

UNIVERSITÄT OSNABRÜCK

SEMINAR VERTEILTE SYSTEME
WINTERSEMESTER 14/15

PROF. NILS ASCHENBRUCK

Opportunistische Netze

Anwendungen, Protokolle und Modelle

Autor:

Christian HEIDEN
cheiden@uos.de
(945303)

Betreuer:

Matthias SCHWAMBORN

7. Dezember 2014

Inhaltsverzeichnis

1	Einleitung	1
2	Anwendungsgebiete	1
2.1	ZebraNet	2
2.2	Haggle Project	2
3	Forwarding-Protokolle	2
3.1	Ohne Kontext - Epidemic	4
3.2	Partieller Kontext - PRoPHET	5
3.2.1	Tabellen-Update	5
3.2.2	Forwarding-Prozess	6
3.3	Voller Kontext - SimBet	6
3.3.1	Betweenness-Index	7
3.3.2	Similarity-Index	7
4	Bewegungsmodelle	7
4.1	Random Walk Mobility Model	8
4.2	Pursue Mobility Model	8
5	Zusammenfassung	9

1 Einleitung

Mobile Endgeräte wie Handys, Tablets oder Laptops sind immer populärer geworden. Die meisten dieser Geräte verfügen über drahtlose Kommunikationsmöglichkeiten, wie Bluetooth, W-Lan, Mobilfunknetzanbindung und vielem mehr. Zusätzlich werden Akkulaufzeit, Speicherkapazität, Kommunikationsqualität, etc. permanent verbessert. Dies ermöglicht die Nutzung einer Klasse von Anwendungen, welche auf dem direkten Kontakt einzelner Endgeräten basiert. Ein *Kontakt* ist definiert, als eine *one-Hop*-Verbindung zweier Endgeräte, z.B. via Bluetooth, sodass ein Datenaustausch auf direktem Weg, also ohne Umwege über eine dritte Knoten, stattfinden kann. Dieser Verbindungstyp wird auch als *ad-hoc* Verbindung bezeichnet.

Eine Klasse von Anwendungen in denen Endgeräte Daten ad-hoc verschicken sind sogenannte *MANETs* (*mobile ad hoc networks*) [1, p. 2]. Einige (häufig die meisten, bis alle) Knoten eines MANETs sind frei beweglich, sodass deren Verbindungen zu anderen Knoten häufig wechseln können. Außerdem fungieren die Knoten eines MANETs sowohl als Sender, als auch als Router. Denn sobald eine Nachricht von Knoten *A* zu Knoten *B* verschickt werden soll, kann es passieren, dass es nie zu einem Kontakt zwischen *A* und *B* kommt. Folglich muss die Nachricht, über eine Menge von Knoten geleitet werden um schlussendlich bei Knoten *B* anzukommen.

Wenn die Zeitspanne zwischen Verschicken und Ankommen einer Nachricht groß ist, und das Prinzip des Forwardings innerhalb des Netzes nach dem store-carry-forward-Prinzip arbeitet, so bezeichnet man ein MANET auch als opportunistisches Netzwerk (*OppNet*) [2]. Erstgenannte Eigenschaft wird auch *end-to-end-delay* genannt. Die Bezeichnung *opportunistisch* leitet sich aus dem englischen Begriff *opportunity* (Möglichkeit, Chance) ab und bedeutet, dass Knoten eines OppNets die auftretenden Kontakte als Chance sehen ihre Daten weiterzuleiten. Anhand des bereits erwähnten *store-carry-forward*-Prinzips, sollen im Folgenden die Grundlagen des Informationsaustauschs innerhalb eines OppNets erläutert werden:

Kommt es zu einem Kontakt zweier Knoten, tauschen sie in irgendeiner Form Informationen aus. Da beide Knoten, sowohl Sender als auch Router darstellen, speichern sie auch Informationen für andere Knoten ab. Durch ihre Mobilität, tragen sie also diese Informationen an andere Orte, auf deren Wegen es zu weiteren Kontakten kommen kann. Schlussendlich ist es möglich, dass so weitergeleitete Informationen, bei dem jeweiligen Zielknoten ankommen. Eine Herausforderung in OppNets ist es, eine Strategie zu finden, welche Nachrichten so versendet, dass das Ziel erreicht wird. Durch die Mobilität einzelner Knoten, ist die Topologie des Netzes veränderlich und es fällt schwer Aussagen über zukünftige Netzwerkzustände und damit verbundene Routen zu treffen.

