



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Waldemar Spät

Entwurf und Realisierung einer  
Verarbeitungskette für reale Internet  
Topologiedaten und deren Anwendung in der  
Strukturanalyse von Multicast Verteilbäumen

Waldemar Spät  
Entwurf und Realisierung einer Verarbeitungskette  
für reale Internet Topologiedaten und deren  
Anwendung in der Strukturanalyse von Multicast  
Verteilbäumen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Thomas Schmidt  
Zweitgutachter : Prof. Dr. rer. nat. Bettina Buth

Abgegeben am 7. August 2007

**Waldemar Spät**

**Thema der Bachelorarbeit**

Entwurf und Realisierung einer Verarbeitungskette für reale Internet Topologiedaten und deren Anwendung in der Strukturanalyse von Multicast Verteilbäumen

**Stichworte**

Internet Topologiedaten, Internet Topologiegeneratoren, Multicast Verteilbäume, Mobile IP

**Kurzzusammenfassung**

Täglich steigt die Anzahl an Benutzern, die sich im Internet befinden. Dabei entstehen neue Angebote und Dienstleistungen, die die Infrastruktur des Internets erweitern. Je komplexer das Internet wird, umso wichtiger ist es, den Aufbau anhand von Messungen zu analysieren und die Verteilung der Routen zu optimieren. In dieser Arbeit wurde eine Verarbeitungskette für reale Internet Topologiedaten erstellt, um mit Analyse- und Simulationswerkzeugen auf aktuelle Meßdaten einfach zugreifen zu können. Diese Daten werden in eine im Rahmen dieser Arbeit erweiterte Version von *BRITE* eingelesen, verarbeitet und gefiltert, so dass anschließend eine Analyse in anderen Werkzeugen wie z.B. der Netzwerksimulationsplattform *OMNeT++* stattfinden kann. Mit diesen Werkzeugen wurde in Abhängigkeit der Netzwerkgröße, die Stabilität von Multicast Verteilbäumen unter Mobilität der Sender sowie Empfänger ermittelt.

**Waldemar Spät**

**Title of the paper**

Design and implementation of a processing chain for real Internet data and its application in a structural analysis of multicast distribution trees

**Keywords**

Internet topology data, Internet topology generators, multicast distribution trees, mobile IP

**Abstract**

The number of Internet users increases daily. At the same time, new services are created and extend the Internet infrastructure. The more complex the Internet evolves, the more important does it become to analyze its structure based on measurements for optimizing the routing. In this thesis a processing chain for real Internet data has been created to improve the access of latest measurements for analysis and simulations. These data are managed by an extended version of *BRITE*, processed and filtered in such a way that they can be subsequently analyzed, e.g., in *OMNeT++*. With the help of this toolchain, the stability of multicast distribution trees under mobility have been analysed, thereby regarding multicast sender and receiver movement.

## **Danksagung**

An erster Stelle möchte ich Herrn Prof. Dr. Thomas Schmidt für die hervorragende Betreuung dieser Arbeit danken. Obwohl er in den letzten Wochen dieser Arbeit in den USA war, konnte ich ihn jederzeit per Email erreichen und erhielt innerhalb kurzer Zeit Feedback.

Ein großer Dank geht an Matthias Wählisch, der mich bei jeglichen Hardwareproblemen sowie bei den Programmen zur Analyse der Multicast-Verteilbäume unterstützt hat.

Der Lerngruppe bestehend aus Boris Woscidlo, Manuel Schneider, Stefan Gehn, Stefan Tamm und Sven Tennstedt verdanke ich, dass es überhaupt zu dieser Arbeit kommen konnte.

Auch möchte ich mich bei Christian Maschmann bedanken, mit dem ich zusammen alle Praktika gemeistert habe.

Nicht vergessen möchte ich meine Familie, die mich während des Studiums unterstützt hat.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>8</b>
<b>Abbildungsverzeichnis</b>	<b>9</b>
<b>1 Einleitung</b>	<b>10</b>
1.1 Motivation . . . . .	10
1.2 Zielsetzung . . . . .	11
1.3 Gliederung der Arbeit . . . . .	13
<b>2 Erfassung von charakteristischen Eigenschaften des Internets</b>	<b>14</b>
2.1 Die Internet Topologie . . . . .	14
2.1.1 Topologie-Modelle . . . . .	14
2.1.2 Reale Internet-Topologiedaten . . . . .	16
2.1.3 Das Problem der Messung . . . . .	17
2.2 CAIDA . . . . .	18
2.3 Mercator . . . . .	19
2.4 DIMES . . . . .	20
2.5 Diskussion . . . . .	21
<b>3 Entwurf einer Verarbeitungskette</b>	<b>22</b>
3.1 Anforderungen . . . . .	22
3.2 Konzeption . . . . .	23
<b>4 Eine Verarbeitungskette für reale Internet Topologiedaten</b>	<b>25</b>
4.1 BRITE . . . . .	25
4.1.1 Architektur . . . . .	27
4.1.2 Daten-Import . . . . .	27
4.1.3 Daten-Filter . . . . .	29
4.1.4 Daten-Export . . . . .	34
4.1.5 Konverter . . . . .	34
4.1.6 Benutzerschnittstelle . . . . .	34
4.2 OMNeT++ . . . . .	36

---

<b>5 Multicast Verteilbäume</b>	<b>38</b>
5.1 Mobile IP . . . . .	38
5.2 Multicast . . . . .	39
5.3 Multicast im Kontext der Mobilität . . . . .	40
5.3.1 Charakteristische Eigenschaften . . . . .	41
5.4 Analyse . . . . .	42
<b>6 Zusammenfassung und Ausblick</b>	<b>60</b>
<b>Literaturverzeichnis</b>	<b>61</b>

# Tabellenverzeichnis

- 4.1 Brite map sampling . . . . . 31
- 4.2 nem map sampling . . . . . 31
- 4.3 Unveränderte Internet-Topologien . . . . . 32

# Abbildungsverzeichnis

2.1	Router- sowie AS-Level . . . . .	15
2.2	Netzwerkstruktur - Quelle: [Watts und Strogatz (1998)] . . . . .	16
2.3	Beispiel für asymmetrisches Routing . . . . .	17
2.4	Beispielgrafik aus Skitter-Daten . . . . .	19
4.1	BRITE Architektur . . . . .	26
4.2	Überarbeitete BRITE Architektur . . . . .	27
4.3	Vergleich zwischen BRITE und nem 'map sampling' mittels des Rangzahl und Grad Korrelationskoeffizienten . . . . .	32
4.4	Vergleich zwischen BRITE und nem 'map sampling' mittels des Baumrangzahl und Baumgröße Korrelationskoeffizienten . . . . .	33
4.5	FileConverter GUI . . . . .	35
4.6	Filter-Einstellungen . . . . .	36
4.7	BRITE Hauptprogramm GUI . . . . .	37
5.1	Verteilung eines Pakets über Unicast (links) vs. Versendung über Multicast (rechts) . . . . .	39
5.2	Senderbewegung - Baum-Koinzidenz - pDR-nDR Distanz 5 Knoten . . . . .	44
5.3	Senderbewegung - neu entdeckte Router - pDR-nDR Distanz 5 Knoten . . . . .	45
5.4	Senderbewegung - Baum-Koinzidenz - 40 Empfänger . . . . .	46
5.5	Senderbewegung - neu entdeckte Router - 40 Empfänger . . . . .	47
5.6	Senderbewegung - Baum-Koinzidenz - pDR-nDR Distanz 10 Knoten . . . . .	48
5.7	Senderbewegung - neu entdeckte Router - pDR-nDR Distanz 10 Knoten . . . . .	49
5.8	Senderbewegung - Baum-Koinzidenz - pDR-nDR Distanz 15 Knoten . . . . .	50
5.9	Senderbewegung - neu entdeckte Router - pDR-nDR Distanz 15 Knoten . . . . .	51
5.10	Empfängerbewegung - Baum-Koinzidenz - pDR-nDR Distanz 5 Knoten . . . . .	53
5.11	Empfängerbewegung - neu entdeckte Router - pDR-nDR Distanz 5 Knoten . . . . .	54
5.12	Empfängerbewegung - Baum-Koinzidenz - pDR-nDR Distanz 10 Knoten . . . . .	55
5.13	Empfängerbewegung - neu entdeckte Router - pDR-nDR Distanz 10 Knoten . . . . .	56
5.14	Empfängerbewegung - Baum-Koinzidenz - pDR-nDR Distanz 15 Knoten . . . . .	57
5.15	Empfängerbewegung - neu entdeckte Router - pDR-nDR Distanz 15 Knoten . . . . .	58

# 1 Einleitung

## 1.1 Motivation

Wir befinden uns in einem Zeitalter, wo das Internet immer mehr an Bedeutung gewinnt. Die wirtschaftliche Globalisierung wird durch das E-Business vorangetrieben und das Internet verbreitet sich weltweit weiter. Dies ermöglicht es immer mehr Menschen, sich untereinander auszutauschen, unabhängig von ihrer Herkunft oder ihrem Standort. Informationen gewinnen durch ihre leichte Verteilbarkeit an globaler Bedeutung. Die Kommunikation mithilfe des Internets oder mobiler Geräte wird sowohl im privaten als auch im beruflichen Umfeld immer mehr zur Selbstverständlichkeit.

Nachdem das Internet in den 90er Jahren ein exponentielles Wachstum zeigte, hat sich dieses in den entwickelten Ländern verlangsamt. So beträgt das Wachstum von Internetzugängen in den Jahren 2000-2007 in Europa 199.5% und in Nord Amerika 115.7%. In den sich entwickelnden Kontinenten Asien mit 248.8%, Latein Amerika mit 433.4% sowie Afrika mit 638.4% findet immer noch ein großer Anstieg statt.<sup>1</sup>

Nicht nur das Internet ist in den letzten Jahren gewachsen, auch der Absatz mobiler Geräte nahm in den letzten Jahren stark zu. So wurde vom Marktbericht 'Telekommunikation 2006' prognostiziert, dass es Anfang 2007 erstmals mehr Mobiltelefone als Bewohner in Deutschland geben wird<sup>2</sup>. Die Geräte werden zudem immer vielseitiger benutzbar. Noch vor wenigen Jahren diente ein Mobiltelefon nur zur sprachlichen sowie auch schriftlichen Kommunikation zwischen zwei Personen. Inzwischen wird es als mobiles Büro und multimediales Gerät verwendet.

Spezifikationen wie IMS (IP Multimedia Subsystem) im 3GPP sorgen dafür, dass eine Standardisierung im Zugriff auf Dienste aus unterschiedlichen Netzwerken stattfinden kann. Dabei erfolgt die Kommunikation IP-basiert. Multimedia Broadcasting und Multicasting (MBMS) sind Bestandteil des UMTS Rel. 6. Aber auch ältere Netze wie GSM werden unterstützt. Der Zugang zum Internet über mobile Geräte wird sowohl einfacher als auch leistungsfähiger. Dienstleistungen und Spiele nutzen solche Funktionen und befördern die notwendige Akzeptanz gegenüber neuer Technik mit.

---

<sup>1</sup>Quelle: <http://www.internetworldstats.com/stats.htm>

<sup>2</sup>Quelle: <http://www.axelspringer.de/inhalte/pressese/inhalte/presse/5463.html>

Ein neuer Übertragungsstandard DVB-H (Digital Video Broadcasting for Handheld Terminals) bietet den mobilen Geräten die Möglichkeit zum Empfang von digitalen Rundfunkprogrammen. Über DVB-H können Multimedienetze wie Fernsehen für Mobiltelefone und andere kleine mobile Geräte über Rundfunk ausgesendet werden. Der Standard ist eine Weiterentwicklung von DVB-T, dem digitalen terrestrischen Antennenfernsehen. DVB-H erweitert DVB-T in 4 wesentlichen Punkten.

- IP Datacasting erlaubt die Nutzung von IP zur Übertragung der Audio- und Videoinhalte an auch selektive Gruppen von Empfängern per Multicast.
- Über Timescaling können die einzelnen Dienste in Blöcke gebündelt werden.
- Durch MPE-FEC ist ein optionaler zusätzlicher Fehlerschutz integriert.
- Neue physikalische Übertragungsmodi sind vorhanden (4k Modus und In-Depth Interleaver).

Zusätzlich können die Videos mit modernen Videokompressionsverfahren wie z.B. H.264/AVC kodiert werden.

So ist es nicht verwunderlich, dass Multicast im Kontext der Mobilität immer mehr an Bedeutung gewinnt. Zwar haben mobile Geräte die Funktionen eines kleinen Büros, aber ihre Bandbreite bleibt im Vergleich zu Festnetzanschlüssen zurück und ist auf effizientes Routing angewiesen. Multicast war ursprünglich nur auf den Einsatz in Netzwerken mit stationären Geräten zugeschnitten. Dabei ermöglicht Multicast eine effiziente und Netzwerk-Ressourcen schonende 1:n Kommunikation. Insbesondere für Anwendungen wie Video-Konferenzen, Softwareverteilung oder verteilte Spiele ist Multicast zur effizienten Kommunikation unersetzbar. Durch die Mobilität der Geräte, sei es nun in der Rolle eines Senders oder eines Empfängers von Multicastpaketen, muss auch der ursprüngliche Ansatz von Multicast für den Einsatz in der Umgebung mit mobilen Geräten neu überdacht werden.

## 1.2 Zielsetzung

Der Aufbau und die Struktur des wachsenden Internets wird immer komplexer. Um diese Struktur zu untersuchen, wird eine Reihe an Werkzeugen benötigt, die verschiedene Messungen durchführen können. Messungen sind erforderlich, da die Informationen über die Topologie des Internets nicht in Form von zentralen Daten gespeichert sind. Es wäre zwar möglich, aus den Informationen der Internet Provider solche Daten zu generieren, doch würde kein Provider Informationen über seine Infrastruktur darlegen. Was bleibt sind Messungen, die von verschiedenen Standorten durchgeführt werden müssen, um aussagekräftige

Informationen über diese Netze zu erhalten. Auch wenn dies bedeutet, dass nur ein Teil der ganzen Topologie dargestellt wird.

Reale Internetdaten ermöglichen Komplexitätsuntersuchungen des Internets. Aufgrund der Ergebnisse aus solchen Untersuchungen ist es möglich zu sehen, welche Leistungsfähigkeiten die Topologie hat und an welcher Stelle Verbesserungen notwendig sind. So ist es z.B. durch Messungen möglich, die Route eines Datenpakets zu verfolgen und durch Optimierung von Routing-Algorithmen zu verbessern.

Um dies zu erreichen, werden unterschiedliche Werkzeuge zur Datenerfassung sowie Verarbeitung benötigt. Hierzu muss eine Verarbeitungskette erstellt und soweit angepasst werden, dass die Daten ohne großen Aufwand von der Erfassung bis zur Analyse verwendet werden können.

Als ein Beispiel ist Multicast im Internet stark von den Routing-Algorithmen und dem Aufbau des Netzwerkes abhängig. Um effizient Pakete zu verteilen, muss ein Verteilbaum gebildet werden. Bei Multicast in Verbindung mit Mobilität kommen noch weitere Probleme dazu. So ist der Einsatz von Multicast unter Echtzeitanforderungen auf mobilen Geräten nicht unmittelbar möglich. Die Multicast Verteilbäume werden dynamisch aufgebaut und hängen zudem von der Netzwerkposition der Sender als auch der Empfänger ab. Sobald ein Subnetzwechsel der Teilnehmer stattfindet, müssen Teilbereiche, bzw. der ganze Baum neu aufgebaut werden, wobei die Errichtung von Multicast Bäumen zeitintensiv ist. Es muss gewährleistet sein, dass ein nahtloser Übergang der Multicast-Session stattfindet, sobald ein Teilnetzwechsel erfolgt. Multicast Routing passt sich dynamisch den Sitzungs-Topologien an, wobei sich diese bei Bewegung verändern können. Abhängig von der Routing-Topologie und dem Protokolls, erreicht die Routing-Konvergenz eine Zeit von Sekunden, aber auch Minuten und ist somit zu langsam, um einen nahtlosen Übergang für interaktive oder auch Echtzeit Anwendungen in Sessions zu ermöglichen.

Das Vorliegen realer Internetdaten ermöglicht uns, Komplexitätsuntersuchungen für den mobilen Multicast durchzuführen. Interessant sind hierbei die Ergebnisse in Bezug auf die Netzwerkgröße und der Senderbewegung. Dabei wird geschaut, wie sich der erstellte Verteilbaum in einem realen Netzwerk verhält, sobald die Sender, bzw. Empfänger ihre Position verändern.

## 1.3 Gliederung der Arbeit

Die Arbeit ist wie folgt aufgebaut: In Kapitel 2 (Erfassung von charakteristischen Eigenschaften des Internets) werden die Grundlagen und Problemstellungen zu der Gewinnung von realen Internet-Daten beschrieben. Zusätzlich werden noch Einrichtungen vorgestellt, die sich dem Thema gewidmet haben und Messungen in diesem Bereich durchführen.

Die Anforderungen sowie der Entwurf der Verarbeitungskette werden in Kapitel 3 (Entwurf einer Verarbeitungskette) vorgestellt.

Kapitel 4 (Eine Verarbeitungskette für reale Internet Topologiedaten) beinhaltet die Implementierung der Verarbeitungskette, die im vorangegangenen Kapitel dargestellt wurde. Dabei werden die jeweiligen verwendeten Werkzeuge vorgestellt und deren Erweiterungen ihrer Funktionen gezeigt.

In Kapitel 5 (Multicast Verteilbäume) werden zunächst die Grundlagen zu Multicast vorgestellt und anschließend werden die gewonnenen Ergebnisse, die durch die Verarbeitungskette entstanden sind, im mobilen Kontext bei der Sender- sowie Empfängerbewegung diskutiert.

Abschließend werden in dem Kapitel 6 (Zusammenfassung und Ausblick) Ergebnisse und Erkenntnisse, welche während der Arbeit gesammelt wurden, zusammengefasst und ein Ausblick auf mögliche Weiterentwicklungen geworfen.

## **2 Erfassung von charakteristischen Eigenschaften des Internets**

In diesem Kapitel werden die Problemstellungen und Möglichkeiten der Ermittlung von realen Internet Topologiedaten dargestellt.

Im ersten Teil dieses Kapitels wird erklärt, was genau reale Internet Topologiedaten sind und welchen Nutzen diese in der Praxis haben.

