.7) compared to IRT

At higher transmission speeds, we get more and more unique edges in IRT, which is due to a shorter transmission range of LG2.7 in open areas. The degree of dissimilarity between IRT and LG2.7 also increases, which indicates that the topologies have more dissimilar edges. This also explains why IRT has a higher edge density than LG2.7 at 2, 5.5, and 11 Mbps.

Like in the comparisons of mobility models, the values of edge density decrease for the same radio propagation model at higher speeds. The transmission range is shorter at higher speeds. Consequently, the number of edges in topology graphs also decreases.

The results of network partitioning for TRG, LG2.7, and IRT are shown in Figure 5.9 and in Figure 5.10. We see that due to its relatively long transmission range, TRG topologies are almost never partitioned. The average number of partitions and the

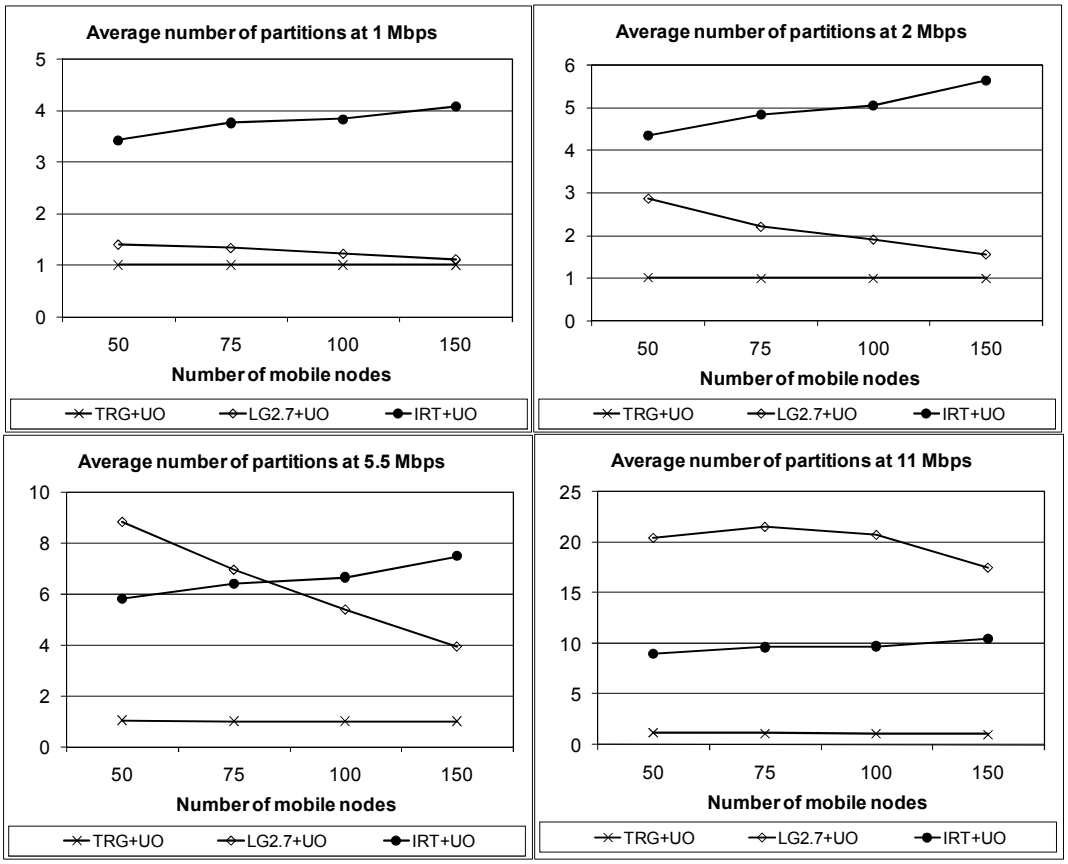


Figure 5.9: Average number of partitions

ratio of nodes in the largest partition for TRG are 1.0 at all transmission speeds (except for some rare cases lasting just a few seconds, which we also observed in the comparison of mobility models).

At 1 Mbps and 2 Mbps, IRT has more partitions than LG2.7 (Figure 5.9) and slightly fewer nodes in the largest partition (Figure 5.10). These additional partitions appear due to obstacles of the spatial environment, which decrease the network connectivity in IRT. These are just few partitions that are formed by a small amount of nodes (since more than 0.9 of all nodes reside in the largest partition). By shortening the maximum transmission range, LG2.7 can only partially approximate the impact of those obstacles.

At 11 Mbps, LG2.7 has more partitions than IRT and fewer nodes in the largest partition. This is due to a shorter transmission range in open areas, which is also

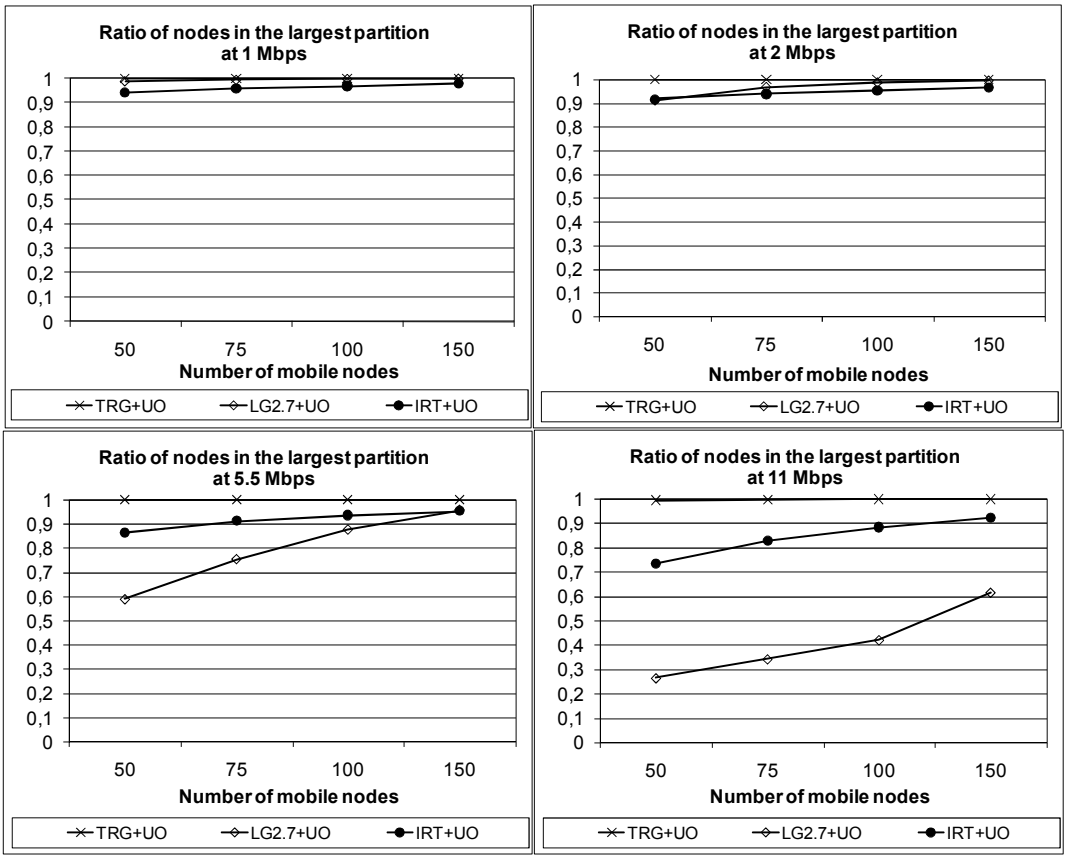


Figure 5.10: Ratio of mobile nodes in the largest partition

confirmed by graph comparisons (degree of dissimilarity and the ratio of unique edges) between LG2.7 and IRT.

At 5.5 Mbps, LG2.7 has more partitions than IRT with 50 and 75 nodes, and fewer partitions with 100 and 150 nodes. The former is due to a shorter transmission range of LG2.7 in open areas. We also see this behavior at 11 Mbps. The latter is due to the obstacles of the spatial environment, which create additional partitions in IRT. For empirical models, the more mobile nodes we have in a simulation, the fewer network partitions and the more nodes in the largest partition we get. IRT shows a different behavior: not only the ratio of nodes in the largest partition but also the total number of partitions increases with more nodes in a simulation. Basically in IRT, the more nodes we have, the higher is the chance that the obstacles block connections to some of them. Our simulation results show that these additional small partitions are

characteristic property of IRT, which simple empirical models cannot reflect. This explains the difference in partitioning results between IRT and LG2.7 at 5.5 Mbps.

To sum up, the application of more realistic physical layer models changes network topologies. Obstacles of the spatial environment block certain connections in the IRT model. This decreases the number of edges in topology graphs compared to TRG. The obstacles also cause additional network partitions to appear. These IRT partitions are formed by relatively small amount of nodes residing behind the obstacles. The number of such partitions increases as more mobile nodes are simulated.

The LG2.7 shortens its transmission range in order to approximate the propagation in obstructed areas. However, then the propagation in open areas is underestimated. As a consequence, IRT model has more network connections than LG2.7 at 2, 5.5, and 11 Mbps, and fewer network partitions at 5.5 (for 50 and 75 nodes) and 11 Mbps.

5.2.4 Discussion

Figure 5.11 − Figure 5.13 summarize the impact of more realistic mobility and physical layer models on network topologies. In previous sections, these parts were considered separately. In this section, we discuss the combined impact. Hence, the diagrams present the results for the combinations of standard ns-2 models, namely: TRG+RW and LG2.7+RW. Due to their simplicity, they are the mostly used in MANET community. They are compared against the combination IRT+UO that are described in this thesis.

In contrast to the random movement of RW, UO relies on the spatial environment, thus considering the movement area constraints. It also reflects user trips between the points of interest. To better mimic the dynamics of pedestrian movement, UO applies the model of Helbing and Molnár [HM95], which is based on real-world observations. This makes the UO movement more realistic.

The IRT uses ray tracing for finding possible radio propagation paths between the sender and the receiver. Unlike TRG and LG2.7, it relies on the map of the area and takes obstacles of the spatial environment into account. The accuracy of the model is

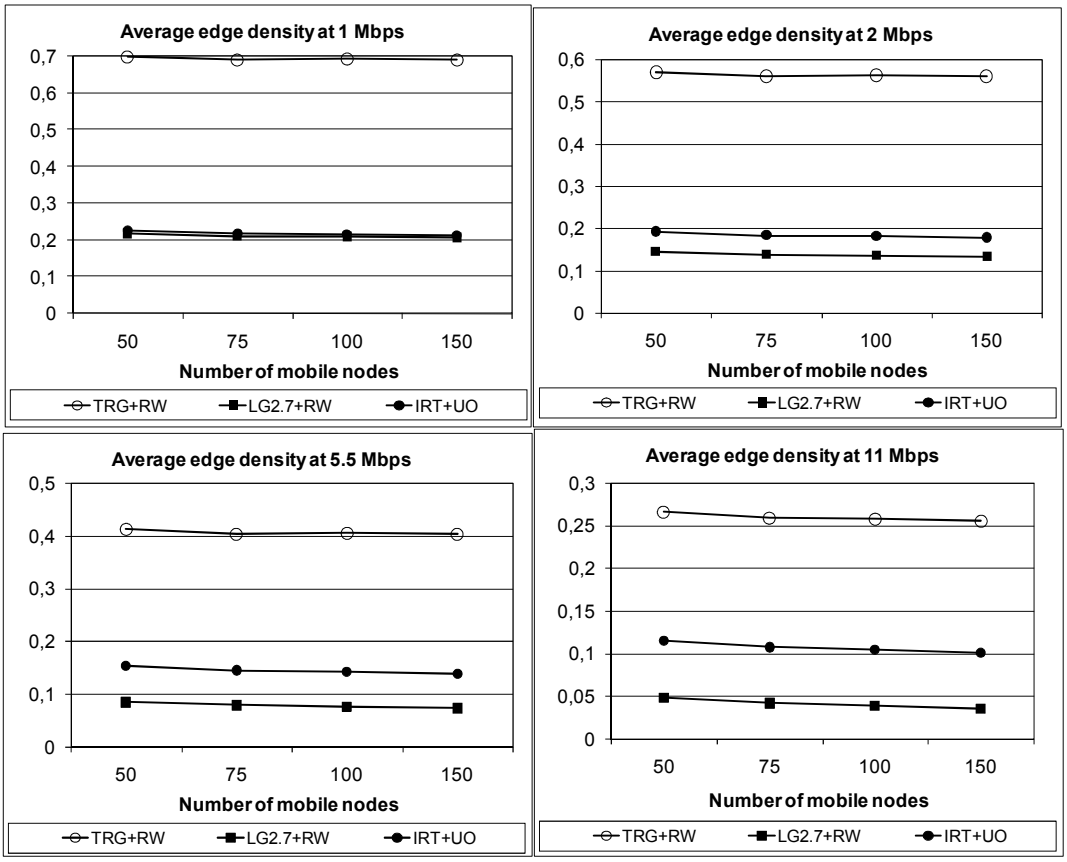


Figure 5.11: Average edge density

proven by measurements in European cities. For Stuttgart city center (Germany), which is used in this chapter, the mean error is 0.3 dB and the standard deviation is 5.8 dB.

By looking at the edge density (Figure 5.11), we see that the TRG+RW topologies contain the most edges, while LG2.7+RW contain the least edges. The results of IRT+UO lie in between and are closer to LG2.7+RW (at 1 Mbps, they are nearly equal to LG2.7+RW). We found the following reasons for those changes.

Obstacles of the spatial environment block certain connections in IRT. This decreases the number of edges in a more realistic model compared to TRG. The LG2.7 shortens its transmission range in order to approximate the propagation in obstructed areas. However, then the propagation in open areas is underestimated. This causes differences between LG2.7 and IRT topologies: some connections are missing in one graph and some are missing in another.

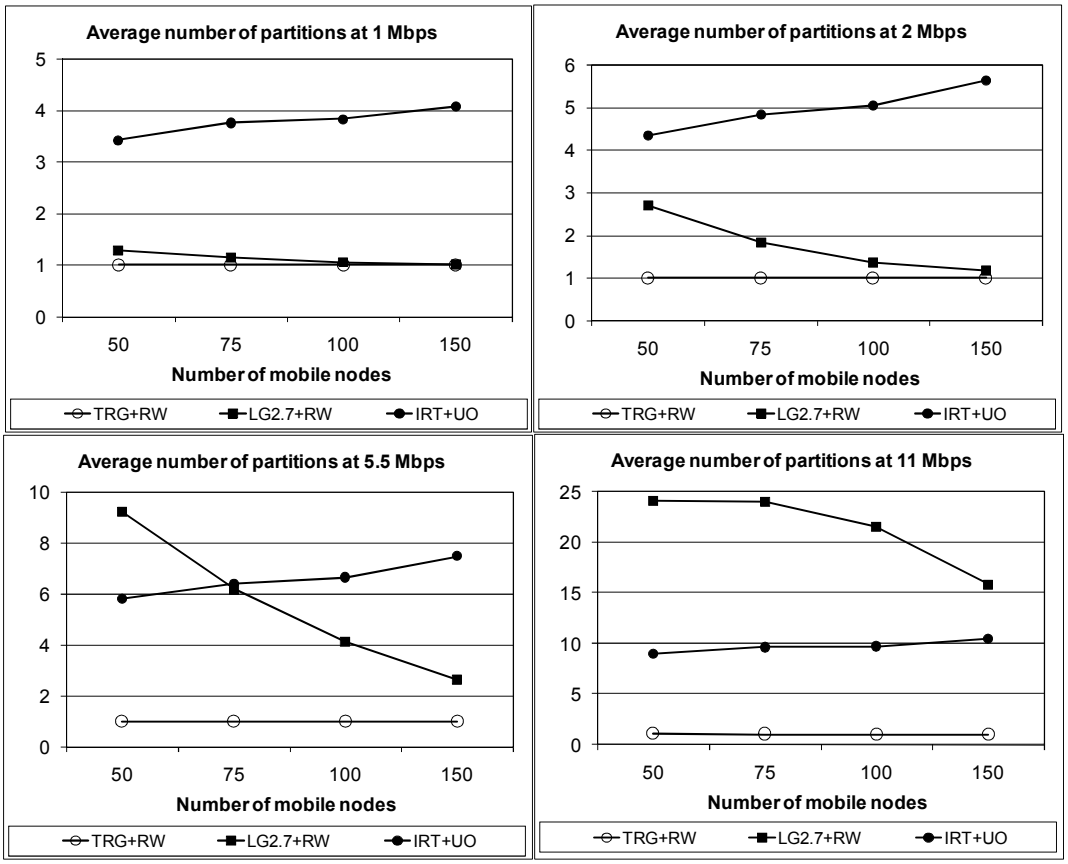


Figure 5.12: Average number of partitions

The application of UO changes a distribution of mobile nodes in the area. The spatial environment limits possible locations of nodes in UO, therefore, the nodes become located closer to each other. This results in more network connections in the topology graph. In RW, mobile nodes are more evenly distributed in the area, which decreases the total number of connections.