Diese Ausarbeitung soll dem Leser einen Überblick über das Feld der OppNets geben. Zu Beginn werden zwei Anwendungen besprochen (Kapitel 2), in denen OppNets genutzt werden. Anschließend wird eine Einführung in die existierenden Forwarding-Protokolle gegeben (Kapitel 3) und es werden exemplarisch drei Protokolle vorgestellt. Daraufhin wird dem Leser ein Einblick in die zur Bewertung der Protokolle genutzten Simulationsmodelle gegeben (Kapitel 4). In Abschnitt 5 werden die in der Ausarbeitung besprochenen Inhalte zusammengefasst.

2 Anwendungsgebiete

Generell eignen sich Opportunistische Netze in Situationen, wo es zu langen kontaktlosen Zeiten zwischen Knoten kommt und diese bei Kontakt in der Lage sind, Daten ad-hoc miteinander auszutauschen. Beispielsweise gibt es Szenarien, in denen Autos die Knoten eines

Netzwerkes darstellen und durch Kontakte mit anderen, z.B. entgegenkommenden, Autos Informationen über Wetter oder Straßenverhältnisse austauschen. Diese Netze werden auch als *VANETs* (*vehicular ad hoc networks*) bezeichnet [2]. Weitere Anwendungsgebiete sind Umgebungen in denen es keine Alternative zu opportunistischen Ansätzen gibt, weil es beispielsweise keine Anbindung an Mobilfunknetze gibt oder ständige Verbindungen aufgrund von Energieknappheit nicht möglich sind. Solche Umgebungen werden auch als *challenged environments* bezeichnet [2]. Im Folgenden werden exemplarisch zwei weitere Szenarien tiefergehend vorgestellt, in denen OppNets zur Anwendung kommen.

2.1 ZebraNet

Ein Anwendungsgebiet ist das Beobachten von Tierverhalten. Beim *ZebraNet* Projekt [3] wurden Zebras Geräte angelegt, welche als Knoten in einem Netz fungieren sollten. Die Umgebung, in der das Experiment stattfand, lässt sich in die Gruppe der *challenging environments* einordnen. Das Ziel war die Analyse ihrer Bewegungsmuster. Die Geräte verfügten über GPS, einem solar-betriebenen Akku, Flash-Speicher, einer kleinen CPU und Hardware zur Datenübertragung an benachbarte Geräte. Gesammelte Daten sollten an ein Auto, die sogenannte *Base-Station*, übertragen werden, welches in gewissen Zeitabständen in Radio-Empfangsreichweite an den Zebras vorbeifuhr. Die Base-Station stellte das endgültige Ziel einer jeden Nachricht dar.

Dadurch ergibt sich die erste Herausforderung dieses Szenarios: Es ist nicht klar, ob ein Tier jemals Kontakt zur Base-Station haben wird. Hierzu müssen Protokolle entwickelt werden, welche dafür sorgen, dass Nachrichten trotzdem an einer Base-Station ankommen, ohne, dass ein Kontakt zwischen Gerät und Base-Station jemals stattgefunden hat. Weitere Herausforderungen sind limitierte Verfügbarkeit von Speicher und Energie, welche ein simples Fluten der Nachrichten an benachbarte Knoten ausschließt.

2.2 Hagle Project

Im Rahmen des Hagle Project [4] wurde eine Software erstellt, welche Nutzer auf ihren Smartphones installieren können. Die Idee ist der Austausch von Inhalten, abhängig von den Interessen der User. Dazu gibt der Nutzer seine eigenen Interessen an und versieht seine Inhalte mit Schlüsselwörtern.

Kommt es zu einem Kontakt zwischen zwei Nutzern, welche die Hagle-Software installiert haben, so findet ein Austausch der Daten statt, die den Interessen des anderen Users entsprechen. Kontakte zwischen Smartphones finden auf Basis von Bluetooth oder W-Lan statt. Hier ist zu beachten, dass Sender und Empfänger einer Nachricht immer die Beteiligten eines Kontakts sind, sodass ein Forwarding über mehrere Knoten nicht nötig ist.