Einerseits möchte man solche Daten so genau wie möglich ermitteln und andererseits auch so schnell wie möglich. Diesem Problem haben sich einige Einrichtungen gewidmet und werden ebenfalls hier vorgestellt.

### **2.1 Die Internet Topologie**

Studien, die sich mit der Abbildung des Internets beschäftigen, haben typischerweise ihren Fokus auf einer Charakterisierung der Internet-Topologie sowie deren Leistungseigenschaft. Um aber eine Charakterisierung durchführen zu können, wird eine Sammlung an Daten benötigt. Diese Daten beinhalten die Internet-Knoten sowie deren Verbindungen zueinander. Anhand dieser Daten kann man einen Graphen erzeugen, der einen Teil des Internets sowie dessen Eigenschaften repräsentiert.

Bei der Ermittlung dieser Daten kommt es aber zu verschiedenen Problemen. Das Internet wächst täglich weiter und es entstehen dabei neu erreichbare Knoten. Allerdings fallen bekannte Knoten auch aus und sind somit nicht mehr erreichbar. Dadurch sehen die Daten aus einer Messung an verschiedenen Zeitpunkten auch anders aus.

#### **2.1.1 Topologie-Modelle**

Das Internet besteht nicht aus einem einzelnen Netzwerk, sondern wächst 'organisch' aus mehreren Netzwerken zusammen. Die dahinter stehende Internet Topologie folgt einem dezentralen hierarchischen Aufbau, bestehend aus Autonomen Systemen, Core- und Edge-

Routern. Die Autonomen Systeme stellen die verschiedenen delegierte Zonen vieler 'autonom' gerouteter Netzwerke dar. Die Core-Router sind für die Verbindungen innerhalb eines Netzwerks zuständig, die Edge-Router hingegen stellen einen Übergang zwischen zwei Netzen oder Domänen her. Modelle die dieses Bild umsetzen, sind Transit-Stub [Zegura u. a. (1997)] oder Tiers, in dem das Netzwerk aus mehreren Zufallsgraphen zusammengesetzt wird. Neben diesen hierarchischen Modell, gibt es noch das nicht-hierarchische Modell. Dieses wird durch Zufallsgraphen entweder auf AS-Level oder Router-Level abgebildet. Dabei sollte aber beachtet werden, dass das Internet aus unterschiedlichen Auflösungsstufen betrachtet werden kann [Heckmann u. a. (2003)]. Auf der Ebene der Autonomen Systeme stellt ein Autonomes System einen Knoten dar, auf der Router-Ebene hingegen stellt ein Router den Knoten dar. Die Semantik eines Knotens ist also dementsprechend unterschiedlich und sollte bei der Datenerfassung beachtet werden.

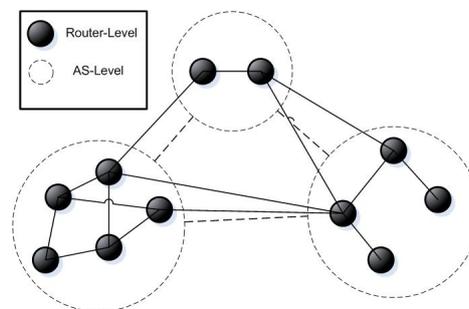


Abbildung 2.1: Router- sowie AS-Level

Bestimmte Eigenschaften eines Topologiemodells können Potenzgesetzen folgen. Hierbei kommt es zu Skaleninvarianz. So zeigt Faloutsos [Faloutsos u. a. (1999)], dass der Knotengrad von Autonomen Systemen einem Potenzgesetz folgt. Künstlich generierte Topologien versuchen diese Verteilung des Potenzgesetzes zu reproduzieren. Dabei liegt die Struktur eines realen Netzwerkes zwischen den Extremen eines regulärem und einem zufälligem Netzwerkes. Abbildung 2.2 zeigt das 'small world network' [Watts und Strogatz (1998)]. Bei einem regulärem Netzwerk ist die mittlere charakteristische Pfadlänge groß. Die Knoten sind beispielsweise jeweils mit dem direkten und übernächsten Nachbarn verbunden. Dabei entsteht eine starke Clusterbildung. Das zufällige Netzwerk hat im Gegensatz zu dem regulärem Netzwerk eine kurze charakteristische Weglänge, da die Kantenverteilung zufällig ist. Es entsteht wenn überhaupt nur eine sehr schwache Clusterbildung. Bei dem 'small world network', welches zwischen den beiden anderen Netzwerken liegt, existiert eine kurze mittlere Weglänge und es findet eine starke Clusterbildung statt. Dabei kommt es zu der Bildung von Hub-Knoten, diese weisen einen hohen Kantengrad auf. Dies führt zum einen zu einer schnellen Informationsverteilung und zum anderen zu einer erhöhten Stabilität gegen Knotenausfälle, wobei dieses Netzwerk bei Ausfall von Hub-Knoten leicht in einzelne Teilnetze zerfallen kann. Das Kleine-Welt-Phänomen [Milgram (1967)] liegt dem 'small world network'

zugrunde. Dabei handelt es sich um die soziale Vernetzung in der modernen Gesellschaft und den hohen Grad an abkürzender Wege durch persönliche Beziehungen.

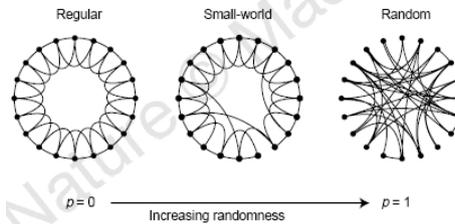


Abbildung 2.2: Netzwerkstruktur - Quelle: [Watts und Strogatz (1998)]

Ein zufällig erzeugter Graph findet seinen Ursprung in den Arbeiten von Erdős und Rényi [Erdős und Rényi (1959)]. So wird in einem Erdős-Rényi Graphen ( $G_p(N)$ ) eine Kante zwischen zwei Knoten mit einer Wahrscheinlichkeit  $p$  gewählt, wobei  $N$  die Gesamtknotenzahl darstellt. Die Verbindungen in einem realen Netzwerk werden aber nicht derartig zufällig erstellt, sondern unter Kostengesichtspunkten erzeugt. Bei so einem geometrischen Zufallsgraphen wird die Entfernung zwischen zwei Knoten berücksichtigt. Der Waxman-Graph [Waxman (1991)] ist wohl der bekannteste Vertreter dieser Zufallsgraphen. In diesem Modell hängt die Wahrscheinlichkeit für eine Kante zwischen zwei Knoten  $u, v$  von ihrem euklidischen Abstand  $d(u, v)$  ab:  $P(u, v) = \beta \exp^{-d(u,v)/L^\alpha}$ ,  $0 < \alpha, \beta \leq 1$ .  $L$  ist hierbei die maximale Entfernung zweier Knoten. Ein größeres  $\beta$  erhöht die Kantendichte, ein kleineres  $\alpha$  hingegen verringert den Anteil längerer zu kürzerer Kanten. Diese Parameter erlauben somit den mittleren Knotengrad zu justieren. Das Waxman-Modell eignet sich sehr gut, um kleine Netzwerk zu repräsentieren, doch das Internet mit der damit verbundenen Größe approximiert er nur schlecht. Aufgrund der Größe ist es bei sehr großen Graphen unzureichend, dass nur der euklidische Abstand für die Kantenbildung relevant ist [Medina u. a. (2000)].

### 2.1.2 Reale Internet-Topologiedaten

Um Internet-Topologiedaten zu erhalten, müssen Messungen im Internet vorgenommen werden. Messungen können aber nur erfolgen, wenn bestimmte Wege im Internet bekannt sind. So wird zunächst ein Standort benötigt von dem diese Messungen vorgenommen werden können. Neben einem Quell-Standort wird auch ein Ziel-Standort benötigt. Dies können bekannte Router als auch normale Webserver sein. Da die Route vom Quell- zum Ziel-Standort nicht bekannt ist, muss diese nun erfasst werden.

Die Internet-Topologiedaten werden erzeugt, indem Traceroute [Jacobson (1989)] Messungen von einem Quell-Host zu einem Ziel-Host durchgeführt werden. Dabei wird ein Paket mit einem TTL (time-to-live) Wert von 1 zum Ziel-Host gesendet. Der erste Router, der das

Paket empfängt verringert den TTL Wert um 1, da der TTL Wert 0 beträgt, wird das Paket verworfen und eine ICMP Nachricht 'Time Exceeded' [Postel (1981)] an den Quell-Host geschickt, dass der TTL Wert überschritten wurde. Damit wird der erste Router auf dem Weg zum Ziel-Host identifiziert. Der Quell-Host erhöht nun den TTL Wert um 1 und es wird der zweite Router auf dem Weg zum Ziel-Host gefunden. Dies wiederholt sich so lange, bis das Paket den Ziel-Host erreicht hat.

Zusätzlich zu den Adressen der Router, wird auch die RTT (round-trip-time) gemessen. Die RTT gibt die Zeit in ms an, die ein Paket braucht, um vom Quell-Host zum entsprechenden Router und zurück zu gelangen. Hierbei sollte beachtet werden, dass der Rückweg auch asymmetrisch [Paxson (1999)] sein kann. So hat Paxson rund 40.000 Ende-zu-Ende Verbindungen analysiert und herausgefunden, dass die Hälfte dieser Verbindungen asymmetrisch verlief. Sofern die Ermittlung der Verzögerung zwischen den einzelnen Verbindungen durch die Division von der RTT durch 2 geschieht, sollte davon ausgegangen werden, dass die Ergebnisse fehlerhaft sein können. Abbildung 2.3 zeigt, wie asymmetrisches Routing aussehen könnte. Dabei wird eine Anfrage vom Quell-Host zum Ziel-Host geschickt und über die Router 1 und Router 2 weitergeleitet. Bei symmetrischem Routing würde nun die Antwort über die Router 2 und Router 1 zu der Quelle weitergeleitet, hier aber wird die Antwort über Router 3 zu der Quelle weitergeleitet. Asymmetrisches Routing entsteht im Internet Backbone vornehmlich durch verschiedenartige BGP4-Policies der Provider.

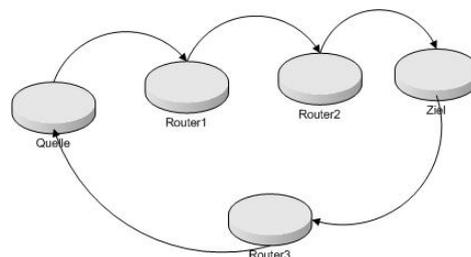


Abbildung 2.3: Beispiel für asymmetrisches Routing

### 2.1.3 Das Problem der Messung

Sicherheit stellt im Internet ein wichtiges Thema dar. Allerdings führt diese Sicherheit in Zusammenhang mit Messungen zu einem Problem. Es befindet sich ein großer Teil der Internet-Hosts hinter Firewalls, die das interne Netzwerk vom Rest des Internets abgrenzen. Die Firewall kann je nach Filtereinstellung, eingehende sowie auch ausgehende Daten anhand von Merkmalen filtern. Ein Merkmal wäre das verwendete Protokoll, wie z.B. ICMP oder UDP. Da aber Traceroute oder äquivalenten Methoden zum größten Teil mit ICMP oder

UDP arbeiten, führt das Sammeln von Daten nicht immer zu richtigen bzw. überhaupt zu Ergebnissen. Pakete werden von Firewall, verworfen.

Ein weiteres Problem, das auftreten kann, haben Janic und van Mieghem [[Janic und Van Mieghem \(2006\)](#)] dokumentiert. So sind etwa 17 % der von Janic und van Mieghem verwendeten Traceroute Daten fehlerhaft, da einige ISP (Internet Service Provider) ihre Router vor Traceroute verstecken und die ICMP Antworten mit falschen Angaben zurücksenden.

Bei der Versendung eines Paketes vom Quell-Host zum Ziel-Host bestimmt normalerweise der jeweilige Router dynamisch über welchen Weg das Paket gesendet wird. IPv4 sowie IPv6 haben aber im Header eine Option, den Weg festzulegen. Source Routing bietet die Möglichkeit, dass das Paket vom Quell-Host über vorgegebene Router zum Ziel-Host gesendet wird. Es gibt zwei Formen von Source Routing:

- **Strict Source Routing**  
Dabei legt der Sender die exakte Route des Paketes fest. Der komplette Weg ist also vorgeschrieben.
- **Loose Source Routing**  
Hierbei werden nur einzelne Router ausgewählt, über die ein Paket zum Ziel gelangen kann. Nur ein Teil des Weges wird also bestimmt.

Allerdings werden oft aus Sicherheitsgründen Pakete mit IPv4 Headern, die zusätzliche Optionen haben, verworfen.

## 2.2 CAIDA

CAIDA (Cooperative Association for Internet Data Analysis) [[Claffy \(2001\)](#)] ist ein Zusammenschluss zwischen mehreren Organisationen, die in unterschiedlichen Bereichen tätig sind. Darunter sind Vertreter aus dem kommerziellen, Regierungs- als auch Forschungssektor. Gegründet wurde die Organisation 1997 in der Universität von Kalifornien in San Diego (UCSD). Ziel ist es, eine gemeinsame Forschungs- und Analyseumgebung entstehen zu lassen, wo verschiedene Messdaten analysiert und geteilt werden sollen. Dabei soll auch die Entwicklung für Methoden zur Steigerung der Datenverkehrs-Leistung sowie Methoden zur Charakterisierung des Datenverkehrs-Flusses gefördert werden.

Um diese Ziele zu erreichen, hat CAIDA eine Vielzahl an Tools entwickelt. Darunter auch Skitter [[Skitter](#)]. Das Skitter-Projekt vermisst kontinuierlich große Teile des Internets seit ca. 9 Jahren, um damit den Weg zwischen einem Quell-Host und mehreren Ziel-Host zu erfassen. Diese Daten können dann von anderen Tools eingelesen, analysiert, bzw. weiterverarbeitet werden.

Die Messung eines Weges zwischen einer Quelle und einem Ziel erfolgt auf ähnliche Weise wie Traceroute. Dabei wird der TTL Wert eines gesendeten 'ICMP echo' Paketes erhöht, bis der Ziel-Host erreicht ist bzw. abgebrochen wenn der TTL Wert größer als 30 oder eine Wiederholung der Router dazwischen erkennbar ist.

Um diese Messungen durchzuführen, werden 17 Quell-Hosts, die in den Vereinigten Staaten, Europa und Asien liegen, verwendet. Als Ziel-Hosts werden 23.000 weltweit verteilte Rechner<sup>1</sup> abgefragt.<sup>2</sup> Da Skitter so entwickelt wurde, dass der aufkommende Datenverkehr minimal bleibt, werden je nach der Anzahl der Ziel-Hosts unterschiedlich oft die Ziel-Hosts abgefragt. Bei 21.000 Zieladressen wurden diese an einem Tag bis zu 20 mal abgefragt. Durch die Wiederholung der Anfragen, ist es Skitter möglich, Änderungen an zurückgelegten Wegen zu erkennen.

Ein typischer Datensatz, der an einem Tag durch Skitter erstellt wird, beinhaltet 300.000 - 500.000 Messungen, wobei ungefähr 19.000 Ziel-Hosts erreicht werden. Dieses Messen ermöglicht es, einen Teil der Internettopologie zu rekonstruieren.

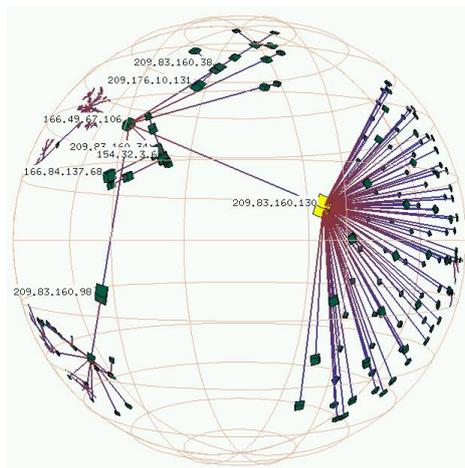


Abbildung 2.4: Beispielgrafik aus Skitter-Daten

## 2.3 Mercator

Das Mercator Project [Govindan und Tangmunarunkit (2000)] ist eines der umfangreichsten Versuche, Messdaten aus dem Internet zu erhalten. Dabei werden bei Mercator UDP Pakete auf die selbe Weise wie bei Traceroute verwendet, um ICMP Pakete zu empfangen.

<sup>1</sup>Hauptsächlich Webserver

<sup>2</sup>Stand 12.05.2007, <http://www.caida.org/home/about/faq.xml>

Mercator erhält die Daten, indem von einem Host-Rechner ein Zielrechner ohne die Verwendung einer Zielrechner-Datenbank ausgewählt wird. Hierbei wird eine Heuristik namens 'informed random address probing' verwendet:

- Sobald eine Antwort von einem Router empfangen wird, wird angenommen, dass dessen Netzwerk-Prefix adressierbare Knoten enthält. Der Prefix wird zudem in einen Pool von gefundenen Prefixen hinzugefügt.
- Für jede Untersuchung, wird ein Prefix aus diesem Pool ausgewählt, und die Ziel-Adresse für diesen Prefix wird zufällig erstellt
- Sollte nach einer bestimmten Zeit der Pool an Prefixen nicht gewachsen sein, wird ein neuer Prefix eingefügt, indem ein Prefix aus dem Pool genommen wird und dessen benachbarter Prefix ausgewählt wird

Mercator verwendet zusätzliche Heuristiken, um Namen aufzulösen, sowie mehrfach-Interfaces (IP-Adressen) zu identifizieren, die einem Router gehören. So wurden im Sommer 1999 nach drei Wochen Laufzeit, nahezu 150.000 Interfaces und fast 200.000 Verknüpfungen entdeckt.

## 2.4 DIMES

Das DIMES-Projekt [[Shavitt und Shir \(2005\)](#)] ist ein verteiltes wissenschaftliches Projekt mit dem Ziel, die Struktur und Topologie des Internets zu erforschen. Gestartet wurde das recht junge Projekt im September 2004.