The results for network partitioning are shown in Figure 5.12 and in Figure 5.13. Due to its longer transmission range, TRG is nearly never partitioned, which is confirmed by the average number of network partitions and the ratio of nodes in the largest partition (both are 1.0). Moreover, as we saw in the comparison of mobility models, even distribution of mobile nodes reduces a chance of network partitioning in RW. This also contributes to the partitioning results of TRG+RW.

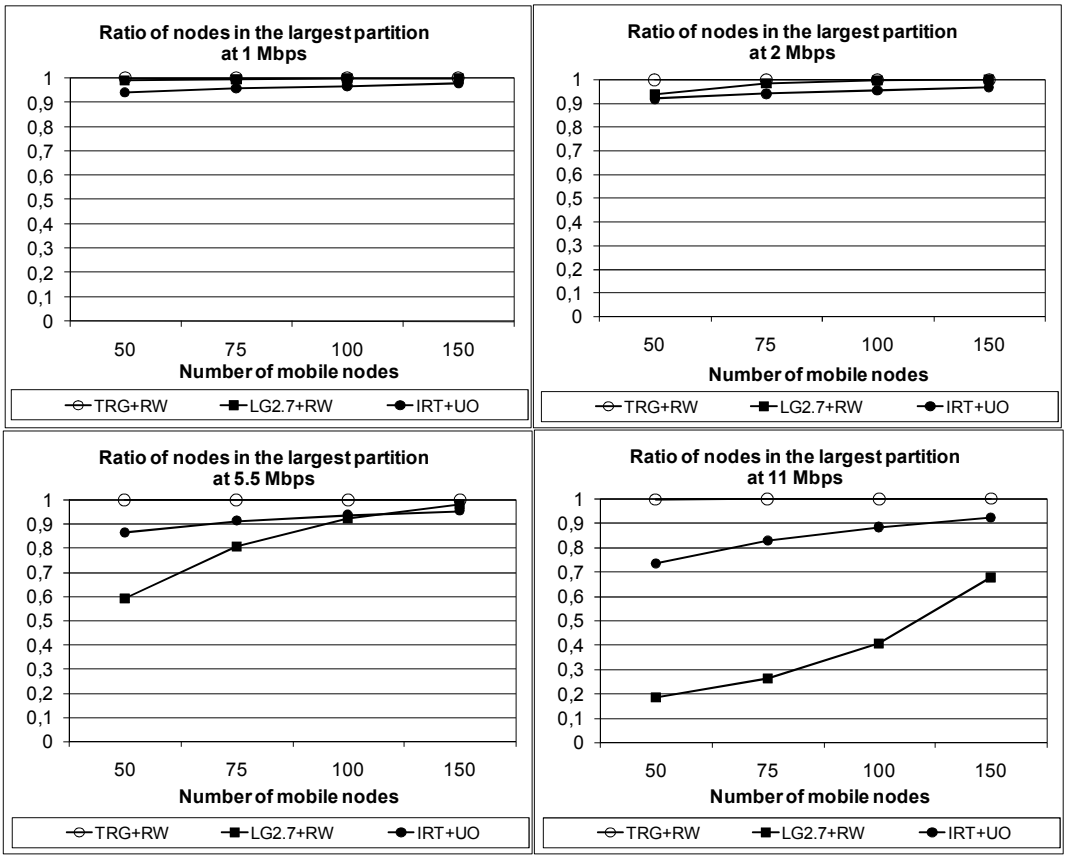


Figure 5.13: Ratio of mobile nodes in the largest partition

Obstacles cause additional network partitions to appear in IRT. These partitions are formed by relatively small amount of nodes that reside behind obstacles. The number of such partitions increases as more mobile nodes are used in a simulation. With more nodes, there is basically a higher chance that some of the nodes will be blocked by obstacles. These partitions cannot be reflected by simple empirical models, and hence, is a characteristic property of the more realistic model. In contrast, the number of partitions for LG2.7 always decreases as more nodes are used in simulations. Together with the changes due to mobility (RW is less partitioned), this causes IRT+UO to have more partitions than LG2.7+RW at 1, 2, and 5.5 (100 and 150 nodes) Mbps. A shorter transmission range of LG2.7 in open areas is the main reason for more partitions and fewer nodes in the largest partition for LG2.7+RW at 5.5 Mbps (50 and 75 nodes) and at 11 Mbps compared to IRT+UO.

Thus, we see the need for in using more realistic models in performance evaluations. The widely used TRG+RW combination shows rather "optimistic" results: there are more network connections and no network partitions, which is mainly due to the radio propagation model. The combination of LG2.7+RW shows rather "pessimistic" results as of the total number of connections in the network topology graph. The results of IRT+UO cannot be easily approximated by simple models. IRT+UO topologies are also more partitioned than LG2.7+RW at lower transmission speeds and with higher node densities. In turn, LG2.7+RW have more partitions at higher speeds and lower node densities. These changes have effect on simulation results, for instance, on routing protocol performance, which are analyzed in the next sections.

5.3 Routing Protocol (AODV) Performance

Next, we analyze the impact of the more realistic models on the performance of AODV routing protocol [PR99]. It is a reactive routing protocol, so it performs on-demand route discovery. It is based on traditional distance vector method. Therefore, each node maintains a routing table containing all reachable destination nodes, the number of hops, and the next hop to the destination. Whenever a node needs to find a path to another node, it broadcasts a Route Request (RREQ) packet to its neighbors. The neighbors flood this packet through the network, until it reaches the destination node (or a node with a fresh route to the destination). By receiving RREQ, intermediate nodes learn a path to the source node. A Route Reply (RREP) packet is sent back to the source node by unicast. By forwarding RREP, the nodes learn a route to the destination node. Nodes discover their direct neighbors by periodically broadcasting HELLO packets.

5.3.1 Metrics

We use the following metrics in comparisons. They are commonly applied for analyzing the performance of MANET routing protocols, e.g., in [BMJ+98], [CBD02], [JBAS03], and [THB+02].

Data packet delivery ratio DR is the ratio between the number of data packets successfully delivered to recipients and the total number of data packets originated by traffic sources:

$$DR = \frac{\left| \, data \;\; packets \;\; delivered \, \right|}{\left| \, data \;\; packets \;\; originated \, \right|} \times 100\% \quad (5.9)$$

In other words, this metric indicates the percentage of packets that were successfully routed to destinations. For instance, in the case of network partitioning, a path to recipient may not exist, so the packets will be dropped after some buffering at intermediate nodes. This metric counts only the data packets between traffic ends and not the packets that are sent by intermediate nodes upon the packet forwarding. Thus, if a routing path contains several hops, the packet would be counted only once, although it might be retransmitted several times by intermediate nodes. Obviously, the metric takes its values in:

$$DR \in [0,100\%] \quad (5.10)$$

The lowest value indicates that none of the packets were delivered. The maximum value tells that all data packets were delivered successfully.

Routing packet overhead RO is the ratio between the number of routing packets sent (RREQ and RREP) and the total number of packets (routing and data):

$$RO = \frac{\left| \, routing \;\; packets \, \right|}{\left| \, routing \;\; packets + data \;\; packets \, \right|} \times 100\% \quad (5.11)$$

Unlike the data packet delivery ratio, this metric also counts the data packets that were forwarded by intermediate nodes. Thus, if a routing path contains several hops, the packet would be sent several times by intermediate nodes, and these additional data packets will be counted. This metric basically indicates the percentage of routing (control) packets to the total number of packets. The metric takes its values in:

$$RO \in [0,100\%] \quad (5.12)$$

The next two metrics are related to routing paths. *Data packet delay* t_d is the time elapsed between the data packet origination at a source and its delivery to the recipient. This metric takes its values in:

$$t_d \in (0, t_{\max}) \tag{5.13}$$

where t_{max} indicates the maximum lifetime for a data packet before it should be discarded. In our simulations, we use the default ns-2 value of 30 s.

Hop count HC is the number of nodes that a data packet traverses on the way to recipient. This metric takes its values in:

$$HC \in [1, N-1] \tag{5.14}$$

where N is the number of nodes in simulation. The lowest value indicates direct communication between the source and recipient. The maximum value indicates the case when a packet traverses all nodes on the way to destination. It is important to note that the last two metrics consider only the packets that were successfully delivered to the recipients.

Simulation results below present arithmetic mean over 20 simulation runs. The results for data packet delay were aggregated by using the *geometric mean*, which is the *nth* root of the product of *n* terms. The delays for different packets can differ significantly, e.g., due to occasional buffering at intermediate nodes. The geometric mean is more resistant to the inequality of samples, and hence, is used in our evaluations upon comparing the packet delay results.

5.3.2 Impact of Mobility Models

To analyze the impact of the more realistic mobility model on AODV performance, we compare RW and UO mobility models with empirical radio propagation models (TRG and LG2.7).

The results for data packet delivery ratio *DR* are shown in Figure 5.14. Surprisingly, the results of TRG+RW and TRG+UO at 1 and 2 Mbps are below LG2.7+RW and LG2.7+UO, although the comparison of network topologies in the previous section shows that TRG topologies have more network connections and are less partitioned. In AODV, mobile nodes perform route discovery by flooding the RREQ packet through the network. TRG has a relatively long transmission range at 1 and 2 Mbps, which results in densely connected topologies with many nodes residing in the range of each other. According to simulator's logs, this leads to frequent collisions when multiple nodes flood the RREQ packets through the network.

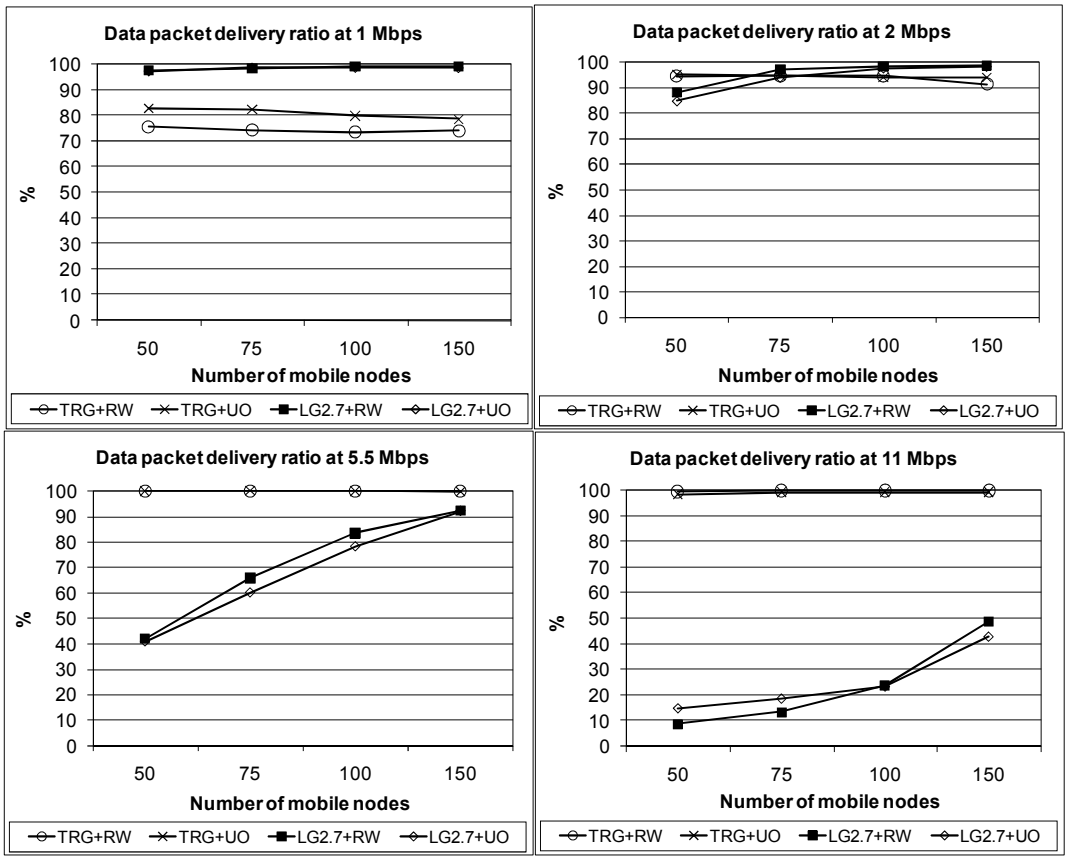


Figure 5.14: AODV data packet delivery ratio

Consequently, many route discovery packets are lost, which has negative impact on the routing protocol performance. The network connectivity decreases at 5.5 and 11 Mpbs. At those speeds (and at 2 Mbps with 50 nodes), TRG performs better than LG2.7, as expected.

From the results of network connectivity, we know that the spatial environment limits possible locations of nodes in UO. Therefore, mobile nodes are closer located to each other in the model, and hence, network topologies have more connections. In RW, mobile nodes are more evenly distributed in the area. This makes the total number of network connections decrease, however, also reduces a chance of network partitioning. Closer node placement seems to have a positive effect on packet delivery in densely connected networks, and thus, DR(TRG+UO) > DR(TRG+RW) at 1 and 2 Mbps. The network connectivity decreases at 5.5 and 11 Mbps. This

makes the situations with many RREQ packet collisions disappear, and hence, about 100% of data packets are delivered in TRG for both mobility models.

In difference to TRG, LG2.7 has a shorter transmission range that results in fewer connections in network graphs. At 1, 2 (100 and 150 nodes), and 11 (100 nodes) Mbps, nearly the same percentage of data packets (≈100%) is delivered in LG2.7 for both mobility models. According to the results of network partitioning, LG2.7 is less partitioned with RW at 2 (50 and 75 nodes), 5.5, and 11 (150 nodes) Mbps. Therefore, more packets are delivered in those scenarios than for LG2.7+ UO. In turn, LG2.7+UO topologies are less partitioned at 11 (50 and 75 nodes) Mbps. In those setups, DR(LG2.7+UO) > DR(LG2.7+RW). In fact, the LG2.7 curves in Figure 5.14 behave similar to LG2.7 curves in Figure 5.4 (ratio of nodes in the largest partition), which indicates on a direct dependency between the data packet delivery ratio and partitioning results. In all scenarios, data packet delivery ratio increases as more nodes are simulated. This is caused by higher network connectivity and fewer partitions.

The results for routing packet overhead *RO* are shown in Figure 5.15. More mobile nodes in the transmission range of each other make more RREQ packets being flooded in the network. Therefore, the routing overhead is generally higher in UO than in RW. The exceptions are the scenarios with TRG+RW at 1 Mbps (more routing packets) and LG2.7+RW with 50 nodes at 11 Mbps. Detailed analysis for the latter case shows that we still have more routing packets being sent in LG2.7+UO. However, the *RO* metric reflects the *ratio* of routing packets to the *total* number of packets. In the mentioned LG2.7+RW scenario, we have poorer network connectivity, and consequently, fewer data packets being sent, since the routing paths to destinations are not found. Thus, the total number of packets (routing and data) is less for LG2.7+RW than for LG2.7+UO, and therefore, the percentage of LG2.7+RW routing packets is higher.

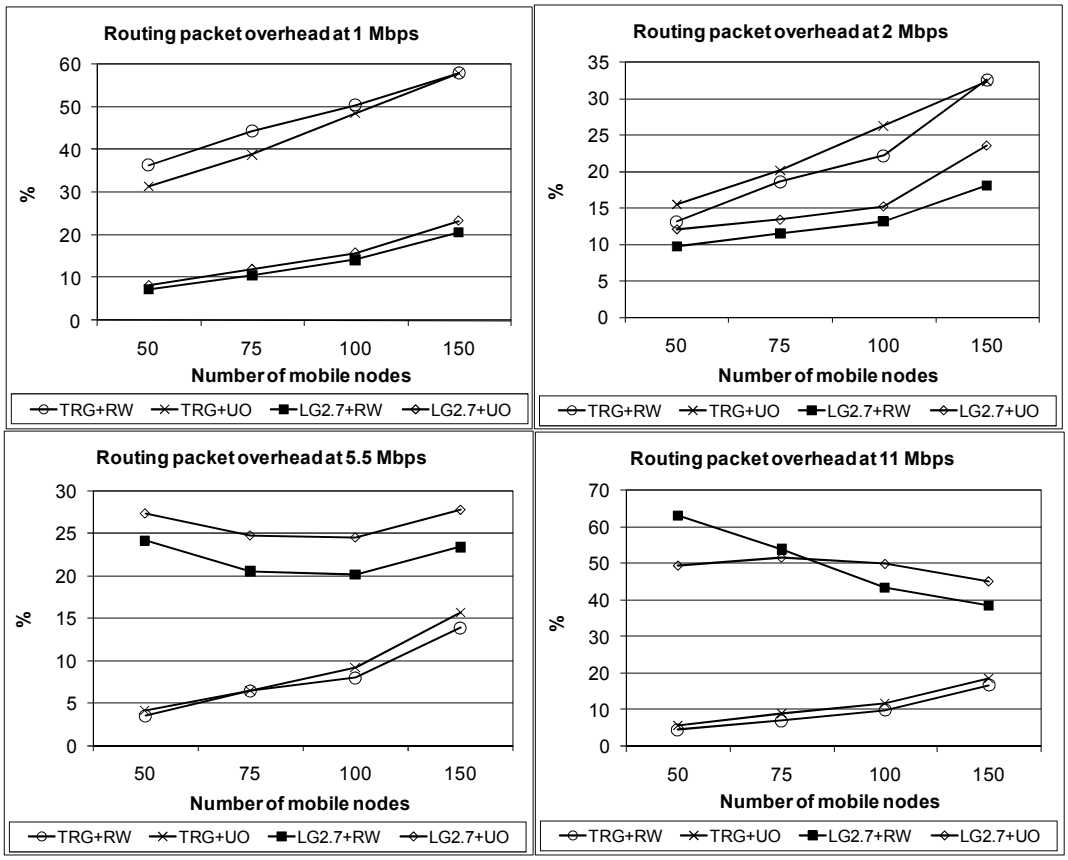


Figure 5.15: AODV routing packet overhead

The routing packet overhead increases with the number of nodes in the scenario. This is expected, since then more packets are required to search and to maintain the routes. Moreover, more routing packets are required for finding paths in the networks with many partitions. Therefore, the routing packet overhead for LG2.7 is relatively high at 11 Mbps. In these simulations, the routing overhead even decreases with the number of nodes. Here we have the situation that the negative impact of poorer network connectivity heavily dominates over the impact caused by more mobile nodes. Those additional nodes increase the network connectivity, thus reducing the routing overhead.