Eine Herausforderung des Hagle Projekts ist die geringe Knotendichte des Netzwerks, da die Software nicht sehr populär ist. Als weitere Schwierigkeit ist die meist begrenzte Energieverfügbarkeit zu benennen. Durch die ständigen Versuche Verbindungen an Schnittstellen wie Bluetooth oder W-Lan aufrecht zu erhalten, wird der Energieverbrauch des Endgeräts mitunter enorm angehoben.

3 Forwarding-Protokolle

Wegfindung stellt in opportunistischen Netzen die wohl größte Herausforderung dar. Betrachtet man die Route einer Nachricht, als Pfad entlang von Routerknoten von einem Startknoten (Sender) zu einem Zielknoten (Empfänger), so lässt sich für Netzwerke wie

z.B. dem Internet mit einer vergleichsweise hohen Wahrscheinlichkeit sagen, entlang welcher Knoten Nachrichten verschickt werden müssen, sodass sie ihr Ziel erreichen. Dies wird unter anderem mithilfe von Routingtabellen realisiert.

OppNets hingegen arbeiten mit der Annahme, dass theoretisch alle Knoten des Netzwerks mobil sein können, wodurch ein Einsatz von klassischen Routingtabellen in vielen Fällen nicht realisierbar ist. Daher verwenden OppNets meist Verfahren, welche den nächsten Hop mit einer gewissen Wahrscheinlichkeit als zielführend bewerten. Manche Forwarding Strategien verwenden *flooding*-Ansätze um die Wahrscheinlichkeit des Erreichens des Ziel zu erhöhen. Andere Strategien basieren auf Mustererkennung von Meta-Informationen wie z.B. vergangener Aufenthaltsorte, um die Wahrscheinlichkeit eines zielführenden nächsten Knotens zu schätzen.

Auf den Begriff *Routing* wird hier absichtlich verzichtet, da dieser impliziert, dass eine konkrete Route entlang von Knoten beim erstmaligen Versenden einer Nachricht bekannt ist. Stattdessen werden Protokolle opportunistischer Netze allgemein als Forwarding-Protokolle bezeichnet ([2, p.10]).

Nach [2] werden Protokolle in drei Kategorien unterteilt:

ohne Kontext

Kontextlose Protokolle wählen den nächsten Hop ohne auf Berechnung von Meta-Informationen zurückzugreifen. Diese Protokolle arbeiten meist mit Zufallskomponenten.