Im Gegensatz zu den schon erwähnten Projekten, verwendet DIMES eine Software, die von jedem Benutzer auf dem eigenen Rechner benutzt werden kann. Dieser DIMES Agent läuft dann ähnlich wie bei dem bekannten SETI@Home Projekt im Hintergrund. Dabei kommt es aber nicht auf die Rechenleistung, sondern viel mehr auf den Standort an. So ist durch eine höhere Benutzeranzahl die Möglichkeit gegeben, präzisere Darstellungen des Internets zu erstellen. Hierbei sollte aber auch beachtet werden, dass nicht nur die Anzahl der Benutzer, sondern auch deren Verteilung auf dem Globus eine wichtige Rolle spielt. Erst dadurch werden die gewonnenen Daten, bei einem verteiltem Projekt interessant. Innerhalb der ersten neun Monate stieg die Anzahl der verwendeten DIMES Agenten auf 3600, inzwischen sind es über 12.500 und es wurden bisher knapp über 2.306.000.000 Messungen durchgeführt.

3

---

<sup>3</sup>Stand 12.05.2007, <http://www.netdimes.org/new/?q=node/5>

## 2.5 Diskussion

Skitter-, Mercator- sowie DIMES-Datensätze sind ziemlich groß und enthalten viele Knoten sowie Kanten. Allerdings werden in den Skitter-Datensätzen die Kanten auch als vollständig, bzw. unvollständig markiert, und dementsprechend verkleinert sich der Datensatz um ca. 40%, sofern die unvollständigen Routen ignoriert werden. DIMES-Daten sind hingegen in Knoten sowie Kanten Dateien unterteilt. Die Kanten werden nur einzeln angegeben und daraus lässt sich nicht schließen, ob ein Traceroute vollständig, bzw. unvollständig ablaufen konnte.

Das Skitter-Projekt hat einerseits den Vorteil, dass die Ziel-Knoten bekannt und auch die Start-Knoten zwar sehr begrenzt, aber unverändert sind. Dies ermöglicht es Skitter, die Daten auf Veränderungen zu überprüfen und eine Entwicklung des Aufbaus aufzuzeichnen. Andererseits besteht bei diesem Verfahren der Nachteil, dass die Ziel-Knoten zwar regelmäßig erweitert werden, aber dennoch die Messungen nur aus einer Perspektive erfolgen und somit begrenzte Informationen über das Internet messen.

Mercator gleicht zwar vom Ablauf her dem Skitter-Projekt, doch sind die Ziel-Knoten nicht durch eine feste Liste aus bekannten Adressen entstanden, sondern durch Heuristiken, die neu adressierbare Knoten suchen. Insofern erfolgt bei Mercator eine dynamische Entdeckung des Internets.

DIMES ist zwar ein recht junges Projekt, doch hat es aufgrund der verteilten Struktur die Möglichkeit, genauere Ergebnisse zu messen. Hier besteht aber das Problem, dass solche Anwendungen überwiegend im wissenschaftlichen Bereichen eingesetzt werden. Dementsprechend kommen diese auch in Universitäten und anderen Forschungseinrichtungen zum Einsatz, die sich in wissenschaftlichen Netzwerken befinden. Dadurch werden trotz Verteilung der Benutzer nur wenige neue Routen aufgezeigt.

## 3 Entwurf einer Verarbeitungskette

Es existiert eine Reihe von Werkzeugen, mit denen ermittelte Daten verarbeitet, konvertierte sowie auch analysiert werden können. Dabei besteht die Schwierigkeit, dass jedes dieser Werkzeuge mit einem bestimmten Datenformat umgehen kann, aber mit einigen Daten nur unzuverlässig oder erst überhaupt nicht anwendbar ist. So ist es oft erforderlich, mehrere dieser Werkzeuge Schritt für Schritt zu benutzen, um zu einem gewünschten Ergebnis zu kommen.

### 3.1 Anforderungen

Es gibt eine Reihe an Anforderungen, die eine Verarbeitungskette erfüllen muss. So dürfen Topologiedaten, die exportiert werden sollen, keine Informationen bei ihrem Zustandswechsel verlieren oder verfälscht werden. Da es sich um relativ große Mengen an Informationen handelt, die verarbeitet werden müssen, sollten auch die Kosten in Form von Rechnerleistung und benötigter Zeit, in einem akzeptablen Rahmen bleiben. Hierbei muss ein Kompromiss gefunden werden, da eine schnelle Verarbeitung mehr Rechenleistung und umgekehrt, eine langsame Verarbeitung weniger Rechenleistung benötigt. Gesammelte Daten aus dem Internet sind zwar in der Regel unvollständig, können aber einen repräsentativen Ausschnitt des Internets darstellen. So befinden sich in einem vollständigem Skitter- sowie DIMES Datensatz bis zu 900.000 Knoten, inklusive derer spezifischer Eigenschaften. Um aber diese Menge an Informationen in einem Netzwerksimulator zu verarbeiten, müssen diese Daten entsprechend gefiltert werden, ohne dass diese ihre charakteristischen Eigenschaften verlieren. Dies hat den Vorteil, dass selbst kleine extrahierte Graphen die Eigenschaften des originalen Graphen besitzen und somit performanter analysiert und auf eine gewünschte Größe gefiltert werden können. Mit einer steigenden Anzahl der verwendeten Werkzeuge, steigt auch die Wahrscheinlichkeit, dass Fehler auftreten können. Deswegen sollte diese Anzahl auf ein Minimum beschränkt werden.

## 3.2 Konzeption

Zur Verwirklichung dieser Anforderungen gibt es verschiedene Möglichkeiten. So existieren mehrere Werkzeuge, die eine Topologie weiterverarbeiten können. Darunter BRITE [Medina u. a. (2001)] sowie nem [Magoni (2002)]. Beide bieten die Möglichkeit, Topologiedaten einzulesen und diese auch in andere Topologiedaten zu exportieren. Hierbei kann keines der beiden Werkzeuge DIMES-Daten einlesen, wobei diese im Gegensatz zu den Skitter-Daten schon im ASCII-Format vorliegen. Ein Skitter-Datensatz besteht aus einem Stream und ist für eine Weiterverarbeitung nicht gut geeignet. Um mit den Daten dennoch arbeiten zu können, bietet CAIDA `sk_analysis_dump`<sup>1</sup> an, mit dem Skitter-Datensätze in das ASCII-Format konvertiert werden können. Somit wäre das erste Schritt in der Verarbeitungskette die Benutzung von `sk_analysis_dump`. BRITE hat zwei Vorteile. Zum einen ist schon ein Import der Skitter-Daten vorhanden und zum anderen existiert auch die Export Funktion für den Netzwerksimulator OMNeT++ [Varga (2005)]. Nem bietet aber im Vergleich dazu die Möglichkeit, Netzwerke samt ihrer Eigenschaften zu filtern. Sowohl BRITE als auch nem sind in C++ implementiert, wobei BRITE zusätzlich in Java programmiert wurde. Daraus ergeben sich nun drei Szenarien.

1. Die Daten werden in BRITE eingelesen und in ein nem Format exportiert. Anschließend wird in nem der Graph gefiltert und in BRITE wieder eingelesen. Zum Schluss wird nun der gefilterte Graph in das OMNeT++ Format exportiert.
2. Die Daten werden in nem eingelesen und gefiltert. Anschließend in das OMNeT++ Format exportiert.
3. Die Daten werden in BRITE eingelesen und gefiltert. Anschließend in das OMNeT++ Format exportiert.

Aufgrund der Anforderungen entfällt Szenario 1, da sich mit der steigenden Anzahl an verwendeten Werkzeugen, die Fehlerwahrscheinlichkeit sowie der Aufwand erhöht. Szenario 2 und 3 setzen voraus, dass in den jeweiligen Werkzeugen die fehlenden Funktionen noch implementiert werden. Aufgrund der schon vorhandenen Import und Export Funktionen in BRITE, sowie dessen modularem Aufbau und somit leichter Erweiterbarkeit, wird Szenario 3 gewählt. Die Schritte in der Verarbeitungskette würden nun wie folgt aussehen:

- `sk_analysis_dump` - Konvertierung der Skitter-Datensätze in das ASCII-Format<sup>2</sup>
- BRITE - Import der Topologiedaten
- BRITE - Filterung der Topologiedaten

<sup>1</sup>[http://www.caida.org/tools/measurement/skitter/sample\\_code/](http://www.caida.org/tools/measurement/skitter/sample_code/)

<sup>2</sup>DIMES-Datensätze benötigen diesen Schritt nicht

- BRITE - Export der Topologiedaten in das OMNeT++ Format
- OMNeT++ - Verarbeitung der Topologiedaten

# 4 Eine Verarbeitungskette für reale Internet Topologiedaten

In diesem Kapitel wird die Architektur der zu entwickelnden bzw. erweiterbaren Komponenten und deren Realisierung erläutert.

## 4.1 BRITE

BRITE (Boston university Representative Internet Topology gEnerator) [Medina u. a. \(2001\)](#) ist ein modular aufgebautes Werkzeug, das primär für die Generierung von Internet Modellen gedacht ist. Durch diese Modularität ist es möglich, jede beliebig unterstützte Topologie einzulesen und daraus einen Graphen zu erstellen. Sobald der Graph vorliegt, kann durch eine Export-Funktion dieser Graph in ein beliebiges andere Format konvertiert werden. Allerdings fehlt hier eine Funktion, die den Graphen nach bestimmten Regeln filtert.

BRITE ist sowohl in C++ als auch in Java implementiert. Durch einen bereits vorhandenen Patch <sup>1</sup>, der für die Java Version verfügbar ist, ist es zudem möglich, die Graphen in das OMNeT++ Format zu konvertieren. Der Skitter Import ist aber nicht vollständig implementiert und muss dementsprechend vervollständigt werden.

BRITE nimmt somit schon in seiner Grundversion eine zentrale Rolle in der Verarbeitungskette ein, bietet aber nicht alle erforderlichen Funktionen, die für eine Weiterverarbeitung in einem Netzwerksimulator relevant sind.

Abbildung 4.1 zeigt die grobe Architektur von BRITE. Die Abbildung ist insofern unvollständig, als sowohl im Import als auch im Export andere Dateiformate unterstützt werden. Es sind lediglich die Formate abgebildet, die für diese Arbeit relevant sind. Eine Topologie besteht aus einem Graphen und einem Modell. Eine Exportfunktion ermöglicht es, die Topologie in das gewünschte Format zu überführen. Der Graph besteht aus Knoten sowie den dazugehörigen Kanten. Das Modell hingegen wird entweder je nach gewähltem Algorithmus generiert, oder durch ein gewähltes Dateiformat erstellt.

---

<sup>1</sup><http://www.omnetpp.org/article.php?story=20030518223255884>

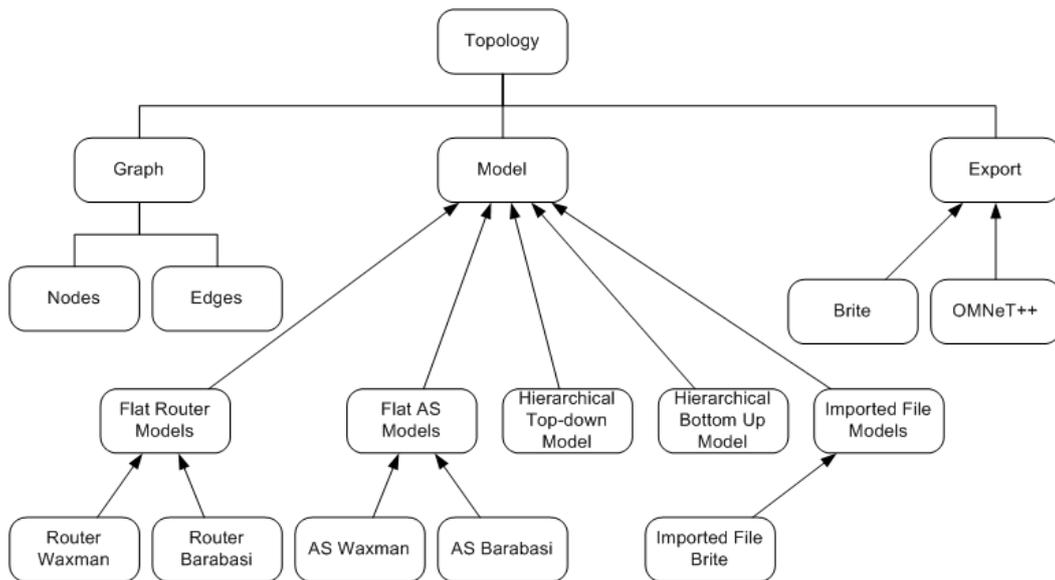


Abbildung 4.1: BRITE Architektur

Folgende Funktionen fehlen oder müssen bei BRITE überarbeitet werden, damit die zentrale Rolle in der Verarbeitungskette eingenommen werden kann:

- Import von Skitter-Datensätzen
- Import von DIMES-Datensätzen
- Eingelesene Verzögerungen speichern, ohne dass diese aus dem euklidischem Abstand berechnet werden. <sup>2</sup>
- Filterung der Netzwerke, ohne dass deren Topologie-Eigenschaften verloren gehen
- Anpassung der Graph-, Edge-, Node- sowie Export Klassen, um die Funktion des Filters zu gewährleisten.
- Überarbeitung des FileModels, um mehrere Dateien einlesen zu können und zur Unterscheidung, ob es sich um eine generierte oder eingelesene Topologie handelt.
- Erweiterung der GUI.

Die Realisierung dieser Punkte wird in den folgenden Unterkapiteln besprochen.

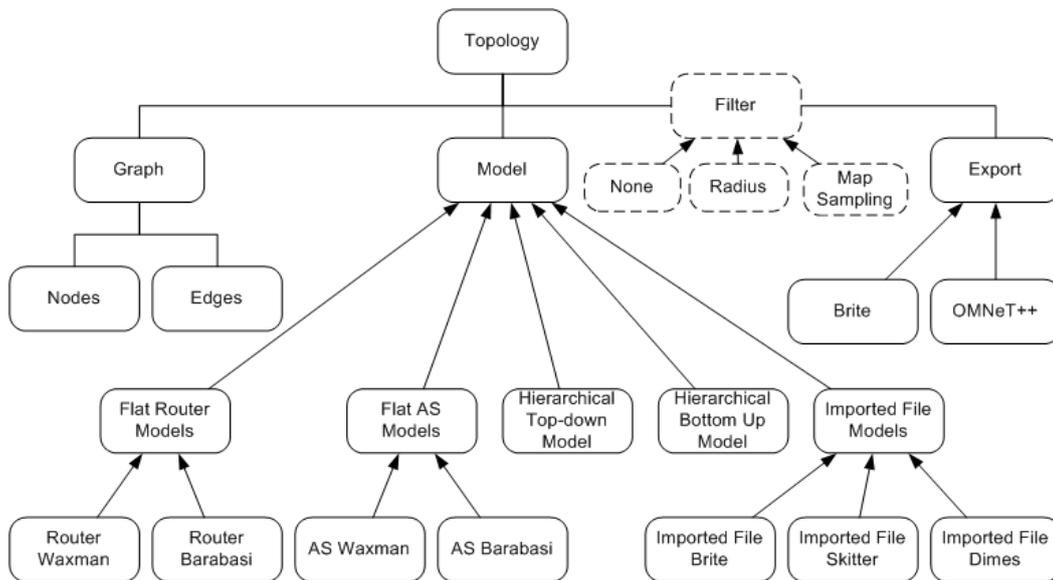


Abbildung 4.2: Überarbeitete BRITE Architektur

### 4.1.1 Architektur

In Abbildung 4.2 werden die wichtigen Funktionen, die zusätzlich implementiert wurden, bzw. überarbeitet gestrichelt dargestellt. Neu hinzugekommen ist die Filter-Funktion, die einen Teilgraphen nach vorgegebenen Filterregeln extrahiert. Auch der Import von Skitter-Daten ist nun in der Java Version vollständig implementiert. Der Import von DIMES-Daten ist ebenfalls möglich.

### 4.1.2 Daten-Import

Der Import einer Datei stellt mit dem gleichzeitigen Aufbau des Graphen das Grundgerüst des Programms dar. Da der Skitter-Import schon vorhanden war, aber in der Java Version nicht fertig gestellt wurde, musste an dieser Stelle der Import umgeschrieben werden. So war es zwar nach einem Eintrag für den Skitter-Import im Model schon möglich, diese Daten einzulesen, doch war das Parsen fehlerhaft. Nach einer Korrektur funktionierte auch das, allerdings kam dabei ein anderes Problem hervor. Das Einlesen der Skitter-Daten ist aufgrund der Daten-Struktur nicht performant. Es kommen Kanten mehrfach in der Datei vor, wobei sie verschiedene Verzögerungen wiedergeben. BRITE besitzt aber nicht die Funktion, die Verzögerung mehrfach für eine Kante zu speichern und berechnet diese ggf. aufgrund der x,

<sup>2</sup>BRITE ist primär als Topologie-Generator konzipiert. Das bedeutet, dass jeder Knoten Koordinaten erhält. Durch diesen Abstand wird die Verzögerung zwischen den Knoten berechnet.

y Koordinaten, die es bei der Generierung einer Topologie, neu erstellt. Skitter-Datensätze besitzen aber nicht solche Koordinaten. Diese werden beim Einlesen zufällig für jeden Knoten erstellt, doch haben diese keine Aussagekraft über die reale Verzögerung. Das mehrfache Vorkommen der Kanten hat zur Folge, dass bei jedem Zugriff auf eine Kante überprüft werden muss, ob diese schon vorhanden ist. Erst nach dem Einlesen muss jede Kante auf ihre eingelesenen Verzögerungen überprüft werden, um anschließend die durchschnittliche Verzögerung zu berechnen. Allerdings kostet jede Abfrage, ob eine Kante schon im Graphen enthalten ist, Zeit. Der Algorithmus zur Überprüfung musste also hinsichtlich der Zugriffszeit verbessert werden. Dies geht aber nicht ohne den benötigten Speicher zu erhöhen. Es musste also ein Kompromiss zwischen Zugriffszeit und verwendetem Speicher gefunden werden.