The results for data packet hop count *HC* and data packet delay t_d are shown in Figure 5.16 and in Figure 5.17 respectively. Mobile nodes are located closer to each other in UO. Therefore, data packets need to traverse fewer hops comparing to RW. Consequently, *HC*(TRG+RW) > *HC*(TRG+UO) and *HC*(LG2.7+RW) >*HC*(LG2.7+

UO) for all transmission speeds. Due to a long transmission range of TRG, the difference between *HC*(TRG+RW) and *HC*(TRG+UO) is very small. For LG2.7, we see a larger difference. The routing path lengths impact the data packet delays. Since the packets need to traverse more hops in RW, t_d(TRG+RW) > t_d(TRG+UO) and t_d(LG2.7+RW) > t_d(LG2.7+UO) for all transmission speeds.

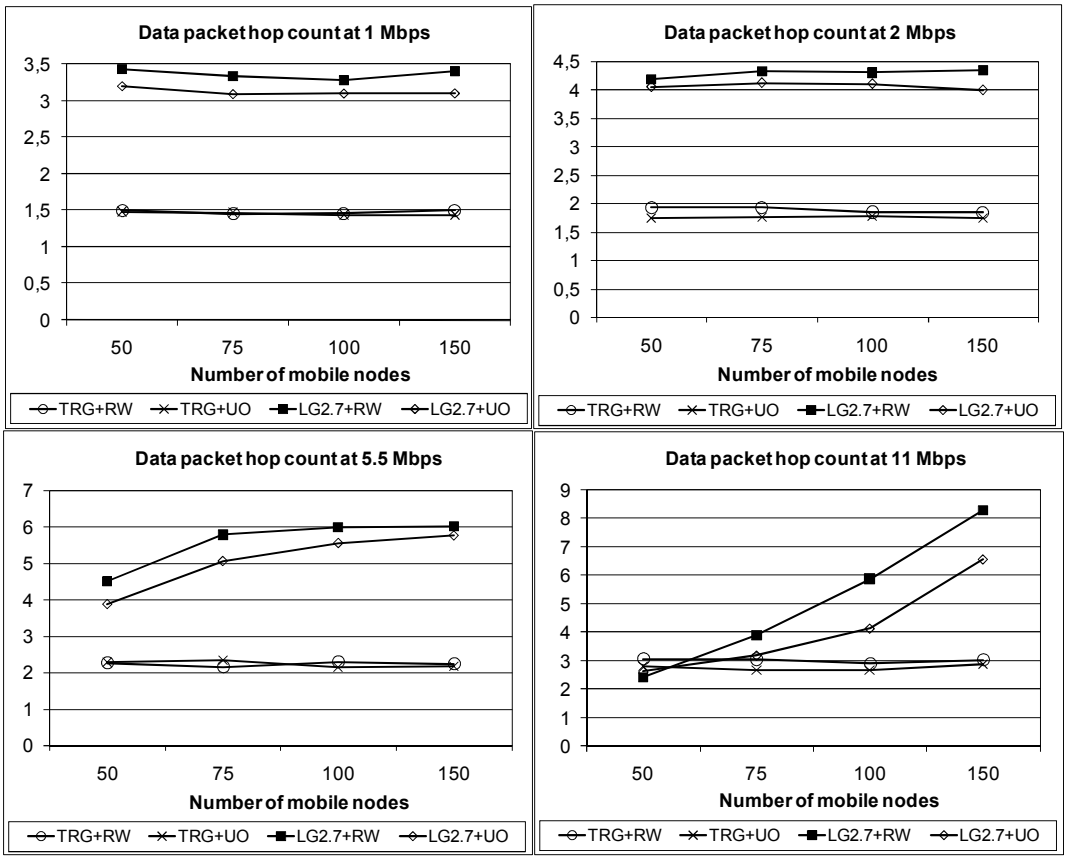


Figure 5.16: AODV data packet hop count

The results for the data packet hop count and data packet delay for TRG and LG2.7 at 1 and 2 Mbps and for TRG at 5.5 and 11 Mbps remain nearly constant for different numbers of nodes. In these setups, the transmission range of the radio propagation models is so long that the diameter of the topology graph stays the same even with a small number of nodes (the graph diameter is then obviously determined by the maximum transmission range of the model). At 5.5 and 11 Mbps, LG2.7 has a

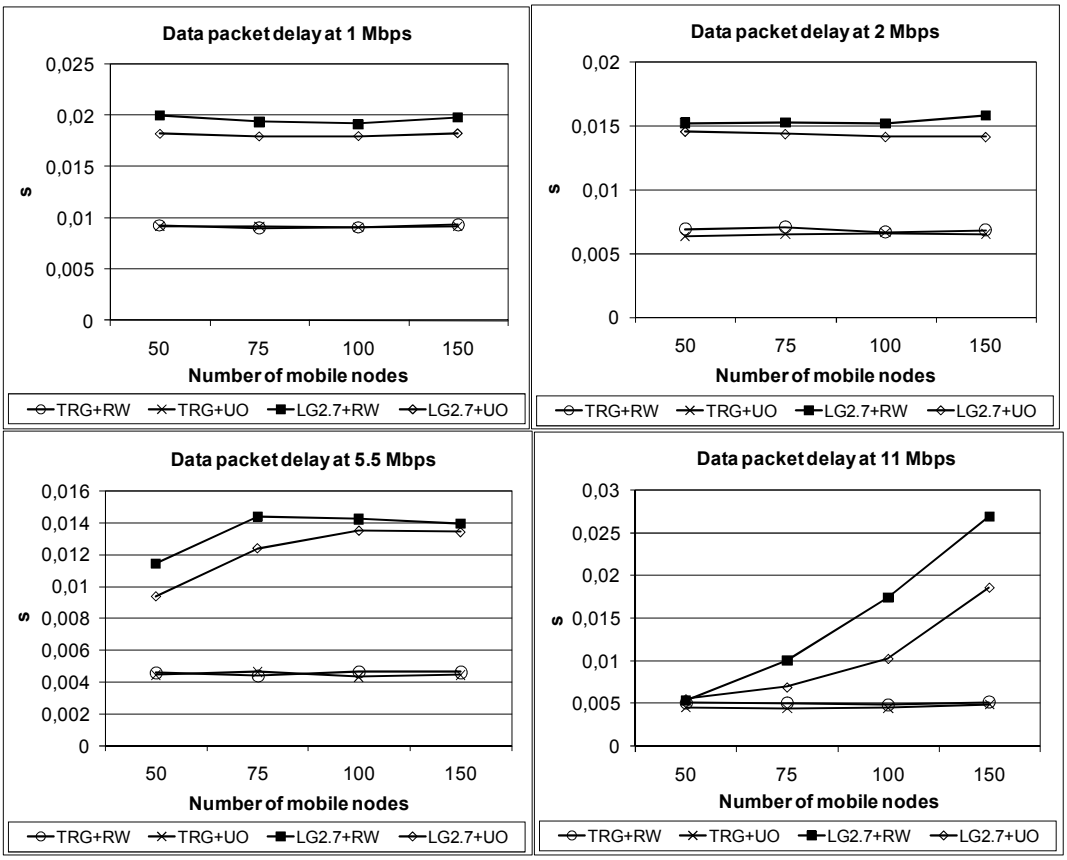


Figure 5.17: AODV data packet delay

considerably shorter transmission range, so the network connectivity is poorer for a small number of nodes. With the increase in the number of mobile nodes, we get better connectivity and more routing paths that connect the parts of the network. From certain amount of nodes (i.e. 100 nodes in 5.5 Mbps scenario), further increase does not bring anything: there is basically enough nodes in-between to establish the shortest path.

To sum up, the application of UO causes the following changes in AODV performance. Mobile nodes are located closer to each other in UO. This improves the packet delivery ratio in densely connected TRG topologies at 1 and 2 Mbps, which experience many collisions upon the flooding of route discovery packets. Due to the changes in network partitioning (fewer partitions and more nodes in the largest partition in RW), LG2.7 normally delivers more packets in RW scenarios at 5.5 and 11 (150 nodes) Mbps. The LG2.7 scenarios at 11 Mbps with smaller number of

nodes (50 and 75 nodes) profit more from the constrained movement area in UO. In general, more routing packets are sent in UO, however, the routing paths and the data packet delays are shorter.

5.3.3 Impact of Physical Layer Models

For analyzing the impact of physical layer models on AODV performance, we compare the results of TRG, LG2.7, and IRT models with the same mobility model (UO).

The results for data packet delivery ratio *DR* are shown in Figure 5.18. Similar to the results that we discussed upon comparing the mobility models, the performance of TRG+UO degrades at 1 and 2 Mbps. This happens due to collisions of discovery packets upon flooding. In these scenarios, LG2.7+UO show higher data packet delivery ratios, except for 50 nodes at 2 Mbps, which still profits from more network

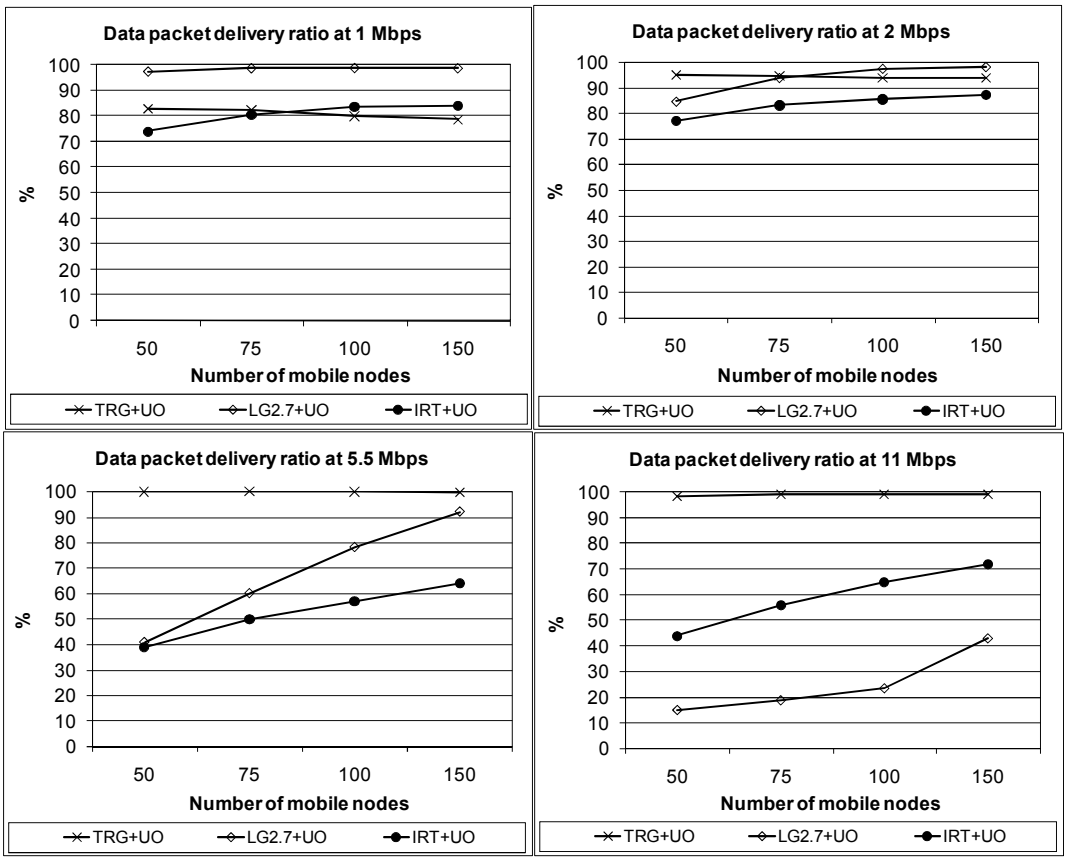


Figure 5.18: AODV data packet delivery ratio

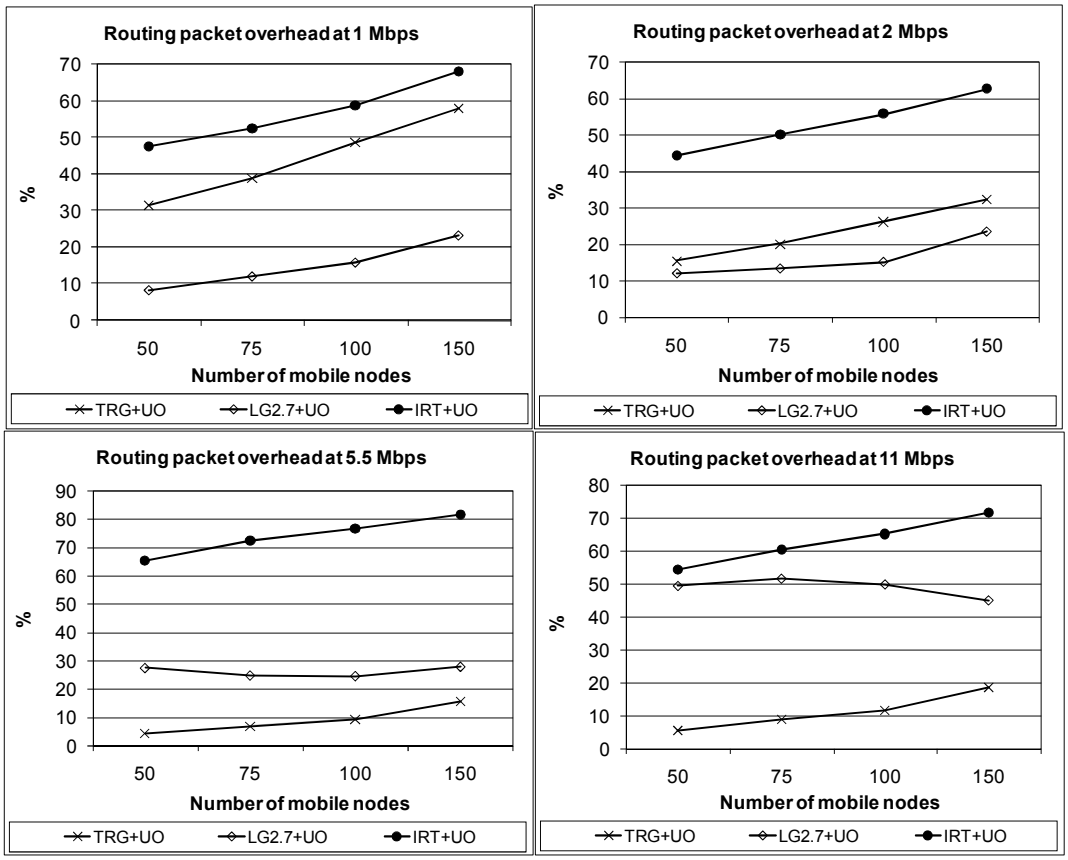


Figure 5.19: AODV routing packet overhead

connections in TRG. TRG outperforms LG2.7 at 5.5 and 11 Mbps, as expected. Due to more network partitions, the results of IRT at 1 and 2 Mbps are normally below TRG and LG2.7, except for the scenarios with 100 and 150 nodes at 1 Mbps. Here *DR*(IRT+UO) > *DR*(TRG+UO).

LG2.7 has much shorter maximum transmission range at 11 Mbps (127 m vs. 400 m, as in Table 5.2). Therefore, the corresponding topologies have fewer network connections and more partitions. Consequently, *DR*(IRT+UO) > *DR*(LG2.7+UO) in these scenarios. Upon the results of topology comparisons, we also expected to see this behavior at 5.5 Mbps. Surprisingly, LG2.7+UO deliver more packets than IRT+UO in these scenarios. To understand a reason for this phenomenon, lets us look at the results of routing packet overhead *RO* in Figure 5.19.