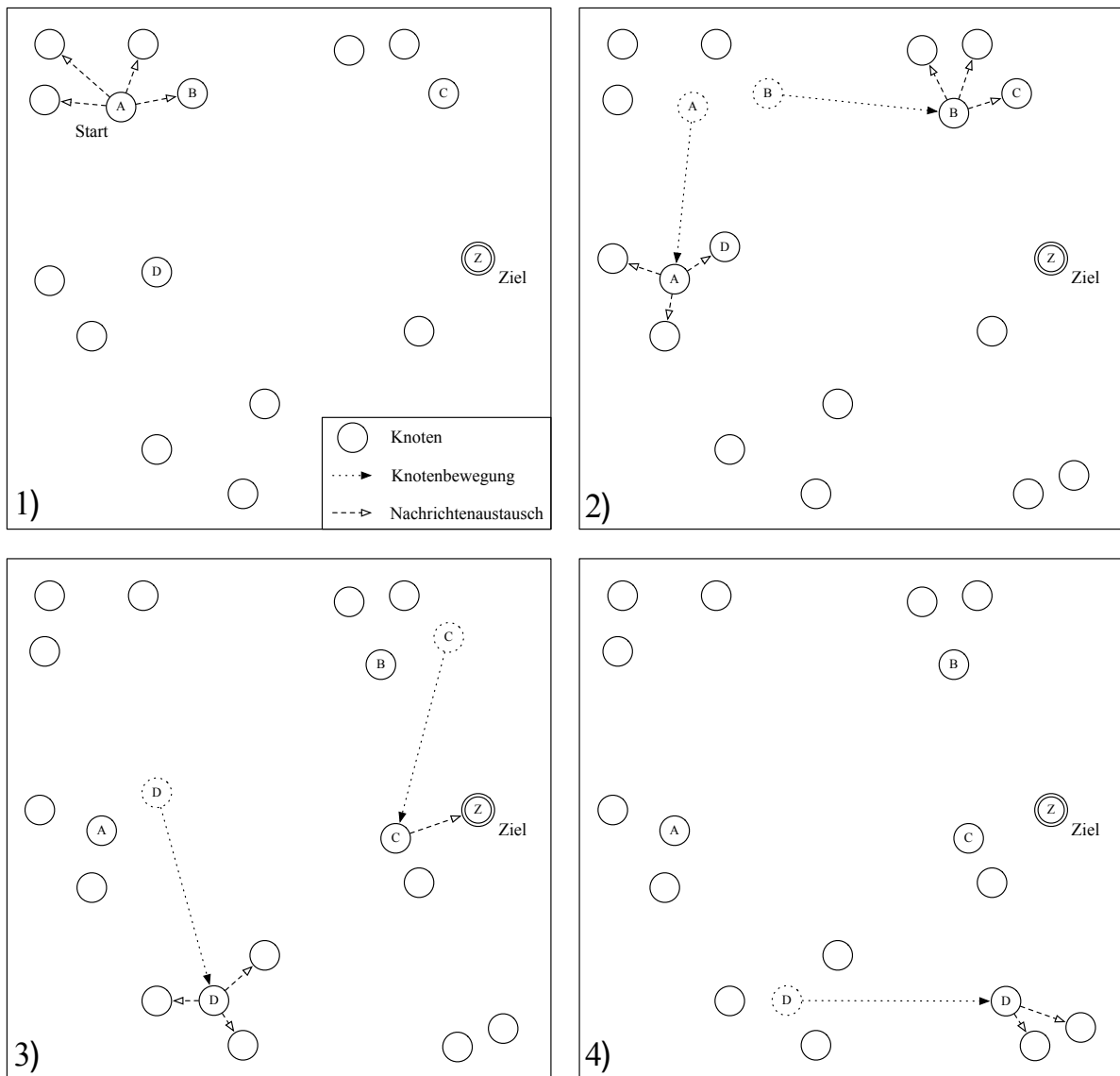
partieller Kontext

Protokolle mit partiellem Kontext beziehen relativ einfach erstellbare Netzwerkinformationen wie Kontakthistorien mit in die Wegfindung ein.

voller Kontext

Protokolle mit vollem Kontext stellen ein komplexes Modell der Knotennachbarschaften auf, sodass daraus Schlüsse gezogen werden können, welcher Knoten sich besser als potenzieller nächster Hop eignet als andere.

Im Folgenden wird zu jeder dieser Kategorien jeweils ein Beispiel gegeben und erklärt.



Hier ist zu sehen, dass der Ursprungsknoten A niemals in die Nähe von seinem gewünschten Ziel gelangt. Zusätzlich kommt es zu keiner durchgehenden Verbindung, denn während sich die Knoten fortbewegen, muss es zu keinem Kontakt zu anderen Knoten kommen. Es ist anzumerken, dass zur Vereinfachung der Darstellung Kontakte zu anderen Knoten nur in Gruppen stattfinden und sich lediglich die gekennzeichneten Knoten bewegen. Dies ist jedoch keine Bedingung für Epidemic-Forwarding. Außerdem ist ein Nachrichtenaustausch in der Abbildung immer nur einseitig dargestellt. Dies ist nur der

Fall, sofern die Gegenseite überhaupt keine Nachrichten zu versenden hat.