So dauerte das Einlesen einer Datei mit 461.763 Knoten auf einem Pentium4 mit 3.000 Mhz und 1 GB Speicher ursprünglich ca. 1 Std. Durch die Verwendung einer zusätzlichen Hash-Map, konnte diese Zeit auf ca. 18 min reduziert werden. Dabei wird als Key die IP-Adresse des Quell-Knotens und als Value eine Zeichenkette, bestehend aus den IP-Adressen der Ziel-Knoten, gespeichert. So wird bei der Überprüfung, ob eine Kante schon vorhanden ist, nicht auf die BRITE interne Überprüfung zurückgegriffen. Diese benötigt die Knoten-Objekte zur Überprüfung, wodurch im Gegensatz zu einem Zeichenkettenvergleich mehr Zeit benötigt wird. Da solche Datensätze ziemlich groß sind und dementsprechend der erstellte Graph auch größer wird, empfiehlt es sich hier, mit mindestens 1536 MB Speicher zu arbeiten.

Zusätzlich zum Skitter-Import wurde der DIMES-Import implementiert. Dieser basiert auf den Algorithmen, die auch im Skitter-Import verwendet werden, wobei das Einlesen auf die Struktur der DIMES-Datensätze angepasst wurde. Hier haben die DIMES-Datensätze einen Vorteil gegenüber den Skitter-Datensätzen, diese sind schon im Vorfeld auf eine Knoten sowie Kanten Datei aufgeteilt. Dadurch werden zuerst alle Knoten eingelesen und danach die Kanten. Dies führt dazu, dass bei gleicher Größe der Datensätze von DIMES und Skitter die DIMES-Daten im Schnitt 5 mal schneller eingelesen werden und der Graph erstellt wird.

Skitter-Datensätze bestehen aus mehreren Dateien, die einen Monitorpunkt repräsentieren. Es kann sowohl nur ein Monitorpunkt als auch mehrere Monitorpunkte eingelesen werden. Dies wird nur vom Speicher und der Laufzeit beschränkt. Die Daten werden Zeilenweise eingelesen und jede dieser Zeilen enthält die Route von dem Start- zu dem Zielknoten. Die Route kann entweder vollständig oder unvollständig sein. Bei einer unvollständigen Route können einige Knoten nicht erreicht werden, dadurch würden beim Einlesen Lücken entstehen und es würden sich einzelne Knoten ohne Kanten im Netzwerk befinden. Deswegen werden die unvollständigen Routen verworfen. Sobald nun der Startknoten eingelesen wurde, wird dieser mit dem nächsten Router verbunden. Die IP-Adressen der jeweiligen Router werden zur Identifikation verwendet. Da wir vom symmetrischen Routing ausgehen, wird die RTT durch zwei geteilt und der Wert wird nun als Verzögerung zwischen den beiden Knoten eingetragen. Da es vorkommen kann, dass aufgrund des asymmetrischen Routings

die RTT der einzelnen trace-hops nicht kontinuierlich wächst, wird ein vorgegebener Wert als Verzögerung bei den betroffenen Kanten eingetragen. Anschliessend wird der Router mit dem nächsten Router in der Route verbunden. Dabei wird die Differenz zwischen alter und neuer Verzögerung als neue Verzögerung gewählt. Dieser Vorgang wiederholt sich so lange, bis der Zielknoten erreicht wurde. Sollten Kanten mehrfach vorkommen, wird die Verzögerung zusätzlich in einer Liste eingetragen um nach dem Einlesen einen Durchschnittswert zu bilden.

DIMES-Datensätze bestehen aus zwei Dateien, eine Datei enthält Informationen über die Knoten, die andere Datei enthält Informationen über die Kanten zwischen den Knoten. Ähnlich wie bei Skitter-Datensätzen wird auch hier Zeilenweise eingelesen. Zunächst wird die Datei mit den Knoten eingelesen um alle IP-Adressen zu erhalten. Anschliessend wird die Datei mit den Kanten eingelesen. Im Gegensatz zu Skitter-Datensätzen, existiert keine Route von einem Start- zu einem Zielknoten. Es wird Zeilenweise nur eine Kante zwischen zwei Knoten sowie deren Durchschnittliche Verzögerung eingelesen. Eine Berechnung der Verzögerung ist nicht notwendig, da diese in den DIMES-Datensätzen schon als Durchschnittsverzögerung zwischen den Kanten vorhanden ist.

BRITE enthält mit dieser Erweiterung die Möglichkeit, sowohl Skitter- als auch DIMES-Datensätze einzulesen, um diese Topologien in ein anderes Format zu exportieren.

### 4.1.3 Daten-Filter

Eine Filter-Klasse ist im originalen BRITE nicht vorhanden. Um aber Daten komfortabel in einem Durchgang einzulesen und entsprechend gefiltert zu exportieren, ohne dabei an wesentlichen Eigenschaften des Graphen zu verlieren, musste diese Klasse implementiert werden. Die Filter-Klasse folgt dem Aufbau des Factory-Pattern [[Gamma u. a. \(1994\)](#)]. Hierbei wird je nach Einstellung des Filter-Typen eine Filter-Klasse generiert. Dabei können neue konkrete Filter-Klassen hinzugefügt werden, ohne dass grundsätzliche Veränderungen am Code vorgenommen werden müssen. Die Filterklasse passt somit gut zum Konzept der Modularität von BRITE. So wurden bisher zwei konkrete Filter implementiert.

Der erste Filter benutzt den vollständig erzeugten Graphen, sucht sich dabei einen vorgegebenen Knoten und zieht einen 'Radius' um diesen. Der Radius wird vor dem Aufruf festgelegt und entspricht der maximalen Anzahl an Hops, die vom vorgegebenen Knoten entfernt sind. Dadurch entsteht ein neuer Graph, der aus dem alten Graphen nach topologischen Eigenschaften extrahiert wurde.

Der zweite Filter wählt nach einem vorgegebenen Algorithmus bestimmte Knoten aus dem vollständig erstellten Graphen aus. Dieser Algorithmus wird auch in einem anderen Topologie Generator namens nem [[Magoni \(2002\)](#)] verwendet. Nem verwendet diesen Filter namens

'map sampling' [Magoni und Pansiot (2002)]. Dabei wird im groben Ablauf, ein Knoten zufällig ausgewählt und dessen Nachbarn sowie Kanten werden als Kandidaten für den neuen Graphen eingetragen. Aus diesen Kandidaten wird dann der nächste Knoten zufällig gewählt. Dies wiederholt sich so lange, bis die gewünschte Größe des neuen Graphen erreicht wurde.

Um aber nun zu überprüfen, ob der 'map sampling' Algorithmus richtig funktioniert, wurden sieben Skitter-Datensätze auf sechs verschiedene Größen mittels BRITE und zum Vergleich mittels nem gefiltert. Anschliessend wurden diese Topologien in nem eingelesen und auf vier Potenzgesetze, die auch in [Magoni und Pansiot (2002)] zur Überprüfung des Algorithmus dienten, verglichen. Aus den sieben Datensätzen wurde für die Auswertung der Mittelwert der Ergebnisse benutzt, um zu verhindern, dass kleine Ausreißer das Endergebnis verfälschen.

- Potenzgesetz 1 (Rangzahl Exponent): Der Ausgangsgrad  $d$  eines Knotens ist proportional zu der Rangzahl  $r$  eines Knotens hoch einer Konstanten  $R$ . So dass gilt:  
$$d \propto r^R$$
- Potenzgesetz 2 (Ausgangsgrad Exponent): Die Häufigkeit  $f$  eines Ausgangsgrades  $d$  ist proportional zu dem Ausgangsgrad hoch einer Konstanten  $O$ . So dass gilt:  
$$f \propto d^O$$
- Potenzgesetz 3 (Baum Rangzahl Exponent): Die Größe  $s$  eines Baumes (z.B. die Anzahl an Knoten) ist proportional zu der Rangzahl  $r$  eines Baumes hoch einer Konstanten  $T$ . So dass gilt:  
$$s \propto r^T$$
- Potenzgesetz 4 (Baum Größe Exponent): Die Häufigkeit  $f$  einer Baumgröße  $s$  ist proportional zu der Baumgröße hoch einer Konstanten  $S$ . So dass gilt:  
$$f \propto s^S$$

Sowohl in Tabelle 4.1, 4.2 und 4.3 wird der absolute Korrelationskoeffizient (ACC) zur Überprüfung der Potenzgesetze benutzt. Ein Wert von 0.95 gilt als Grenzwert für die Bestätigung einer linearen Regression. Sowohl in der Größe von 15400 Knoten und der ursprünglichen Skitter-Topologie fehlen einige Werte, die aufgrund der Größe nicht berechnet werden konnten.

Tabelle 4.1 zeigt die Ergebnisse aus der Filterung der Skitter-Datensätze mittels BRITE. Auffällig ist hierbei der Korrelationskoeffizient der Baumgröße bei 154 Knoten. Dieser ist im Vergleich zu den anderen Werten ziemlich gering. Dies tritt aber nur bei Graphen auf, deren Anzahl an Knoten unter 1000 liegen. Dieses Problem trat auch in [Magoni und Pansiot (2002)] auf, wobei Tabelle 4.2 dies nicht bestätigt. Hier ist der Wert sogar höher als von einigen größeren extrahierten Graphen. Erst bei über 1000 Knoten liegen die Ergebnisse von nem und BRITE wieder nahe beieinander.

Map type	Skitter-2006 - Brite map sampling					
Number of nodes	154	1540	3080	6160	9240	15400
Number of trees	17	164	299	664	1008	1625
Rank ACC	0.959	0.959	0.962	0.960	0.960	0.961
Degree ACC	0.966	0.976	0.970	0.973	0.979	0.980
Tree rank ACC	0.963	0.972	0.978	0.974	0.969	0.976
Tree size ACC	0.759	0.908	0.925	0.949	0.962	0.963
Average path length	5.70	10.79	12.40	15.99	16.558	-
Average eccentricity	10.16	20.14	22.97	29.75	31.21	-

Tabelle 4.1: Brite map sampling

Map type	Skitter-2006 - nem map sampling					
Number of nodes	154	1540	3080	6160	9240	15400
Number of trees	12	196	405	869	1329	2239
Rank ACC	0.958	0.959	0.959	0.959	0.959	0.959
Degree ACC	0.948	0.965	0.962	0.963	0.975	0.973
Tree rank ACC	0.959	0.985	0.986	0.986	0.984	0.983
Tree size ACC	0.958	0.932	0.952	0.959	0.968	0.972
Average path length	5.33	10.50	11.52	12.74	13.48	-
Average eccentricity	9.81	19.03	21.51	23.08	24.54	-

Tabelle 4.2: nem map sampling

Die Tabelle 4.3 dient lediglich als Überblick über die ursprünglichen Topologie-Daten, ohne dass diese gefiltert wurden. Zusätzlich sind drei andere Topologie-Daten zum Vergleich aufgelistet, die auch in [Magoni und Pansiot (2002)] zum Einsatz kamen.

Abbildung 4.3 zeigt den Vergleich zwischen nem und BRITE hinsichtlich der Potenzgesetze 1 und 2. Man sieht hier deutlich, dass sowohl der 'map sampling' Algorithmus in BRITE als auch nem zu fast identischen Ergebnissen kommen. Nem rutscht zwar hier bei 154 Knoten unter die 0.95 Grenze, dies liegt aber an der sehr geringen Anzahl an Knoten.

Auch die Ergebnisse aus Abbildung 4.4, welches die Potenzgesetze 3 und 4 darstellt, zeigen, dass nem und BRITE ziemlich dicht beieinander liegen. Allerdings sind auch hier die 154 Knoten zu gering und zeigen einen großen Unterschied zwischen nem und BRITE auf.

Map type	LSIIT-95	Lucent-99	Skitter-2006	ISI-99
Number of nodes	3888	112969	160631	228263
Number of trees	391	15289	20594	23136
Rank ACC	0.945	0.967	0.966	0.975
Degree ACC	0.967	0.984	0.981	0.987
Tree rank ACC	0.981	0.986	0.985	0.978
Tree size ACC	0.891	0.986	0.983	0.982
Average path length	11.59	9.94	-	9.51
Average eccentricity	21.86	19.19	-	20.79

Tabelle 4.3: Unveränderte Internet-Topologien

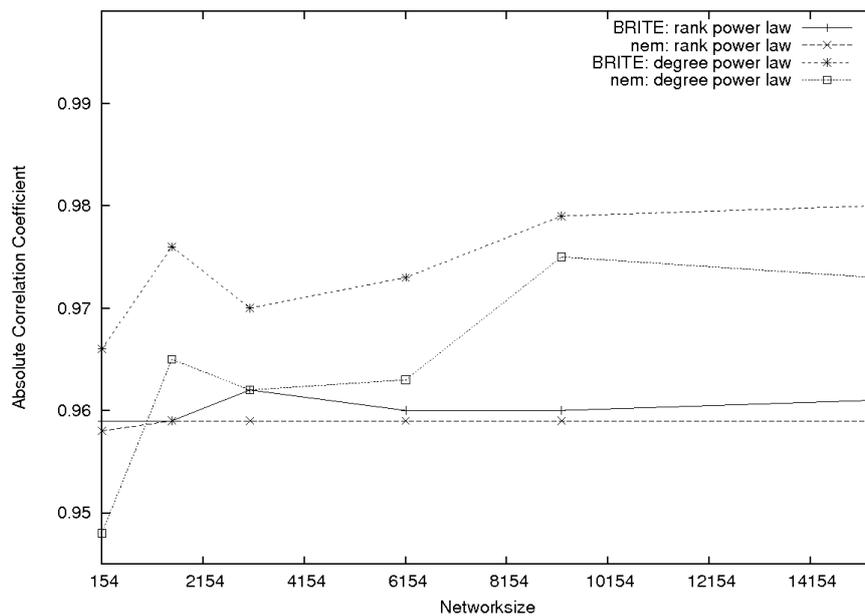


Abbildung 4.3: Vergleich zwischen BRITE und nem 'map sampling' mittels des Rangzahl und Grad Korrelationskoeffizienten

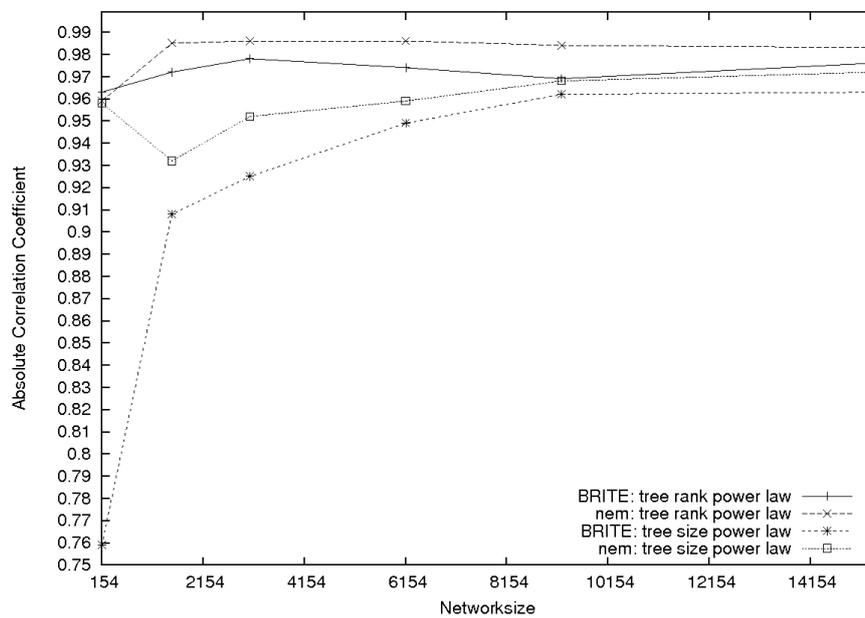


Abbildung 4.4: Vergleich zwischen BRITE und nem 'map sampling' mittels des Baumrangzahl und Baumgröße Korrelationskoeffizienten

#### 4.1.4 Daten-Export

Die Export Funktion ermöglicht es, den eingelesen bzw. selbst erstellten Graphen in ein Format zu bringen, welches z.B. in einem Netzwerksimulator weiterverarbeitet werden kann. Durch einen Patch<sup>3</sup> ist es möglich, den Graphen in das OMNeT++ Format zu exportieren. Allerdings war in BRITE ursprünglich kein Filter vorhanden, so dass nur der Graph vollständig exportiert wurde. Um nun den Filter zu verwenden, wird im Export überprüft, welches Filterformat verwendet wird. Je nach verwendetem Filter wird ein neuer Graph erzeugt und dieser exportiert. So bleibt bei keinem Filter die Funktionalität von BRITE erhalten und bei einem ausgewähltem Filter wird die Filter-Funktion aufgerufen.

#### 4.1.5 Konverter

BRITE ist primär als Topologie-Generator erstellt worden. Dabei können die erstellten Topologien in bestimmte Formate exportiert werden. Die Verwendung des Filters aus Abschnitt 4.1.3 macht für diese generierten Topologien keinen Sinn, da diese schon nach bestimmten ausgewählten Eigenschaften generiert werden. Somit wird der Filter nur bei Konvertierungen verwendet. Da eine Konvertierung unabhängig von der Generierung einer Topologie ist, wurde die Klasse *FileConverter* implementiert. Diese Klasse verwendet die Klassenbibliothek von BRITE und stellt bei der Konvertierung das Hauptprogramm dar. Parameter bestimmen den Programmablauf und rufen die Methoden auf, die für den Import sowie Export einer Datei benötigt werden. Bei dem ersten Programmaufruf wird zusätzlich für die Kommandozeileneingabe eine Datei mit den Filter-Einstellungen generiert.

#### 4.1.6 Benutzerschnittstelle

Es existiert sowohl eine Kommandozeileneingabe als auch eine grafische Benutzeroberfläche. Bei der Kommandozeileneingabe werden beim Starten des Hauptprogramms Parameter mitgegeben, um eine Konvertierung durchzuführen. Aufgrund der möglichen Filter-Einstellungen würden diese Parameter zu einer schlechten Bedienbarkeit führen, deswegen werden diese Einstellungen in einer Datei gespeichert, die nur bei der Kommandozeileneingabe verwendet wird. Um nun eine Konvertierung zu starten, werden folgende Parameter benötigt:

1. Name der Ausgabe-Datei.
2. Router- oder AS-Level.

---

<sup>3</sup><http://www.omnetpp.org/article.php?story=20030518223255884>

3. Art der Konvertierung (Import- sowie Exportformat).
4. Name(n) der einzulesenden Datei(en).

Soll ein Skitter-Datensatz auf Router-Ebene, bestehend aus drei Dateien in das OMNeT++ Format konvertiert werden, würde dies so aussehen: `java FileConverter omnet.ned RT skitter2omnet Teil1.skitter Teil2.skitter Teil3.skitter` Filter-Einstellungen werden direkt in der Datei vorgenommen.

