

UNIVERSITÄT SIEGEN

BACHELORARBEIT

---

# Simulation von crowd-basiertem Transport auf Straßennetzen

---

*Autor:*

Nils STAHLHUT

*Erstprüfer:*

Prof. Dr. Volker WULF

*Zweitprüfer:*

Juniorprof. Dr. Gunnar STEVENS

*Betreuer:*

Dr. Lars FISCHER

Matrikelnr.: 862251

7. April 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
<b>2</b>	<b>Grundlagen</b>	<b>6</b>
2.1	DTNs und MANETs . . . . .	6
2.2	Simulationen . . . . .	7
2.3	Projekt POST . . . . .	8
<b>3</b>	<b>Simulationsdesign</b>	<b>9</b>
3.1	Transportkonzept . . . . .	9
3.2	Simulation oder reale Kontakttereignisse . . . . .	10
3.2.1	Contact Traces . . . . .	10
3.2.2	Simulation . . . . .	11
3.3	Algorithmen . . . . .	12
3.3.1	Random Waypoint Algorithmus . . . . .	12
3.3.2	Cluster Waypoint Algorithmus . . . . .	12
3.4	Straßennetz . . . . .	13
3.4.1	Wegfindung . . . . .	14
3.5	Simulationszyklus . . . . .	15
<b>4</b>	<b>Implementation</b>	<b>16</b>
4.1	GUI . . . . .	16
4.1.1	InitialGUI . . . . .	16
4.1.2	MapGUI . . . . .	16
4.2	Objects . . . . .	17
4.2.1	Coordinate . . . . .	17
4.2.2	SimNode . . . . .	17
4.2.3	SimWay . . . . .	17
4.2.4	SimMap . . . . .	18
4.2.5	Packet . . . . .	18
4.2.6	User . . . . .	19
4.3	Simulation . . . . .	19
4.3.1	MapSimulation . . . . .	19
4.4	IO . . . . .	20
4.4.1	CSVLogging . . . . .	20
4.4.2	XMLParser . . . . .	20
<b>5</b>	<b>Ergebnisse der Simulation</b>	<b>21</b>
5.1	Simulationsparameter . . . . .	21
5.2	Ergebnisse . . . . .	24
5.2.1	Rennerod-RWP . . . . .	24
5.2.2	Neubrandenburg-RWP . . . . .	24
5.2.3	Rennerod-Cluster . . . . .	29

5.2.4	Neubrandenburg-Cluster . . . . .	29
5.3	Analyse . . . . .	34
<b>6</b>	<b>Fazit und Ausblick</b>	<b>38</b>
6.1	Fazit . . . . .	38
6.2	Ausblick . . . . .	39

# 1 Einleitung

Der Wunsch, eine Sache von A nach B befördern zu lassen, wird heutzutage von Postdienstleistern erfüllt. Besagtes Gut wird in ein Paket verpackt und dem Dienstleister überlassen, welcher für ein Entgelt ermöglicht, das Paket mit Hilfe eines zentralisierten Prozesses zu transportieren und auszuliefern. Das zu verschickende Paket einfach einer anderen Person auf der Straße mitzugeben, oder am Straßenrand abzustellen, woraufhin dieses sich durch ein robustes Routing-Protokoll und ein Netz von dezentralisierten Trägern den Weg zum Ziel selbst sucht, ist eine interessante Alternatividee. Ähnliche Konzepte werden bereits erfolgreich beim Carsharing oder bei Mitfahrzentralen verfolgt, allerdings mit einem variierenden Grad der Zentralisierung.

Eine Umsetzung dieser Alternatividee bietet der Transport physischer Objekte mit Hilfe von ad hoc Netzwerken. Durch das Aufkommen von allgegenwärtigen Geräten, die eine günstige Möglichkeit zur Kurzstreckenkommunikation bieten (Smartphones mit Bluetooth oder WLAN), ist es möglich, dass Menschen durch ihre lokale physische Präsenz ad hoc Netzwerke bilden, in denen Daten, oder aber auch physische Objekte, ausgetauscht werden können.

Ein solches Netz bietet vielseitige Anwendungsmöglichkeiten. Es kann beispielsweise den Transport von Gütern in Gegenden mit geringer oder temporär ausgefallener Infrastruktur realisieren oder zur Paketbeförderung in Organisationen eingesetzt werden, die die Möglichkeit einer „Betriebspost“ nutzen möchten. Möglicherweise gäbe es auch Anwendungen mit Anbindung an klassische zentralisierte Transportsysteme.

Während über Transportprotokolle und Methoden für das Versenden digitaler Informationen in mobilen ad hoc Netzwerken (MANET) bereits einige Forschungsarbeiten publiziert worden sind, ist die Verwendung solcher Netze für den Transport physischer Objekte ein bisher weniger beachtetes Feld.

Durch die Unterschiede zwischen digitalen und physischen Objekten, die beim Transport eine Rolle spielen, stellen sich interessante Forschungsfragen, die bislang noch nicht ausführlich beantwortet wurden. Beispielsweise zu Privatheits- und Sicherheitsaspekten, der potentiellen Nutzergruppe eines solchen Transportsystems und den Anreizmechanismen zu seiner Nutzung. Zusätzlich stellt sich die Frage nach der Performance des Transports, besonders bei Berücksichtigung der fehlenden Möglichkeit physische Pakete, wie digitale, bei der Übertragung einfach zu kopieren.

Die vorliegende Arbeit beschäftigt sich vor allem mit der Performance von ad hoc Netzwerken beim Transport physischer Objekte. Zu diesem Zweck wird ein in einer anderen Arbeit vorgestelltes Protokoll zum Routing physischer Pakete in MANETs verwendet und versucht, dessen Performance in unterschiedlichen Szenarien auszuwerten. Dieses Protokoll wurde bereits bei seiner Vorstellung durch eine Simulation auf seine grundsätzliche Anwendbarkeit geprüft. Auch in dieser Arbeit wird ein Simulationsansatz gewählt, welcher aber unter Einbeziehung von Straßennetzen und zwei verschiedenen Bewegungsalgorithmen versucht, eine realistischere Vorhersage zur Leistung des Protokolls in einem praktischen Szenario zu treffen.

Die Leistung des Protokolls wird anhand verschiedener Erfolgskriterien wie Lieferzeit und zurückgelegter Distanz der Pakete quantifiziert und kann somit in Relation zur Leistung klassischer zentralisierter Transportsysteme gesetzt werden. Dabei wird besonders auf die einzelnen Faktoren der Simulation und ihre Auswirkungen auf die Ergebnisse des Transportvorgangs eingegangen. Hauptziel der Arbeit ist es, durch diese Analyse eine Vorbereitung für etwaige Feldtests zu bieten, mögliche Schwächen des Protokolls aufzudecken und daraus Verbesserungsmöglichkeiten zu erschließen sowie Szenarien aufzuzeigen, in denen das Protokoll besonders effizient eingesetzt werden könnte.

Nach dieser Einleitung wird in Abschnitt 2 auf verwandte Arbeiten eingegangen, wobei sowohl allgemeinere Arbeiten zu MANETs und DTNs (Delay- and Disruption-Tolerant Networks) vorgestellt werden als auch konkrete Simulationen dieser Netze, besonders in automobilen ad hoc Netzwerken, behandelt werden. Außerdem beschrieben werden die Arbeiten zu Projekt POST, dessen Transportansatz Gegenstand dieser Arbeit ist. In Abschnitt 3 wird das Konzept und der Aufbau der erstellten Simulationsanwendung erläutert, während sich in Abschnitt 4 mit der konkreten Implementation beschäftigt wird. Die Ergebnisse der Simulationsläufe werden anschließend in Abschnitt 5 vorgestellt. Dazu werden die genauen Simulationsszenarien und -parameter genannt und erläutert, außerdem die Simulationsergebnisse präsentiert und analysiert. Den Schluss bildet Abschnitt 6 mit einem Fazit der Ergebnisse sowie einem Ausblick auf potentielle zukünftige Forschungsarbeiten.

## 2 Grundlagen

Zur Leistungsanalyse von MANETs beim Transport physischer Objekte auf Straßennetzen liegen dem Autor zum aktuellen Zeitpunkt keine bereits veröffentlichten Arbeiten vor. Im folgenden Kapitel werden aber verwandte Forschungsarbeiten vorgestellt, auf denen diese Arbeit aufbaut. Im ersten Unterabschnitt wird allgemeine Literatur zu DTNs und MANETs präsentiert, im zweiten geht es um Simulationen und Bewegungsalgorithmen und im dritten Unterabschnitt um Projekt POST, dessen Routing-Protokoll in dieser Arbeit analysiert wird.

### 2.1 DTNs und MANETs

Forschungen zu Delay- and Disruption-Tolerant Networks (DTN) sind der Ursprungspunkt aus dem Untersuchungen zu Mobile ad hoc Networks (MANET) hervorgegangen sind. Einen Überblick über die Forschung zu DTNs im Laufe der Zeit sowie über aktuellere Entwicklungen bietet [1]. Die Idee ein Kommunikationsnetzwerk aufzubauen, das resistenter gegenüber Unterbrechungen der Verbindung als das Internet wäre entstand in den 1990er Jahren im Rahmen der Weltraumforschung. Aufgrund von, im Gegensatz zur Kommunikation auf der Erde, erschwerten Bedingungen im Weltraum existierte eine direkte Verbindung zwischen zwei bestimmten Kommunikationsknoten oft gar nicht oder nur für kurze Zeit. Dadurch wurde ein Routing-Ansatz, wie er im Internet angewendet wird, unpraktikabel. Aus diesen Erfahrungen wurde eine Spezifikation für ein Netzwerk entwickelt, welches auch für die interplanetare Kommunikation funktionieren würde. Es wurde festgestellt, dass es auch auf der Erde Anwendungsmöglichkeiten für solche Netze, zum Beispiel in infrastrukturschwachen Gegenden, geben könnte [1]. Mit dem Fortschreiten der Entwicklung von immer kleiner werdenden Computern mit der Möglichkeit zur Kurzstreckenkommunikation begannen sich neue Forschungsgruppen zu bilden.

Diese Forschungsgruppen arbeiteten an den ersten MANETs. Als theoretische Grundlage diente dabei zum Beispiel die Arbeit [2], welche den Gedanken von Co-Presence-Netzwerken als verteiltes Netz von Kontaktereignissen beschreibt. In [3] wird ein ad hoc Netzwerk präsentiert, in welchem Daten opportunistisch über die Bewegung der Nutzer weitergeleitet werden, sowie ein Vergleich zwischen der Leistung verschiedener Routing-Protokolle anhand von Kontaktdaten, die mit PDAs gesammelt wurden.

Eine Analyse zu effizienten Forwarding-Entscheidungen für digitale Pakete in MANETs findet sich in [4]. Während [5] die Performance verschiedener opportunistischer Routing-Protokolle anhand von mehreren Datensets vergleicht. Beide Arbeiten verwenden in Feldversuchen gesammelte Kontaktereignisdaten von Testpersonen als Grundlage ihrer Untersuchungen. Diese sogenannten Contact Traces sind neben Simulationen ein weiteres beliebtes Mittel zur Analyse von MANET-Protokollen.

Konkret implementierte Anwendungsbeispiele für MANETs finden sich beispielsweise in [6], in welchem die regelmäßige Verwendung von öffentlichen Verkehrsmitteln ausgenutzt wird, um ein Netzwerk zum Teilen von Medien aufzubauen und in [7], in welchem gezeigt wird, wie MANETs in Katastrophenrettungsszenarien eingesetzt werden können, um die Kommunikation der Hilfskräfte auch bei Infrastrukturausfällen zu gewährleisten. Ein weiteres Beispiel bietet [8], in dem ein dezentraler und zensurresistenter Microbloggingdienst präsentiert wird, der sich über ein mobiles ad hoc Netzwerk propagiert.

## 2.2 Simulationen

Bewegungsalgorithmen sind für Simulationen außerordentlich wichtig. Die Bewegung der Agenten in einer Simulation ist meist der entscheidende Faktor für ihre Realitätsnähe. Einen guten Überblick über verschiedene Standardalgorithmen, die bei der Simulation von ad hoc Netzwerken verwendet werden bietet [9]. Die gängigsten Algorithmen und ihre Funktionsweise sowie ihre Vor- und Nachteile werden darin vorgestellt. Eine besonders wichtige Stellung nimmt dabei der Random Waypoint Algorithmus ein. Dieser wird aufgrund seines simplen Konzepts und der rechenzeitgünstigen Implementierbarkeit in vielen Arbeiten verwendet. Unter anderem auch in der ersten POST-Simulation [10] sowie in der vorliegenden Arbeit (siehe Abschnitt 3.3.1).

Aufgrund der schwerwiegenden Bedeutung von Bewegungsalgorithmen für Simulationen gibt es viele Arbeiten, die sich mit der Entwicklung neuer Bewegungsalgorithmen, die menschliche Bewegungsmuster möglichst adäquat abbilden sollen, beschäftigen. [11] analysiert verschiedene Sätze von Contact Traces, um die Gemeinsamkeiten zu einem Modell zusammenzufassen, und entwickelt aus diesen Erkenntnissen einen Algorithmus. Ein weiterer Bewegungsalgorithmus namens SMOOTH wird in [12] beschrieben. SMOOTH versucht einen Kompromiss zwischen dem Realismus der Ergebnisse und der Simplizität der Nutzbarkeit bei der Konstruktion eines Bewegungsalgorithmus für ad hoc Netzwerke herzustellen. Dies wird hauptsächlich durch die Anwendung von Clustern als Eingabeparameter erreicht. In der vorliegenden Arbeit wird als zweites Bewegungsmodell ein vereinfachter, von SMOOTH abgeleiteter Algorithmus in der Simulation verwendet (siehe Abschnitt 3.3.2).

Aufgrund des hohen Aufwands von Feldversuchen bieten sich Simulationen oder die Nutzung von Contact Traces aus bereits durchgeführten Feldversuchen, bei der Evaluation von DTNs und MANETs an. Besonders bei einer Sonderform der MANETs, den VANETs (Vehicular ad hoc Networks), die Fahrzeugen auf der Straße ermöglichen sollen miteinander zu kommunizieren, werden häufig Simulationen verwendet, da das Durchführen von Feldversuchen in diesem Bereich sehr kostenintensiv ist. Weil VANETs in ihren Bewegungen an ein Straßennetz gebunden sind, bietet die, im Folgenden genannte, Literatur über sie einen für die vorliegende Arbeit interessanten Forschungsbereich.