Jeder Knoten besitzt einen Buffer, welcher sowohl eigene Nachrichten, als auch Nachrichten anderer Knoten speichert. Kommt es zu einem Kontakt, werden beide Buffer verglichen und es findet ein Austausch der Nachrichten statt, welche dem jeweils anderen Knoten fehlen. Um zu verhindern, dass Nachrichten, welche evtl. bereits erfolgreich übertragen wurden, sich in anderen Teilen des Netzwerks weiter verbreiten, können ACK-Requests oder Hop-Counts implementiert werden. Beispielsweise ist in Abbildung 1 die Nachricht von Knoten A bereits erfolgreich an Z übertragen, während sie jedoch von Knoten D weiter verteilt wird. Durch die Implementation eines Hop-Counts (vergleichbar mit TTL in IP) würden Nachrichten aus dem Buffer entfernt, sobald sie eine gewisse Anzahl an Übertragungen überschritten haben. Mit einem ACK-Request, welcher vom Zielknoten in das Netzwerk gegeben wird, würden alle Knoten, welche diesen Request erhalten, die ursprüngliche Anfrage aus ihrem Buffer entfernen. Zusätzlich bekommt der Sender (falls er den ACK erhält) die Information, dass seine Nachricht erfolgreich übermittelt wurde.

Ein Vorteil dieses Verfahrens ist die hohe Wahrscheinlichkeit, mit der Nachrichten erfolgreich an ihr Ziel gebracht werden, denn je mehr Knoten die Nachricht verbreiten, desto mehr Knoten werden damit erreicht. Dies resultiert jedoch in einem der größten Nachteile dieses Verfahrens: Es findet mit jedem Kontakt ein Nachrichtenaustausch statt, was dazu führt, dass die in der Regel limitierte Speicherkapazität überschritten und ein weiteres Aufnehmen von Informationen somit unmöglich wird. Durch die häufigen Datenübertragungen kristallisiert sich dieses Protokoll in manchen Anwendungen als wenig praktikabel heraus. Häufige Datentransfers sorgen neben Speicherlimitation für hohen Energieverbrauch.

Das in Abschnitt 2.2 vorgestellte Haggle Project verwendet Epidemic-Forwarding insofern, als dass Nachrichten an *alle* User geschickt werden, welche übereinstimmende Interessen besitzen. Hierbei ist jedoch zu bemerken, dass Sender und Empfänger der Daten die beiden im Kontakt beteiligten Knoten sind.

3.2 Partieller Kontext - PROPHET

PROPHET [6] steht für *Probabilistic Routing Protocol using History of Encounters and Transitivity* und der Kern dieses Forwarding-Protokolls ist, Nachrichten nur an jene Knoten weiter zu leiten, welche mit einer gewissen Wahrscheinlichkeit die Nachricht bis zum Ziel bringen werden. Dieses Protokoll basiert auf der Annahme, dass Bewegungsmuster von Knoten nicht randomisiert sind, sondern eine gewisse Struktur aufweisen. Struktur bedeutet, dass Knoten, welche sich oft gesehen haben, sich auch in Zukunft oft sehen werden.

Wie in [6] beschrieben, besitzt jeder Knoten A eine Tabelle, in der für alle anderen Knoten B ein Wert $0 \leq p(A, B) \leq 1$ eingetragen ist, welcher die Wahrscheinlichkeit einer erfolgreichen Übertragung von A nach B angibt. Dieser Wert wird *delivery predictability* genannt.

In den nächsten beiden Kapiteln wird zunächst auf das Aktualisieren der Tabelle (Kapitel 3.2.1) und anschließend auf den generellen Ablauf des Protokolls eingegangen (Kapitel 3.2.2).

3.2.1 Tabellen-Update

Bei jedem Kontakt werden die Wahrscheinlichkeitstabellen ausgetauscht und es findet ein Update der Werte statt (die folgenden Gleichungen sind aus [6] entnommen). Zunächst wird der aktuelle Kontakt zwischen den Knoten A und B mithilfe des Terms (1) bewertet. Dazu wird in der Tabelle die Wahrscheinlichkeit, dass dieser Kontakt erneut auftritt

erhöht. $p(A, B)_{old}$ stellt den Wert in der Tabelle dar, welcher vor dem Kontakt existierte und $P_{init} \in (0, 1]$ ist eine Konstante, welche die positive Auswirkung auf die Wahrscheinlichkeit eines erneuten Treffens reguliert. Wie $p(A, B)_{old}$ initial gewählt wird hängt vom Szenario ab.