Die grafische Benutzeroberfläche kann entweder direkt gestartet oder über das BRITE Hauptprogramm (Abbildung 4.7) aufgerufen werden.

Abbildung 4.5 zeigt die grafische Oberfläche des Konverters. Die Einstellungen sind die selben, die auch bei der Parameterübergabe benutzt werden. Unter *Source File* wird die einzulesende Datei angegeben, mit *add* wird diese Datei in eine Liste eingefügt, damit mehrere Dateien eingelesen werden können. *Destination File* gibt den Ausgabe-Dateinamen an. Durch ein DropDown-Menü wird die Art der Konvertierung festgelegt. Die zwei Radio-Buttons legen fest, ob es sich um ein Router- oder AS-Level handelt. Mit *convert* wird der Vorgang gestartet. Abbildung 4.6 zeigt die Filter-Einstellungen, die auch per DropDown-Menü ausgewählt werden. Je nach Filter-Format werden unterschiedliche Optionen der Einstellungen dargestellt.

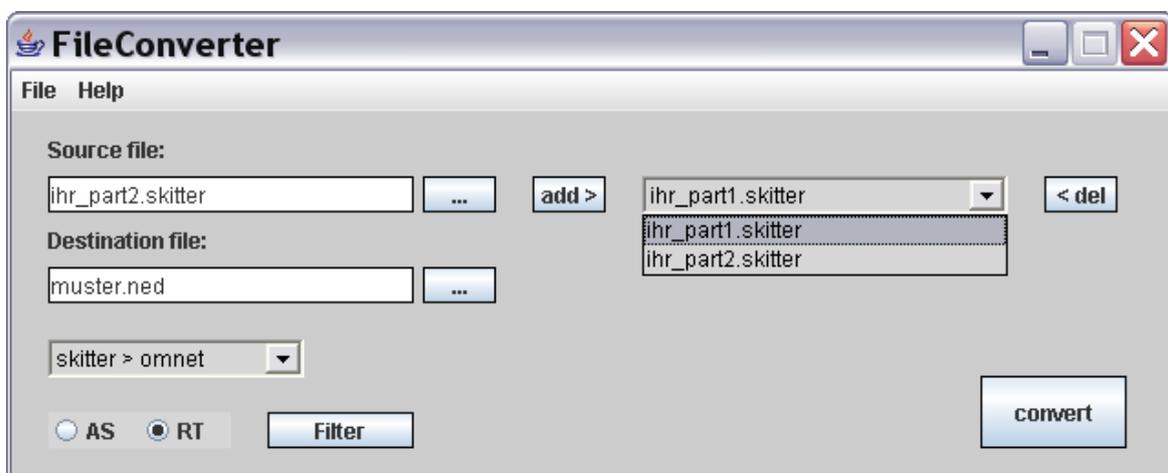


Abbildung 4.5: FileConverter GUI

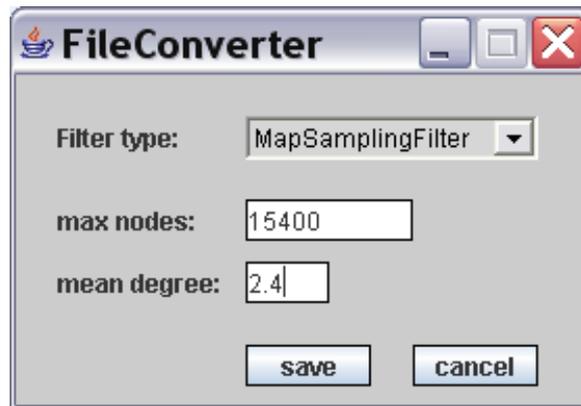


Abbildung 4.6: Filter-Einstellungen

## 4.2 OMNeT++

OMNeT++ (Objective Modular Network Testbed in C++) [Varga (2005)] ist eine auf C++ basierende objektorientierte Simulationsumgebung zur diskreten ereignisorientierten Simulation. Hauptanwendungen von OMNeT++ sind Netzwerksimulationen, doch durch die generische und flexible Architektur wird OMNeT++ auch zur Simulation von komplexen IT-Systemen, sowie Hardware verwendet. Es ist zudem modular aufgebaut. Die Module bauen auf der OMNeT++ Klassenbibliothek auf und werden in C++ programmiert. Simulationsmodelle sind aus hierarchisch verbundenen Modulen zusammengesetzt. Es gibt zwei Typen von Modulen, einfache und zusammengesetzte. Einfache Module bilden die unterste hierarchische Ebene ab und implementieren die Aktivität eines Moduls. Zudem können diese beliebig kombiniert werden und bilden dann ein zusammengesetztes Modul ab. Die Module kommunizieren durch Nachrichten untereinander. Nachrichten können durch die Verbindungen der Module geleitet werden oder direkt an das Zielmodul, ohne dass andere Module betroffen werden. Verbindungen werden in OMNeT++ zwischen den Anschlüssen von Modulen geschaltet. Ihnen kann eine Übertragungsverzögerung, ein maximaler Durchsatz und eine Bitfehlerrate zugewiesen werden, um reale Verbindungen zu simulieren. Der Benutzer legt fest, welche Struktur das Modell hat, indem die Topologiebeschreibungssprache NED verwendet wird. Der Aufbau des gesamten Simulationsmodells wird ebenfalls in NED beschrieben. Die Sprache NED stellt dabei Funktionen zur Verfügung, über die eine variable Anzahl von Untermodulen und Verbindungen erzeugt werden können. Parameter, die nicht im Simulationsmodell in den NED-Dateien festgelegt werden, können entweder in der Initialisierungsdatei `omnetpp.ini` gesetzt oder beim Start der Simulation mitgegeben werden.

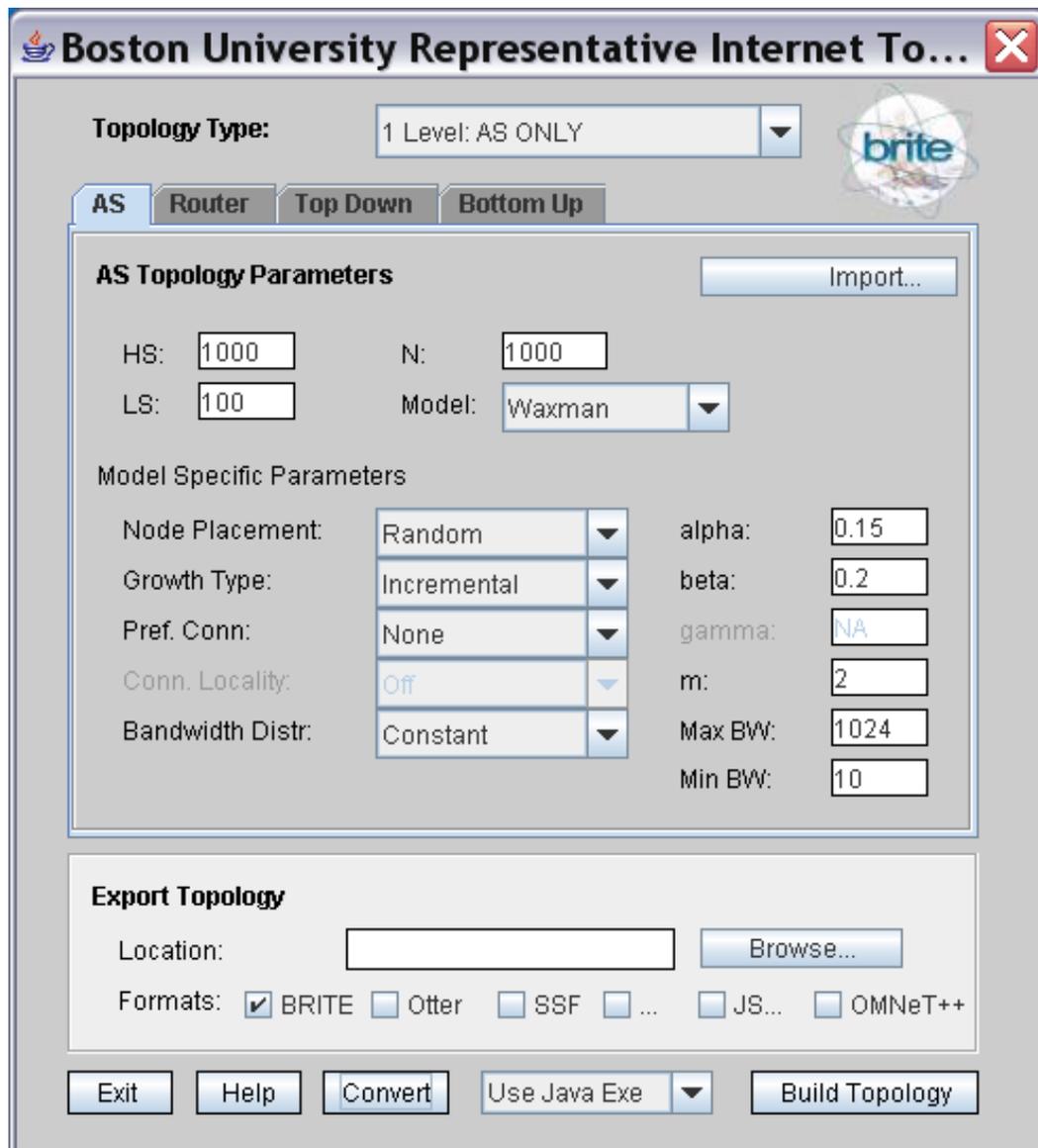


Abbildung 4.7: BRITE Hauptprogramm GUI

# 5 Multicast Verteilbäume

Video-Konferenzen oder auch verteilte Spiele würden bei normalen Punkt-zu-Punkt Verbindungen, in denen jeder Empfänger direkt mit jedem Sender kommuniziert, ein hohes Datenaufkommen auf Seiten des Senders verursachen, da der Sender jeden einzelnen Empfänger mit den angeforderten Daten versorgen muss. Aber auch die Router, die zwischen Sender und Empfänger liegen, werden vom erhöhtem Datenaufkommen betroffen. Multicast bietet hier einen Lösungsweg und wird in diesem Kapitel sowohl in der Ursprungsform, als auch in der mobilen Variante vorgestellt. Anschliessend wird die Stabilität von Multicast-Verteilbäumen bei einer Sender- sowie Empfängerbewegung anhand realer Internetdaten analysiert.

## 5.1 Mobile IP

Ein Ziel von Mobile IP [Perkins (2002)] ist es, mobile Geräte unabhängig von ihrem Netzanschlusspunkt, unter einer konstanten IP-Adresse erreichbar zu machen. Wechselt ein mobiles Gerät sein Heimat-Netzwerk in ein anderes, so wird ihm normalerweise per DHCP [Drom (1993)], den Foreign Agent oder durch manuelle Konfiguration eine neue IP-Adresse zugewiesen. Durch den Wechsel der IP-Adresse würde nun eine Trennung aller bestehender Verbindungen stattfinden. Bei Mobile IP wird nun jedem mobilen Gerät eine Home Address in Form einer IP-Adresse in seinem Heimat-Netzwerk zugeordnet. Unter dieser Adresse ist der mobile Host unabhängig von seinem aktuellem Aufenthaltsort für Kommunikationspartner erreichbar. Im Heim-Netzwerk befindet sich auch der Home Agent, diese Rolle übernimmt normalerweise der Router des Heimat-Netzwerks. Dieser ist für die Verwaltung der Aufenthaltsorte der mobilen Hosts, sowie die Weiterleitung von IP-Paketen für die mobilen Hosts verantwortlich. Verbindet sich nun ein mobiler Host mit einem fremden Netzwerk, so registriert sich dieser bei einem Foreign Agent, der auch der Router des Netzwerks ist. Dabei teilt der Foreign Agent dem mobilen Host eine Care-Of Address zu, die dem aktuellen Aufenthaltsort entspricht. Die Home Address sowie die Care-Of Address wird als Binding bezeichnet und ist mit einem vom mobilen Host ausgestellten Zeitstempel versehen, so dass die Gültigkeit beschränkt wird. Beim Datenverkehr leitet nun der Home Agent ankommende Datenpakete an die Care-Of Address und damit an den Mobile Host weiter. Verschickt hingegen der mobile Host Pakete, kann dieser im Normalfall, das auch ohne den Umweg über den

Home Agent tun. Sobald der mobile Host wieder in sein Heimat-Netzwerk zurückkehrt, meldet sich dieser bei dem Foreign Agent wieder ab, so dass die Pakete wieder ohne Umweg zu diesem geleitet werden.

## 5.2 Multicast

Multicast Anwendungen benötigen oft eine hohe Bandbreite, so dass eine triviale Abbildung der 1:n Kommunikation auf n Unicast-Verbindungen zu einer Überlastung des Netzwerks führen kann, da der Sender das zu sendende Paket vervielfältigt und jedem Empfänger eine separate Kopie schickt. Abbildung 5.1 zeigt die Problematik und den Lösungsansatz von IP Multicast.

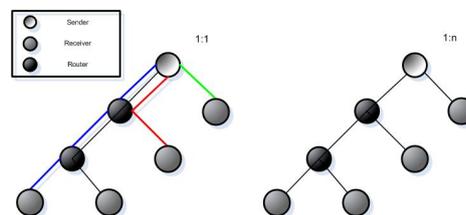


Abbildung 5.1: Verteilung eines Pakets über Unicast (links) vs. Versendung über Multicast (rechts)

Die Abbildung stellt den Versand eines Pakets an vier Empfänger, die auch über zwei Router verbunden sind, über Unicast-Verbindungen (links) und über IP Multicast (rechts) dar. Schon bei der ersten Teilstrecke ist die Netzwerkbelastung im Vergleich zu Multicast dreimal so hoch, da dort jedes Paket nur einmal pro Teilstrecke versandt wird.

Bei der Verwendung von IP Multicast als Routing Algorithmus wird häufig bei der Ankunft eines Multicast Pakets in jedem Router ein Verteilzustand kürzester Wege berechnet, der als Wurzel den Sender und als Blätter die Empfänger enthält. Der zusammengesetzte Baum wird als Multicast Delivery Tree bezeichnet. Dabei verwenden alle Router den gleichen Algorithmus zur Erstellung des Baums. Die Multicast-Pakete werden dann von allen Routern entlang des Baums auf ihrem Pfad nur einfach gesendet. Dadurch sinkt das Datenaufkommen. Pakete die an eine bestimmte, dynamische Gruppe von Geräten adressiert werden sollen, werden bei IP Multicast über Class D-Adressen identifiziert. Um nun ein Paket an alle Empfänger der Multicast-Gruppe zu senden, genügt es, das Paket an die Multicast-Adresse dieser Gruppe zu adressieren. Die Router erkennen dann am Aufbau der Empfängersadresse, dass es sich um ein Multicast-Paket handelt und routen es ihren Forwarding States entsprechend weiter. Empfänger, die einer bestimmten Multicast-Gruppe beitreten möchten, teilen dies ihren nächsten Routern über das Internet Group Management Protocol [Cain u. a. (2002)]

mit. Die Verweildauer ist dynamisch, ein Empfänger kann jederzeit einer Gruppe beitreten bzw. diese wieder verlassen. Dabei existieren zur eigentlichen Abwicklung des Multicast-Routing, also dem Aufbau des Multicast Delivery Trees, verschiedene Algorithmen:

- Multicast Open Shortest Path First [[Moy \(1994\)](#)]
- Distance-Vector Multicast Routing Protocol [[Waitzman u. a. \(1988\)](#)]
- Core Based Trees [[Ballardie \(1997\)](#)]
- Protocol Independent Multicast [[Deering u. a. \(1994\)](#)]

Die jeweiligen Funktionen dieser Algorithmen werden nicht näher erläutert, da diese für den weiteren Verlauf der Arbeit nicht relevant sind.

### 5.3 Multicast im Kontext der Mobilität

Bei normalen Multicast-Übertragungen wird davon ausgegangen, dass sich sowohl die Sender als auch die Empfänger an festen Orten aufhalten und sich deren Standpunkt nicht ändert. Die Mobilität der Kommunikationspartner ist aber eine zwingende Voraussetzung für die Umsetzung von Multicast-Übertragungen, in mobilen Umgebungen. So sind einige Änderungen notwendig, um in mobilen Bereich Multicast zu unterstützen. Bei der Umsetzung treten aber einige Problemstellungen auf [[Schmidt und Wählisch \(2007\)](#)]:

- Befindet sich ein Teilnehmer in einem Subnetz, bei dem der Multicast-Service gesperrt oder sogar nicht vorhanden ist, kann dieser keine Daten über Multicast empfangen.
- Bei der Multicast-Übertragung mit mobilen Geräten werden häufig Tunneling-Mechanismen genutzt. Dadurch steigen die Anforderungen an Bandbreite und Zeit.
- Durch die Bewegung der Teilnehmer müssen die Verteilbäume stets neu aufgebaut bzw. aktualisiert werden. Dies kostet Zeit und kann unter zu einem Abbruch der Verbindung kommen.

Ein wesentliches Ziel bei der Umsetzung von Multicast für die mobile Umgebung ist der nahtlose und vor allem verlustfreie Übergang von einem Subnetz zum nächsten. Ein Subnetz-Wechsel bedeutet, dass der Multicast-Verteilbaum neu aufgebaut werden müsste und die Teilnehmer sich jedes mal beim Multicast-Router anmelden. Diese Vorgänge benötigen aber zu viel Zeit, so dass bei einem Wechsel des Netzwerks die Datenverbindung zum mobilen Endgerät abbrechen würde.