Similar to the results with different mobility models, more packets are sent in TRG at 1 and 2 Mbps due to many RREQ packet collisions. This is not a case for 5.5 and

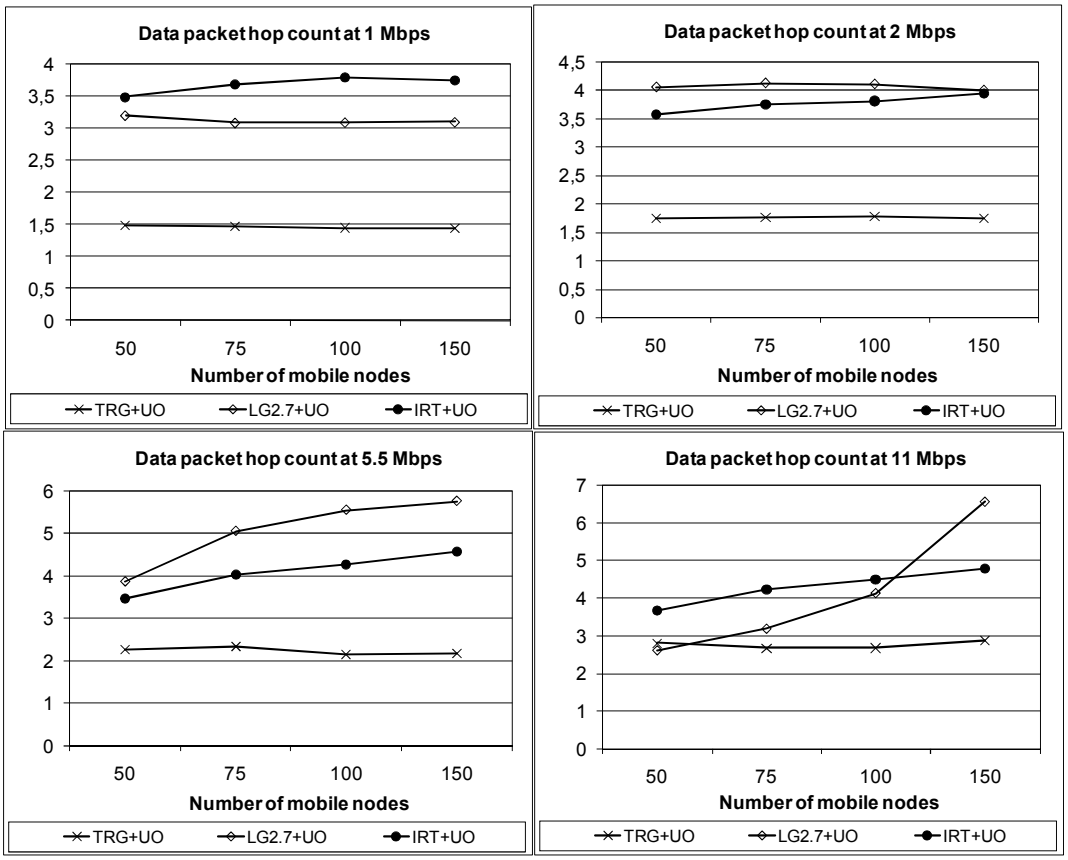


Figure 5.20: AODV data packet hop count

11 Mbps, in which less routing packets are required for route search and maintenance. Hence, *RO*(TRG+UO) < *RO*(LG2.7+UO) in these scenarios. IRT shows the highest routing overhead for all four speeds. The percentage of routing packets is above 50% in most cases. Here we see another important effect that is introduced by the more realistic radio propagation model: coping with obstacles of the propagation environment requires more routing packets being sent. The established routing paths are not "direct" any more but rather must "bend" those obstacles. Obstacles also make the topologies change more quickly upon the user movement. This requires the routing protocol to perform additional route rediscoveries. The frequently changing network topologies are the reason why IRT delivers fewer packets than LG2.7 at 5.5 Mbps, although the topologies contain more network connections.

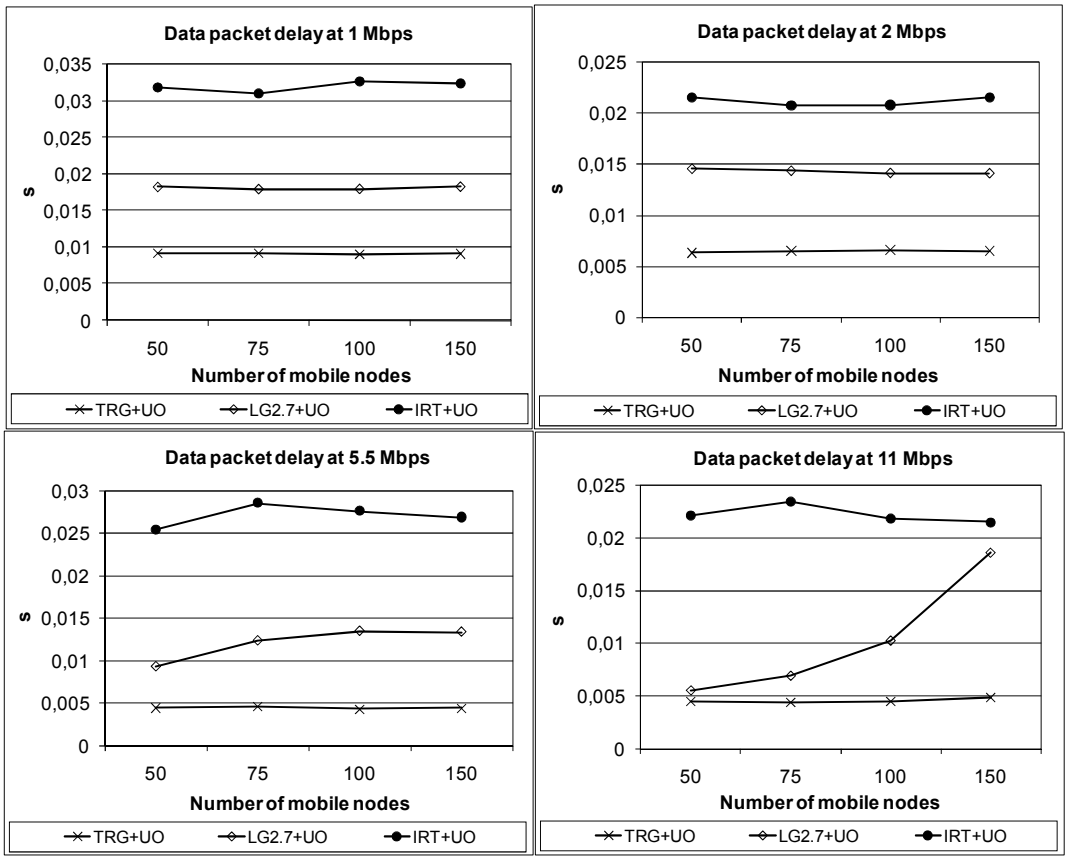


Figure 5.21: AODV data packet delay

The results for the data packet hop count *HC* are shown in Figure 5.20. The shortest network paths are obviously in the TRG model. The paths in IRT must "bend" the obstacles, therefore, IRT shows the biggest value of *HC* among the three models at 1 Mbps. Due to its shorter transmission range, LG2.7 contains less network connections at 2, 5.5, and 11 (150 nodes) Mbps, and thus, *HC*(LG2.7+UO) > *HC*(IRT+UO).

In LG2.7 studies with 50, 75, and 100 nodes at 11 Mbps, data packets are only delivered among nearby nodes (longer network paths basically do not exist for such low node densities). Therefore, the routing paths contain only few hops in these scenarios. With more mobile nodes (150 nodes), the network connectivity increases, which results in longer LG2.7 network paths and higher data packet delivery ratios.

The results for data packet delay t_d are shown in Figure 5.21. For empirical models, the results are directly dependent on the lengths of the network paths.

Therefore, t_d(TRG+UO) < t_d(LG2.7+UO) at all transmission speeds. The delay results for IRT are the highest among the models. Unlike in empirical models, the packet delays in IRT are also introduced by more frequently changing network topologies, which is caused by user movements in the areas with obstacles.

To sum up, dense network topologies of TRG at 1 and 2 Mbps experience many packet collisions upon the flooding-based route discovery. This increases the routing packet overhead and hardens the discovery process, thus negatively impacting the data packet delivery ratio. This effect disappears at 5.5 and 11 Mbps, therefore, nearly 100% of data packets are delivered in these TRG scenarios. Obstacles of the propagation environment harden route search and maintenance in IRT. Moreover, the obstacles make routing paths to change more often upon the movement of users. Thus, IRT delivers fewer packets comparing to TRG and LG2.7 at 1, 2, and 5.5 Mbps, and shows the highest values of the routing packet overhead and data packet delay among the models. Since TRG and LG2.7 neglect the obstacles of spatial environment, they are unable to reproduce this effect in simulations. LG2.7 has much shorter transmission range at 11 Mbps (remember, the transmission range of TRG and IRT is the same in open areas), therefore, IRT delivers more data packets in these scenarios.

The shortest routing paths are in TRG. In IRT, the routing paths must "bend" the obstacles, thus traversing more hops. Therefore, IRT has the largest routing path lengths among the three models at 1 Mbps. At higher transmission speeds, the shorter transmission range of LG2.7 has more impact on the results. For that reason, the routing paths are shorter in IRT at 2, 5.5, and 11 Mbps comparing to LG2.7.

5.3.4 Discussion

From the comparisons of network topologies, we know that UO changes the distribution of mobile users, so they get located closer to each other. This results in more connections in network graphs. However, since the mobile nodes are not evenly distributed in the area any more, UO topologies are more partitioned (so they contain more partitions and fewer nodes in the largest partition). Also obstacles of the propagation environment cause additional network partitions to appear in IRT.

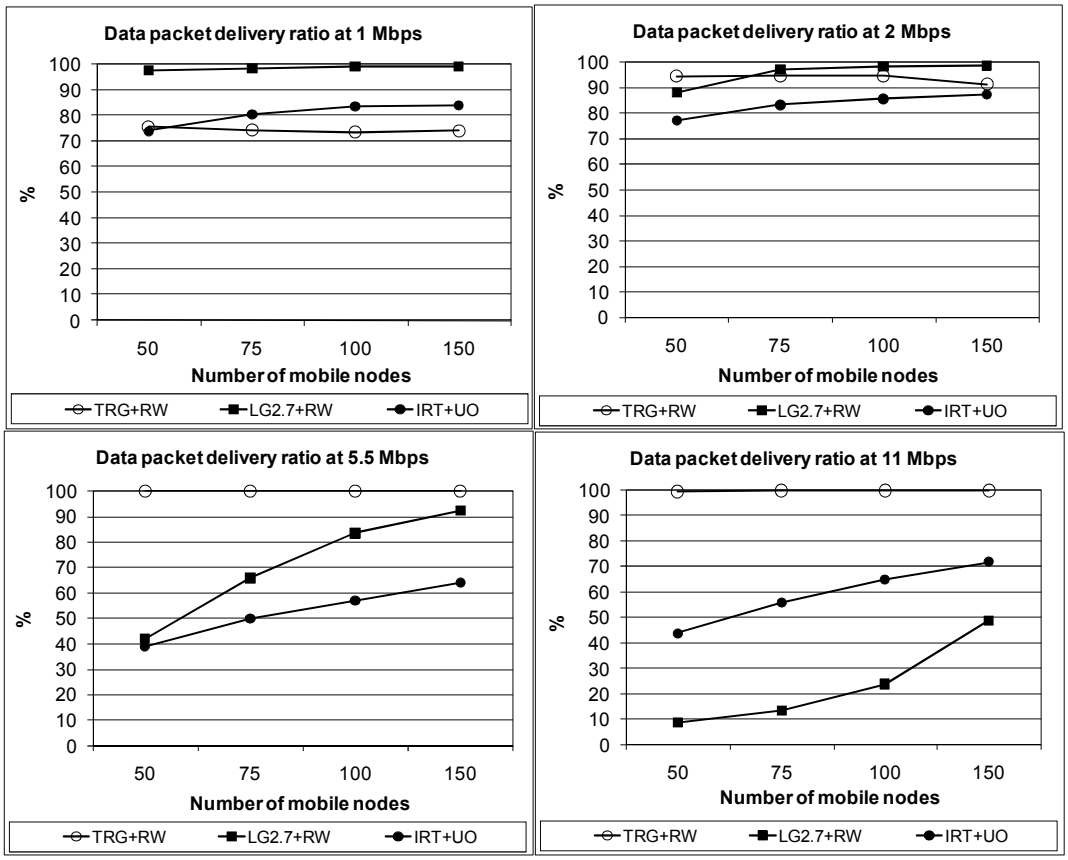


Figure 5.22: AODV data packet delivery ratio

These partitions are formed by relatively small amount of nodes that reside behind obstacles. TRG has a long transmission range, thus delivering rather "optimistic" results (more network connections and no network partitions). LG2.7 has a shorter transmission range, thus approximating the radio propagation in obstructed areas. However, then the propagation in open areas in underestimated. This results in fewer connections in network graphs and more network partitions, especially at higher transmission speeds and with smaller number of nodes.

Let us now look at how this impacts AODV performance. Figure 5.22 − Figure 5.25 summarize the simulation results for TRG+RW, LG2.7+RW, and IRT+UO.

Dense network topologies of TRG+RW at 1 and 2 Mbps experience many collisions upon the flooding-based route discovery (Figure 5.22). This increases the routing overhead (Figure 5.23) and hardens the path discovery in the network, thus causing a negative impact on the data packet delivery ratio. Network connectivity

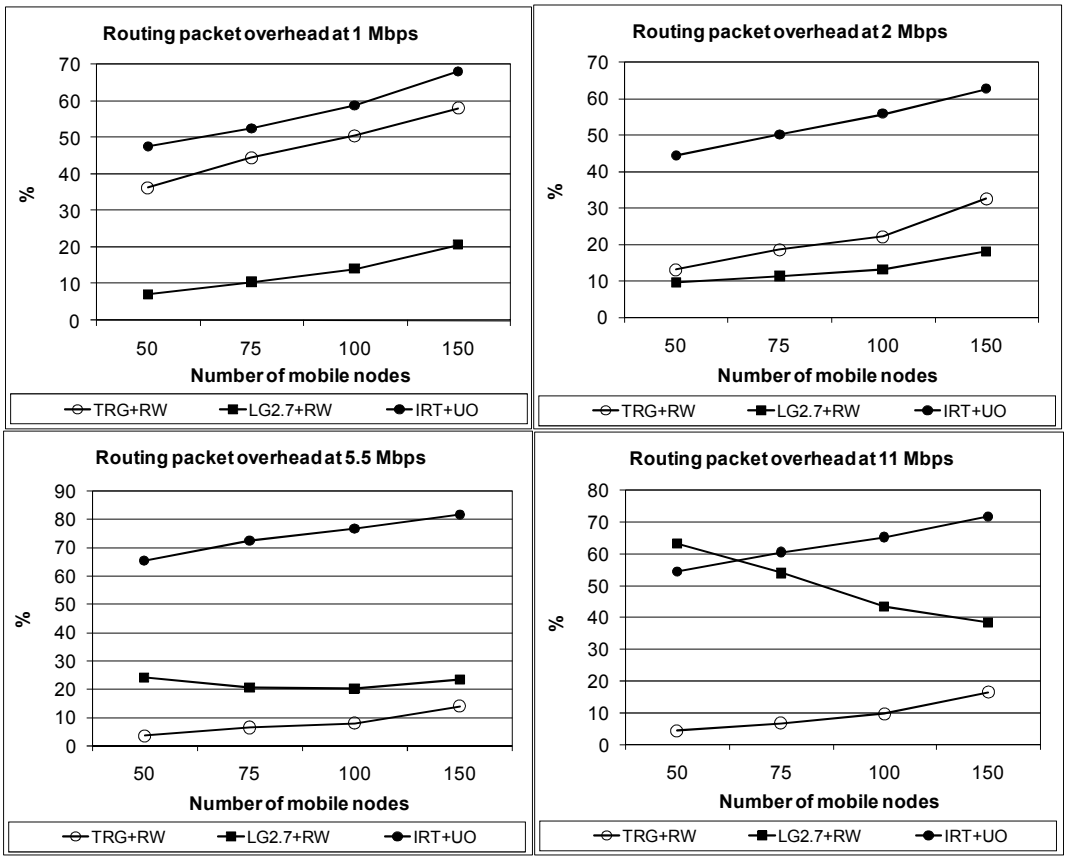


Figure 5.23: AODV routing packet overhead

decreases at higher speeds. Therefore, nearly 100% of data packets are delivered in those TRG+RW scenarios. Obstacles of the propagation environment and more network partitions make the route search in IRT+UO more difficult. The obstacles also cause the routing paths to change more often upon the user movement. This results in lower delivery ratios in IRT+UO at 1 and 2 Mbps comparing to TRG+RW and LG2.7+RW. IRT+UO also show much higher routing overhead (up to 80%). The shorter transmission range of LG2.7 effects in fewer data packets being delivered at 11 Mbps.