SUMO [13] ist ein Mikro-Simulator zur Fahrzeugsimulation, welcher jedes Fahrzeug einzeln simuliert und in vielen Forschungsarbeiten bei der Simulation von VANETs zum Einsatz kommt. Auch für VANETs sind Bewegungsalgorithmen sehr wichtig und nehmen daher in der Forschung über sie einen bedeutenden Platz ein. Das Ziel ist auch hier oft eine bessere Alternative zum Random Waypoint Algorithmus zu finden. [14] sowie [15] und [16] stellen Bewegungsalgorithmen für Fahrzeuge vor, die zu einem realistischerem Ergebnis als das Random Waypoint Modell führen sollen.

## 2.3 Projekt POST

In [10] wird die Idee eines Netzwerkes aus Personen beschrieben, die sich bereit erklären, die physischen Pakete anderer Mitglieder des Netzwerkes zu transportieren. Dabei wird zunächst von einer altruistischen Handlungsmotivation der Teilnehmer ausgegangen und das eventuelle Entwickeln von Konzepten für andere Motivationen auf weiterführende Arbeiten verlegt. Mit der Hilfe von Smartphones soll ein opportunistisches Routing über die Mitglieder des Netzwerkes Pakete an ihr Ziel bringen. Dieses Konzept wird als POST (Physical Object Sneaker Transport) bezeichnet und es werden mögliche Einsatzgebiete für ein solches POST-System genannt. Ein Verhandlungsprotokoll sowie ein Routing-Protokoll werden vorgestellt, welche die Pakete so zwischen den Mitgliedern des Netzwerkes austauschen sollen, dass alle Pakete ihr Ziel erreichen.

Das verwendete Routing-Protokoll wurde in einer ersten Simulation getestet, welche auf einer quadratischen Fläche von  $100 \text{ km}^2$  und mit dem Random Waypoint Algorithmus arbeitete. In mehreren Simulationsläufen wurde mit unterschiedlichen Kommunikationsreichweiten die Machbarkeit eines solchen Systems geprüft. Die Ergebnisse waren vielversprechend, aber es wurde darauf hingewiesen, dass weitere Tests nötig wären. Besonders um die Alltagsperformance eines solchen Netzes zu zeigen und die kritische Masse an Nutzern festzustellen, die für einen reibungslosen Pakettransport benötigt werden.

Als Fortsetzung wurde in [17] in einer studentischen Projektgruppe ein Prototyp entwickelt, in welchem das Verhandlungsprotokoll nochmals erweitert und mit der entwickelten App erste Feldtests durchgeführt wurden. Damit wurde eine Grundlage zur weiteren Entwicklung der App zu Testzwecken geschaffen.



## 3 Simulationsdesign

In diesem Abschnitt werden die Designentscheidungen behandelt, die getroffen wurden, um eine Simulation von POST auf Straßennetzen durchzuführen.

### 3.1 Transportkonzept

Die Simulation soll die Bewegungen mobiler Agenten beim Transport physischer Objekte auf einer Straßenkarte darstellen. Dafür muss zunächst festgelegt werden, in welcher Form sich die Agenten beim Transport verhalten. Es wird davon ausgegangen, dass die Agenten ein Smartphone oder ähnliches Gerät besitzen, welches über die Fähigkeit zur Kurzstreckenkommunikation verfügt, zum Beispiel über WLAN oder Bluetooth. Auf diesem Smartphone befindet sich eine App, die den POST-Transport regelt.

Pakete sind zu Beginn über das gesamte Simulationsgebiet verteilt und werden von App-Nutzern aufgenommen, wenn sie in die Nähe des Pakets kommen. Ebenso werden die Pakete an ihren Zielort gebracht, wenn die Nutzer in Reichweite sind. Die Reichweite, die für Aufnehmen und Abliefern des Paketes maßgebend ist, ist hierbei die Kommunikationsreichweite der Geräte. Befinden sich zwei Nutzer in dieser Kommunikationsreichweite sind sie in der Lage eine Verbindung aufzubauen, um eine Routing-Verhandlung über ein Paket durchzuführen.

Eine solche Routing-Verhandlung wird in zwei Schritten ausgeführt. Es wird dabei davon ausgegangen, dass der App das nächste Ziel des Nutzers (zum Beispiel der Supermarkt) bekannt ist. Der momentane Träger des Pakets (Nutzer A) erhält von seinem Verhandlungspartner (Nutzer B) das Ziel, zu dem sich Nutzer B bewegen wird. Nutzer A gleicht dies mit seinem Zielort und dem des Pakets ab. Dabei betrachtet er eine Luftlinie der Strecke zu seinem Ziel und eine Luftlinie der Strecke zum Zielort des Pakets und errechnet den Winkel zwischen den beiden Strecken. Gleichfalls geht er mit der Strecke zum Ziel von Nutzer B vor. Wenn der Winkel von Nutzer B zum Paket kleiner ist als der Winkel von Nutzer A ist die Verhandlung erfolgreich und das Paket wird übergeben. Diese Variante eines Greedy-Routing-Protokolls versucht so die Pakete an ihren Zielort zu bringen. (siehe Abbildung 1)

Die Performance eines solchen Systems im Vergleich zu einem traditionellen zentralisierten System, wie bei Postdienstleistern üblich, hängt von vielen Faktoren ab. Neben der Anzahl und Geschwindigkeit der Nutzer, spielt auch die Beschaffenheit der Infrastruktur eine Rolle. Die Kommunikationsreichweite der benutzten Geräte und natürlich die Bewegungsmuster der Nutzer sind ebenfalls von Bedeutung.

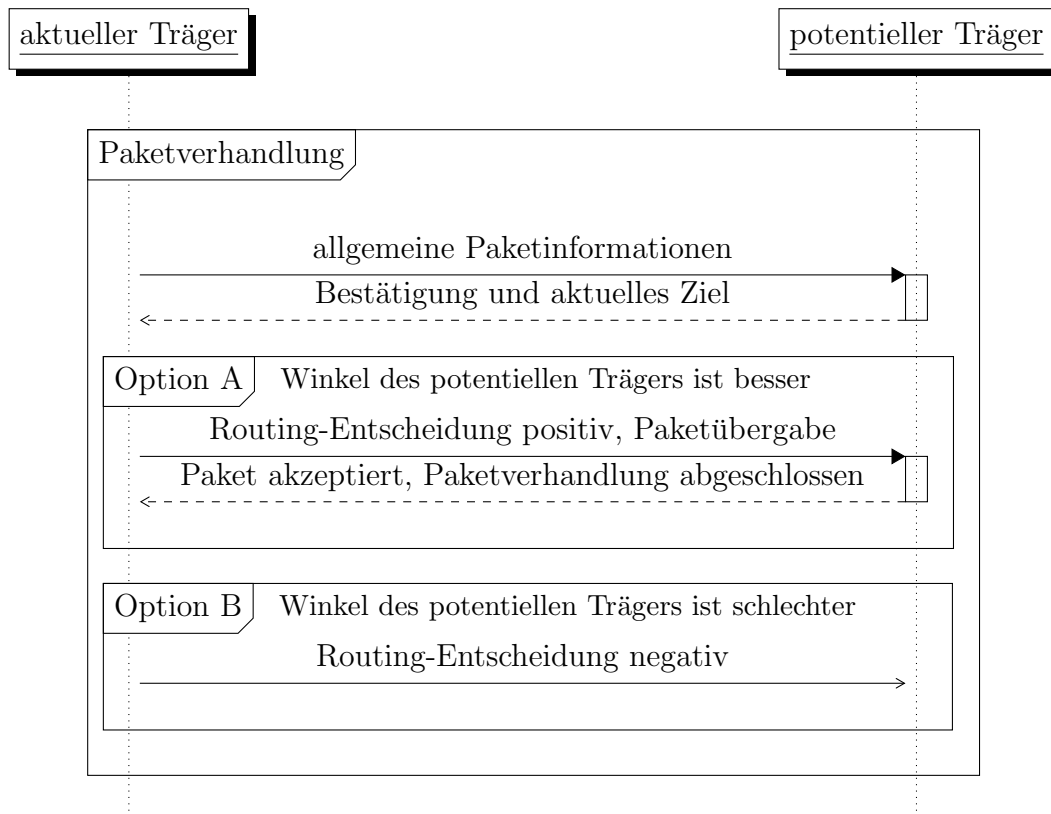


Abbildung 1: Greedy Routing Protokoll

## 3.2 Simulation oder reale Kontakttereignisse

Zur Evaluation von Routing-Protokollen in MANETs bieten sich verschiedene Möglichkeiten an. Die erste ist das Simulieren des Verhaltens von Nutzern und die Beobachtung, wie erfolgreich sich das Routing-Protokoll dabei verhält. Alternativ können gesammelte Datensätze von Contact Traces verwendet werden, um darauf die Effizienz des Routing-Protokolls zu testen, oder es kann ein Prototyp entwickelt werden, mit dem man einen tatsächlichen Testlauf durchführt. Letzteres ist außerhalb des Rahmens dieser Arbeit, es bleibt daher zu klären, warum ein Simulationsansatz im Gegensatz zu einem Ansatz mit Contact Traces verwendet wurde.

### 3.2.1 Contact Traces

Simulationen kranken an einem großen Problem: Das Vorhersagen realitätsnaher menschlicher Bewegungen mit Hilfe einfacher Algorithmen ist sehr schwierig [9]. Contact Traces umgehen dieses Problem, indem sie die Kontakte einer Testgruppe von Menschen bei ihren alltäglichen Bewegungen mit Hilfe von Geräten mit der Fähigkeit zur Kurzstrecke-

ckenkommunikation aufzeichnen und diesen Kontaktdatensatz speichern.

Die so gewonnenen Daten werden daraufhin verwendet, um zu analysieren, wie sich das eigene Protokoll in diesem Datensatz verhalten hätte und wie weit auf diesen konkreten Bewegungen von Menschen die Daten oder Objekte transportiert worden wären. Ein Problem stellt bei Contact Traces dar, dass sie stets nur ein diskretes Szenario darstellen, in denen kleine Anomalien oft große Folgen haben können. Dass zum Beispiel ein wichtiger Hop für Pakete seinen Zug verpasst, kann über mehrere Simulationsiterationen ausgeglichen werden, bei einem Datensatz bleiben die Datenpunkte aber immer gleich. Sie sind also nicht variabel und können natürlich auch wenn sie realistische Bewegungen darstellen nur eine Momentaufnahme bieten. Hinzu kommt, dass beim Sammeln von Contact Traces leicht eine Stichprobenverzerrung durch die Auswahl der Personen, die sich bereit erklären, ihre Bewegungen aufzeichnen zu lassen entstehen kann. Im schlechtesten Fall sind die Daten damit ebenso wenig realitätsnah, wie eine Simulation mit künstlichem Bewegungsmodell.

### **3.2.2 Simulation**

Ein Vorteil von Simulationen gegenüber Contact Traces ist, dass diese Traces immer nur ein konkretes Szenario beinhalten. Im Gegensatz dazu sind Simulationen in ihren Parametern einfach variierbar. Die Anzahl der Nutzer oder das Straßennetz der Simulation können beliebig geändert werden, während sie bei Traces feststehen und bei einer Änderung dieser Daten neue Traces gesammelt werden müssen. Findet man kein passendes Traceset, muss man ein eigenes erstellen und das Sammeln von einer ausreichenden Anzahl an Contact Traces ist meist ein wesentlich zeitaufwendigeres Unterfangen als diese durch einen Bewegungsalgorithmus in der Simulation zu substituieren. Hinzu kommt, dass bei einer Simulation mehrfache Iterationen möglich sind, die zufällige Ereignisse dämpfen, die im Traceset große Auswirkungen haben könnten.

Simulationen sind also variabel, während Traces dafür eine Garantie zumindest eines konkreten, realistischen Szenarios geben können. Es wurde sich in der vorliegenden Arbeit letztlich aufgrund der Flexibilität für eine Simulation mit künstlichem Bewegungsmodell entschieden, um zum Beispiel verschiedene selbst gewählte Straßennetze auf ihre Tauglichkeit mit POST testen zu können. Dies wäre ohne künstliches Bewegungsmodell schwer möglich, da die Anzahl an geeigneten und verfügbaren Contact Traces, vor allem aus dem deutschen Raum, sehr gering ist und eine eigene Sammlung von Contact Traces, wie zu Anfang des Kapitels erwähnt, nicht im Rahmen dieser Arbeit wäre.

## 3.3 Algorithmen

Die in dieser Arbeit vorgestellte Simulation nutzt zwei verschiedene Bewegungsalgorithmen, die im folgenden Abschnitt näher erläutert werden.

### 3.3.1 Random Waypoint Algorithmus

Der Random Waypoint Algorithmus ist ein Algorithmus, der in verschiedenen Variationen und Konfigurationen in vielen Forschungsarbeiten eingesetzt wird. Die Grundidee dieses Bewegungsmodells ist, dass alle Agenten einen zufälligen Wegpunkt und eine zufällige Geschwindigkeit erhalten und beim Erreichen ihres Wegpunkts für eine zufällige Zeitspanne stoppen. Danach ändern sie wieder ihre Geschwindigkeit auf einen neuen zufälligen Wert und erhalten einen neuen Wegpunkt.

Sowohl Wegpunkte, Geschwindigkeit als auch das Stoppintervall sind normalerweise in einem gewissen Bereich eingegrenzt, beispielsweise ein Feld von 600x600 Metern mit 0-10 m/s Geschwindigkeit und 60-360 Sekunden Stoppintervall. Abhängig von den gewählten Parametern entstehen unterschiedliche Bewegungsmuster.

Für diese Arbeit wurde eine Geschwindigkeit von 1,4 m/s und ein Stoppintervall von 0 Sekunden festgelegt. Die Wegpunkte sind eingegrenzt durch die Straßenkarte auf der die Simulation ausgeführt wird. Mit diesen Einstellungen wird eine konstante Bewegung mit 1,4 m/s gewährleistet, was ungefähr der Geschwindigkeit eines Menschen beim Gehen entspricht. Diese Werte wurden gewählt, um das erste Bewegungsmodell der Simulation ähnlich zum Bewegungsmodell im ersten Paper zu POST [10] zu gestalten, mit Ausnahme natürlich der Einschränkung der Wegpunkte durch eine Karte.