### 5.3.1 Charakteristische Eigenschaften

Multicast Verteilbäume wurden zunehmend im Bezug auf ihre Netzwerk- und damit Kosteneffizienz hin untersucht. Basierend auf den empirischen Beobachtungen von Chuang Sirbu [Chuang und Sirbu (2001)], wurde ein Potenzgesetz für die gesamte Anzahl an Kanten in einem Multicast Shortest Path Tree mit  $m$  Empfängern vorgeschlagen ( $m^k$ ). Die Autoren ermittelten übereinstimmend für die Konstante  $k$  den Wert  $k = 0.8$ . Dies deutet darauf hin, dass Multicast Shortest Path Trees, ähnlich dem Internet, viele Knoten mit einem niedrigem, aber wenige Knoten mit einem hohen Knotengrad aufweisen. Somit müssten die Bäume mehr in die Tiefe, als in die Breite gehen. [Van Mieghem u. a. (2001)] zeigten, dass dieses Potenzgesetz nicht für ein wachsendes Internet oder sehr große Multicast-Gruppen stimmen kann. Allerdings ist dieses Potenzgesetz für eine moderate Anzahl an Empfängern und derzeitigen Internetgröße  $N = 10^5$  Core-Routern anwendbar. Janic und Van Mieghem [Janic und Van Mieghem (2006)] untersuchten die Selbstähnlichkeit von Shortest Path Trees und kamen zum Schluss, dass diese mit einer angemessenen Genauigkeit durch 'Uniform Recursive Trees' [Van Mieghem (2006)] modelliert werden können, sofern  $m$  kleiner gegen  $N$  ist.

Der Blickwinkel der Mobilität für Shortest Path Trees richtet sich auf die Veränderung dieser. Ein Beispiel dafür wäre das Ausmaß an topologischen Veränderungen, die durch Bewegung im mobilen Kontext verursacht werden. Sowohl für Empfänger als auch Sender kann dies als Maß für die Routing-Komplexität dienen. Mobile Empfänger, die sich in benachbarte Netzwerke begeben, verändern nur einige Äste des Baumes. Senderspezifische Multicast-Bäume, die nach einer Senderübergabe generiert werden, sind nicht unabhängig, sondern in hohem Masse korreliert. Diese stimmen meistens bei einigen Knotenpunkten der identischen Empfänger überein. Aufgrund der selbstähnlichen Struktur zeigen die Teilbäume ein Verhalten, das dem Potenzgesetz entspricht. Baumänderungen im mobilen Kontext wurden durch Simulation als auch analytisch in [Schmidt und Wählisch (2005)] erforscht. Dabei wurde dokumentiert, dass selbst bei großen Netzwerken und auch bei einer mittelmäßigen Anzahl an Empfängern, mehr als 80% der Multicast-Router ihren Zustand beibehalten, trotz einer Senderübergabe.

## 5.4 Analyse

In diesem Abschnitt wird die Stabilität von Multicast Verteilbäumen unter der Sender- und Empfängermobilität untersucht, dabei werden die Topologiedaten von Skitter und Dimes verwendet und deren Ergebnisse anschliessend verglichen.

Die folgenden Simulationsergebnisse stammen aus dem diskreten ereignisorientierten Simulator OMNeT++. Der Multicast-Verteilbaum setzt sich aus den kürzesten Wegen von der Quelle (Sender) zu den Empfängern zusammen, wobei wir vom symmetrischen Routing ausgehen. Sowohl die Quelle als auch die Empfänger werden gleich verteilt an Knoten mit dem Grad 1 gewählt, die die Edge-Router darstellen. Als Netzwerktopologien wurden die realen Internet-Daten aus dem Skitter-Monitorpunkt <sup>1</sup> *ihf* (Oktober 2006) sowie ein DIMES-Datensatz von Oktober 2006 genommen. Es wurde der gleiche Zeitraum gewählt, um die Topologien leichter vergleichen zu können. Mittels unseres erweiterten BRITE wurden Teilgraphen der Größe 154, 1540 sowie 15400 erstellt <sup>2</sup> und in das für OMNeT benötigte ned-Format exportiert.

Bei der Senderbewegung wird vom Programmablauf zunächst das ganze Netzwerk eingelesen und die Core- sowie Edge-Router bestimmt. Anschließend wird der erste Shortes Path Tree (SPT) zwischen altem Sender und den Empfängern aufgebaut. Nach diesem Schritt wechselt der Sender seine Position und ein zweiter SPT wird zwischen dem neuen Sender und den Empfängern aufgebaut. Die Baum-Übereinstimmung des ersten und zweiten SPT sowie nur der neue SPT werden in Vektordateien gespeichert. Anhand dieser Daten kann nun die Baum-Koinzidenz sowie die neu entdeckten Router für sich bewegende Sender berechnet werden. Die Parameter, die dabei verwendet werden, sind zunächst die Anzahl an Wiederholungen, die in einem Durchlauf berechnet werden. Dies stellt sicher, dass der Verlauf eines Graphen, der ja Mittelwerte über die iterierten Simulationsereignisse darstellt, nicht durch statistische Fluktuationen gestört wird. Ein Durchlauf bezieht sich immer auf die variierenden Empfänger (1-200) bzw. auf die variierende pDR (alter designierter Router) - nDR (neuer designierter Router) Distanz. Dabei bezieht sich bei der Sendermobilität die pDR-nDR-Entfernung auf eine wechselnde Quelle. Bei der Empfängerbewegung wechselt aus n Empfängern immer nur ein Empfänger seine Position, der Sender bleibt unverändert. Der restliche Programmablauf gleicht der Senderbewegung, nur dass sich hier die pDR-nDR-Entfernung auf einen wechselnden Empfänger bezieht. Beide Programme wurden von Matthias Wählisch programmiert.

---

<sup>1</sup> Es wurde nur ein Monitorpunkt benutzt, da der Speicher von 2Gigabyte für mehr Monitorpunkte nicht gereicht hat

<sup>2</sup> Die Simulationszeiten ließen es nicht zu, alle Szenarien mit 15400 Knoten zu berechnen

Abbildung 5.2 vergleicht die Senderbewegung der verschiedenen Topologiedaten in Abhängigkeit einer variierenden Empfängeranzahl und einer festen Distanz zwischen pDR und nDR (alter sowie neuer designierter Router) des mobilen Knotens (Sender). Die steigende Empfängeranzahl erhöht hier die Wahrscheinlichkeit für die Wiederverwendung eines bereits erstelltem Teilbaumes, da mehr Edge-Router besetzt sind und somit mehr Teilzweige erstellt werden, denen sich die Multicast-Quelle nähern kann. Hier zeigen sich zunächst keine signifikanten Unterschiede zwischen den Skitter- sowie DIMES-Topologien, allerdings hat das Skitter-Netzwerk der Größe 15400 eine etwas höhere Baum-Übereinstimmung als das DIMES-Netzwerk. Abbildung 5.3<sup>3</sup> zeigt, dass die Anzahl an neu entdeckten Routern in den jeweiligen Netzwerkgrößen relativ konstant bleibt, obwohl die Empfängeranzahl zunimmt. Lediglich bei den 15400er Netzwerken findet eine Steigung statt. Prozentual gesehen nimmt aber die Anzahl an neu entdeckten Routern bei steigenden Empfängern ab. Dabei zeigt sich, dass die Steigung bei den neu entdeckten Routern im DIMES-15400-Netzwerk wesentlich höher ausfällt, als im Skitter-15400-Netzwerk, wobei mehr als doppelt so viele neue Router in dem DIMES-Netzwerk enthalten sind. In der DIMES-Topologie sind im Vergleich zu der Skitter-Topologie ca. 1000 Edge-Router mehr enthalten, dafür aber auch 1000 Core-Router weniger. Da der Knotengrad bei den extrahieren Topologien gleich bleibt, bedeutet dies, dass in dem Skitter-Netzwerk die inneren Router, die auch am Verteilbaum beteiligt sind, mehr Kanten zu anderen Knoten haben. Das würde darauf hindeuten, dass sich der Verteilbaum im DIMES-Netzwerk aus mehr Routern zusammensetzt, als im Skitter-Netzwerk. Dadurch ist die Wahrscheinlichkeit höher, dass die Koinzidenz abnimmt und die Anzahl an neu entdeckten Routern steigt.

Zur Gegenkontrolle wurde nun in Abbildung 5.4 für eine feste Empfängeranzahl die Schrittweite (pDR-nDR-Distanz) des Senders verändert. Dabei verhalten sich die Ergebnisse qualitativ ähnlich zu den Abbildungen 5.2 und 5.3. Je weiter sich die Quelle wegbewegt, desto mehr neue Teilzweige entstehen. Dies wird nochmal in der Abbildung 5.5 bestätigt, dabei nimmt die Anzahl an neu entdeckten Routern zu, je weiter sich die Quelle entfernt. Bei den Abbildungen 5.4 und 5.5 sollte beachtet werden, dass für die Netzwerkgrößen 154, die maximale Distanz 10 und für die Netzwerkgröße 1540 schon 20 betrug. Erst bei einer Netzwerkgröße von 15400 konnte die volle Distanz 1-30 berechnet werden. Abbildungen 5.4 und 5.5 zeigen einen größeren Unterschied zwischen den beiden 15400er Netzwerken. Hier nimmt die Übereinstimmung der Verteilbäume im DIMES-Netzwerk im Vergleich zu den Skitter-Netzwerk stark ab. Dies zeigt sich auch an der Anzahl an neu entdeckten Router, die im DIMES-Netzwerk fast doppelt so hoch ist, wie im Skitter-Netzwerk. Dieses Verhalten zeigte sich auch in den beschriebenen Abbildungen 5.2 und 5.3.

Zusätzlich zu den Ergebnissen aus den Abbildungen 5.2 und 5.3, wurden nun die pDR-nDR Distanzen auf jeweils 10 sowie 15 erhöht, wobei ab einer Distanz von 15, die Netzwerkgröße

---

<sup>3</sup>Aufgrund des Abgabedatums, konnte die Skitter-15400-Topologie nicht zu ende berechnet werden

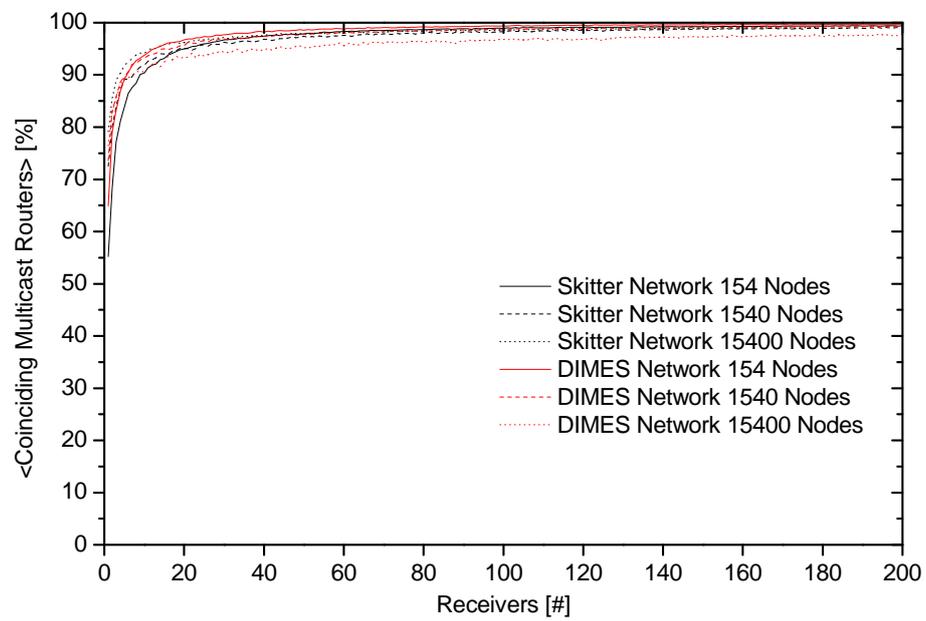


Abbildung 5.2: Senderbewegung - Baum-Koinzidenz - pDR-nDR Distanz 5 Knoten

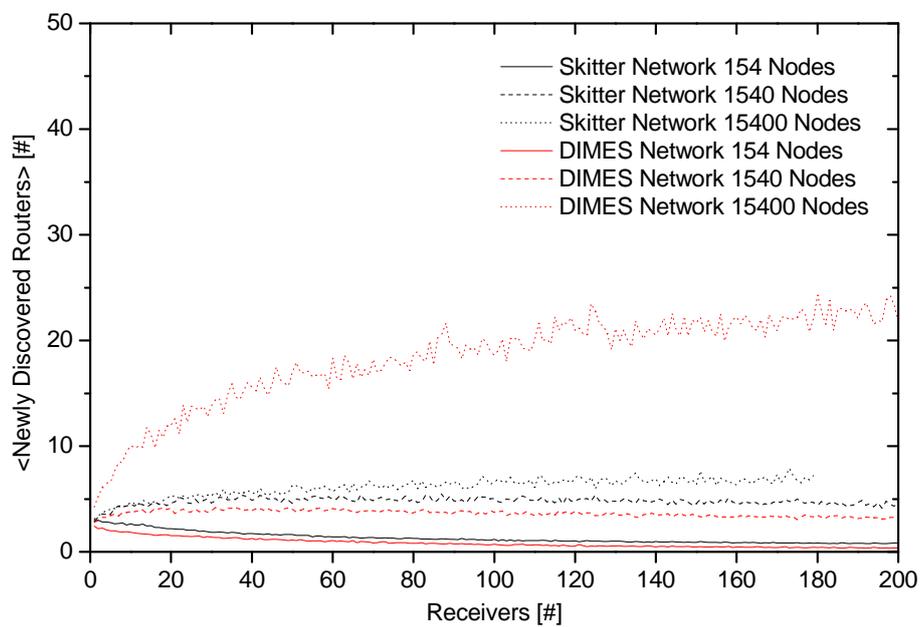


Abbildung 5.3: Senderbewegung - neu entdeckte Router - pDR-nDR Distanz 5 Knoten

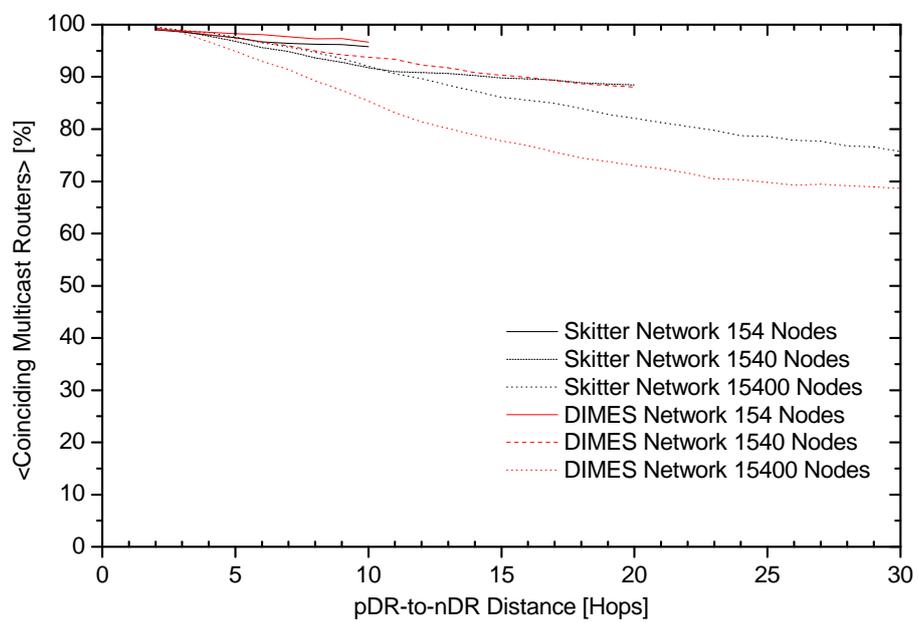


Abbildung 5.4: Senderbewegung - Baum-Koinzidenz - 40 Empfänger

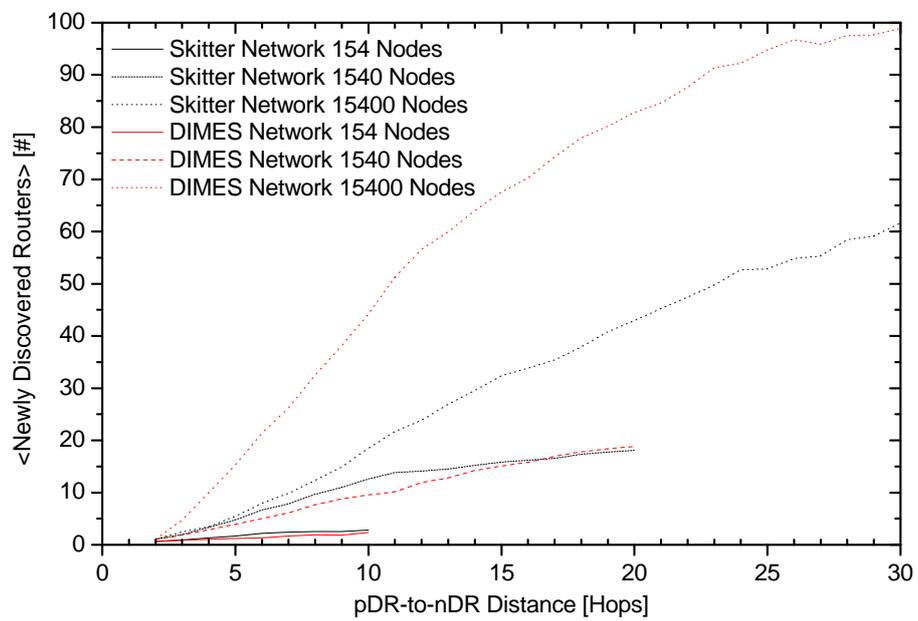


Abbildung 5.5: Senderbewegung - neu entdeckte Router - 40 Empfänger

154 aufgrund der geringen Größe nicht berechnet werden konnte. Dabei zeigen Abbildungen 5.6 und 5.8, dass die Baum-Koinzidenz trotz Erhöhung der pDR-nDR Distanz und einer steigenden Empfängeranzahl quantitativ gleich bleibt. Die Baum-Übereinstimmung liegt immer über 95%. Lediglich bei sehr wenigen Empfängern (1-20) nimmt die Koinzidenz im Vergleich zu Abbildung 5.2 deutlich ab, was aber aufgrund der Ergebnisse aus der Gegenkontrolle normal ist, da sich bei einer weiter entfernenden Quelle, mehr Teilzweige erzeugt werden. Dies wird auch in den Abbildungen 5.6 und 5.7 nochmal dargestellt. Interessant ist hierbei zu sehen, dass die Anzahl an den neu entdeckten Routern Anfangs leicht zunimmt, aber mit steigender Empfängeranzahl wieder leicht abnimmt. Trotzdem nimmt der prozentuale Anteil an den neuen Routern ab. Dabei zeigt der Vergleich zwischen den DIMES- und Skitter-Messungen, dass sich beide Netzwerke in den unterschiedlichen Größen bei der Baum-Übereinstimmung ähneln. Die Ergebnisse aus den Graphen mit den neu entdeckten Routern zeigen allerdings einen geringen Unterschied zwischen beiden Netzwerken. So ist die Anzahl an neu entdeckten Router in den Skitter-Netzwerken geringfügig höher als in den Dimes-Netzwerken. Da es sich aber um einen Unterschied von maximal vier Router in einem Netzwerk von über 1540 Knoten handelt, hat dies keine wesentlichen Auswirkungen auf die Baumübereinstimmung, was auch Abbildungen 5.6 und 5.8 zeigen.