The shortest routing paths are in TRG+RW (Figure 5.24). In IRT, the paths must "bend" the obstacles of the propagation environment. Therefore, IRT+UO show the longest routing paths at 1 Mbps. At higher speeds, LG2.7 has much shorter transmission range, and consequently, the packets must traverse more hops. Also the application of RW increases the lengths of routing paths. In LG2.7+RW simulation

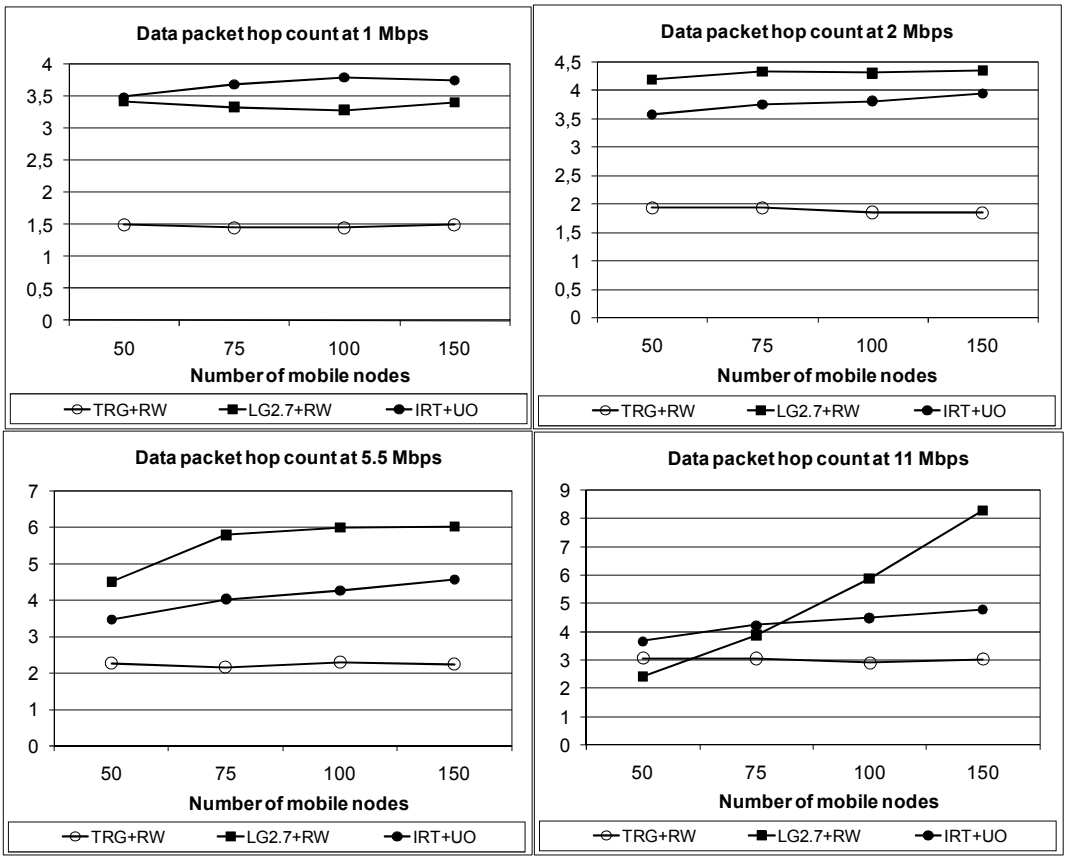


Figure 5.24: AODV data packet hop count

studies with 50 and 75 nodes at 11 Mbps, the data packets are only delivered among nearby nodes (longer network paths basically do not exist for such low node densities). Therefore, the routing paths contain only few hops in these scenarios (2-3 hops maximum).

The obstacles also make the routing paths in IRT+UO change more often. This results in longer data packet delays (Figure 5.25), since more route discoveries need to be performed. For empirical models, the results are mainly dependent on the routing paths lengths.

Thus, due to the application of more realistic models we observe a difference in AODV data packet delivery ratios and in the routing path lengths. The most significant changes, however, are in routing packet overhead and in data packet delays. They occur mainly due to obstacles of the propagation environment. The impact of those obstacles cannot be easily imitated by the simple mobility and radio

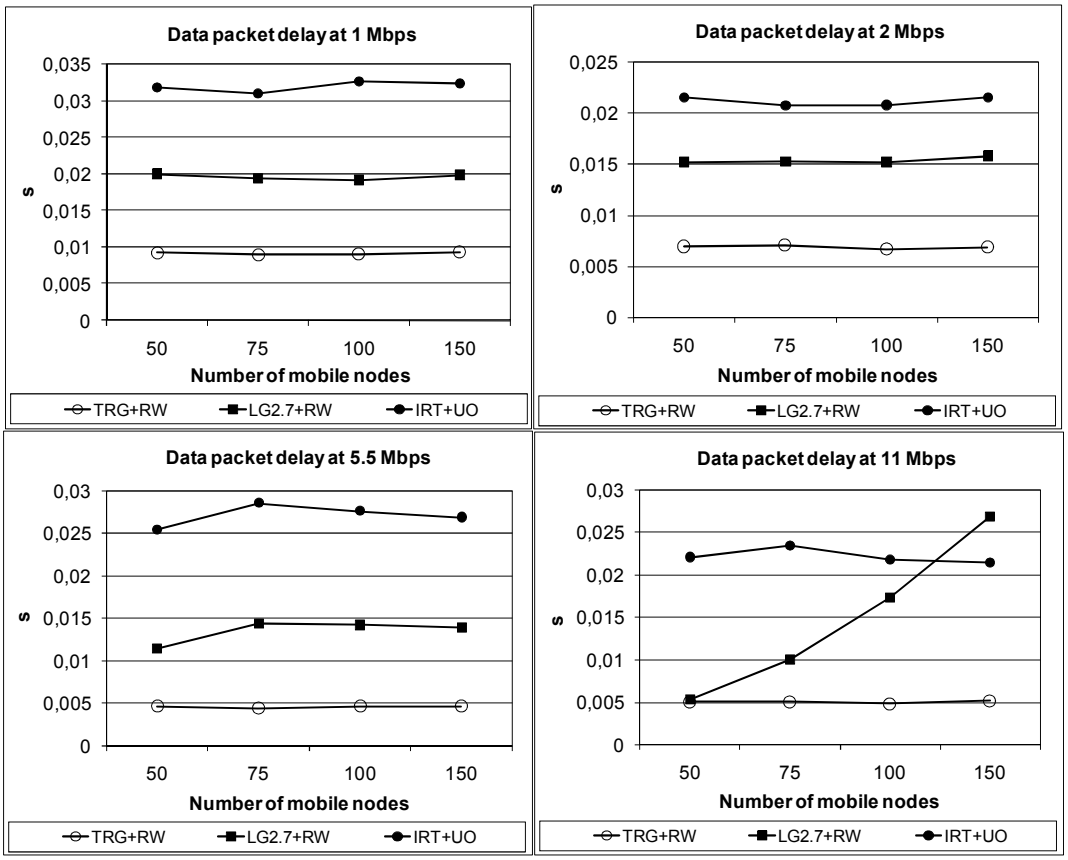


Figure 5.25: AODV data packet delay

propagation models. More realistic models help us to obtain more realistic simulation results.

5.4 Application Performance

In the previous section, we used synthetic (constant-bit rate) network traffic for analyzing the routing protocol performance. This allows comparing the results in this thesis with other routing protocol performance studies. In this section, we use the emulation facility of ns-2 (nse) for integrating an unmodified mobile application (Usenet-on-the-Fly), its execution environment, and the real implementations of network protocols into a MANET simulation. This allows us investigating the impact of more realistic models on the performance of real mobile application, and not only on the synthetic traffic. We then compare the results with standard ns-2 models and with more realistic models.

5.4.1 Integrating Mobile Applications into ns-2

The simulator ns-2 also supports the emulation mode where simulator performs in real time, thus reproducing a timing behavior similar or equal to the timing behavior of simulated entities. This allows using the network traffic of real mobile devices in a simulation, thus making it possible to imitate different scenarios in a controllable environment with relatively small effort. For mobile communication, ns-2 simulates the mobility of network nodes, data link layer (MAC), and physical layer (PHY). The traffic of higher layers then comes from real network devices. Each of these devices, like user-carried PDAs, has a dedicated simulated network node in ns-2. Network/tap object layer of ns-2 is responsible for injecting the network traffic from the external devices into the simulator. The packets that are received by the simulated nodes are forwarded back to the real devices (Figure 5.26). This allows having unmodified mobile applications and network protocol stack up to OSI layer 3 of real devices in a simulation. ns-2 only models the data link layer, the physical layer, and mobility of nodes.

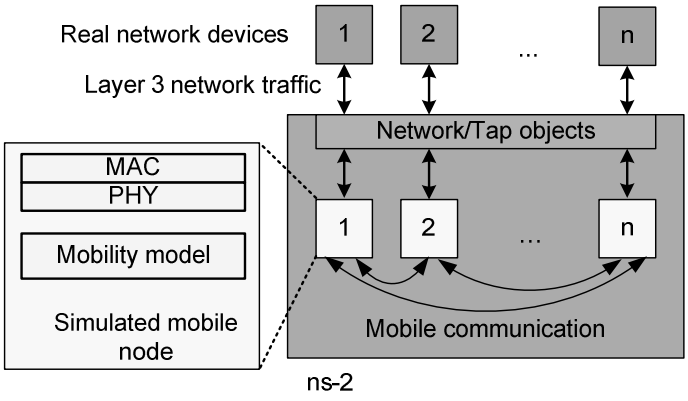


Figure 5.26: Network emulation in ns-2

However, this architecture has obvious scalability limitations, since we need the same number of real devices as the number of simulated mobile nodes. In our studies, each network device represents a single user-carried PDA or a laptop running Linux operating system. They are imitated with the help of Linux-based personal computers, which execute mobile applications under study. To be able to

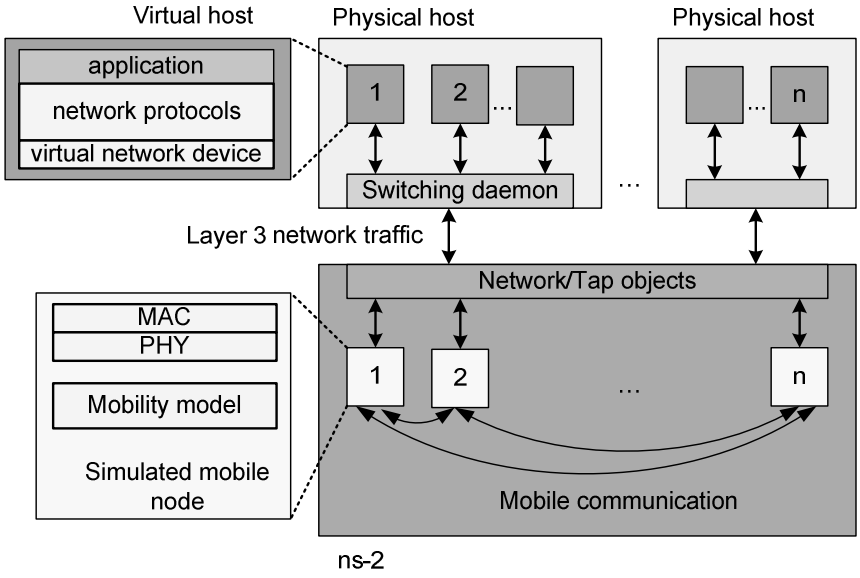


Figure 5.27: Virtualized network emulation in ns-2

support more mobile devices than the number of physical computers that we have, we extend the described architecture by using the User-mode Linux (UML) [Dik00] virtual machines. The virtual machines run as user processes on top of the operating systems of physical hosts. For an application being executed inside of it, the virtual machine appears to be a single computer with own network interface. Having several virtual machines per physical host allows us emulating more mobile devices.

The architecture of the approach is shown in Figure 5.27 [SR06b], [SR07]. Mobile applications are executed on separate virtual machines that correspond to individual network nodes. In this case, we also need to tunnel the layer 3 traffic of the virtual machines to ns-2. Virtual network devices and switching daemons are a standard way of providing the virtual UML hosts with an access to the outer network. Therefore, we modify the implementation of those switching daemons[1] to tunnel the traffic to the network/tap objects of ns-2. The latter inject the traffic into simulated mobile devices. ns-2 then models the physical layer and the data link layer of Wireless LAN,

[1] The modified version is available for download at: http://www.ipvs.uni-stuttgart.de?id=illya.stepanov&lang=en

as well as the mobility of network users. The packets received by the simulated devices are tunneled to the dedicated virtual hosts in a similar fashion.

The tunnel between the ns-2 and switching daemons is implemented by using the User Datagram Protocol (UDP) [Pos80]. This protocol guarantees neither message reliability nor message ordering. However, the use of Transmission Control Protocol (TCP) [Pos81] cause stacked TCP connections in the case when applications under study also use the TCP communication. Each of those stacked connections independently maintains TCP timeout timers and performs retransmissions. This introduces significant packet delays, which degrade network performance and affect the simulation results.

Virtual hosts and ns-2 perform in real time. To assure that the physical hosts running the virtual machines and ns-2 stay below the load limit, we monitor their CPU load and the packet drops at the ends of tunnel. To improve the accuracy of ns-2 real-time scheduler, the extensions from [MI05] are used. They contain performance and virtual clock drift correction improvements. Those extensions also provide the "network" and "tap" objects that serve as a basis for implementing the tunnel between the physical hosts and ns-2.

The described architecture allows us using unmodified mobile applications with real traffic patterns in a simulation instead of synthetic network traffic. Let us have a look at the application that we use in our evaluations.

5.4.2 "Usenet-on-the-Fly" Application

The Usenet-on-the-Fly application [BBH02] is developed in our department. It is an implementation for ad-hoc networks of the well-known Usenet system [Hor83]. The corresponding client application is implemented in Java. Therefore, it runs on various hardware platforms, in particular, on PDAs.

The basic idea is that we have different newsgroups (or news channels). The users can subscribe to desired channels to start receiving new messages that are dedicated to these channels. The users can publish new messages, which are then distributed among all subscribed users. For the dissemination, each message is distinguished by a unique ID. To stop receiving the messages, the users can also unsubscribe from

Figure 5.28: Usenet-on-the-Fly screenshot

unwanted channels. The application Graphical User Interface (GUI) (Figure 5.28) provides the user with this functionality.

Unlike in wired networks, MANET devices must cope with limited connectivity and frequent topology changes. Therefore, the messages are disseminated through diffusion: mobile devices periodically exchange their messages with other devices that are in transmission range. For supporting the message diffusion, the application includes three modules, which are implemented in separate threads:

- Neighbor discovery sender
- Neighbor discovery receiver
- SOAP engine

The neighbor discovery sender is responsible for periodic broadcasting of discovery packets that contain the device's IP address. Other mobile devices that are in transmission range receive these packets, thus learning about the devices in their proximity. It is important to note that the broadcast transmissions are unacknowledged according to the IEEE 802.11 standard [IEE99]. Hence, there are no retransmissions performed in the case of collisions.

The neighbor discovery receiver is responsible for receiving and processing the broadcasted discovery packets from the neighbors. Upon receiving such a packet, the thread algorithm issues a Simple Object Access Protocol (SOAP) call "Get All Message IDs" addressed to the neighbor. The call returns the IDs of all the messages

that are stored at that neighbor. Then they are compared with the receiver's own messages. If the neighbor has any new messages, the discovery receiver issues a SOAP call "Get Message Bodies." The IDs of the required messages are passed as parameters. The received messages are added to the local database. In the current implementation, the database stores all the messages. The GUI performs the necessary filtering to display only the messages from the subscribed channels.

The SOAP engine thread is responsible for receiving, processing, and answering the sequentially incoming SOAP requests, such as "Get All Message IDs" and "Get Message Bodies."

The mentioned application parts have the following important parameters that are specified in the configuration file:

- *DiscoveryInterval (BroadcastInterval)*: time interval between broadcasting two successive neighbor discovery packets. More often neighbor discoveries generate additional network traffic, thus requiring mobile devices to spend more energy. However, this speeds up the spreading of newsgroup messages in the network.
- *ServerSocketTimeout*: TCP timeout that the SOAP engine waits for a client to establish a connection. This timeout is required for timely aborting a TCP connection attempt, for example, when the peer node moves out the transmission range during the connection establishing.
- *ClientSocketTimeout*: TCP timeout that the SOAP engine waits for a request to be fully received. This timeout is required for timely aborting a blocking socket read operation, for instance, when the communication peer moves out the transmission range during transmitting a SOAP request.
- *SOAPTimeout*: TCP timeout that the neighbor discovery receiver waits for a SOAP response to be received on local side. This includes the time for connection establishment, the message delays for transmitting request and response, and the message processing on the remote side. This timeout is also needed for aborting a blocking socket read operation.