### 3.3.2 Cluster Waypoint Algorithmus

Der Cluster Waypoint Algorithmus ist eine Abwandlung des SMOOTH Algorithmus, der in Kapitel 2.2 erwähnt wird. Im Gegensatz zum Random Waypoint Algorithmus werden die nächsten Wegpunkte der Agenten nicht zufällig aus einer Liste aller begehbarer Knoten ausgewählt, sondern aus einer Anzahl an unterschiedlich gewichteten Clustern. Abhängig von der Dichte des Straßennetzes wird eine Anzahl an Clustern mit unterschiedlicher Gewichtung generiert, die Fixpunkte menschlicher Bewegung darstellen sollen, beispielsweise eine Bank, Rathaus oder eine Privatwohnung. Je nachdem wie stark der Cluster gewichtet ist, desto wahrscheinlicher ist, dass der Agent ihn als nächsten Wegpunkt auswählt, so dass ein Einfamilienhaus im Durchschnitt von weniger Menschen besucht wird als der Supermarkt. Zusätzlich dazu besitzt jeder Agent eine Liste an bekannten Clustern, die er bereits besucht hat. Bei jeder Entscheidung für einen neuen Wegpunkt wählt der Agent abhängig davon, wie viele Cluster er kennt, ob er einen

neuen Cluster besucht oder einen bereits bekannten. Dort verbringt er dann eine gewisse Zeit, bevor der Selektionsprozess für den nächsten Wegpunkt von Neuem startet.

Für die vorliegende Arbeit wurden bei jeder Simulation eine Anzahl von Clustern abhängig von der Anzahl von Knoten im gewählten Straßennetz erzeugt (Knotenanzahl/50) und zufällig auf der Karte verteilt. Diese Knoten werden in normale (60 Prozent), mittelgroße (25 Prozent) und große Cluster (15 Prozent) eingeteilt. Bei der Auswahl der Weg- und Startpunkte ist es gegenüber einem kleinen Cluster zweimal so wahrscheinlich, einen mittelgroßen beziehungsweise viermal so wahrscheinlich, einen großen Cluster auszuwählen. Für eine realistischere Simulation auf einer Karte wäre es natürlich notwendig den untersuchten Kartenausschnitt zu prüfen und die realen Cluster, die Menschen besuchen zu erforschen und entsprechend in der Simulation einzutragen und zu gewichten. Dies wäre allerdings mit einem beträchtlichem Aufwand verbunden und ist außerhalb des Rahmens dieser Arbeit. In der vorliegenden Simulation soll es lediglich möglich sein direkt mit importierten Karten Simulationen durchzuführen, daher werden die Cluster zufällig platziert und gewichtet. In Abschnitt 6.2 wird allerdings auf mögliche Verbesserungen dieses Systems anhand vorhandener Kartendaten eingegangen.

Die Geschwindigkeit der Nutzer beträgt 1,4 m/s und das Stoppintervall beträgt zu 50 Prozent zwischen 5 und 20 Minuten, zu 30 Prozent zwischen 30 Minuten und 2 Stunden und zu 20 Prozent zwischen 4 und 8 Stunden. Diese Stoppintervalle sollen die Zeit simulieren, die Menschen in Clustern verbringen. Wobei Menschen tendenziell in wenigen Clustern sehr viel Zeit verbringen (Wohnung, Arbeitsstelle), in etwas mehr eine mittlere Zeitspanne (Fußballtraining, Fitnessstudio, Restaurant) und in vielen nur kurz vorbeischauen (Bäckerei, Drogerie, Bankautomat, Café). Bei den gewählten Zeitintervallen bleibt ein Nutzer im Durchschnitt 100 Minuten in einem Cluster und besucht somit 12-14 Cluster pro Tag, je nachdem wie viel Zeit er auf dem Weg zu ihnen verbringt.

Die Simulationen mit dem Cluster Waypoint Algorithmus sollen eine ungefähre Vorhersage für die Alltagsperformance des POST-Protokolls und Hinweise auf eine mögliche kritische Masse an Nutzern, eventuelle Änderungen im Protokoll oder Testläufe mit Traces geben.

### 3.4 Straßennetz

Im Alltag bewegen sich Menschen auch zu Fuß meist anhand von Straßennetzen, da diese in Ballungsgebieten die ausgewiesene (und nicht bebaute) Möglichkeit zur Bewegung darstellen. Dieser Umstand ist auch für den Transport physischer Objekte mit MANETs zu beachten. In einer ersten Simulation im ersten Paper zum Thema POST [10] wurde eine leere Fläche verwendet, auf der sich die Nutzer bewegten. Neben einem realistischeren Bewegungsalgorithmus ist ein Straßennetz ein weiterer Schritt in Richtung Wirklichkeitsnähe beim Evaluieren des POST-Protokolls.

Um eine Simulation auf einem Straßennetz auszuführen, werden Rohdaten eines solchen benötigt. Dafür wurde in dieser Arbeit OpenStreetMap verwendet. „OSM ist ein im Jahre 2004 gegründetes internationales Open-Source-Projekt mit dem Ziel, eine freie Weltkarte zu erschaffen“<sup>1</sup>. Die Daten des OpenStreetMap-Projekts sind in verschiedenen Formen exportierbar und aufgrund mehrerer anderer Projekte gibt es Bibliotheken für verschiedene Programmiersprachen, unter anderem Java, die beim Importieren dieser Daten in eine Anwendung unterstützen.

Durch die Möglichkeit, jeden Kartenausschnitt aus OSM zu importieren, besitzt die Simulation eine hohe Flexibilität. Es kann sowohl das Verhalten von Nutzern in einem relativen kleinen Raum, wie zum Beispiel einem Werksgelände, welches ein MANET zum opportunistischen Transport bestimmter physischer Objekte in ihrer Firma nutzen möchte simuliert werden, als auch das Verschicken von Paketen in einer größeren Stadt. Dabei ist neben der Fläche auch die Knoten- und Straßendichte, die ungefähr mit der Besiedlung korreliert, ein wichtiger Faktor für die Effizienz des POST-Protokolls.

Da die in dieser Arbeit genutzte Simulation in Java geschrieben ist, wird die Bibliothek *osmosis*<sup>2</sup> für Java verwendet, um .osm-Dateien (XML-Format) für das Simulationsprogramm einzulesen.

### 3.4.1 Wegfindung

Zur Wegfindung auf der Karte wird eine Implementation des A\*-Algorithmus [18] verwendet. Der A\*-Algorithmus nutzt eine Heuristik um den optimalen Weg zwischen zwei Knoten zu finden. Dabei untersucht er meist nur einen kleinen Teil der Knoten des Graphen und ist deshalb sehr performant. Der Algorithmus teilt die Knoten des Kartengraphen in unbekannte, bekannte und untersuchte Knoten ein.

Zu Beginn ist lediglich der Startknoten bekannt, alle anderen Knoten sind unbekannt. Jeder bekannte Knoten besitzt einen f-Wert, dieser besteht aus dem kürzesten bekannten Weg vom Startknoten zu diesem Knoten und der Entfernung der Luftlinie zum Zielknoten. Um nun den optimalen Weg zum Zielknoten zu finden wird stets der Knoten mit dem niedrigsten f-Wert aus der Liste der bekannten Knoten ausgewählt und in die Liste der untersuchten Knoten verschoben. Dabei wird der f-Wert für die Nachbarn des ausgewählten Knotens errechnet und diese werden in die Liste der bekannten Knoten verschoben. Befindet sich ein Nachbar bereits in der Liste der bekannten Knoten wird sein f-Wert gegebenenfalls aktualisiert, sollte der Weg über den ausgewählten Knoten kürzer sein als der bisherige Weg zu diesem Knoten. Der Algorithmus endet, wenn der Zielknoten in die Liste der untersuchten Knoten verschoben wird.

---

<sup>1</sup><http://www.openstreetmap.de/>

<sup>2</sup><http://wiki.openstreetmap.org/wiki/Osmosis>

### 3.5 Simulationszyklus

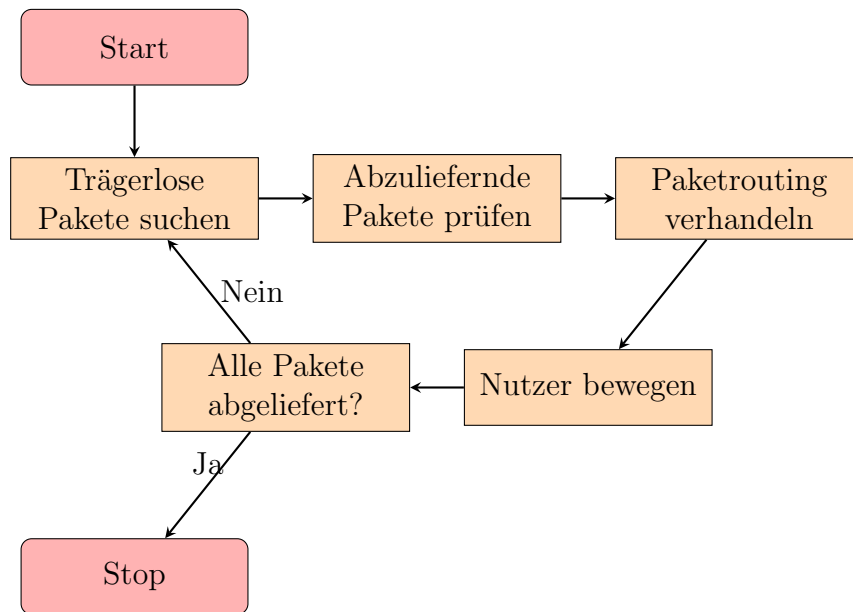


Abbildung 2: Simulationszyklus

In Abbildung 2 wird die Reihenfolge der Aktionen innerhalb einer Simulationsschleife dargestellt. In einem Zyklus, also der kleinsten simulierten Zeiteinheit, werden vier Schritte ausgeführt und eine Abbruchbedingung untersucht.

Im ersten Schritt sollen alle Nutzer nach trägerlosen Paketen in ihrer Kommunikationsreichweite suchen. Wenn sich ein Paket in ihrer Reichweite befindet, sollen sie dessen Position als nächstes Ziel deklarieren bzw. es aufnehmen, wenn sie bereits auf der Position des Pakets sind.

Im zweiten Schritt sollen alle Nutzer die Zielposition ihrer getragenen Pakete untersuchen und prüfen, ob sie in Kommunikationsreichweite der Zielposition eines ihrer Pakete sind. Wenn sie eine Zielposition in ihrer Reichweite finden, sollen sie dessen Position als nächstes Ziel deklarieren bzw. das Paket abliefern, wenn sie sich bereits auf der Zielposition des Pakets befinden.

Im dritten Schritt prüfen alle Nutzer, ob sich andere Nutzer in ihrer Kommunikationsreichweite befinden. Falls ja, werden sie eine Verhandlung durchführen, ob Pakete ausgetauscht werden sollen. Sollte diese Verhandlung positiv enden, werden die beiden Nutzer auf ihrer Position für eine kurze Zeit eingefroren, um die physische Paketübergabe zu simulieren.

Im vierten Schritt werden alle Nutzer bewegt. Sollte ein Nutzer dabei an seinem aktuellen Ziel ankommen, wird ihm anhand des ausgewählten Bewegungsalgorithmus ein neues Ziel vorgegeben.

## 4 Implementation

Das Implementationskapitel besteht aus einer Beschreibung der Simulationsanwendung und den Herausforderungen beim Programmieren dieser Anwendung. Das Programm selbst beinhaltet die vier Pakete GUI, Objects, Simulation und IO.

Die Anwendung beginnt mit der Initialisierung, bei der zuerst die notwendigen Parameter, wie Anzahl der Nutzer und Pakete sowie die zur Simulation genutzte Karte angegeben werden. Diese Karte wird daraufhin eingelesen und die Nutzer und Pakete auf ihr platziert. Danach wird die eigentliche Simulation durchgeführt und zum Schluss wird eine Heatmap erstellt sowie einige abschließende Gesamtstatistiken gespeichert.

### 4.1 GUI

Das GUI-Paket enthält die Klassen, die zum Anzeigen einer graphischen Benutzeroberfläche verwendet werden und somit der Darstellung von Simulationskarten und Heatmaps, aber auch der Eingabe von Parametern dienen.

#### 4.1.1 InitialGUI

Die InitialGUI ist der Startpunkt des Programms, in dem die Simulationsparameter festgelegt werden. Sie öffnet einen Dialog, bei dem eine .osm-Datei ausgewählt werden kann, aus der das Straßennetz für die Simulation extrahiert werden soll. Außerdem werden in ihr die Startwerte für Anzahl der Nutzer, Pakete und die Kommunikationsreichweite sowie der Bewegungsalgorithmus angegeben. Wenn die Simulation initialisiert ist, öffnet die InitialGUI eine MapGUI-Instanz, die die gewählte Karte anzeigt.

#### 4.1.2 MapGUI

Die MapGUI-Klasse erbt von der Java-Klasse JApplet und stellt eine Karte dar, auf der die Nutzer sich bewegen. Das MapGUI-Applet iteriert bei jeder paint()-Operation durch die Liste aller Wege, Benutzer und Pakete und zeigt diese an. Als graue Linien werden Straßen dargestellt, rote, ausgefüllte Kreise symbolisieren die Nutzer und Pakete erscheinen als schwarze, ausgefüllte Quadrate. Im Cluster-Bewegungsalgorithmus werden zusätzlich die Cluster durch größere grüne, nicht ausgefüllte Kreise markiert.

Zum Abschluss der Simulation zeigt das Applet eine Heatmap an, bei der die Breite und Farbe der Straßen davon abhängt, wie oft sie von Agenten in der Simulation genutzt



wurden. Wenig genutzte Straßen gehen von dünn und blau zu viel genutzten Straßen in rot und breit über.

## 4.2 Objects

Das objects-Paket kümmert sich um die Objekte, die in der Simulation verwendet werden, und enthält mit `Coordinate`, `SimNode`, `SimWay`, `SimMap`, `User` und `Paket` 6 Klassen.

### 4.2.1 Coordinate

Die Klasse `Coordinate` stellt eine Koordinate im zwei-dimensionalen Raum dar und enthält neben einem x und y-Wert, der Längen- und Breitengrad angibt lediglich eine Funktion zur Berechnung der Distanz zu einer anderen Koordinate auf der Erdkugel.

### 4.2.2 SimNode

`SimNode` stellt einen Knoten im verwendeten Straßennetz dar. Ein `SimNode` Objekt hat neben einer eindeutigen ID und einer Position (als `Coordinate`), eine `HashMap` mit den Nachbarn dieses Knotens, wobei ein `SimNode` Objekt den Schlüssel darstellt und die Distanz zu diesem Knoten den dazugehörigen Wert. Außerdem implementiert die Klasse `SimNode` das Interface `Comparable`, um eine natürliche Ordnung anhand einer Variable `f` herzustellen, die im A\*-Algorithmus gesetzt wird und in einer `TreeMap` ausgenutzt wird, um die passende Reihenfolge für den nächsten zu prüfenden Weg im Algorithmus zu erhalten. Ein Objekt speichert außerdem, wie oft der Knoten durch Nutzer besucht wurde; dieser Wert wird dann bei der Heatmap-Erstellung verwendet.