$$p(A, B) = p(A, B)_{old} + (1 - p(A, B)_{old}) \times P_{init} \quad (1)$$

Anschließend wird ein Alterungsprozess simuliert, indem jede Wahrscheinlichkeit reduziert wird, abhängig davon wie lange ein Kontakt zurückliegt. In Term (2) steht $\lambda \in (0, 1)$ für eine Konstante, welche diesen Prozess reguliert und k für die Zeiteinheiten seit der letzten Alterung.

$$p(A, B) = p(A, B)_{old} \times \lambda^k \quad (2)$$

In der letzten Phase des Updates werden die Daten des im Kontakt stehenden Knotens verwendet um transitive Beziehungen zu Knoten miteinzubeziehen. Das bedeutet, dass $p(A, C)$ abhängig von den Werten $p(A, B)$ und $p(B, C)$ geändert wird. In Term (3) stellt $\beta \in [0, 1]$ eine Konstante dar, die den Einfluss der Transitivität reguliert.

$$p(A, C) = p(A, C)_{old} + (1 - P(A, C)_{old}) \times P(A, B) \times P(B, C) \times \beta \quad (3)$$

Das Festlegen jeder Konstante ist stark von dem jeweiligen Szenario, in dem das Protokoll eingesetzt wird, abhängig.

3.2.2 Forwarding-Prozess

Nach dem Update der jeweiligen Tabellen entscheidet das Protokoll, ob und welche Nachrichten an den anderen Knoten übertragen werden sollen. Eine Möglichkeit wäre eine Grenze $g \in (0, 1)$ festzulegen, welche eine Nachrichtenübertragung herbeiführt, sobald $p(A, B) > g$. Hier ist die Schwierigkeit g korrekt festzulegen, da ohne weiteres keine Aussage über die im Netz vorkommenden p -Werte gemacht werden kann. Auch hier ist zu betonen, dass die jeweiligen Einstellungen stark von dem jeweiligen Anwendungsgebiet abhängen und dementsprechend gewählt werden müssen.

3.3 Voller Kontext - SimBet

Der Name des Protokolls *SimBet* [7] ist eine Zusammensetzung der Worte *similarity* und *betweenness*. Diese Ausdrücke stehen für die Metrik, die verwendet wird um eine Forwarding-Entscheidung zu treffen. Die Idee hinter SimBet ähnelt der von PROPHET, indem angenommen wird, dass Nutzer in einem Netzwerk eine gewisse Sozialstruktur aufweisen. Im Falle von SimBet bedeutet dies, dass sich Knoten vornehmlich in sogenannten *Communities* aufhalten. Das sind Gruppen von Knoten, welche häufig miteinander verbunden sind. In Abbildung 2 gibt es zwei Communities, eine bestehend aus $\{1, 2, 3, 4\}$ und eine andere bestehend aus $\{5, 6, 7\}$.

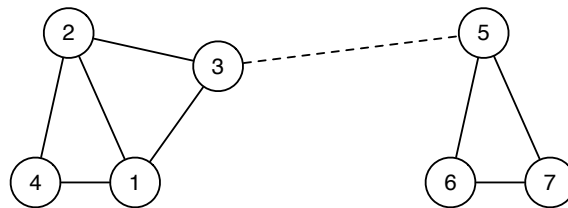


Abbildung 2: Beispiel einer Beziehungskonstellation in SimBet

Die zum Forwarding verwendete Metrik $SimBetUtil_n(d)$, die ein Maß dafür gibt, wie gut sich ein Knoten n als nächster Hop zum Zielknoten d eignet, besteht aus dem *Betweenness*- und dem *Similarity-Index*. Der Betweenness-Index sagt aus, wie stark ein Knoten die eigene Nachbarschaft an andere Nachbarschaften anknüpft. Der Similarity-Index gibt Auskunft darüber, wie ähnlich die Nachbarschaft eines Knotens zu der des Zielknotens ist. Der Ablauf lässt sich wie folgt darstellen: Zunächst versucht SimBet eine Nachricht mithilfe des Betweenness-Index in die Community des Zielknotens weiterzuleiten. Anschließend wird innerhalb der Community die Nachricht mithilfe des Similarity-Index zum Zielknoten weitergeleitet. Mathematisch lässt sich diese Vorschrift in (4) ablesen. Die Konstanten α und β regulieren den Einfluss, den der Similarity- bzw. der Betweenness-Index auf die Berechnung hat.