5.4.3 Emulation Setup

Usenet-on-the-Fly is simulated in the same Stuttgart city scenario that in used in the whole chapter. In addition to the parameters in Section 5.1, we have the following parameters that are specific for our emulation setup (Table 5.3). We already know that there is a difference in topologies among TRG+RW, LG2.7+RW, and IRT+UO setups. Therefore, we only compare standard ns-2 models LG2.7+RW against IRT+UO. LG2.7 was chosen since it delivers closer topology comparison results to IRT as of Figure 5.11-Figure 5.13 for the transmission speed that we use in this evaluation.

Parameter	**Value**
Simulation time	8100 s
Number of mobile users	30, 50, 75, and 100
Mobility models	Random waypoint mobility model (RW) User-oriented mobility model (UO)
Radio propagation models	Log-distance path-loss model with $\beta=2.7$ (LG2.7) Intelligent ray tracing model with small-scale fading and BER packet reception (IRT)
iPAQ battery capacity	3.7 V * 1350 mAh
Transmission power	15 dBm
Radio frequency	2.442 GHz
Transmission speed	1 Mbps
WLAN power consumption	idle mode: 0.045 W receive mode: 0.925 W transmit mode: 1.425 W
Neighbor discovery interval	1, 2, 5, 10, and 20 minutes
Server socket timeout	10 s
Client socket timeout	20 s
SOAP timeout	30 s

Table 5.3: Emulation parameters

The simulated users carry mobile devices running Linux (the same operating system is started on our virtual hosts). We use the parameters of Compaq© iPAQ

3660 PocketPCs with external ORiNOCO© 802.11b WLAN card [Pro03]. Since ns-2 does not support the auto-fallback between speeds, we use a fixed speed of 1 Mbps, which is the lowest transmission rate in IEEE 802.11. It is sufficient for our prototype, since we exchange relatively short newsgroup messages. We currently do not support message attachments, which have the biggest contribution to message size.

In our evaluation, we assume that all simulated mobile users are subscribed to the same newsgroup channel about free-time activities in a city. The published messages resemble 2-hour traffic we took from real newsgroup archive. In total, 18 messages are posted to the newsgroup and have to be disseminated among the users. We set the total simulation time to 8100 s, so all the messages have a chance to get spread in the network.

Up to 5 virtual hosts are created on each physical host. The number was chosen according to the number of mobile users and the number of physical hosts available. In spite of 5 parallel running virtual machines, CPU load of virtual hosts was low (on the order of $1*10^{-2}$), since the Usenet-on-the-Fly application performs little computational activity.

5.4.4 Results

In the evaluations of the Usenet-on-the-Fly application, we are mostly interested in *message spreading* in time. It is basically a ratio of mobile users who received a new newsgroup message to the total number of users. The spreading is analyzed for each message independently. The faster the newsgroup messages spread in the network, the better our Usenet-on-the-Fly application performs.

Besides the message spreading, we also look at the energy consumption of mobile devices. ns-2 estimates the amount of energy that a mobile device spends for communication (without considering the other energy-consuming components, such as CPU, display, etc.). We use also this metric in our comparisons. Clearly, the less energy mobile devices consume the better.

The results in charts present an average over 18 messages that are posted in the newsgroup and are disseminated in the network.

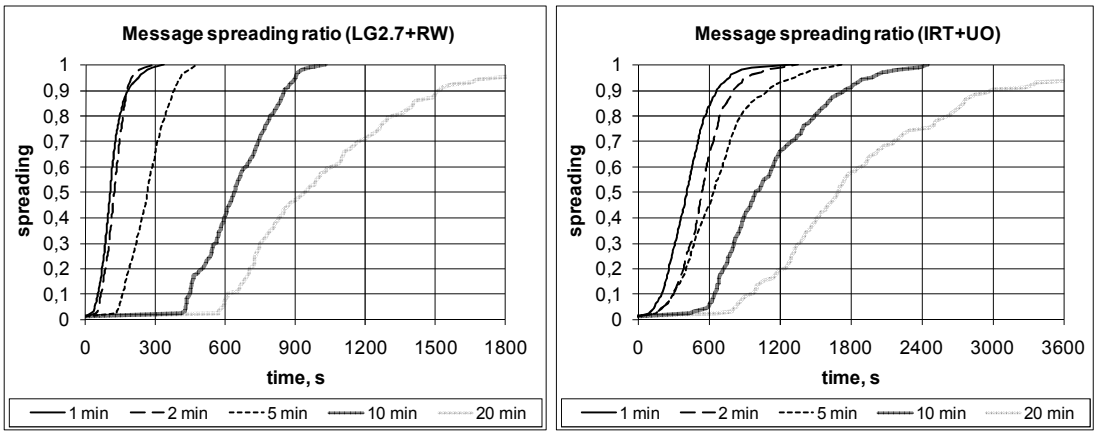


Figure 5.29: Message spreading ratio between 100 users for different neighbor discovery intervals

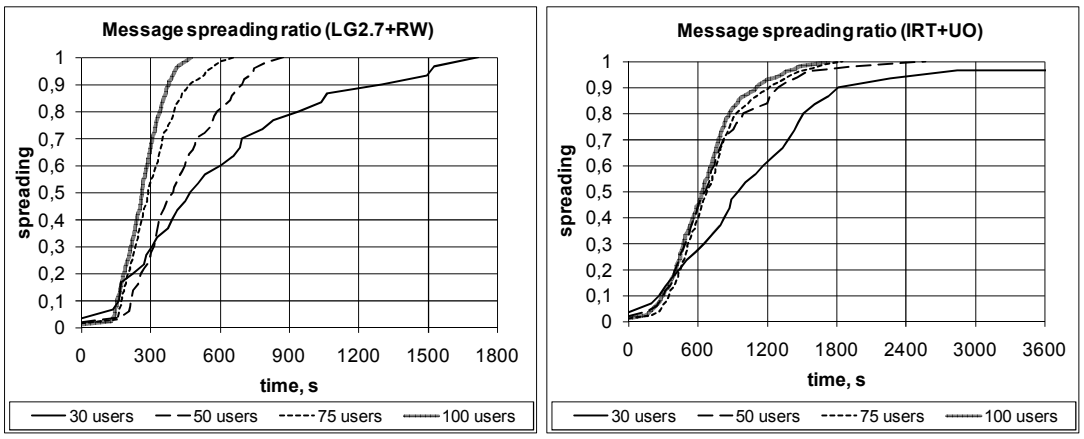


Figure 5.30: Message spreading ratio for different numbers of mobile users, neighbor discovery interval is 5 minutes

Figure 5.29 and Figure 5.30 show the message spreading results for different setups with LG2.7+RW and IRT+UO models. Clearly, the shorter is the neighbor discovery interval, more often mobile nodes synchronize their newsgroup message databases. Consequently, new messages spread faster among the nodes. We also see that the messages are distributed equally fast for LG2.7+RW with 1 and 2 minutes neighbor discovery intervals. In these cases, the discovery interval is short enough to spread the messages very fast. The messages are disseminated nearly immediately as soon as the message originator performs its next neighbor discovery broadcast.

Besides, the messages spread faster in the scenarios with more mobile nodes, which can be easily observed in LG2.7+RW results. Newsgroup messages are disseminated much slower for IRT+UO. Depending on a scenario, the spreading takes 2-4 times more in IRT+UO than in LG2.7+RW.

Other studies in this chapter have already pointed out a negative impact of obstacles in IRT, which cause network topologies to change more often. To investigate this issue more, we also examine the times that mobile nodes stay in the transmission range of each other (contact times, Figure 5.31). We see that in IRT+UO, contact times are below 30 seconds in more than 60% of cases. The contacts are much longer in LG2.7+RW, therefore, the nodes have more time to exchange their messages. According to application logs, communication errors occur more often in IRT+UO, which are caused by socket timeouts when the peer node unexpectedly moves out the transmission range. This explains the differences in message spreading results among IRT+UO and LG2.7+RW. More frequent topology changes (and consequently, shorter contact times) also limit the application performance in IRT+UO scenarios with more mobile nodes. Therefore, the results are nearly equal for IRT+UO in the scenarios with 50, 75, and 100 mobile nodes.

The results in Figure 5.32 and Figure 5.33 show the energy consumption for mobile devices. The results are presented both in Joules and in percent of the iPAQ battery capacity. We again see a difference between the simpler and more realistic models. The network connectivity is higher without obstacles. Hence, more mobile devices receive the broadcasted neighbor discovery packets. This causes more SOAP calls being issued, and consequently, a 2-4 times higher energy consumption in LG2.7+RW comparing to IRT+UO. For the same model, shorter neighbor discovery intervals cause more communication among the nodes, thus increasing the overall energy consumption. The energy consumption is also higher in the scenarios with more mobile nodes, since the newsgroup databases then need to be synchronized with more nodes.

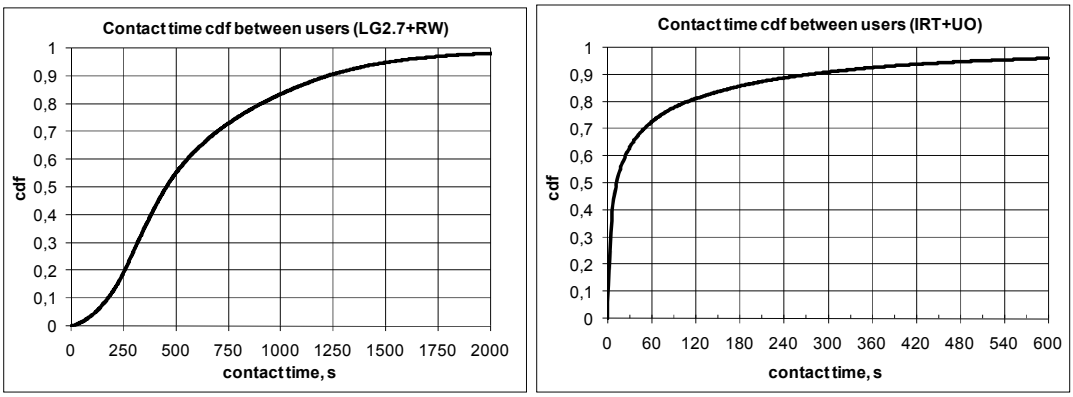


Figure 5.31: Cumulative distribution function of contact times between mobile users

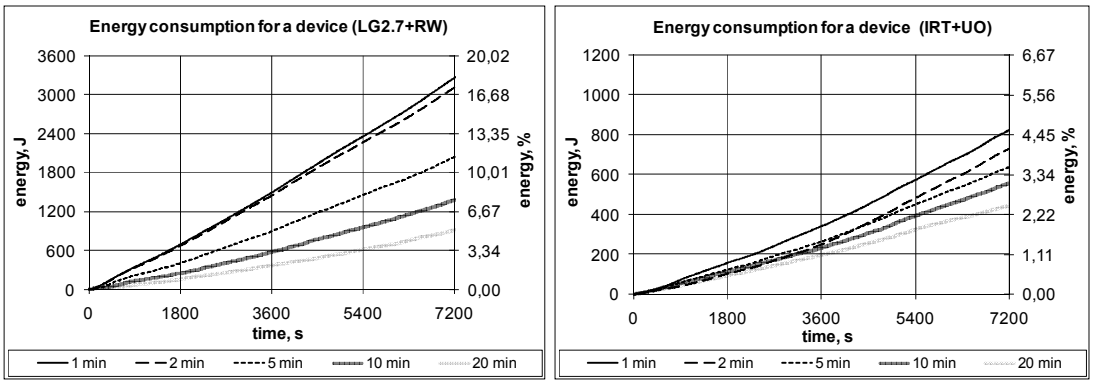


Figure 5.32: Average energy consumption for a device, different neighbor discovery intervals

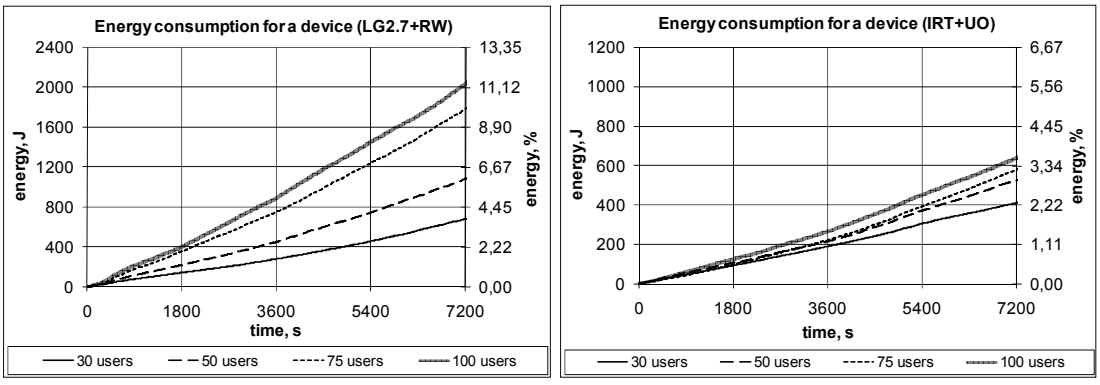


Figure 5.33: Average energy consumption for a device, different numbers of mobile users, neighbor discovery interval is 5 minutes

The simulation results obtained with IRT+UO give us more realistic impression of Usenet-on-the-Fly performance and energy consumption in a city scenario. The results show that a newsgroup message reaches 90% of users within an hour even with a 20 min neighbor discovery interval. Shorter intervals lead to a faster neighbor discovery, and hence, to a faster message exchange at a cost of higher energy consumption. For instance, an interval of 1 min requires about 4.5% of the device energy spent on communication in 2 hours, while a 20 min interval requires about 2.5%. Clearly, the discovery interval must be configured by taking the desired message spreading delay and the energy consumption into account.

5.5 Discussion and Summary

In this chapter, the more realistic user-oriented mobility model (UO) and the intelligent ray tracing model (IRT) were compared against the random waypoint mobility model (RW), the two-ray ground model (TRG), and the log-distance path-loss model with β=2.7 (LG2.7). The latter are default models in network simulation tools.

The RW model simply assumes the movement between the randomly chosen points of the area. Neither spatial environment nor user travel decisions, like certain trip sequences or path selections, are considered by this model. In contrast, UO relies on the spatial environment, so it takes area constraints into account upon the movement path construction. The model also reflects user trips between the points of interest. Therefore, mobile users move between meaningful locations like shops, restaurants, and museums, and not just between the randomly chosen places of the rectangular area. To better mimic the dynamics of pedestrian movement, we apply the model of Helbing and Molnár [HM95], which is based on real-world observations. Obviously, the consideration of users moving along streets with more realistic trips and dynamics makes the movement in the user-oriented mobility model closer to reality.

TRG makes rather simplified assumptions regarding the radio wave propagation. It assumes a path between the transmitter and the receiver that is clear from obstacles.

Consequently, the distance between the transmitter and the receiver is the only dynamic parameter of this model. The communication range can be represented by a perfect disc around the transmitter. All the nodes residing within this disc receive the signals from that transmitter. Another default radio propagation model, which is the log-distance model, uses a path-loss exponent to shorten the maximum transmission range like in obstructed areas. The communication range of a node in the log-distance model is also a circle, but just of a smaller radius.

The more realistic intelligent ray tracing model uses ray tracing for finding possible radio propagation paths between the sender and the receiver. Unlike TRG and LG2.7, it relies on the map of the area and takes detailed model of the spatial environment into account. The accuracy of the model is proven by measurements in European cities. For Stuttgart city center (Germany), which is used in this chapter, the mean error is 0.3 dB and the standard deviation is 5.8 dB. To make the simulation results even more realistic, we combine this intelligent ray tracing model with the small-scale fading model and a bit-error model of frame reception. Both the small-scale fading model and the bit-error models are correlated with measurements in related studies.

We see that the application of the more realistic models changes network topologies in a simulation. The application of the user-oriented mobility model affects a distribution of mobile nodes in the area. The spatial environment limits possible locations of nodes in UO, therefore, the nodes become located closer to each other. This results in more network connections in the topology graph. In RW, mobile nodes are more evenly distributed in the area, which decreases the total number of connections. This also reduces a chance of network partitioning in RW, except for high transmission speeds (5.5 and 11 Mbps) with relatively smaller numbers of mobile nodes (50 nodes at 5.5 Mbps; 50, 75, 100 nodes at 11 Mbps.). These scenarios profit more from the constrained movement area in UO.

Obstacles of the spatial environment block certain connections in the intelligent ray tracing model. This decreases the total number of network connections in the topology compared to TRG. The LG2.7 shortens its transmission range in order to approximate the propagation in obstructed areas. However, then the propagation in

open areas is underestimated. This causes differences between LG2.7 and IRT topologies: some connections are missing in one graph and some are missing in another. At higher transmission speeds (5.5 and 11 Mbps), there are more connections that are missing in LG2.7 due to its much shorter transmission range.

Obstacles also cause additional network partitions to appear in IRT. These partitions are formed by relatively small amount of nodes that reside behind obstacles. The number of such partitions increases as more mobile nodes are used in a simulation. With more nodes, there is basically a higher chance that some of the nodes will be blocked by obstacles. These partitions cannot be reflected by simple empirical models, and hence, is a characteristic property of the more realistic radio propagation model. In contrast, the number of partitions for LG2.7 always decreases as more nodes are used in simulations. TRG topologies are nearly never partitioned due to its long transmission range.