### 4.2.3 SimWay

`SimWays` sind Sammlungen von `SimNodes`, die eine Straße bilden. Sie werden zur Visualisierung im GUI und beim Übertragen der Karte aus den OpenStreetMap Daten verwendet. In der Wegfindung werden sie nicht genutzt, weil `SimNodes` ihre Nachbar-knoten speichern und die jeweiligen Nachbarn eines Knotens und ihre Distanz zu ihm für die Wegfindung mit dem A\*-Algorithmus ausreichend sind.

#### 4.2.4 SimMap

Die Klasse SimMap speichert die gesamte Karte. In ihr sind sämtliche SimNodes und SimWays enthalten sowie die Angaben über die Ränder der Karte. Die Klasse besitzt verschiedene Methoden; unter anderem befindet sich hier die Implementation der Wegfindung, welche mit Hilfe der Implementation eines A\*-Algorithmus den kürzesten Weg zwischen zwei Knoten im Netzwerk berechnet.

Im Konstruktor der Klasse werden ihr die mit Hilfe des XMLParsers aus den OSM-Daten extrahierten SimWays übergeben. Zur Gewährleistung der Integrität der Karte werden dann zunächst alle SimNodes aus den SimWays extrahiert, um nur SimNodes zu haben, die auch auf einer Straße liegen. Danach werden alle Duplikate in der SimNode-Liste entfernt, um doppelte Nodes zu vermeiden. Dann werden die SimWays genutzt, um allen SimNodes ihre passenden Nachbarn zuzuweisen, um nachher auf die SimWay-Objekte in der Wegfindung verzichten zu können. Sollten sich danach noch SimNodes finden, die keine Nachbarn haben, werden diese entfernt, da man sich nicht von ihnen herunterbewegen könnte. Zum Schluss wird die Karte auf den größten zusammenhängenden Graphen geprüft.

Wenn ein beliebiger Kartenausschnitt für die Simulation gewählt wird kann es passieren, dass sich zwei oder mehr unzusammenhängende Graphen im Kartenausschnitt befinden. Da kein Austausch zwischen diesen Graphen stattfinden kann und daher grundsätzlich einige Pakete unauslieferbar bleiben würden, gilt es den zusammenhängenden Graphen mit den meisten SimNodes zu finden und nur auf diesem Pakete und Nutzer zu platzieren sowie nur Knoten auf diesem Graphen als Ziele für diese auszugeben. Eine Möglichkeit für die Suche nach einem zusammenhängenden Graphen bietet der Dijkstra-Algorithmus; da der A\*-Algorithmus aber bereits implementiert war, wurde er ebenfalls hierfür verwendet. Es wird eine zufällige SimNode ausgewählt und von dieser der kürzeste Weg zu einer anderen zufälligen SimNode gewählt. Alle in diesem Weg vorhandenen Knoten werden dann entfernt und zum aktuellen Teilgraph hinzugefügt. Es wird weiter iteriert bis der größte Graph festgestellt ist.

#### 4.2.5 Packet

Die Packet-Klasse wird zur Darstellung eines Pakets in der Simulation verwendet. Pakete besitzen eine Position, ein Ziel (beides Coordinates) sowie einen Träger. Außerdem besitzen Pakete verschiedene für die End-Statistik relevante Attribute, wie zurückgelegter Transportweg, Anzahl der Handovers, Zeitpunkt der Auslieferung und ursprüngliche Startposition.

#### 4.2.6 User

Die Klasse User stellt einen in der Simulation aktiven Nutzer dar. Ein User-Objekt verfügt über eine Position (als Coordinate), eine Liste von Paketen, die es gerade transportiert sowie den Weg zum nächsten Wegpunkt. Ein solcher Weg besteht aus einer Anzahl von SimNodes, die der User nacheinander abarbeitet, indem er sich gerade darauf zu bewegt. Außerdem besitzen User die beiden für die Statistik relevanten Attribute zurückgelegte Strecke und zurückgelegte Strecke als Paketträger.

### 4.3 Simulation

Im Simulation-Paket befindet sich mit der Klasse MapSimulation die Hauptklasse der Anwendung, in der alle für sie relevanten globalen Werte gespeichert werden.

#### 4.3.1 MapSimulation

Als Attribute verfügt die Klasse über die Karte sowie eine Liste der User und Pakete, als auch Geschwindigkeit der User, Kommunikationsreichweite und Zeit, die pro Simulationszyklus vergeht. Eine Simulation beginnt im Konstruktor mit der Erstellung der gewünschten Zahl von Nutzern und Paketen. Je nach Bewegungsalgorithmus werden sie unterschiedlich auf der Karte platziert.

Der Random Waypoint Algorithmus verteilt die Nutzer und Pakete zufällig über die Knoten des größten zusammenhängenden Graphen im Kartenausschnitt (siehe SimMap in 4.2.4) und gibt einem Nutzer, der sein Ziel erreicht hat, ein zufälliges neues aus.

Der Cluster Waypoint Algorithmus verteilt seine Nutzer und Pakete beim Start in gewichteten Clustern. Diese Cluster stellen Interessenzentren der Nutzer dar, die diese öfters besuchen. In dieser Simulation existiert ein Cluster pro 50 Knoten, die im größten zusammenhängenden Graphen vorhanden sind. Die Gewichtung der Knoten wird ebenfalls bei der Platzierung der Nutzer und Pakete beachtet.

Die Simulationsmethode findet sich ebenfalls in dieser Klasse wieder. Sie geht als Schleife durch die Methoden für das Suchen nach trägerlosen Paketen, dann das Suchen nach ablieferbaren Paketen, danach die Suche nach Verhandlungspartnern für die Übergabe von Paketen und zuletzt durch die Methode zum Bewegen der Nutzer. Beim Suchen nach Verhandlungspartnern ist wichtig, dass nach dem erfolgreichen Abschließen des Tausches eine kleine Pause eingelegt wird, um zu simulieren, dass die Nutzer physisch das Paket austauschen.

## 4.4 IO

Das IO-Paket beherbergt die Logging-Klasse und die XMLParser-Klasse. Diese beiden Klassen beschäftigen sich mit Input und Output der Anwendung.

### 4.4.1 CSVLogging

Die Klasse CSVLogging wird verwendet um eine .csv-Datei zu erstellen, in welche mit dem write-Befehl der Klasse geschrieben werden kann. Geloggt werden während der Simulation sämtliche Paketaufnahmen von trägerlosen Paketen, alle Paketablieferungen sowie alle erfolgreichen Verhandlungen über Pakete zwischen zwei Nutzern. Es wird ein Timestamp und die beteiligten Nutzer und Pakete sowie ihre Position geloggt. Nach Ende der Simulation werden als finale Daten sämtliche SimNodes und wie oft sie besucht worden sind in die Datei geschrieben sowie die Liste der Nutzer und abgelieferten Pakete mit zurückgelegter Strecke, Lieferzeit und anderen Daten. Außerdem werden die Anzahl der gelieferten und nicht gelieferten Pakete, sowie die durchschnittliche Lieferzeit und durchschnittliche Anzahl an Handovers angegeben.

### 4.4.2 XMLParser

Im XMLParser wird die .osm-Datei, welche die Kartendaten für die Simulation enthält, mit Hilfe der osmosis Java-Bibliothek verarbeitet. Osmosis besitzt ein eigenes Klassenschema für OpenStreetMap-Daten. Die Daten der .osm-Datei werden ausgelesen und lediglich die als Highway markierten Wege, werden extrahiert. Grenzen, Strommasten oder Flüsse sind in OSM ebenfalls als Wege gespeichert, aber durch eine Beschränkung auf das Tag Highway kann vermieden werden, sie auch mit auszulesen. Außerdem werden alle Knoten aus den Kartenrohdaten gespeichert. Dann werden die relevanten Daten der Knoten (ID und Position) in SimNodes gepackt und aus allen Wegen werden SimWays erstellt, in denen die SimNodes gespeichert werden, die auf diesem Weg liegen. Die SimWays werden dann zusammen mit den Begrenzungskordinaten des Kartenausschnitts zur Erstellung der SimMap verwendet. Das nur die Wege übergeben werden und die Knoten, die in ihnen sind, und nicht die ganze Knotenliste verhindert, dass SimNodes in die Simulation geraten, die auf keiner Straße liegen und somit für diese Arbeit nutzlos sind.

## 5 Ergebnisse der Simulation

In diesem Kapitel werden die genauen Simulationsparameter angegeben sowie die Ergebnisse der durchgeführten Simulationsläufe dargestellt und analysiert.

### 5.1 Simulationsparameter

Für die in diesem Kapitel vorgestellten Ergebnisse wurden Simulationen in 16 unterschiedlichen Szenarien durchgeführt. Dabei wurde jede Kombination aus zwei Straßennetzen, den zwei Bewegungsalgorithmen und einer Anzahl von 10, 25, 50 und 100 Nutzern verwendet. In allen Simulationen wurden 250 auszuliefernde Pakete angesetzt. Jedes Szenario wurde 5 mal simuliert. Warum der Random Waypoint und Cluster Waypoint Algorithmus gewählt wurden und welche Parameter bei den beiden Algorithmen verwendet worden sind, ist bereits in Kapitel 3.3 beschrieben; im Folgenden werden die Straßennetze und alle weiteren Parameter genannt.

Die Geschwindigkeit der Nutzer beträgt in allen Simulationen 1,4 m/s, die Zeit, die in einem Simulationszyklus vergeht beträgt 1 Sekunde und die Kommunikationsreichweite der Geräte der Nutzer beläuft sich auf 10 m. Bluetooth kann zwar theoretisch eine größere oder eine kleinere Reichweite erreichen (besonders in Gebäuden), aber für die Anwendung im Außenbereich haben sich, nach Erfahrung des Autors, 10 Meter als eine konservative Schätzung für eine Arbeitsreichweite als realistisch erwiesen. Die Simulation bricht ab, wenn alle Pakete an ihren Zielort geliefert worden sind oder nach 604800 Zyklen bzw. Sekunden, also nach genau einer Woche Simulation. Die Stoppzeit, die die Nutzer anhalten, wenn sie ein Paket aufgrund einer erfolgreichen Routing-Verhandlung austauschen wollen, wurde auf 10 Sekunden pro Paket festgelegt. Dies ist vermutlich nicht genügend Zeit um in einer realen Anwendung Pakete auszutauschen, aber die genauen physischen Abläufe beim Austauschen von Paketen sind bisher in Projekt POST nicht festgelegt und auch nicht Gegenstand dieser Arbeit. Daher wurden 10 Sekunden als Platzhalter gewählt.

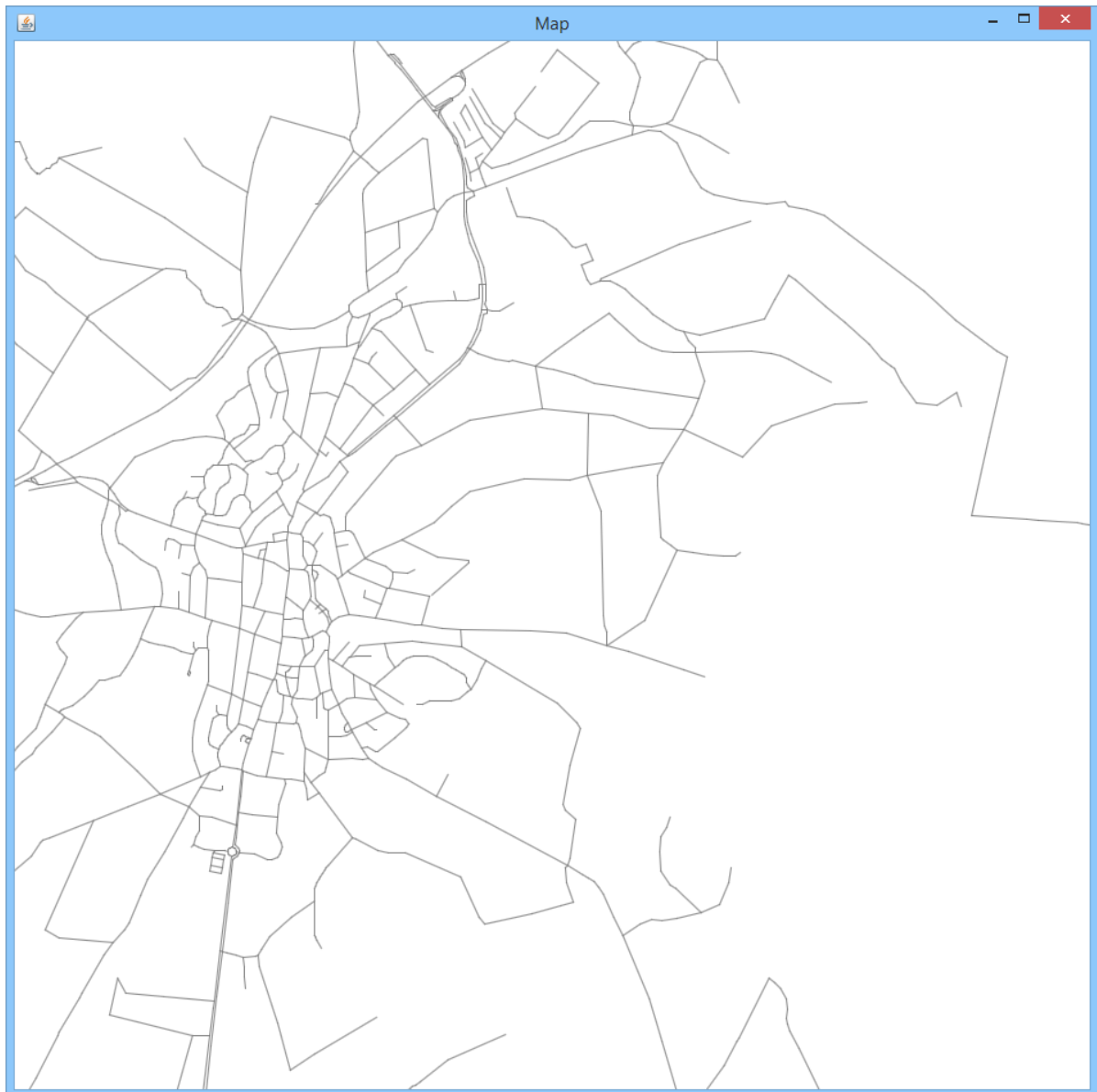


Abbildung 3: Szenario Rennerod

Das erste Kartenszenario bildet die Gegend um Rennerod in Rheinland-Pfalz ab (siehe Abbildung 3), eine Kleinstadt mit etwas über 4000 Einwohnern. Der Kartenausschnitt ist in Nord-Süd-Richtung 3,9 km lang und in Ost-West-Richtung 2,6 km lang. Er wurde als Beispiel für ein Straßennetz mit einer geringen Knotendichte und damit auch geringer Besiedelung ausgewählt. Ein Ort dieser Größe besitzt meist eine grundlegende Struktur an Dienstleistungszentren und Einzelhandelsgeschäften, die es ermöglichen anzunehmen, dass die Nutzer diesen Raum während der Woche Simulationszeit nicht verlassen.