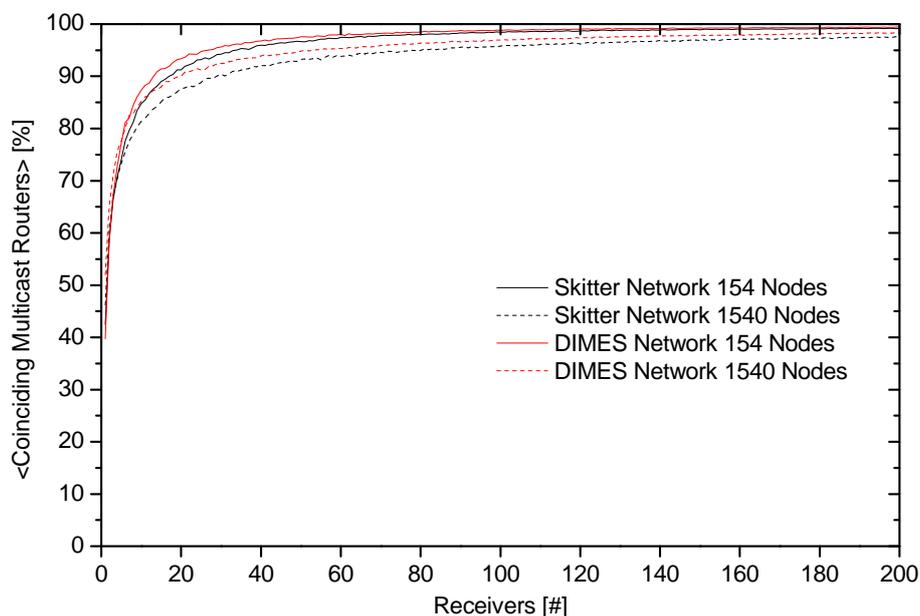


Abbildung 5.6: Senderbewegung - Baum-Koinzidenz - pDR-nDR Distanz 10 Knoten

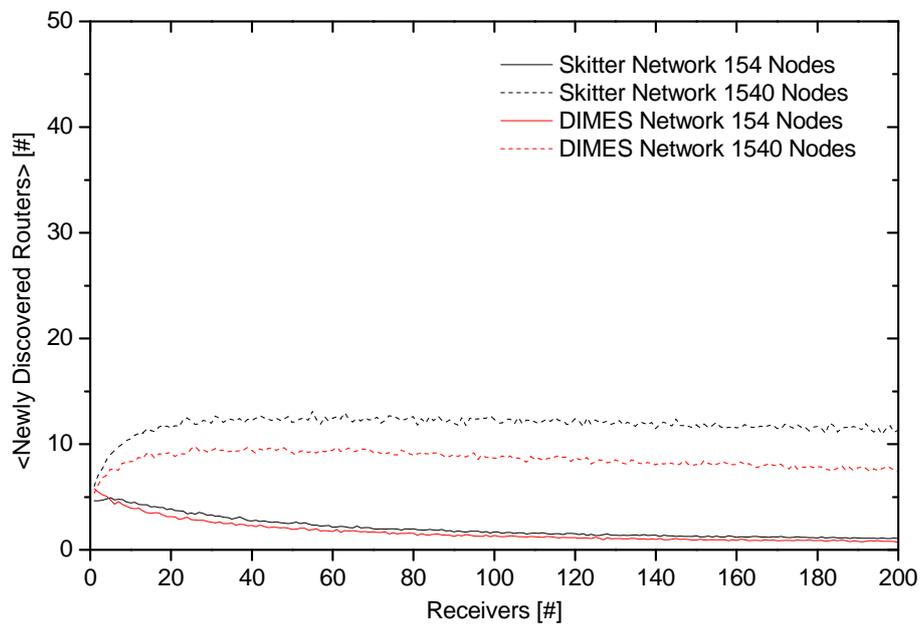


Abbildung 5.7: Senderbewegung - neu entdeckte Router - pDR-nDR Distanz 10 Knoten

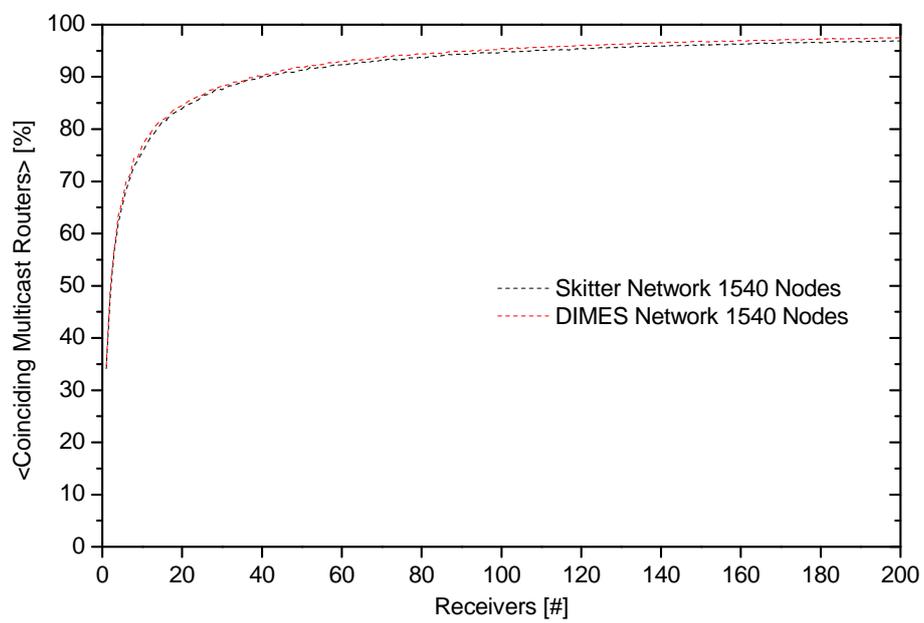


Abbildung 5.8: Senderbewegung - Baum-Koinzidenz - pDR-nDR Distanz 15 Knoten

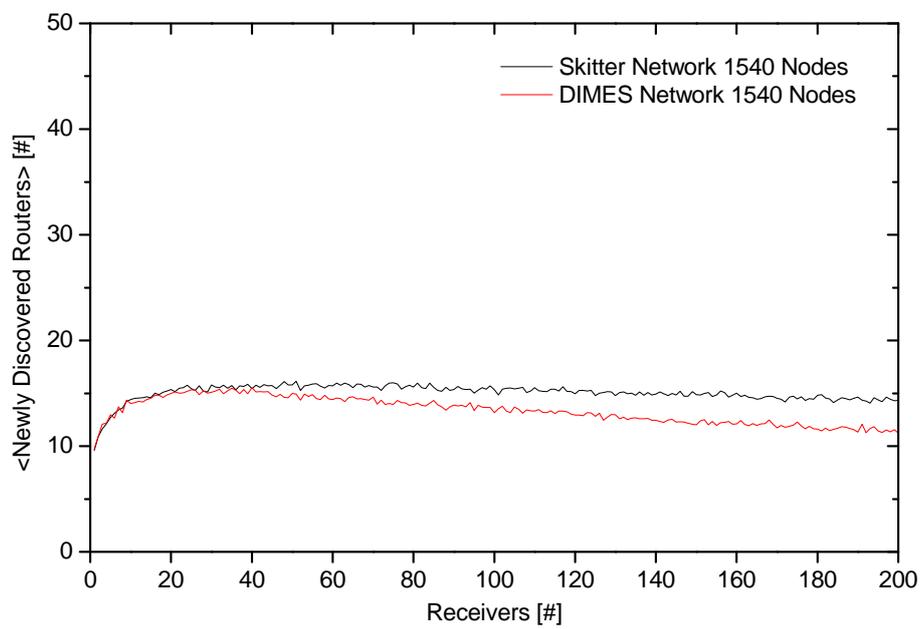


Abbildung 5.9: Senderbewegung - neu entdeckte Router - pDR-nDR Distanz 15 Knoten

Abbildung 5.10 vergleicht die Empfängerbewegung in Abhängigkeit einer variierenden Empfängeranzahl und einer festen Distanz zwischen pDR und nDR des mobilen Knotens. Die steigende Empfängeranzahl erhöht, wie auch bei der Senderbewegung, die Wahrscheinlichkeit für die Wiederverwendung eines bereits erstelltem Teilbaumes. So zeigen die 154er Netzwerkgrößen, dass die Übereinstimmung schon nach ca. 20 Empfängern einen sehr hohen Wert erreicht, bei 80 Empfängern scheint das Netzwerk voll besetzt zu sein. Allerdings konnten keine weiteren Werte mehr bestimmt werden, dies deutet darauf hin, dass es bei den kleinen Topologien vermutlich nicht mehr als 80 verschiedene äußere Router gibt. Die größeren Topologien verhalten sich wie erwartet und die Koinzidenz nimmt kontinuierlich zu, je mehr Empfänger hinzukommen. Dabei zeigt das Skitter-15400-Netzwerk ein etwas seltsames Verhalten gegenüber dem entsprechenden DIMES-Netzwerk. Hier ist die Baum-Übereinstimmung im 15400er Netzwerk höher als in den kleineren Netzwerken, wobei diese mit zunehmender Empfängeranzahl wie erwartet unter den kleineren Netzwerken liegt. Allerdings liegt diese Übereinstimmung um ca 5% höher als in der DIMES-Topologie. Abbildung 5.11 zeigt, dass die Anzahl an neu entdeckten Routern in den jeweiligen Netzwerkgrößen konstant bleibt, wobei die kleinen Topologien stark abnehmen. Dies liegt allerdings an der geringen Anzahl der Edge-Router, die bei ca. 80 Empfänger das Maximum erreicht und danach keine Empfänger mehr aufgenommen werden können. Dadurch sind alle Edge-Router besetzt und bei einer Empfängerbewegung würden nun zwei Empfänger die Position tauschen und es existieren keine neuen Router. Dabei zeigen die zwei unterschiedlichen Topologien keine wesentlichen Unterschiede, lediglich die unterschiedlichen Netzwerkgrößen unterscheiden sich. Dies liegt aber an der sehr kleinen Netzwerkgröße von 154, wo die oben genannten 'Randeffekte' einen großen Einfluss auf das Ergebnis haben.

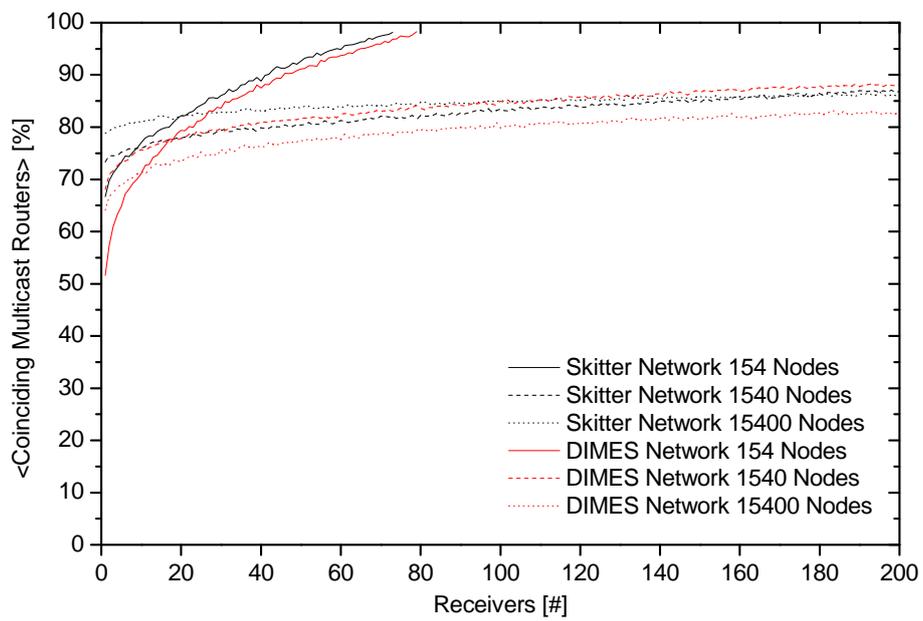


Abbildung 5.10: Empfängerbewegung - Baum-Koinzidenz - pDR-nDR Distanz 5 Knoten

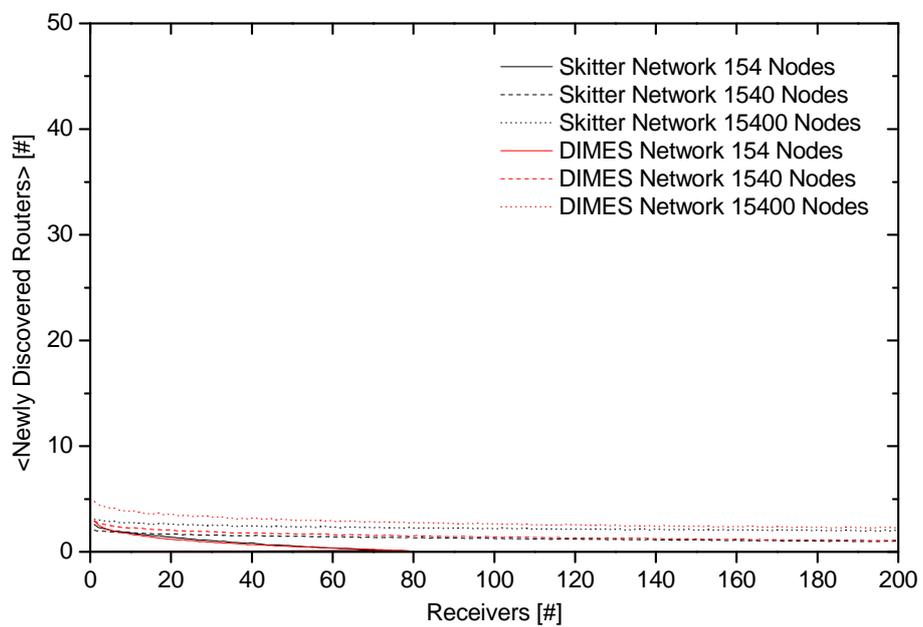


Abbildung 5.11: Empfängerbewegung - neu entdeckte Router - pDR-nDR Distanz 5 Knoten

Um nun zu sehen, wie die pDR-nDR Distanz das Ergebnis beeinflusst, wurde diese von 5 auf 10 erhöht. So zeigt Abbildung 5.12 qualitativ ähnliche Ergebnisse wie Abbildung 5.10. Allerdings ist die Baum-Übereinstimmung, aufgrund der höheren Distanz, im Anfangsbereich gering, nimmt aber mit steigender Empfängeranzahl zu. Abbildung 5.13 zeigt den gleichen Verlauf wie in Abbildung 5.11, lediglich die Anzahl an neuen Router nimmt zu, da sich aufgrund der erhöhten Distanz, größere Verteilbäume bilden und mehr Router beteiligt sind. Interessant ist hier zu sehen, dass die DIMES-Topologie im Vergleich zu der Skitter-Topologie eine höhere Baum-Übereinstimmung erreicht, obwohl die Anzahl an neuen Routern in beiden Netzwerken gleich ist.

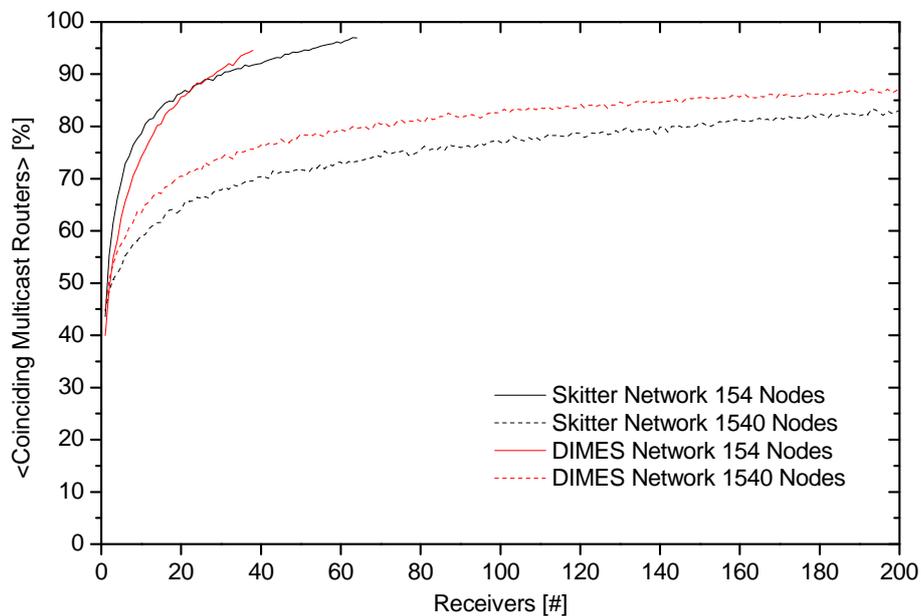


Abbildung 5.12: Empfängerbewegung - Baum-Koinzidenz - pDR-nDR Distanz 10 Knoten

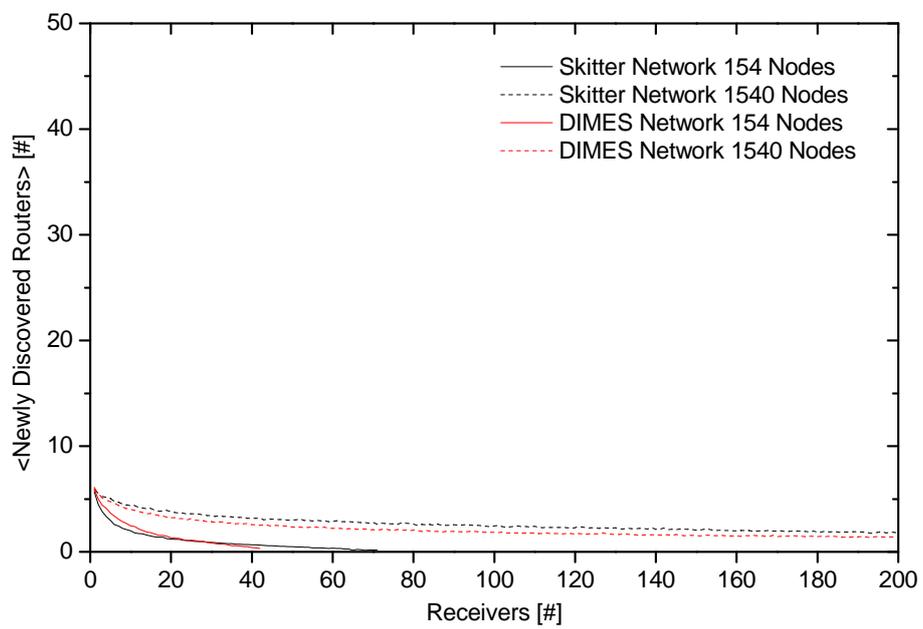


Abbildung 5.13: Empfängerbewegung - neu entdeckte Router - pDR-nDR Distanz 10 Knoten

Die pDR-nDR Distanz wurde nun von 10 auf 15 erhöht. Die dabei erzielten Ergebnisse entsprechen der vorherigen Erhöhung der Distanz und sind in den Abbildungen 5.14 sowie 5.15 zu sehen. Dabei nimmt die Anzahl an neuen Routern um die Hälfte ab. Hier ist die Baum-Übereinstimmung in der DIMES-Topologie zwischenzeitlich höher als in der Skitter-Topologie.