Hence, the widely used TRG+RW combination delivers rather "optimistic" simulation results: there are more network connections and no network partitions, which is mainly due to the radio propagation model. The combination of LG2.7+RW shows rather "pessimistic" results as of the total number of connections in the network topology graph. We see that the results of the more realistic IRT+UO combination cannot be easily approximated by simple models. IRT+UO topologies are also more partitioned than LG2.7+RW at lower transmission speeds and with higher node densities. In turn, LG2.7+RW have more partitions at higher speeds and lower node densities.

The changes in network topologies affect the routing protocol and mobile application performance. In our evaluations, we used the AODV routing protocol, which combines the strengths of two other popular routing protocols for mobile ad-hoc networks: table-driven DSDV and on-demand DSR. As the mobile application, we used Usenet-on-the-Fly, which was taken unmodified and emulated by ns-2.

As of AODV, dense network topologies of TRG at 1 and 2 Mbps experience many packet collisions upon the flooding-based route discovery. This increases the routing packet overhead and hardens the discovery process, thus negatively impacting the data packet delivery ratios. This effect disappears at 5.5 and 11 Mbps, therefore,

nearly 100% of data packets are delivered in those TRG scenarios. Obstacles of the propagation environment harden route search and maintenance in IRT. Moreover, the obstacles make routing paths to change more often upon the movement of users, which is also confirmed by analyzing the times that mobile nodes spend in the transmission range of each other. Thus, IRT delivers fewer packets comparing to TRG and LG2.7 at 1, 2, and 5.5 Mbps, and shows the much higher values of the routing packet overhead and data packet delay among the models. Since TRG and LG2.7 neglect the obstacles of spatial environment, they are unable to reproduce this effect in simulations. LG2.7 has much shorter transmission range at 11 Mbps, therefore, the IRT model delivers more data packets in these scenarios.

The shortest routing paths are in TRG. In IRT, the routing paths must “bend” the obstacles, thus traversing more hops. Therefore, IRT has the largest routing path lengths among the three models at 1 Mbps. At higher transmission speeds, the shorter transmission range of LG2.7 has more impact on the results. For that reason, network packets need to traverse fewer hops in IRT at 2, 5.5, and 11 Mbps comparing to LG2.7. For empirical models, the results for data packet delay are directly dependent on the lengths of routing paths. In the more realistic IRT model, additional packet delays are introduced by more frequently changing network topologies.

Due to the changes in network partitioning, UO makes fewer packets being delivered at higher transmission speeds and with more mobile nodes (5.5 Mbps and 11 Mbps for 150 nodes). The scenarios at 11 Mbps with smaller number of nodes (50 and 75 nodes) profit more from the constrained movement area in UO. In general, more routing packets are sent in UO, however, the routing paths and the data packet delays are shorter.

The application of more realistic models also changes the results of Usenet-on-the-Fly evaluation. Depending on a scenario, it takes 2-4 times longer until a newsgroup message gets spread in the network. Since the network connectivity of IRT+UO is lower, less communication takes place among the nodes, which results in lower energy consumption for a mobile device.

The results in this chapter show that the impact of obstacles of the spatial environment and the changed distribution of mobile users cannot be easily imitated

by simple models. More realistic models are required for getting more realistic simulation results. This thesis describes such models, their integration into a network simulation tool, and demonstrates their practical usage for evaluations of mobile networks.

6 Conclusion and Future Work

Simulation tools are frequently used for the performance analysis of mobile networks. Their common shortcoming lies within the approaches they use for the modeling of user mobility and radio wave propagation. These models do not take the spatial environment into account, although it has a significant impact on network performance. In this thesis, more realistic models were described and integrated into a simulation tool for mobile networks.

At first, geospatial data standards were analyzed. They serve for specifying spatial objects such as the movement constraints and obstacles in digital maps. The overview of commonly used geospatial standards showed their conceptual similarities. This allowed for the creation of generic geographic data model, which supports diverse geographic data formats and provides spatial information to the network simulation. This model is used by the described mobility and radio propagation modeling approaches. The implementation of the generic geographic model also provides parsers for some geospatial standards. This allows using existing data sources in network simulations, thus reducing the overhead for defining a custom geospatial model for the target scenario.

Next, mobility modeling in network simulations was discussed. It was shown that most of studies rely on rather simple models. These models neglect spatial environments, assume totally random trips in the area, or oversimplify a user movement dynamics. Therefore, a more realistic user-oriented mobility model was introduced. Our model reflects the main factors that impact a user behavior in real life, such as a spatial environment, user travel behavior, and user movement dynamics. Consequently, the model consists of three sub-models: the spatial model, the user trip model, and the movement dynamics model. The spatial model provides a description of the movement area. It is automatically generated from the generic geographic model. The user trip model reflects user travel decisions. It performs the modeling of user trips and movement path selections. The user trips are modeled

with the help of activity-based travel demand modeling approach [Kit96], which defines travel as the demand to participate in activities. The movement path selections are modeled with the help of shortest-path choice or probabilistic path choice approaches. The corresponding algorithms were described along with their integration into the proposed mobility model. The movement dynamics model reflects the user position changes along the chosen path. It is based on the approaches from traffic modeling. The user-oriented mobility model includes different movement dynamics models to support the modeling of pedestrians and vehicles in diverse scenarios. The model is implemented with all sub-models (and with the generic geographic model and geospatial data parsers) in a framework for user mobility modeling CANUMobiSim. It also includes the module of parameter derivation, which helps to obtain some trip model parameters from user position traces.

In the next part, physical layer modeling for mobile networks was discussed. Network simulation tools commonly offer only simple empirical models. Such models, like the free space model or the two-ray ground model, normally assume a clear path between the transmitter and the receiver. Therefore, the distance between the transmitter and the receiver is the only dynamic parameter of these models. The communication range can be represented by a perfect disc around the transmitter. All the nodes residing within this disc receive the signals from that transmitter. These models deliver good results in open areas, but become unrealistic in many scenarios, like a city center, in which buildings significantly affect mobile communication, or in indoor scenarios with walls. The log-distance model is another commonly used model. It uses a path-loss exponent to shorten the transmission range like in obstructed areas. However, this underestimates the open space propagation. The communication range of a node in the log-distance model is also a circle, but just of a smaller radius.

Ray optical models rely on ray tracing or similar techniques from geometric optics. They consider a map of the propagation area to determine all possible signal paths between the transmitter and the receiver. These models deliver very accurate results, however, require much longer computation time. We integrated the intelligent ray

tracing model into network simulation. This model was developed by our colleagues at the Institute of Radio Frequency Technology, Universität Stuttgart. Due to many optimizations, it performs 600-1600 times faster than the standard ray tracing approach. The accuracy of the model is proven by measurements in European cities. For Stuttgart city center (Germany), which we used in the evaluation part of this thesis, the mean error is 0.3 dB and the standard deviation is 5.8 dB. To integrate the intelligent ray tracing model into a simulation, we precompute the receive power values for all possible transmitter-receiver pairs in a grid and to save them in a database. This eliminates the need of ray tracing during the simulation and makes the simulation time comparable to the time with an empirical model. Together with the small-scale fading model and a BER-based frame reception, which were also described, it allows for more realistic simulation studies of mobile networks in indoor and outdoor scenarios.

In the evaluations, a mobile network was simulated with both simple (default models that are used in network simulations) and the more realistic mobility and physical layer models. The results showed various differences in network topologies as well as in the performance of routing protocols and mobile applications. This basically means that the impact of obstacles of the spatial environment on the radio propagation and the changes in node distribution caused by mobility cannot be easily imitated by simple models. In spite of their complexity, e.g., more parameters or higher computation overhead, more realistic models must be used for getting more realistic simulation results. It was also shown that it is feasible to integrate such more realistic models into a network simulator.

The model implementations that are described in this thesis are freely available for download from the author's homepage[1]. They are utilized in different research projects by the Universität Stuttgart. Other institutions all over the world also use these implementations, like the AMADEOS project [SC05] of the Université de Sherbrooke, Canada, or VanetMobiSim [FHFB07], which is a joint project between the Institut Eurécom, France, and Politecnico Di Torino, Italy. It is interesting to add

[1] http://www.ipvs.uni-stuttgart.de?id=illya.stepanov&lang=en

that the CANUMobiSim was proven to be useful not only for the modeling of mobility of human beings. In [RW05], the authors exploit the simulator for generating movement patterns for cows, thus presenting a new aspect for using this tool.

For future research, one topic would be the integration of real travel survey or position trace data into a simulation. CANUMobiSim supports it by providing the described trip parameter derivation module. Since at the time this thesis was written we did not have the real data, we used random trip generation between certain points of the area. The application of survey data would make the produced mobility traces even more realistic. By sharing such simulation scenario in the community, we would establish a common scenario that researchers could use for comparing the performance of their algorithms, protocols, and mobile applications under more realistic conditions.

Another important research topic is storing the precomputed radio propagation data in a more efficient way. We compute the receive power values for the area with the intelligent ray tracing model and save them to files. The data are stored for every transmitter and receiver cell pair. The total size of data for the scenario used in this thesis is about 120 GB. However, many cells are located far away from each other, so the power of the received signal is a way below the receive threshold. Such signals can hardly be heard by the receiver and have nearly no impact on network simulation results. By eliminating such entries, we could substantially reduce our database size, thus possibly enabling finer grid granularities (better than 5 m × 5 m that is used in this thesis). Such finer granularities are required for short-range low-power devices.

List of Abbreviations

AODV Ad-hoc On-Demand Distance Vector Routing Protocol (a routing protocol used in mobile ad-hoc networks)

API Application Programming Interface

AWML Augmented World Modeling Language (used to model geospatial data in the Nexus project of the Universität Stuttgart)

BER Bit-Error Ratio (used in telecommunication for measuring a quality of data transmission)

BPR Bureau of Public Roads (US transportation system authority)

cdf Cumulative Distribution Function (used in probability theory to describe the probability distribution of a real-valued variable)

CRC Cyclic Redundancy Code (used in communication engineering for detecting data corruption)

DSDV Destination-Sequenced Distance-Vector Routing (a routing protocol used in mobile ad-hoc networks)

DSR Dynamic Source Routing (a routing protocol used in mobile ad-hoc networks)

FER Frame-Error Ratio (used in telecommunication for measuring a quality of data transmission)

GDF Geographic Data Files (standard of describing geospatial data)

GIS Geographic Information System (a computer system for storing and managing spatial data)

GML Geography Markup Language (standard of describing geospatial data)

GPS General Positioning System (Earth's satellite navigation system)

GTD Geometrical Theory of Diffraction (a method for computing wave propagation in inhomogeneous medium)

IRT Intelligent Ray Tracing Model (used in communication engineering for predicting a received signal power)

LAN Local Area Network

LGxx Log-Distance Path-Loss Model with β=xx (used in communication engineering for predicting a received signal power)

MANET Mobile Ad-Hoc Network

Mbps Megabit per Second (a unit of data transfer rate, which is equal to 1,000,000 bits per second)

pdf Probability Density Function (used in probability theory to describe the probability distribution of a real-valued variable)

RW Random Waypoint Mobility Model (a typical model for simulating user mobility in simulations of wireless networks)

SINR Signal-to-Interference-and-Noise Ratio (used in communication engineering for deciding on a signal reception)

SRS Spatial Reference System (used for specifying coordinates in geospatial data sources)

TCP Transmission Control Protocol (a transport layer protocol used in computer networks)

TRG Combination of Two-Ray Ground and Free Space Radio Propagation Models (used in communication engineering for predicting a received signal power)

UDA Urban Database ASCII (the format used for specifying building data in the tool WinProp)

UDP User Datagram Protocol (a transport layer protocol used in computer networks)

UHF Ultra High Frequency (a band of electromagnetic waves between 300 MHz and 3 GHz)

UML User-mode Linux (a modification of Linux kernel that allows it to be run as a user process under a Linux operating system)

UTD Uniform Theory of Diffraction (a method for solving electromagnetic scattering problems)

WGS World Geodetic System (used for specifying coordinates in geospatial data sources)

WLAN Wireless LAN

XML Extensible Markup Language (general purpose markup language for describing tree-structured information)

XSL Extensible Stylesheet Language (family of languages for describing formatting and transformation of XML-encoded documents)

XSLT XSL Transformations (language for transforming XML documents)

List of Figures

List of Tables

Bibliography

[Abr00] J. Abraham. Parameter Estimation in Urban Models: Theory and Application to a Land Use Transport Interaction Model of the Sacramento, California Region. Ph.D. Thesis, Department of Civil Engineering, University of Calgary, Canada, January 2000.

[AHL96] I. Akyildiz, S. Ho, and Y. Lin. Movement-Based Location Update and Selective Paging for PCS Networks. *IEEE/ACM Transactions on Networking*, vol. 4, pages 629-638, August 1996.

[Akç91] R. Akçelik. Travel Time Functions for Transport Planning Purposes: Davidson's Function, Its Time-Dependent Form and an Alternative Travel Time Function. *Australian Road Research*, vol. 21, no. 3, pages 49-59, 1991.

[AS03] D. Ashbrook and T. Starner. Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. *Personal Ubiquitous Computing*, vol. 7, pages 275-286, 2003.

[AWE03] AWE Communications. *WinProp – ProMan User Reference*, 2003.

[BAB99] M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to short-term travel decisions. In R. Hall, editor, *Handbook of Transportation Science*, International Series in Operations Research and Management Science, vol. 23, Kluwer, 1999.

[BBH02] C. Becker, M. Bauer, and J. Hähner. Usenet-on-the-fly: Supporting Locality of Information in Spontaneous Networking Environments. In *Proceedings of CSCW 2002 Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments*, New Orleans, USA, November 2002.

[BEF+00] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in

Network Simulation. *IEEE Computer*, vol. 33, no. 5, pages 59-67, May 2000.

[Bet01] C. Bettstetter. Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks. In *Proceedings of the 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM'01)*, Rome, Italy, July 2001.

[BMJ+98] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, USA, October 1998.

[BPR64] Bureau of Public Roads. *Traffic Assignment Manual.* U.S. Department of Commerce, Urban Planning Division, Washington D.C., USA, 1964.

[BRS03] C. Bettstetter, G. Resta, and P. Santi. The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pages 257-269, July-September 2003.

[CBD02] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pages 483-502, 2002.

[CEN95] CEN TC 278. *Geographic Data Files*, 1995.

[Chi96] C. Chiang. Wireless networks multicasting. Ph.D Thesis, University of California, Los Angeles, USA, 1996.

[CJB+01] 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 7th ACM International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001.

[COST91] COST 231. *Urban transmission loss models for mobile radio in the 900- and 1,800 MHz bands (Revision 2)*. TD (90) 119, September 1991.

[DCYS98] S. Das, R. Castaneda, J. Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile ad hoc networks. In *Proceedings of the 7th International Conference on Computer Communications and Networks (ICCCN)*, Lafayette, USA, October 1998.

[DFG95] E. Del Re, R. Fantacci, G. Giambene. Handover and Dynamic Channel Allocation Techniques in Mobilecellular Networks. *IEEE Transactions on Vehicular Technology*, vol. 44, iss. 2, pages 229-237, May 1995.

[Dia71] R. Dial. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, no. 5, 1971.

[Dij59] E. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1, pages 269-271, 1959.

[Dik00] J. Dike. A User-mode Port of the Linux Kernel. In *Proceedings of the 4th Annual Linux Showcase & Conference*, Atlanta, USA, October 2000.

[DOD01] Department of Defense. *Global Positioning System Standard Positioning Service Performance Standard*, 2001.

[DPR97] G. Durgin, N. Patwari, and T. Rappaport. An advanced 3D ray launching method for wireless propagation prediction. In *Proceedings of the 47th IEEE Vehicular Technology Conference* (VTC'97), Phoenix, USA, May 1997.

[Epp94] D. Eppstein. Finding the k shortest paths. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, Santa Fe, USA, 1994.

[FHFB07] M. Fiore, J. Härri, F. Fethi, and C. Bonnet. Vehicular mobility simulation for VANETs. In *Proceedings of the 40th IEEE Annual Simulation Symposium (ANSS'07)*, Norfolk, USA, March 2007.

[For98] S. Fortune. Efficient algorithms for prediction of indoor radio propagation. In *Proceedings of the 48th IEEE Vehicular Technology Conference (VTC'98)*, Ottawa, Canada, May 1998.

[Fri46] H. Friis. A note on a simple transmission formula. In *Proceedings of IRE*, vol. 41, pages 254–256, 1946.

[GJ99] T. Gibson and D. Jenn. Prediction and measurement of wall insertion loss. *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 1, January 1999.

[GKL04] I. Gruber, O. Knauf, and H. Li. Performance of Ad Hoc Routing Protocols in Urban Environments. In *Proceedings of European Wireless 2004 (EW'2004)*, Barcelona, Spain, February 2004.