Abbildung 4: Szenario Neubrandenburg

Der zweite Kartenausschnitt (siehe Abbildung 4) zeigt die Stadt Neubrandenburg in Brandenburg mit ca. 63000 Einwohnern. Dieser Ausschnitt wurde gewählt, da er eine wesentlich höhere Knotendichte und Besiedelung als der erste Ausschnitt aufweist und außerdem über ein sehr gut in OpenStreetMap eingetragenes Datenset verfügt. Der Ausschnitt ist in Nord-Süd-Richtung 3,4 km lang und in Ost-West-Richtung 3,1 km. Damit besitzen die beiden Kartenausschnitte eine vergleichbare Fläche.

Einen Ausschnitt aus einer größeren Stadt zu wählen wurde ebenfalls erwogen, aber verworfen. Die Städte werden hierbei als teilweise geschlossenes System betrachtet, bei der sich bei der Simulation keine Nutzer aus der Stadt heraus oder wieder herein bewegen. Ein 3x3 km Abschnitt einer größeren Stadt wie Berlin kann aber kaum die ganze Stadt oder den Großteil der Innenstadt abbilden. Zusätzlich würden öffentliche Fortbewegungsmittel, wie U- und S-Bahnen die Bewegung sehr stark und unabhängig vom Straßennetz beeinflussen.

## 5.2 Ergebnisse

### 5.2.1 Rennerod-RWP

Die Ergebnisse für den Random Waypoint Algorithmus im Straßennetz Rennerod zeigen, dass Transportzeit (siehe Abbildung 5) und Anzahl der abgelieferten Pakete relativ regelmäßig mit der Anzahl der eingesetzten Nutzer skalieren. Zwar liefern schon 10 Nutzer mit 90 Prozent der Pakete eine sehr große Anzahl aus, allerdings steigt dieser Wert bei 100 Nutzern nicht nur auf 100 Prozent, sondern dieses Szenario ist auch das einzige, welches frühzeitig durch Auslieferung aller Pakete stoppt. Dabei sind im Durchschnitt nach nur 3 1/2 Tagen alle Pakete ausgeliefert. Die Handovers (siehe Abbildung 6) sinken interessanterweise mit der Zeit, obwohl eine gewisse Erhöhung durch die erhöhte Anzahl an Nutzern erwartet werden könnte. Die Heatmap für Rennerod (siehe Abbildung 7) zeigt beim Random Waypoint Algorithmus in allen Konfigurationen ein ähnliches Bild. Beliebte sind die Hauptstraßen, da auf ihnen der kürzeste Weg mit dem A\*-Algorithmus zu vielen Punkten liegt.

### 5.2.2 Neubrandenburg-RWP

In Neubrandenburg werden ähnliche Ergebnisse mit dem Random Waypoint Algorithmus wie in Rennerod erzielt, wobei allerdings hier die Handovers (siehe Abbildung 9) genauso mit der Anzahl der Nutzer ansteigen, während zurückgelegte Distanz und benötigte Lieferzeit (siehe Abbildung 8) der abgelieferten Pakete sinken. Insgesamt werden weniger Pakete abgeliefert, bei 10 Nutzern nur etwa die Hälfte der Pakete, bei 100 ca. 91 Prozent. Dies dürfte aber eine erwartete Auswirkung der höheren Knotendichte sein. Auf der Heatmap (siehe Abbildung 10) kann wieder erkannt werden, welche Straßen besonders oft Teil der kürzesten Wege mit dem A\*-Algorithmus sind, wobei bedacht werden muss, dass in der OSM-Karte teilweise einzelne Fahrbahnen mehrspuriger Straßen als einzelner Weg angezeigt werden, womit sich das Volumen dort auf die separaten Wege verteilt.