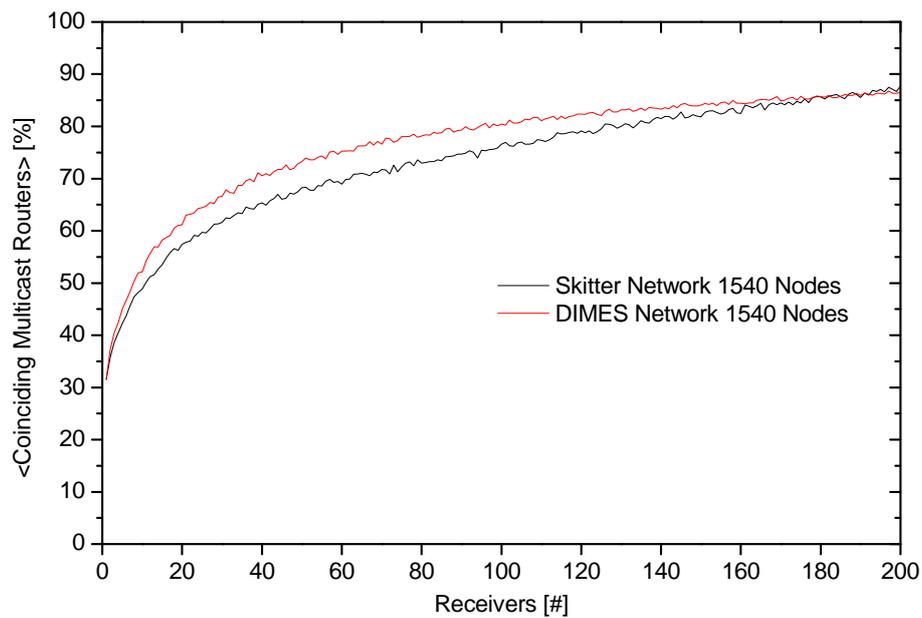


Abbildung 5.14: Empfängerbewegung - Baum-Koinzidenz - pDR-nDR Distanz 15 Knoten

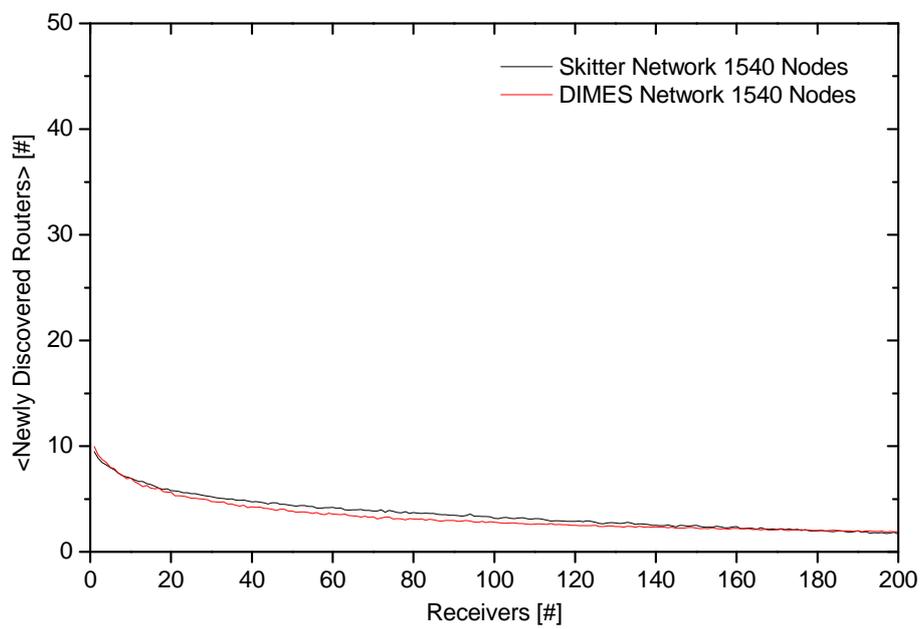


Abbildung 5.15: Empfängerbewegung - neu entdeckte Router - pDR-nDR Distanz 15 Knoten

Es war anzunehmen, dass die Baum-Koinzidenz bei einer Empfängerbewegung einen höheren Wert erreicht, als bei der Senderbewegung. Bei der Senderbewegung muss immer ein neuer Teilzweig aufgebaut werden, sobald der Sender die Position verändert. Sobald aber die Empfänger ihre Position verändern, können diese auch untereinander die Position wechseln, was zur Folge hat, dass kein neuer Teilzweig entsteht und damit eine sehr hohe Koinzidenzverhalten gezeigt wird. Allerdings zeigt sich in den Ergebnissen, zumindest im Koinzidenz-Verhalten, dass dies nicht ganz zutrifft. Bei der Senderbewegung nimmt die Baum-Übereinstimmung schon bei wenigen Empfängern stark zu, bei der Empfängerbewegung nimmt diese Übereinstimmung nur langsam zu. Dies liegt aber daran, dass bei wenigen Empfängern sehr oft ein neuer Teilbaum erzeugt werden muss, da sich der Empfänger an einer neuen Position befindet, lediglich der Teilzweig beim Sender bleibt gleich. Bei der Senderbewegung bleibt der Teilzweig im Empfängerbereich unverändert und scheint einen stärkeren Einfluss auf die Baum-Koinzidenz zu haben als erwartet. Die neu entdeckten Router zeigen im Vergleich zwischen Sender- und Empfängerbewegung ein großen Unterschied. Dabei steigt bei der Senderbewegung zunächst die Anzahl an Router in den größeren Topologien und nimmt anschliessend wieder leicht ab. Bei der Empfängerbewegung nimmt die Anzahl geringfügig ab, bleibt aber im Vergleich zu der Senderbewegung in den größeren Topologien gering.

## 6 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Verarbeitungskette für reale Internet Topologiedaten entworfen und realisiert. Dabei wurden zunächst Anforderungen an die Verarbeitungskette gestellt. Aufgrund dieser Anforderungen wurden einige Werkzeuge, die dafür in Frage kommen, näher betrachtet und anhand von Vor- und Nachteilen ausgewählt. Diese Werkzeuge mussten teilweise in ihrer Funktion sowohl überarbeitet als auch erweitert werden um den Anforderungen zu entsprechen. Die Ergebnisse wurden anschließend für eine Analyse der Stabilität von Multicast Verteilbäumen, im Bezug auf die Sender- sowie Empfängermobilität verwendet.

Dabei wurden in Kapitel 2 die Grundlagen zu der Gewinnung von realen Internet Topologiedaten erarbeitet und deren Problemstellungen im Bezug auf das Messen erörtert. Einrichtungen, die kontinuierlich das Internet auf unterschiedliche Weise vermessen, wurden vorgestellt und Vor- und Nachteile ihrer jeweiligen Messdaten diskutiert.

Aus diesen Grundlagen entstanden Anforderungen an die Verarbeitungskette, die in Kapitel 3 dargestellt sind. Anhand dieser Anforderungen wurde die Verarbeitungskette realisiert.

In Kapitel 4 werden die notwendigen Implementierungen und die daraus resultierenden Ergebnisse sowie Probleme gezeigt. Dabei wurde BRITE erweitert, um sowohl Skitter- als auch DIMES-Datensätze einlesen zu können. Auch wurde ein Filter implementiert, um große Netzwerke anhand von bestimmten Eigenschaften auf kleine Netzwerke zu extrahieren.

Kapitel 5 bildet den Abschluss der Arbeit. Dabei wurden zunächst die Grundlagen zu Multicast vorgestellt. Anschließend wurden die gewonnenen Ergebnisse aus der Verarbeitungskette, im Bereich der Sender- sowie Empfängerbewegung bei mobilen Multicast, analysiert.

Dabei zeigte sich, dass die Ergebnisse aus der Verarbeitungskette ähnliche Resultate wie in [Schmidt und Wählich (2005)] erzielten. Neu ist hier die Erkenntnis, dass selbst bei großen Netzwerken und auch bei einer mittelmäßigen Anzahl an Empfängern, die Anzahl an neu entdeckten Multicast-Router konstant bleibt, wobei diese aber Prozentual gesehen abnimmt. Somit werden nicht mehr als 10% der Multicast-Router, in dem neuen Multicast-Verteilbaum nach einer Senderübergabe, aufgenommen. Das Koinzidenz-Verhalten ist also nicht für alle Graphen abhängig von der Netzwerkgröße. Eine weitergehende Analyse sollte stärker auf theoretische Aspekte eingehen und primär große Netzwerke untersuchen.

# Literaturverzeichnis

- [Skitter ] *Skitter*. <http://www.caida.org/tools/measurement/skitter/>.  
– Zugriffsdatum: 2007-05-12
- [Ballardie 1997] BALLARDIE, A.: *RFC 2189: Core Based Trees (CBT version 2) Multicast Routing*. September 1997
- [Cain u. a. 2002] CAIN, B. ; DEERING, S. ; KOUVELAS, I. ; FENNER, B. ; THYAGARAJAN, A.: Internet Group Management Protocol, Version 3 / Internet Engineering Task Force. URL <http://www.rfc-editor.org/rfc/rfc3376.txt>, Oktober 2002 (3376). – RFC. – 53 S
- [Chuang und Sirbu 2001] CHUANG, John C. I. ; SIRBU, Marvin A.: Pricing Multicast Communication: A Cost-Based Approach. In: *Telecommunication Systems* 17 (2001), Nr. 3, S. 281–297. – Presented at the INET'98, Geneva, Switzerland, July 1998
- [Claffy 2001] CLAFFY, K. C.: CAIDA: Visualizing the Internet. In: *IEEE Internet Computing* 5 (2001), Nr. 1, S. 88. – ISSN 1089-7801
- [Deering u. a. 1994] DEERING, Stephen E. ; ESTRIN, Deborah ; FARINACCI, Dino ; JACOBSON, Van ; LIU, Ching-Gung ; WEI, Liming: An Architecture for Wide-Area Multicast Routing. In: *SIGCOMM*. London, UK, August 1994, S. 126–135
- [Drom 1993] DROM, R.: Dynamic Host Configuration Protocol. Oktober 1993 (1541). – RFC
- [Erdős und Rényi 1959] ERDŐS, Paul ; RÉNYI, Alfréd: On Random Graphs. In: *Publ. Math., Debrecen* 6 (1959), S. 290–297
- [Faloutsos u. a. 1999] FALOUTSOS, Michalis ; FALOUTSOS, Petros ; FALOUTSOS, Christos: On power-law relationships of the Internet topology. In: *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication* Bd. 29. New York, NY, USA : ACM Press, October 1999, S. 251–262. – URL <http://portal.acm.org/citation.cfm?id=316229>. – ISBN 1581131356
- [Gamma u. a. 1994] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns: Elements of Reusable Object-Oriented Software*. Massachusetts : Addison Wesley, 1994

- [Govindan und Tangmunarunkit 2000] GOVINDAN, Ramesh ; TANGMUNARUNKIT, Hongsuda: Heuristics for Internet Map Discovery. In: *Proceedings of the 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00)*. Los Alamitos : IEEE, März 26–30 2000, S. 1371–1380
- [Heckmann u. a. 2003] HECKMANN, Oliver ; PIRINGER, Michael ; SCHMITT, Jens ; STEINMETZ, Ralf: On Realistic Network Topologies for Simulation. In: *MoMeTools '03: Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*. New York, NY, USA : ACM Press, August 2003, S. 28–32
- [Jacobson 1989] JACOBSON, Van: *Traceroute Tool*. <ftp://ftp.ee.lbl.gov/traceroute.tar.gz>. 1989
- [Janic und Van Mieghem 2006] JANIC, Milena ; VAN MIEGHEM, Piet: On properties of multicast routing trees. In: *Int. J. Commun. Syst.* 19 (2006), Nr. 1, S. 95–114. – ISSN 1074-5351
- [Magoni 2002] MAGONI, Damien: *nem: A Software for Network Topology Analysis and Modeling*. In: *Proceedings of the 10th IEEE Symposium on Modeling, Analysis and Simulation of Computer & Telecomm. Systems (MASCOTS'02)*. Fort Worth, Texas, USA : IEEE Computer Society, October 2002, S. 364–371
- [Magoni und Pansiot 2002] MAGONI, Damien ; PANSIOT, Jean-Jacques: Internet Topology Modeler Based on Map Sampling. In: *iscc 00 (2002)*, S. 1021. – ISSN 1530-1346
- [Medina u. a. 2001] MEDINA, Alberto ; LAKHINA, Anukool ; MATTA, Ibrahim ; BYERS, John: BRITE: An Approach to Universal Topology Generation. In: *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*. Washington, DC, USA : IEEE Computer Society, August 2001, S. 346
- [Medina u. a. 2000] MEDINA, Alberto ; MATTA, Ibrahim ; BYERS, John: On the origin of power laws in Internet topologies. In: *SIGCOMM Comput. Commun. Rev.* 30 (2000), Nr. 2, S. 18–28. – ISSN 0146-4833
- [Milgram 1967] MILGRAM, S.: The small world problem. In: *Psychology Today* 1 (1967), S. 61
- [Moy 1994] MOY, John: Multicast Routing Extensions for OSPF. In: *Commun. ACM* 37 (1994), Nr. 8, S. 61–66
- [Paxson 1999] PAXSON, Vern: End-to-End Internet Packet Dynamics. In: *IEEE/ACM Trans. Netw.* 7 (1999), Nr. 3, S. 277–292. – ISSN 1063-6692
- [Perkins 2002] PERKINS, Charles: IP Mobility Support for IPv4 / Internet Engineering Task Force. URL <http://www.ietf.org/rfc/rfc3344.txt>, August 2002 (3344). – RFC

- [Postel 1981] POSTEL, John: Internet Control Message Protocol / Internet Engineering Task Force. URL <http://www.rfc-editor.org/rfc/rfc792.txt>, September 1981 (792). – RFC
- [Schmidt und Wählisch 2005] SCHMIDT, T.C. ; WÄHLISCH, M.: Predictive versus Reactive – Analysis of Handover Performance and Its Implications on IPv6 and Multicast Mobility. In: *Telecommunication Systems* 30 (2005), November, Nr. 1-3, S. 123–142
- [Schmidt und Wählisch 2006] SCHMIDT, T.C. ; WÄHLISCH, M.: Morphing Distribution Trees – On the Evolution of Multicast States under Mobility and an Adaptive Routing Scheme for Mobile SSM Sources. In: *Telecommunication Systems* 33 (2006), Dezember, Nr. 1-3, S. 131–154
- [Schmidt und Wählisch 2007] SCHMIDT, Thomas C. ; WÄHLISCH, Matthias: Multicast Mobility in MIPv6: Problem Statement and Brief Survey / MobOpts. URL <http://tools.ietf.org/html/draft-irtf-mobopts-mmcastv6-ps>, July 2007 (01). – IRTF Internet Draft – work in progress
- [Shavitt und Shir 2005] SHAVITT, Yuval ; SHIR, Eran: DIMES: let the internet measure itself. In: *SIGCOMM Comput. Commun. Rev.* 35 (2005), Nr. 5, S. 71–74. – URL <http://www.arxiv.org/abs/cs.NI/0506099>. – ISSN 0146-4833
- [Van Mieghem 2006] VAN MIEGHEM, Piet: *Performance Analysis of Communications Networks and Systems*. Cambridge : Cambridge University Press, 2006
- [Van Mieghem u. a. 2001] VAN MIEGHEM, Piet ; HOOGHIEMSTRA, Gerard ; HOFSTAD, Remco van der: On the Efficiency of Multicast. In: *IEEE/ACM Trans. Netw.* 9 (2001), Nr. 6, S. 719–732. – ISSN 1063-6692
- [Varga 2005] VARGA, András: *The OMNeT++ discrete event simulation system*. <http://www.omnetpp.org>. 2005
- [Waitzman u. a. 1988] WAITZMAN, D. ; PARTRIDGE, C. ; DEERING, S.E.: Distance Vector Multicast Routing Protocol. November 1988 (1075). – Forschungsbericht
- [Watts und Strogatz 1998] WATTS, Duncan J. ; STROGATZ, Steven H.: Collective dynamics of ‘small-world’ networks. In: *Nature* 393 (1998), S. 440–442
- [Waxman 1991] WAXMAN, Bernard M.: Routing of multipoint connections. (1991), S. 347–352. ISBN 0-8186-8926-9
- [Zegura u. a. 1997] ZEGURA, Ellen W. ; CALVERT, Kenneth L. ; DONAHOO, Michael J.: A quantitative comparison of graph-based models for Internet topology. In: *IEEE/ACM Trans. Netw.* 5 (1997), Nr. 6, S. 770–783. – ISSN 1063-6692

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 7. August 2007

Ort, Datum

Unterschrift