[Gué87] R. Guérin. Channel Occupancy Time Distribution in a Cellular Radio System. *IEEE Transactions on Vehicular Technology*, vol. 36, iss. 3, pages 89-99, August 1987.

[GWBL95] B. Gschwendtner, G. Wölfle, B. Burk, and F. Landstorfer. Ray Tracing vs. Ray Launching in 3-D Microcell Modelling. In *Proceedings of the 1st European Personal and Mobile Communications Conference (EPMCC)*, Bologna, Italy, April 1995.

[HA95] J. Ho and I. Akyildiz. A Mobile User Location Update and Paging Mechanism under Delay Constrains. *Wireless Networks*, vol. 1, no. 4, pages 413-425, December 1995.

[HGL99] R. Hoppe, G. Wölfle, F. Landstorfer. Fast 3-D Ray Tracing for the Planning of Microcells by Intelligent Preprocessing of the Data Base. In *Proceedings of the 3rd European Personal and Mobile Communications Conference (EPMCC)*, Paris, France, March 1999.

[HM95] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, vol. 51, no. 5, pages 4282-4286, May 1995.

[HNR68] P. Hart, N. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, vol.4, no. 2, pages 100-107, 1968.

[Hor83] M. Horton. Standard for interchange of USENET messages. RFC 850, IETF, June 1983.

[Hor91] A. Horowitz. *Delay-Volume Relations for Travel Forecasting: Based on the 1985 Highway Capacity Manual*. Department of Civil Engineering and Mechanics, University of Wisconsin-Milwaukee, USA, 1991.

[HR86] D. Hong and S. Rappaport. Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Nonprioritized Handoff Procedures. *IEEE Transactions on Vehicular Technology*, vol. 35, iss. 3, pages 77-92, August 1986.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, USA, 1979.

[HWL03] R. Hoppe, P. Wertz, F.M. Landstorfer, and G. Wölfle. Advanced Ray Optical Wave Propagation Modelling for Urban and Indoor Scenarios Including Wideband Properties. *European Transactions on Telecommunications (ETT)*, no. 1, pages 61-69, November 2003.

[IEE99] IEEE. *802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[Int00] Intersil. *HFA3861B Direct Sequence Spread Spectrum Baseband Processor Data Sheet*. File Number 4816, January 2000.

[ISO04] ISO/TC 211/WG 4/PT 19136 and OGC. *Geographic information – Geography Markup Language (GML)*, 2004.

[IYU84] F. Ikegami, S. Yoshida, and M. Umehira. Propagation factors controlling mean field strength on urban streets. *IEEE Transactions on Antennas and Propagation*, vol. 32, no. 8, pages 822-829, August 1984.

[JBAS03] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, S. Suri. Towards Realistic Mobility Models for Mobile Ad Hoc Networks. In *Proceedings of the 9th Annual ACM International Conference on Mobile Computing and Networking (Mobicom 2003)*, San Diego, USA, September 2003.

[JHP+03] J. Jetcheva, Y. Hu, S. PalChaudhuri, A. Saha, and D. Johnson. Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture. In *Proceedings of the 5th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA)*, Monterey, USA, October 2003.

[JJ00] W. Jeon and D. Jeong. Performance of Improved Probabilistic Location Update Scheme for Cellular Mobile Networks. *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pages 2164-2173, November 2000.

[JLH+99] P. Johanson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'99)*, Seattle, USA, August 1999.

[JM96] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[KBH98] R. Kates, K. Bogenberger, and M. Hoops. Mesoscopic simulation with ANIMAL: Optimal utilization of downstream traffic detector data and the propagation of information. In M. Schreckenberg and D. Wolf, editors, *Traffic and Granular Flow 1997*, pages 453-468, Springer, Singapore, 1998.

[KBST04] U. Kubach, C. Becker, I. Stepanov, and J. Tian. Simulation Model and Tool for Mobile Location Dependent Information Access. In M. Ilyas and I. Mahgoub, editors, *Mobile Computing Handbook*, CRC Press, New York, 2004.

[Khi98] C. Khisty. *Transportation Engineering: an Introduction*. Prentice Hall, Upper Saddle River, USA, 2nd edition, international edition, 1998.

[Kit96] R. Kitamura. Applications of Models of Activity Behavior for Activity Based Demand Forecasting. In *Proceedings of Activity-Based Travel Forecasting Conference*, New Orleans, USA, June 1996.

[KNG+04] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental Evaluation of Wireless Simulation Assumptions. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04)*, Venice, Italy, October, 2004.

[Kra97] S. Krauß. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. Ph.D. Thesis, University of Cologne, Cologne, Germany, 1997.

[KV98] Y. Ko and N. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, USA, October 1998.

[LB98] T. Liu and P. Bahl. Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks. *IEEE Journal on Special Areas in Communications: Special Issue on Wireless Access Broadband Networks*, vol. 16, no. 6, August 1998.

[LFR96] O. Landron, M. Feuerstein, T. Rappaport. A comparison of theoretical and empirical reflection coefficients for typical exterior wall surface in a mobile radio environment. *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 3, March 1996.

[Li99] M. Li. Asymmetrically-Distributed Variations in Traveler-Perceived Travel Times in Stochastic User-Equilibrium Traffic Assignment. Ph.D. Thesis, University of Florida, Florida, USA, 1999.

[LM95] G. Liu and G. Maguire. Efficient Mobility Management Support for Wireless Data Services. In *Proceedings of the 45th IEEE Vehicular Technology Conference (VTC'95)*, Chicago, USA, July 1995.

[LR99] E. Lieberman and A. Rathi. Traffic Simulation. In N. Gartner, C. Messer, and A. Rathi, editors, *Traffic flow theory: a monograph*, chapter 10, Springer-Verlag, 1999.

[MI05] D. Mahrenholz and S. Ivanov. Adjusting the ns-2 Emulation Mode to a Live Network. In *Proceedings of Communication in Distributed Systems 2005 (KiVS'05)*, Kaiserslautern, Germany, March 2005.

[Mic04] Microsoft. *Microsoft Encarta Dictionary: The First Dictionary for the Internet Age*. St. Martin's Paperbacks, New York, USA, rev. ed. 2004.

[MMB01] M. Takai, J. Martin, and R. Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc'01)*, Long Beach, USA, October 2001.

[MMLR05] D. Minder, P. Marrón, A. Lachenmann, K. Rothermel. Experimental construction of a meeting model for smart office environments. In *Proceedings of the First Workshop on Real-World Wireless Sensor Networks (REALWSN'05)*, Stockholm, Sweden, June 2005.

[MW02] Merriam-Webster. *Merriam-Webster's Collegiate Dictionary*. Merriam-Webster, Inc., 10th edition, 2002.

[Nel67] E. Nelson. *Dynamical Theories of Brownian Motion*. Princeton University Press, 1967.

[New93] G. Newell. A simplified theory of kinematic waves in highway traffic. *Transportation Research*, vol. 27b, no. 4, pages 281-313, 1993.

[NG03] J. Nuevo and J. Grégoire. Analysis of the Effects of Entity Mobility Models on Ad Hoc Network Communication. In *Proceedings of the 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'03)*, Montreal, Canada, July 2003.

[NIMA00] National Imagery and Mapping Agency, Department of Defense World Geodetic System. 1984, Its Definition and Relationships with Local Geodetic Systems. Technical Report 8350.2, 3rd edition, 2000.

[NM01] D. Nicklas and B. Mitschang. The Nexus Augmented World Model: An Extensible Approach for Mobile, Spatially-Aware Applications. In *Proceedings of the 7th International Conference on Object-Oriented Information Systems (OOIS'01)*, Calgary, Canada, August 2001.

[Opp95] N. Oppenheim. *Urban Travel Demand Modeling: From Individual Choices to General Equilibrium*. Wiley-Interscience, New York, USA, 1995.

[OS01] Ordnance Survey. *OS MasterMap™ Real-World Object Catalogue*, 2001.

[Ove67] K. Overgaard. Urban Transportation Planning: Traffic Estimation. *Traffic Quarterly*, pages 197-218, 1967.

[OW01] J. Ortuzar and L. Willumsen. *Modelling Transport*. Wiley & Sons, New York, USA, 2001.

[Pap83] M. Papageorgiou. Applications of Automatic Control Concepts to Traffic Flow Modeling Control. In A. Balakrishnan and M. Thoma, editors, *Lecture Notes in Control and Information Sciences*, vol. 50. Springer-Verlag, Berlin, Germany, 1983.

[Pas96] E. I. Pas. Recent Advances in Activity-Based Travel Demand Modeling. In *Proceedings of Activity-Based Travel Forecasting Conference*, New Orleans, USA, June 1996.

[PNS00] R. Punnoose, P. Nikitin, and D. Stancil. Efficient Simulation of Ricean Fading within a Packet Simulator. In *Proceedings of the 52nd IEEE Vehicular Technology Conference (VTC'00)*, Boston, USA, September 2000.

[Pos80] J. Postel. User Datagram Protocol. RFC 768, IETF, August 1980.

[Pos81] J. Postel. Transmission Control Protocol. RFC 793, IETF, September 1981.

[PB94] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proceedings of the ACM SIGCOMM '94 Conference on

Communications Architectures, Protocols and Applications (SIGCOMM'94), London, United Kingdom, 1994.

[PR99] C. Perkins and E. Royer. Ad Hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, USA, February 1999.

[Pro03] Proxim Corporation. *Orinoco® 11b Client PC Card Specification*, 2003.

[PZ75] B. Pushkarev and J. Zupan. *Urban Space for Pedestrians*. The MIT Press, Cambridge, USA, 1975.

[Ril03] G. Riley. The Georgia Tech Network Simulator. In *Proceedings of ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools'03)*, Germany, August 2003.

[RMK97] T. Rappaport, R. Muhamed, and V. Kapoor. Propagation Models. In J. Gibson, editor, *The Communications Handbook*, chapter 84, pages 1182-1196, CRC Press, 1997.

[RMM+00] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz. CarTALK 2000 - Safe and Comfortable Driving Based Upon Inter-Vehicle-Communication. In *Proceedings of IEEE Intelligent Vehicle Symposium*, Versailles, France, June 2002.

[Rup00] E. Ruppert. Finding the k Shortest Paths in Parallel. *Algorithmica*, vol. 28, no. 2, pages 242-254, 2000.

[RW05] M. Radenkovic and B. Wietrzyk. Increasing Scalability and Reliability of Self-Organizing Grids for Life Science. In *Poster Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005)*, Melbourne, Australia, December 2005

[RWH02] T. Rautiainen, G. Wölfle, and R. Hoppe. Verifying Path Loss and Delay Spread Predictions of a 3D Ray Tracing Propagation Model in Urban Environments. In *Proceedings of the 56th IEEE Vehicular*

Technology Conference (VTC'02) 2002 - Fall, Vancouver, Canada, September 2002.

[Sch95] M. Schopp. User Modelling and Performance Evaluation of Distributed Location Management for Personal Communications Services. In *Proceedings of the 15th International Teletraffic Congress (ITC'15)*, Washington, D.C., USA, June 1997.

[SC05] A. Souley and S. Cherkaoui. Advanced mobility models for ad hoc network simulations. In *Proceedings of Systems Communications 2005 (ICW'2005)*, Montreal, Canada, August 2005.

[SDR92] K. Schaubach, N. Davis, and T. Rappaport. A Ray Tracing Method for Predicting Path Loss and Delay Spread in Microcellular Environments. In *Proceedings of the 42nd IEEE Vehicular Technology Conference (VTC'92)*, Denver, USA, May 1992.

[SHB+03] I. Stepanov, J. Haehner, C. Becker, J. Tian, and K. Rothermel. A Meta-Model and Framework for User Mobility in Mobile Networks. In *Proceedings of the 11th IEEE International Conference on Networks 2003 (ICON 2003)*, Sydney, Australia, October 2003.

[SHR05] I. Stepanov, D. Herrscher, and K. Rothermel. On the Impact of Radio Propagation Models on MANET Simulation Results. In *Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communications Networks (MWCN 2005)*, Marrakech, Morocco, September 2005.

[SK99] J. Scourias and T. Kunz. An Activity-Based Mobility Model and Location Management Simulation Framework. In *Proceedings of the Second ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM'99)*, Seattle, USA, August 1999.

[SMR05] I. Stepanov, P. Marrón, and K. Rothermel. Mobility Modeling of Outdoor Scenarios for MANETs. In *Proceedings of the 38th Annual Simulation Symposium (ANSS'38)*, San Diego, USA, April 2005.

[Spi90] H. Spiess. Conical Volume-Delay Functions. *Transportation Science*, vol. 24, no. 2, 1990.

[SR06a] I. Stepanov and K. Rothermel. On the Impact of a More Realistic Physical Layer on MANET Simulations Results. *Ad-Hoc Networks*, vol. 6, iss. 1, pages 61-78, January 2008.

[SR06b] I. Stepanov and K. Rothermel. Simulating Mobile Ad-Hoc Networks in City Scenarios. In *Proceedings of the 4th International Conference on Wired/Wireless Internet Communications (WWIC 2006)*, Bern, Switzerland, May 2006.

[SR07] I. Stepanov and K. Rothermel. Simulating Mobile Ad-Hoc Networks in City Scenarios. *Computer Communications*, vol. 30, iss. 7, pages 1466-1475, May 2007.

[THB+02] J. Tian, J. Haehner, 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'35)*, San Diego, USA, April 2002.

[THH00] M. Treiber, A. Hennecke, and D. Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, vol. 62, iss. 2, pages 1805-1824, 2000.

[TJ99] I-F. Tsai and R.-H. Jan. The Lookahead Strategy for Distance-Based Location Tracking in Wireless Cellular Networks. *ACM Mobile Computing and Communications Review*, vol. 3, pages 27-38, 1999.

[Tob77] R. Tobin. An Extension of Dial's Algorithm Utilizing a Model of Tripmakers' Perceptions. *Transportation Research*, vol. 11, 1977.

[Tra89] Tranplan Associates. *Waterloo Region Travel Survey 1987: An Overview of the Survey Findings*. Regional Municipality of Waterloo, Department of Planning and Development, 1989.

[Tra03] K. Train. *Discrete Choice Methods with Simulation.* Cambridge University Press, 2003.

[VGH+02] S. Volz, M. Grossmann, N. Hönle, D. Nicklas, and T. Schwarz. Integration mehrfach repräsentierter Straßenverkehrsdaten für eine

föderierte Navigation (Integration of Road Data in Multiple Representations for Federated Navigation). *it - Information Technology*, vol. 44, iss. 5, 2002.

[W3C99] W3C. *XSL Transformations (XSLT) 1.0*, 1999.

[W3C04] W3C. *Extensible Markup Language (XML) 1.0 (Third Edition)*, 2004.

[WA04] X. Wu and A. Ananda. Link Characteristics Estimation For IEEE 802.11 DCF Based WLAN. In *Proceedings of the 29th Annual IEEE Conference on Local Computer Networks (LCN 2004)*, Tampa, USA, November 2004.

[Wal96] V. Walter. Zuordnung von raumbezogenen Daten - am Beispiel von ATKIS und GDF. Ph.D. Thesis, Deutsche Geodätische Kommission, Reihe C, Nr. 480, München, Deutschland, 1996.

[WB88] J. Walfisch and H. Bertoni. A theoretical model of UHF propagation in urban environments. *IEEE Transactions on Antennas and Propagation*, vol. 36, no. 12, pages 1788-1796, December 1988.

[Wei93] U. Weidmann. Transporttechnik der Fußgänger. *Schriftenreihe des Instituts für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau*, Nr. 90, ETH, Zürich, Switzerland, 1993.

[WGL97] G. Wölfle, B. Gschwendtner, and F. Landstorfer. Intelligent Ray Tracing - A New Approach for the Field Strength Prediction in Microcells. In *Proceedings of the 47th Vehicular Technology Conference (VTC'97)*, Phoenix, USA, May 1997.

[WHBL00] G. Wölfle, R. Hoppe, T. Binzer, and F. Landstorfer. Radio Network Planning and Propagation Models for Urban and Indoor Wireless Communication Networks. In *Proceedings of Millennium Conference on Antennas & Propagation (AP2000)*, Davos, Switzerland, 2000.

[WHL99] G. Wölfle, R. Hoppe, and F. Landstorfer. A Fast and Enhanced Ray Optical Propagation Model for Indoor and Urban Scenarios, Based on an Intelligent Preprocessing of the Database. In *Proceedings of the 10th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'99)*, Osaka, Japan, September 1999.

[Woe00] G. Wölfle. Propagation Models for Indoor Radio Network Planning Including Tunnels. In *Proceedings of Millennium Conference on Antennas & Propagation (AP2000)*, Davos, Switzerland, April 2000.

[WSS+03] J. Wolf, S. Schönfelder, U. Samaga, M. Oliveira, and K. Axhausen. Eighty Weeks of GPS Traces: Approaches to Enriching Trip Information. In *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington D.C., USA, January 2003.

[ZBG98] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS'98)*, Banff, Canada, May 1998.

[ZD97] M. Zonoozi and P. Dassanayake. User Mobility Modeling and Characterization of Mobility Patterns. *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pages 1239-1252, September 1997.