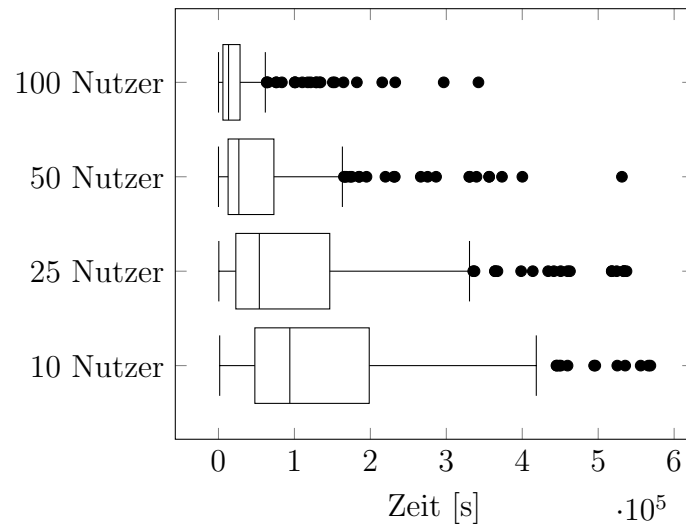


Abbildung 5: Lieferzeit Rennerod RWP

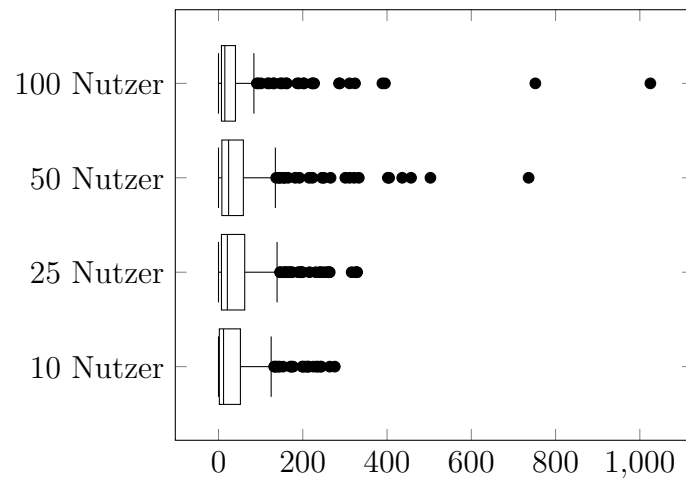


Abbildung 6: Handovers Rennerod RWP

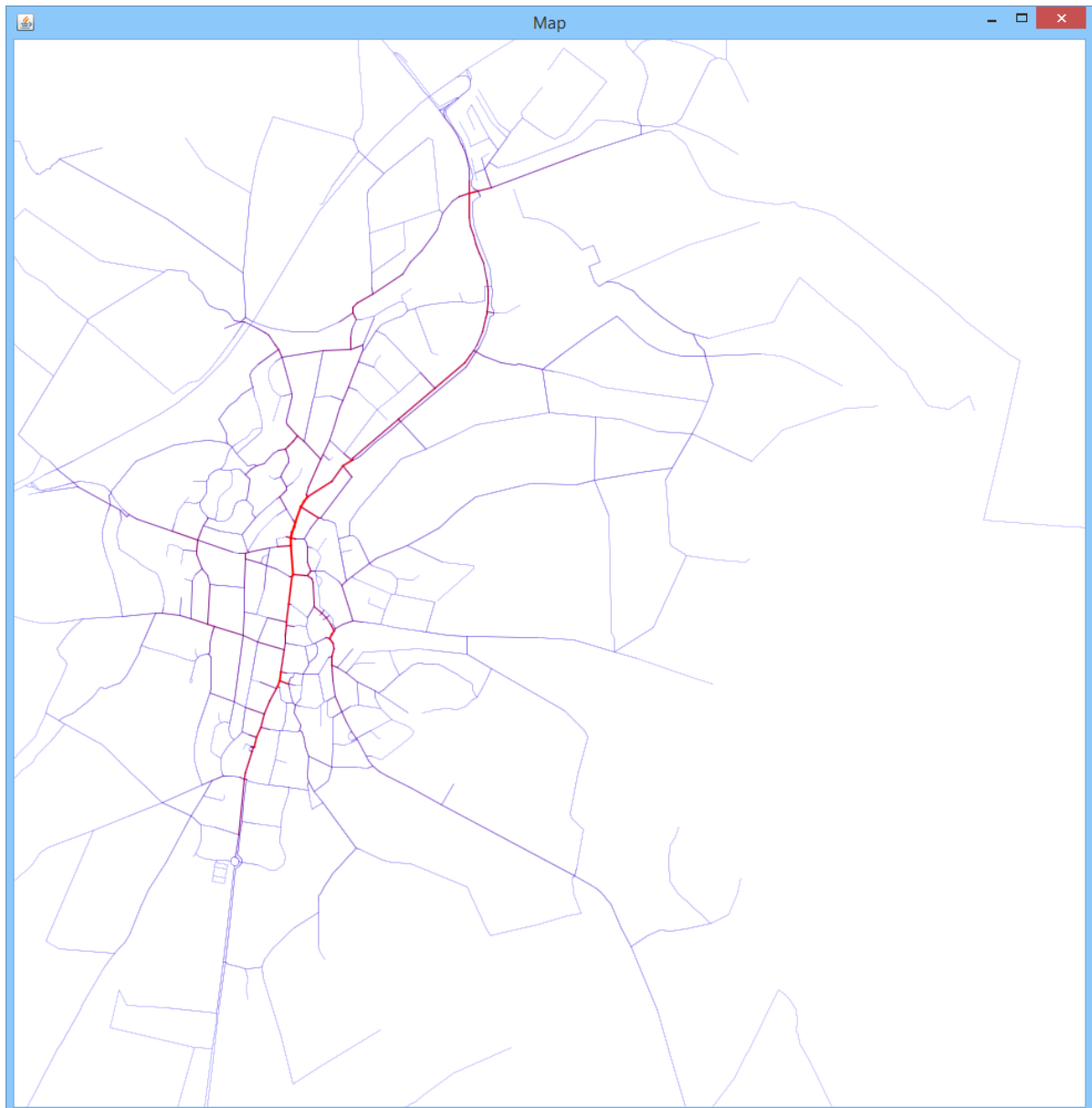


Abbildung 7: Heatmap Rennerod RWP 100 Nutzer

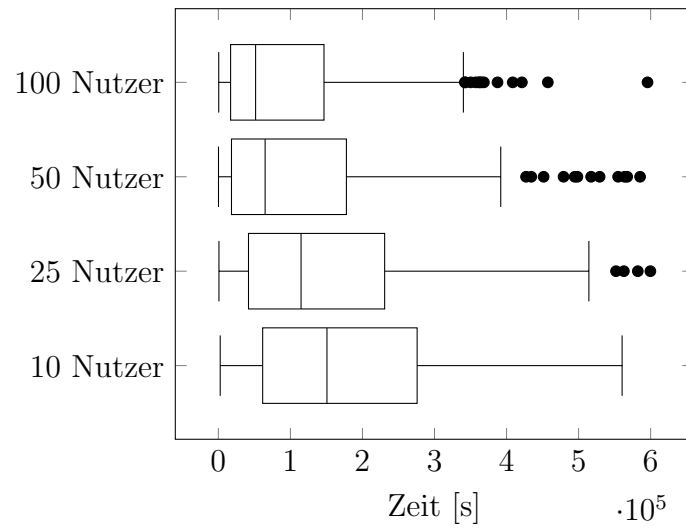


Abbildung 8: Lieferzeit Neubrandenburg RWP

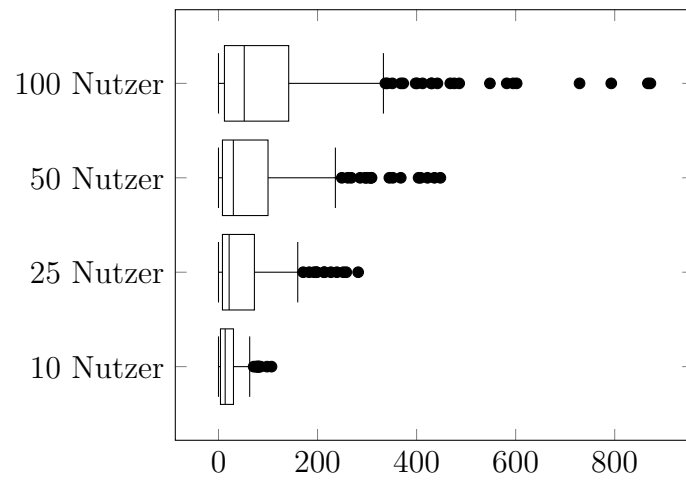


Abbildung 9: Handovers Neubrandenburg RWP

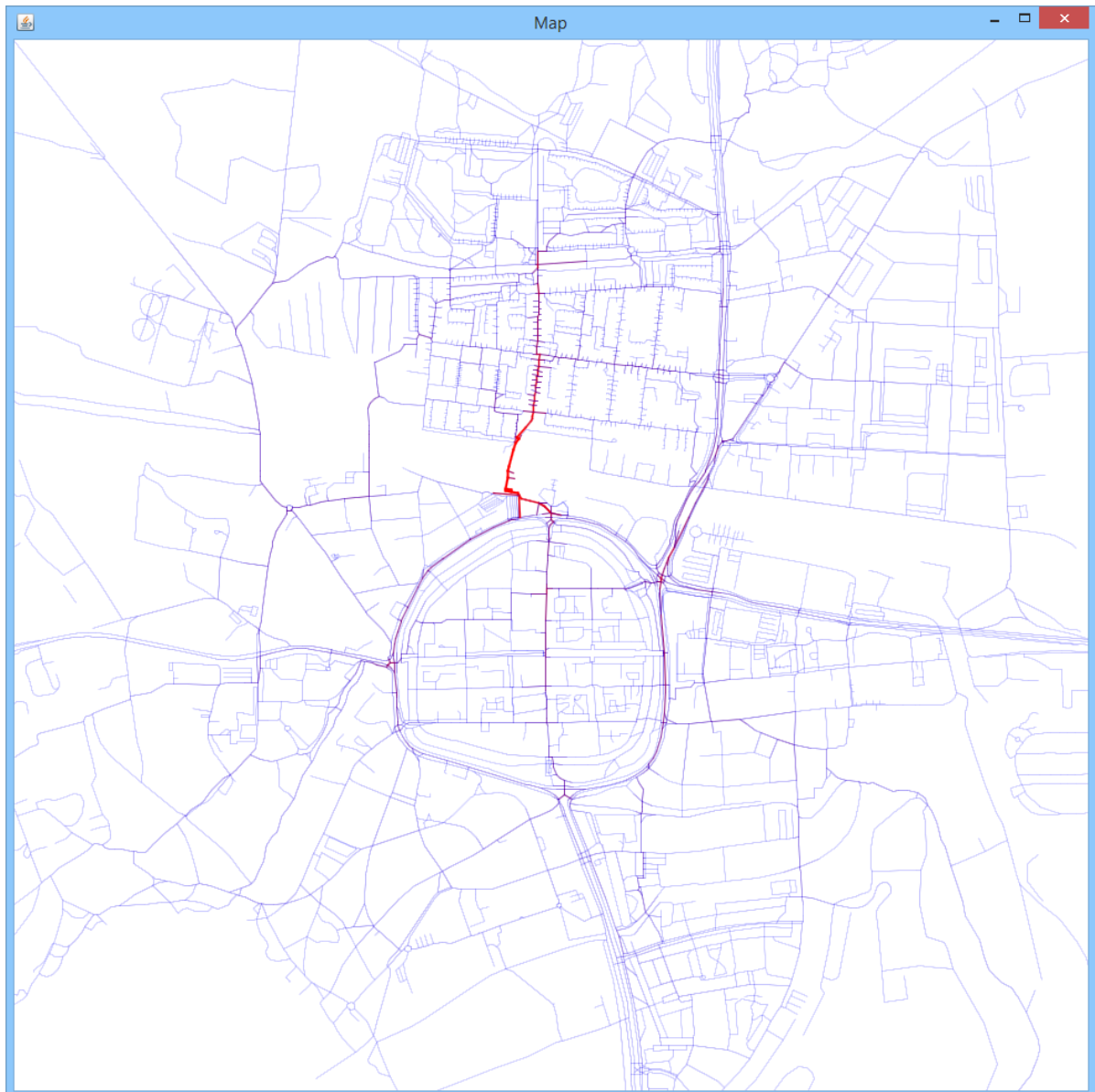


Abbildung 10: Heatmap Neubrandenburg RWP 100 Nutzer

### 5.2.3 Rennerod-Cluster

Der Cluster-Algorithmus erzielt insgesamt schlechtere Ergebnisse als der RWP-Algorithmus, was aber zu erwarten war, da die Nutzer sich hier nicht permanent bewegen und daher kaum so viele Pakete abliefern können. Trotzdem gilt auch hier allgemein, dass die Ergebnisse positiv mit der Anzahl der Nutzer skalieren. Das Szenario Rennerod zeigt ebenfalls gut, dass auch bei realistischerer Bewegung der Nutzer gute Performan-cewerte erreicht werden können. Ab 50 Nutzern werden über 80 Prozent der Pakete innerhalb einer Woche ausgeliefert. Besonders ist, dass bei 100 Nutzern die Anzahl der abgelieferten Pakete lediglich minimal besser ist als bei 50 Nutzern. Auch die Lieferzeit sinkt nicht (siehe Abbildung 11), sondern steigt im Durchschnitt sogar geringfügig. Genau so, wie die Handovers von 25 auf 50 Nutzer nicht ansteigen (siehe Abbildung 12), was ebenfalls nicht mit dem beobachteten Verhalten bei den anderen 3 Nutzergruppen übereinstimmt. Auslöser dafür könnten sehr ungünstige Clusterplatzierungen bei den 100 Nutzer-Simulationen und extrem günstige bei den 50 Nutzer-Simulationen sein. Die Heatmap, hier aus einer Simulation mit 10 Nutzern (siehe Abbildung 13), zeigt auffällig, wie die Clusterplatzierung die beliebten Wege im Gegensatz zum RWP beeinflussen kann.

### 5.2.4 Neubrandenburg-Cluster

Bei der hohen Knoten- und damit Clusteranzahl in Neubrandenburg registriert man den Performanceverlust durch den Algorithmus besonders stark. 10 Nutzer reichen gerade aus, um in einem solch komplexen Straßennetz in einer Woche 15 Prozent der Pakete ausliefern zu können. 100 Nutzer schaffen etwas über 50 Prozent, was zeigt, dass die Skalierung des Routings gut funktioniert. Allerdings werden in einer Stadt mit 50000 Einwohnern wohl über 100 Nutzer benötigt, um eine zuverlässige Leistung in einer vertretbaren Zeitspanne zu erzielen. Der Median der Lieferzeit sinkt mit der erhöhten Anzahl der Nutzer zwar (siehe Abbildung 14), aber insgesamt benötigen die abgelieferten Pakete wesentlich mehr Zeit als in den anderen 3 Szenarien. Die Handovers steigen allerdings durch die erhöhte Anzahl an Treffen bei mehr Nutzern (siehe Abbildung 15) und die Heatmap zeigt auch, wie in Rennerod, den Einfluss der Clusterplatzierung auf die beliebtesten Wege (siehe Abbildung 16).

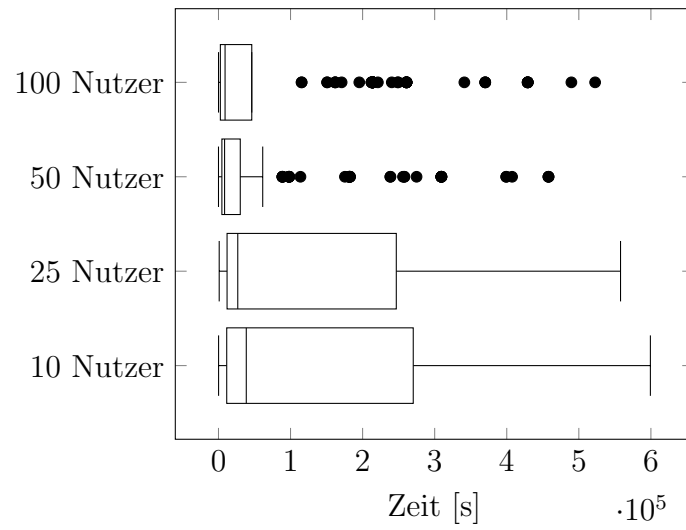


Abbildung 11: Lieferzeit Rennerod Cluster

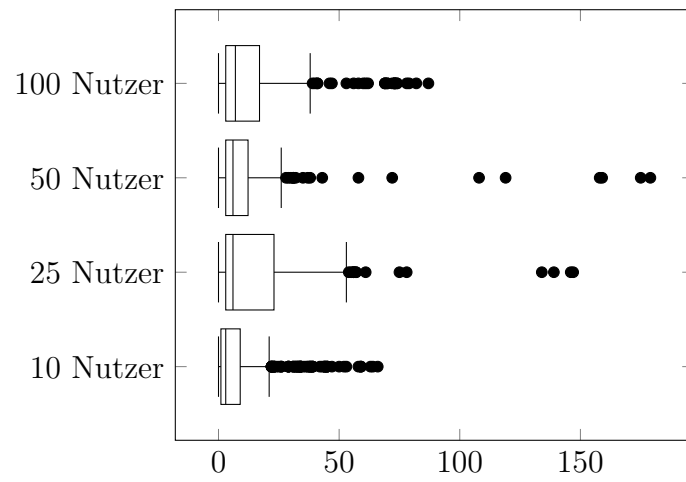


Abbildung 12: Handovers Rennerod Cluster

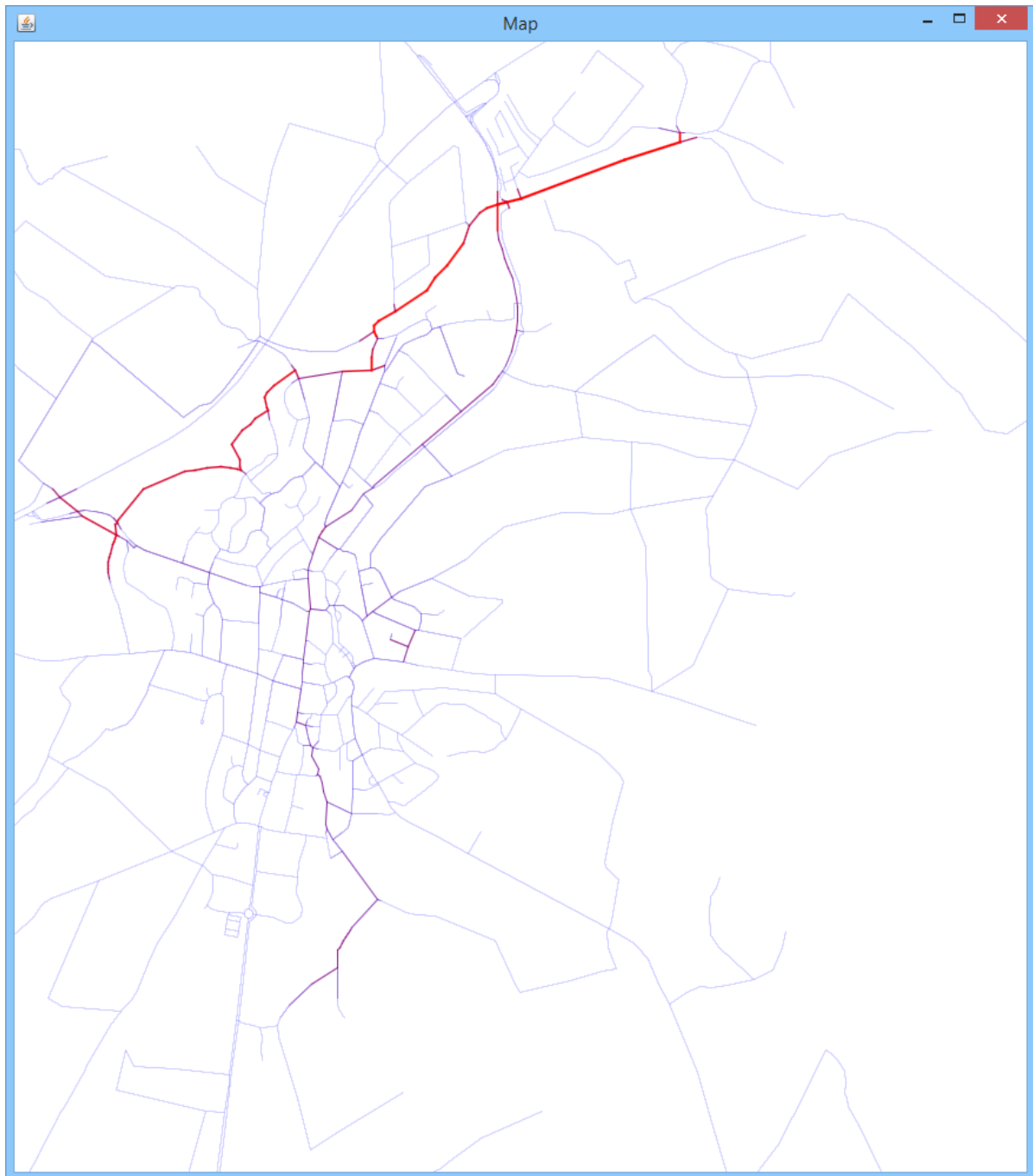


Abbildung 13: Rennerod Heatmap Cluster 10 Nutzer

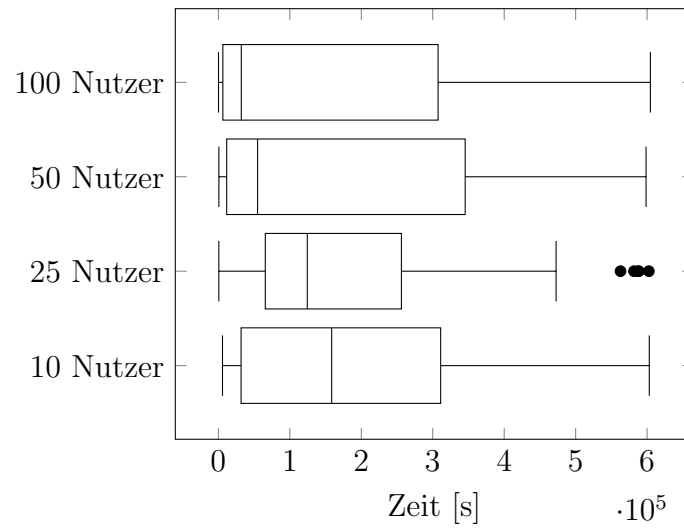


Abbildung 14: Lieferzeit Neubrandenburg Cluster

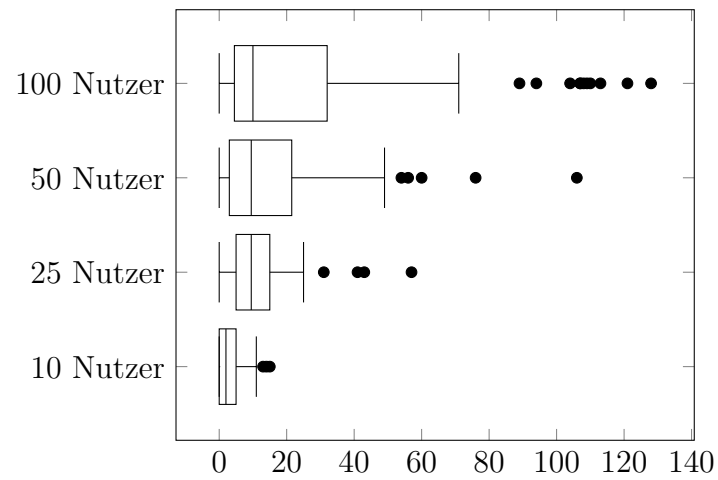


Abbildung 15: Handovers Neubrandenburg Cluster



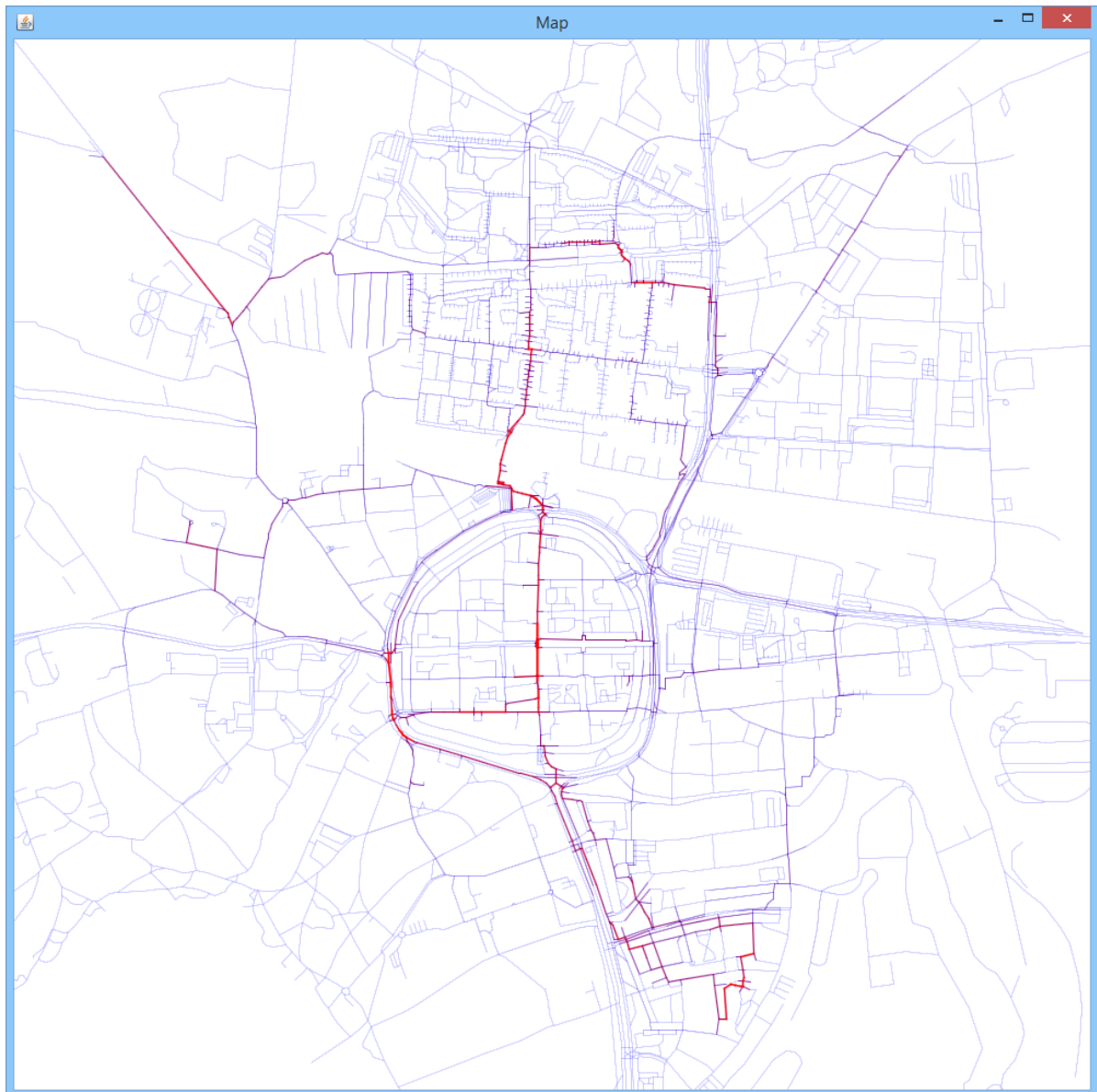


Abbildung 16: Neubrandenburg Heatmap Cluster 25 Nutzer

## 5.3 Analyse

Nach Betrachtung der Simulationsergebnisse lassen sich einige allgemeine, auf alle simulierten Szenarien zutreffende Aussagen machen. Erstens, die Anzahl der Nutzer erhöht sowohl die Anzahl der erfolgreich abgelieferten Pakete als auch die Anzahl der durchgeführten Handovers. Zweitens, je mehr Knoten und Wege ein Straßennetz besitzt, desto schwieriger wird der Transport mit POST. Drittens, der Random Waypoint Algorithmus erzielt bessere Ergebnisse als der Cluster Waypoint Algorithmus.

Zur ersten Aussage lässt sich sagen, dass dieses Verhalten zu erwarten ist. Je mehr Nutzer Pakete transportieren, desto wahrscheinlicher wird es, dass sie am Zielort des Paketes vorbeikommen und dieses abliefern können. Auch, dass die Pakete schneller abgeliefert werden ist nachvollziehbar, da die Wahrscheinlichkeit erhöht ist, dass ein Nutzer früher den Paketort passiert, wenn viele Nutzer unterwegs sind. Das Ansteigen der Anzahl der Handovers ist erklärbar durch die erhöhte Wahrscheinlichkeit einen anderen Nutzer zu treffen, wenn mehr Nutzer unterwegs sind. Bei den Handovers kann allerdings eine leichte Abweichung beobachtet werden. Während zwischen Rennerod RWP mit 50 Nutzern und Rennerod RWP mit 100 Nutzern, die mittlere Transportlänge sinkt und die Anzahl der abgelieferten Pakete steigt, sinkt gleichzeitig die Anzahl der Handovers. Hier kann man beobachten, dass die Anzahl der Handovers sinkt, sobald die Anzahl der abgelieferten Pakete das Maximum erreicht. Fügt man Nutzer hinzu, wenn die bereits vorhandenen Nutzer ausreichen, um alle Pakete im Simulationszeitraum abzuliefern, wird die Zahl der Handovers bei weiterhin sinkender Lieferzeit zurückgehen. Plausibel erklärt sich dieses Verhalten daraus, dass die Nutzer, wenn sie die Pakete schneller abliefern, diese nicht so lange herumtragen und daher nicht so viele Gelegenheiten zum Austauschen bestehen.

Die zweite Beobachtung deckt sich ebenfalls mit der Erwartung. Wenn es mehr potentielle Ziele und Wege für Nutzer gibt ist es schwieriger sich zu treffen, da die Nutzer sich über eine größere Fläche verteilen. Dies führt zu einer geringeren Anzahl an ausgelieferten Paketen sowie einer niedrigeren Anzahl an Handovers, da weniger Treffen auch weniger Gelegenheiten zur Routing-Verhandlung bieten. Anzumerken bei den Straßennetzen ist auch, dass der Medianwert für die abgelieferten Pakete über die gesamte Simulation betrachtet sehr niedrig ist. Dies dürfte damit zusammenhängen, dass ein Paket, welches als Ziel eine Hauptbewegungsader hat, meist sehr schnell abgeliefert wird. Die verbliebenen, nicht abgelieferten Pakete nach einer Woche sind meist lediglich die, welche an Knoten geliefert werden müssen, die selten besucht werden.

Die unterschiedliche Leistungsfähigkeit beider Bewegungsalgorithmen ist ebenfalls zu erwarten. Die Pakete (siehe Abbildung 17 und 18) und Nutzer (siehe Abbildung 21 und 22) bewegen sich im Cluster Waypoint Algorithmus wesentlich weniger, als im RWP. Beim Cluster Waypoint muss außerdem beachtet werden, dass eine ungünstige zufällige Platzierung der Cluster einen Einfluss auf die Werte haben kann.

Im Vergleich zwischen Rennerod mit 50 und 100 Nutzern ergibt sich nur eine gering erhöhte Anzahl von abgelieferten Paketen (siehe Abbildung 20), die auf ungünstige Clusterplatzierung bei den 100 Nutzer Simulationsläufen zurückzuführen ist. Dieser Effekt wäre vermutlich bei einer erhöhten Anzahl an Simulationsläufen weniger stark ausgeprägt, aber es bleibt festzuhalten, dass die 100 Nutzer trotzdem eine Verbesserung bei der Anzahl der abgelieferten Pakete erreichen.

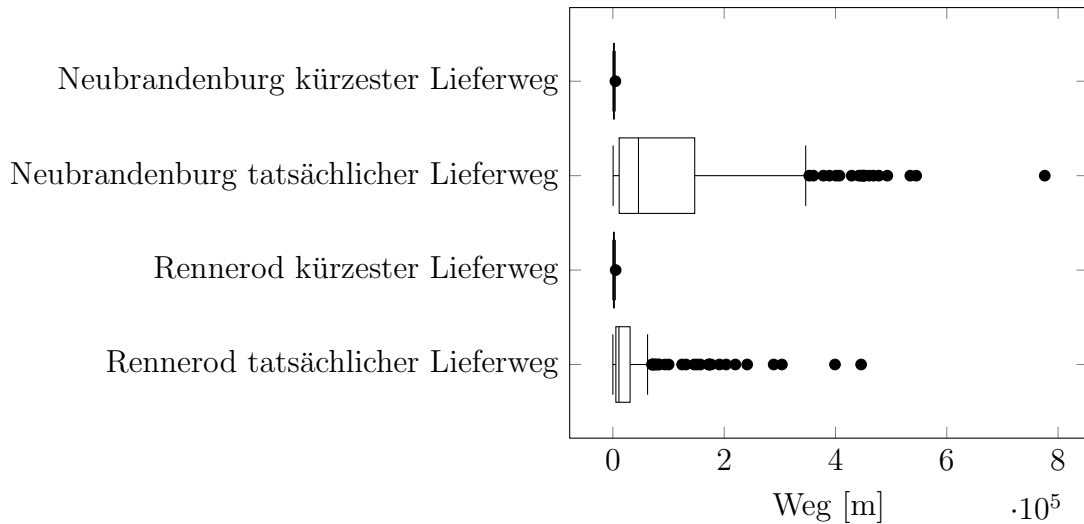


Abbildung 17: Vergleich Weglänge Random Waypoint

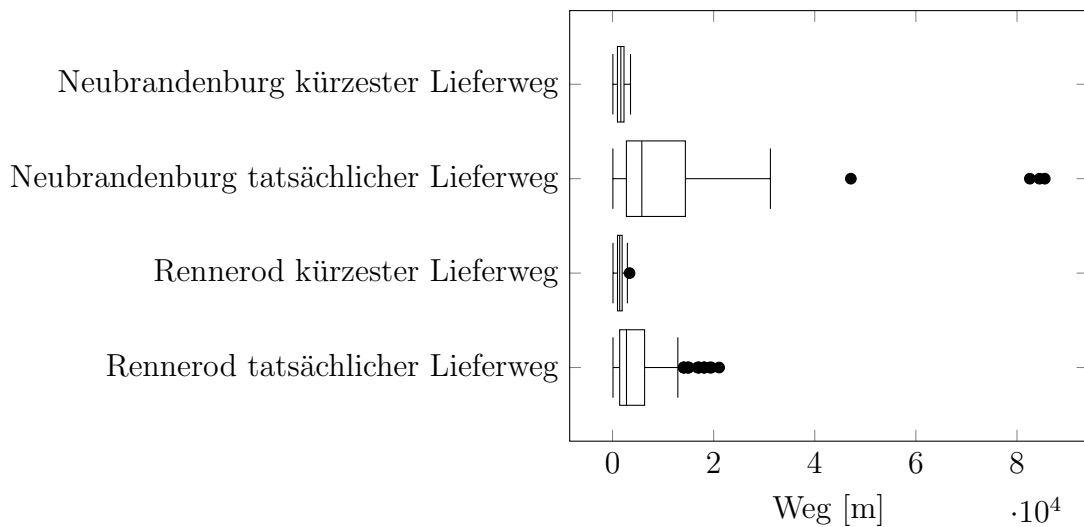


Abbildung 18: Vergleich Weglänge Cluster Waypoint

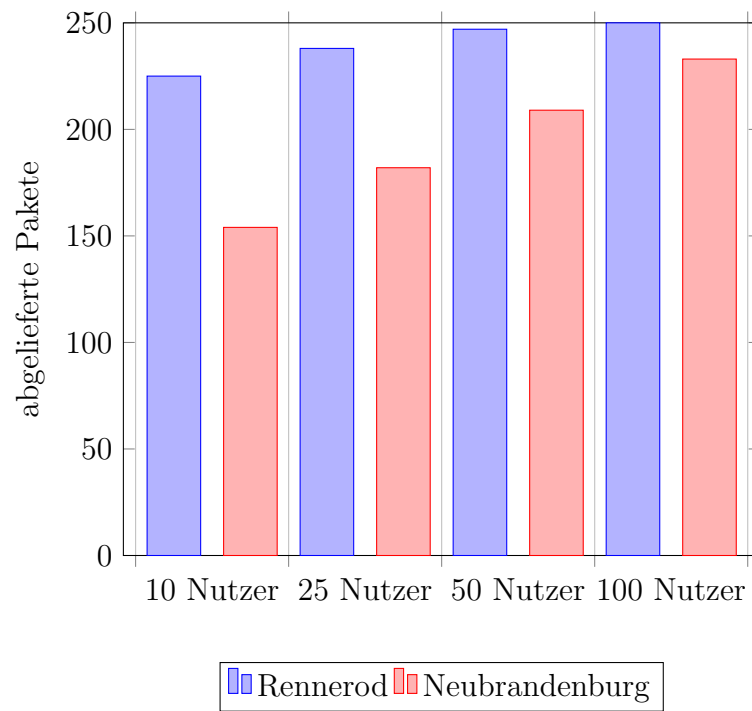


Abbildung 19: Abgelieferte Pakete im Random Waypoint Algorithmus

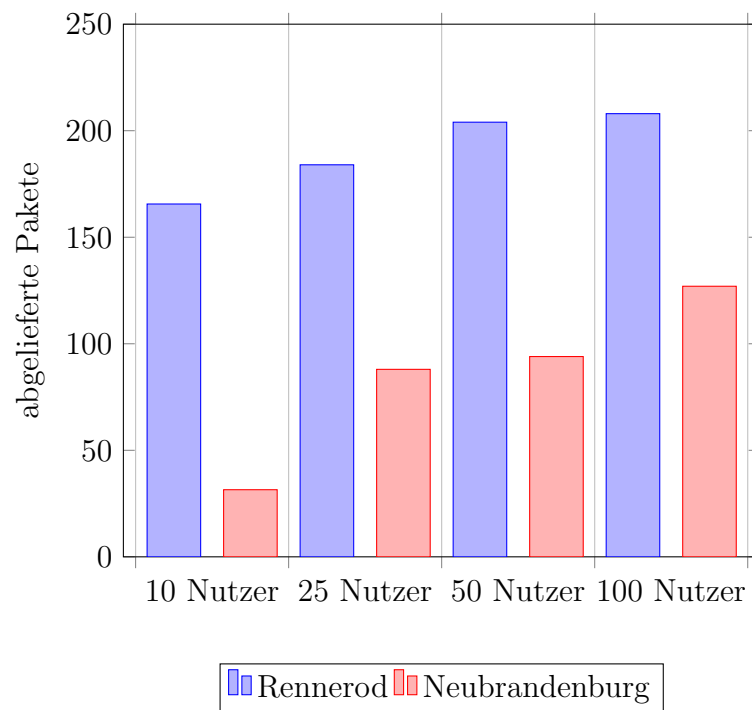


Abbildung 20: Abgelieferte Pakete im Cluster Waypoint Algorithmus

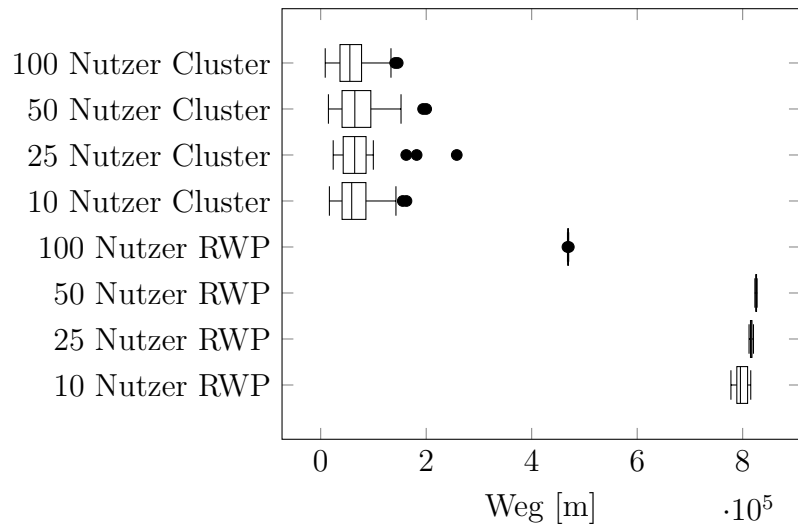


Abbildung 21: Zurückgelegte Strecke der Nutzer in Rennerod

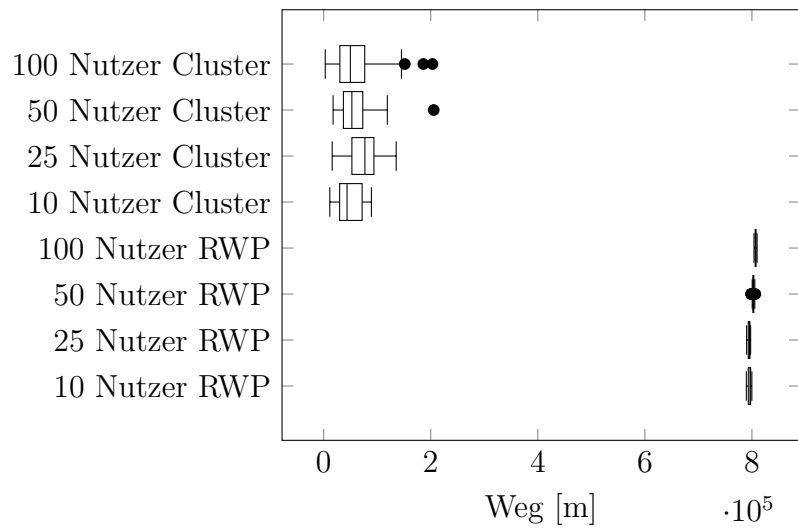


Abbildung 22: Zurückgelegte Strecke der Nutzer in Neubrandenburg

## 6 Fazit und Ausblick

### 6.1 Fazit

Insgesamt ist festzuhalten, dass die Performance von POST in dieser Arbeit vielversprechend ist. Die Simulationen haben gezeigt, dass durch POST auch auf Straßennetzen und bei einer Bewegung, die realistischen Bewegungsabläufen ähnelt, respektable Ergebnisse bei der Paketdistribution in einer annehmbaren Zeitspanne erreicht werden können. Die Simulation hat auch Hinweise auf die nötige Anzahl von Nutzern gegeben, die für ein erfolgreiches, lokales POST-Netzwerk nötig sind. In unseren Simulationen mit dem Cluster Waypoint Algorithmus zeigte sich, dass in einer Kleinstadt mit etwas über 4000 Einwohnern nur 50 Nutzer nötig waren, um über 75 Prozent aller Pakete innerhalb einer Woche auszuliefern. Im zweiten Kartenszenario ergab sich bei einer größeren Stadt mit über 60000 Einwohnern bei 100 Nutzern nur eine Quote von knapp 50 Prozent ausgelieferten Paketen.

Ein ideales Anwendungsgebiet für POST ist eines, in dem Nutzer sich oft treffen und welches eher über wenige Hauptwege, als viele weniger besuchte Straßen verfügt. Das Routing profitiert ebenfalls davon, wenn das Gebiet kompakter ist. Interessante und vielversprechende Anwendungsgebiete wären zum Beispiel ein Werksgelände oder ein Universitätscampus, da in beiden relativ viel Bewegung, bei oftmals wenigen Hauptverkehrsadern, vorhanden ist. Dort wäre die kritische Masse für einen Transport mit hoher Effizienz vermutlich schnell zu erreichen und diese würden damit ein ideales Anwendungsgebiet für einen Feldtest bieten.

Verbesserungen beim Routing-Protokoll von POST sind ebenfalls möglich und könnten zum Beispiel über eine Speicherung von Clustern realisiert werden. Beim Cluster Waypoint Algorithmus besitzen Nutzer verschiedene bekannte Cluster, die sie immer wieder besuchen, zum Beispiel ihre Wohnung, Arbeitsstelle, ein Schwimmbad oder Supermarkt. Würde das Protokoll diese Cluster kennen und berücksichtigen, könnte ein effektiveres Routing entstehen, da man Nutzern Pakete übertragen könnte, die zu einem ihrer bekannten Cluster geliefert werden müssen, auch wenn diese gerade nicht auf dem Weg zu diesem Cluster sind, es aber wahrscheinlich ist, dass sie ihn in nächster Zeit besuchen werden. Weiterhin könnte die POST-App beim Routing eine Straßenkarte der Umgebung berücksichtigen und so bei Paketverhandlungen besser die Vereinbarkeit der Ziele der Nutzer mit den Zielen der Pakete prüfen, als nur die reine Luftlinie zu betrachten.

Die Performance ist zwar soweit vielversprechend, dass weitere Forschung lohnend erscheint, aber solange nicht eine hohe Zahl an Nutzern erreicht wird, wäre POST kaum in der Lage einen zentralisierten Postdienstleister bei gleicher Leistung zu ersetzen. Es könnte diesen aber in gewissen Szenarien durchaus unterstützen.

## 6.2 Ausblick

Die tatsächliche Umsetzung einer Nutzung von MANETs zum Transport physischer Objekte steht noch am Anfang, bleibt aber ein interessantes Feld, welches mit der Weiterentwicklung von Smartphones und drahtloser Kurzstreckenkommunikation nur noch vielversprechender wird.

Im Bezug auf diese Arbeit bietet die erstellte Simulationsanwendung ein solides Fundament, auf welchem weitere Simulationen zum Projekt POST aufgebaut werden könnten. Weiterentwicklungen beim POST-Konzept, zum Beispiel eine genauere Beschreibung der physischen Paketübergabe, könnten in die Simulation eingepflegt werden. Zusätzlich könnten weitere Bewegungsalgorithmen eingebaut werden, welche das Verhalten von Nutzern in allgemeinen oder bestimmten Situationen besser abbilden. Der Cluster-Algorithmus könnte verbessert werden, indem die genutzten Parameter verändert werden und die Platzierung von Clustern optimiert wird. Die Nutzung der OSM-Kartendaten könnte verbessert werden. In diesen Daten sind nicht nur die Straßen eines Gebiets gespeichert, sondern oft auch andere Informationen, wie öffentliche Gebäude oder Vergnügungsparks. Möglich wäre es diese Zusatzinformationen zu nutzen, um direkt relevante und tatsächlich im Gebiet vorhandene Cluster zu bestimmen. Allerdings würde der Erfolg einer solchen Methodik auch von der Qualität der OSM-Kartendaten abhängen.

Mit den Ergebnissen der Simulationsläufe dieser Arbeit oder eventuellen verbesserten Simulationsläufen könnten dann auch besonders vielversprechende Testumgebungen isoliert werden, in welchen Erkenntnisse für die weitere Entwicklung des Protokolls gewonnen werden könnten.

Da eine allgemeine Machbarkeit des erfolgreichen Routings festgestellt werden konnte, sollten andere potentielle Probleme beim Versand physischer Objekte ebenfalls untersucht werden. Wie kann zum Beispiel die Sicherheit der Pakete gewährleistet werden? Dort würden sich zum Beispiel Arbeiten in Richtung eines Vertrauens- oder Reputations-systems ohne zentralisierte Instanz anbieten. Alternativ wäre auch die Möglichkeit eines Systems mit Bezahlung und bezahlten Trägern zu untersuchen. Ein weiterer Aspekt ist die Privatheit der Nutzer und wie viele Informationen die POST-Applikation über sie besitzt. Dabei ist besonders zu beachten, wie viele und wie detaillierte Informationen beim Routing übertragen werden. Hier gibt es ebenfalls potentielle Arbeiten, was das Abwägen zwischen effizientem Routing und effizientem Schutz der Privatsphäre angeht.

Sollten diese Probleme zufriedenstellend gelöst sein, wird das Entwickeln oder Weiterentwickeln und Testen einer nutzbaren POST-Implementierung der nächste notwendige Schritt zu einer erfolgreichen Nutzung von mobilen ad hoc Netzwerken für das Transportieren physischer Objekte sein.

## Literatur

- [1] Artemios Voyiatzis. „A survey of delay-and disruption-tolerant networking applications“. In: *Journal of Internet engineering* 5.1 (2012). URL: <http://www.jie-online.org/index.php/jie/article/view/91>.
- [2] Shanyang Zhao. „Toward a Taxonomy of Copresence Presence“. In: (2003). URL: [http://astro.temple.edu/~bzhao001/Taxonomy\\_Copresence.pdf](http://astro.temple.edu/~bzhao001/Taxonomy_Copresence.pdf).
- [3] Jing Su u. a. „User Mobility for Opportunistic Ad-Hoc Networking“. In: (2004). URL: <http://www.eecg.toronto.edu/~ashvin/publications/wmcsa-2004.pdf>.
- [4] A. Chaintreau u. a. „Impact of Human Mobility on Opportunistic Forwarding Algorithms“. In: *IEEE Transactions on Mobile Computing* 6.6 (2007), S. 606–620. URL: <http://www.cs.columbia.edu/~augustin/pub/chaintreau07impact.pdf>.
- [5] Libo Song und David F. Kotz. „Evaluating Opportunistic Routing Protocols with Large Realistic Contact Traces“. In: (2007).
- [6] Liam McNamara, Cecilia Mascolo und Licia Capra. „Media Sharing based on Co-location Prediction in Urban Transport“. In: (2009). URL: <https://www.cl.cam.ac.uk/~cm542/papers/mobicom08.pdf>.
- [7] Phuong T. Nguyen, Volkmar Schau und Wilhelm R. Rossak. „Towards an adaptive Communication model for mobile agents in highly dynamic networks based on swarming behaviour“. In: *Proceedings of the 9th European Workshop on Multi-agent Systems*. 2011. URL: [http://www.researchgate.net/publication/259198098\\_Towards\\_an\\_Adaptive\\_Communication\\_Model\\_for\\_Mobile\\_Agents\\_in\\_Highly\\_Dynamic\\_Networks\\_based\\_on\\_Swarming\\_Behaviour](http://www.researchgate.net/publication/259198098_Towards_an_Adaptive_Communication_Model_for_Mobile_Agents_in_Highly_Dynamic_Networks_based_on_Swarming_Behaviour).
- [8] Marius Senftleben u. a. „MoP-2-MoP - Mobile private microblogging“. In: *Financial Cryptography and Data Security (FC'14)*. Springer, 2014. URL: [https://www.informatik.tu-darmstadt.de/fileadmin/user\\_upload/Group\\_TRUST/PubsPDF/fc14-paper99-final\\_draft\\_v3.pdf](https://www.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_TRUST/PubsPDF/fc14-paper99-final_draft_v3.pdf).
- [9] Tracy Camp, Jeff Boleng und Vanessa Davies. „A Survey of Mobility Models for Ad Hoc Network Research“. In: (2002).
- [10] Lars Fischer. „Evolving Logistics: Physical-Objects Sneaker Transport (POST)“. In: *Ortsbezogene Anwendungen und Dienste, Fachgespräch der GI-Fachgruppe KuVS*. 2011. URL: [http://larsipulami.de/papers/fischer\\_gi2011\\_post.pdf](http://larsipulami.de/papers/fischer_gi2011_post.pdf).
- [11] Injong Rhee u. a. „Human Mobility Patterns and Their Impact on Routing in Human-Driven Mobile Networks“. In: (2007). URL: <http://www.cs.columbia.edu/~augustin/pub/chaintreau07impact.pdf>.
- [12] Aarti Munjal, Tracy Camp und William C. Navidi. „SMOOTH: A Simple Way to Model Human Mobility“. In: (2011).



- [13] Daniel Krajzewicz u. a. „SUMO (Simulation of Urban MObility)“. In: (2002). URL: [www.researchgate.net/publication/224793504\\_SUMO\\_\(Simulation\\_of\\_Urban\\_MObility\)\\_-\\_an\\_open-source\\_traffic\\_simulation/file/9fcfd50f5af33f1ce8.pdf](http://www.researchgate.net/publication/224793504_SUMO_(Simulation_of_Urban_MObility)_-_an_open-source_traffic_simulation/file/9fcfd50f5af33f1ce8.pdf).
- [14] Amit Kumar Saha und David B. Johnson. „Modeling Mobility for Vehicular Ad Hoc Networks“. In: (2004). URL: [http://pdf.aminer.org/000/505/769/modeling\\_mobility\\_for\\_vehicular\\_ad\\_hoc\\_networks.pdf](http://pdf.aminer.org/000/505/769/modeling_mobility_for_vehicular_ad_hoc_networks.pdf).
- [15] Alexander von Bodisco und Thomas Oberwallne. „Different Strokes for Different Folks: An Evaluation of Road Networks and their Impact on VANET Mobility“. In: *PIK - Praxis der Informationsverarbeitung und Kommunikation* 2.37 (2014).
- [16] Fritz Kristiano Karnadi, Zhi Hai Mo und Kun-chan Lan. „Rapid Generation of Realistic Mobility Models for VANET“. In: (2007). URL: <http://www.csie.ntpu.edu.tw/~yschen/course/96-2/Wireless/papers/Simulation-2.pdf>.
- [17] Lars Fischer und Julia Dauwe. „Design and Implementation of Co-Presence Transportation for Physical Objects“. In: (2014).
- [18] P.E. Hart, N.J. Nilsson und B. Raphael. „A Formal Basis for the Heuristic Determination of Minimum Cost Paths“. In: *Systems Science and Cybernetics, IEEE Transactions on* 4.2 (1968), S. 100–107. ISSN: 0536-1567. DOI: 10.1109/TSSC.1968.300136.

## Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen aus dem Internet. Diejenigen Paragraphen der für mich gültigen Prüfungsordnung, welche etwaige Betrugsversuche betreffen, habe ich zur Kenntnis genommen. Der Speicherung meiner Bachelorarbeit zum Zweck der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

---

Liebenscheid, den 7. April 2015