$$SimBetUtil_n(d) = \alpha Sim_n(d) + \beta Bet_n(d) \quad \text{mit} \quad 1 = \alpha + \beta \quad (4)$$

3.3.1 Betweenness-Index

Jeder Knoten besitzt eine Matrix, in der die direkten Nachbarn eingetragen sind. In (5) sind zwei Matrizen exemplarisch aufgeführt, welche die Nachbarschaftsstrukturen der Knoten 2 und 3 aus Abbildung 2 darstellen. Wenn $a_{i,j} = 1$, existiert eine direkte Verbindung zwischen den Knoten i und j .

$$N_{k_2} = \begin{matrix} & k_1 & k_2 & k_3 & k_4 \\ \begin{matrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix} \quad N_{k_3} = \begin{matrix} & k_1 & k_2 & k_3 & k_5 \\ \begin{matrix} k_1 \\ k_2 \\ k_3 \\ k_5 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix} \quad (5)$$

Mithilfe des Betweenness-Index kann das Protokoll entscheiden, wie gut eine Nachricht zwischen den Communities transportiert werden kann. Betrachte man den Knoten 3 aus Abbildung 2, so ist für die Berechnung des Index wichtig, wieviele Verbindungen c Knoten 3 zu anderen Knoten hat, die Knoten 1 und 2 nicht haben. Je größer also c , desto größer ist auch der Betweenness-Index.

Die genauen mathematischen Formeln sind in [7, p. 35f] nachzulesen.

3.3.2 Similarity-Index

Die Berechnung des Similarity-Index wird durch Vergleich der Einträge einer Reihe in der Matrix erstellt. Betrachte man die Matrix N_{k_3} in (5), so hat k_1 einen gleichen Eintrag, k_2 einen gleichen Eintrag und k_5 keinen gleichen Eintrag mit k_3 . Am Beispiel betrachtet bedeutet dies, dass k_3 einen Knoten mit k_1 , einen Knoten mit k_2 und keinen Knoten mit k_5 gemein hat. Im Forwarding wird diese Größe dazu verwendet um innerhalb der Community eine sinnvolle Route zum Ziel zu finden. Als nächster Hop werden vornehmlich Knoten ausgewählt, welche mit dem Zielknoten eine ähnliche Nachbarschaftsstruktur aufweisen. So kann sichergestellt werden, dass der nächste Hop Kontakt zu einem Knoten hat, welcher dem Ziel ein Stück näher ist.

Auch hier wird für die genauen Berechnungen auf [7, p. 35f] verwiesen.

4 Bewegungsmodelle

Um die Performance eines Protokolls zu evaluieren ist es notwendig dieses zu testen. Da OppNets aus mobilen Knoten bestehen, müssen den Knoten Bewegungsmuster zugeteilt

werden. Aus den Bewegungen werden Daten gewonnen, wie die Anzahl, die Dauer oder die Beteiligten von Kontakten. Die Bewegungsmuster können entweder durch Messungen gesammelt, oder künstlich berechnet worden sein [2].

Messungen von Knotenbewegungen haben den Vorteil, dass sie die realen Bewegungen der Knoten in einem gewissen Szenario wiedergeben. Eine synthetische Generierung der Daten hingegen kann niemals dieselbe Güte an Realität bieten. Auf der anderen Seite ist die Menge an existierenden Messungen begrenzt. Daraus entsteht das Problem, dass nicht alle Messungen passend für das jeweilige Szenario sind. So stellen Messungen von Menschenbewegungen eine schlechte Grundlage zur Auswertung eines OppNets dar, welches für die Kommunikation zwischen Autos verwendet wird. Aufgrund der Vielzahl unterschiedlicher Protokollarten werden daher häufig synthetisch generierte Bewegungsmodelle zur Simulation verwendet.

Im Folgenden werden zwei synthetische Bewegungsmodelle vorgestellt.

4.1 Random Walk Mobility Model

Das *Random Walk Mobility Model* [8] ist ein simples Bewegungsmodell, in der ein Knoten von seiner aktuellen Position randomisiert eine Richtung und eine Geschwindigkeit festlegt, mit der er sich zur nächsten Position bewegt. Dieses Modell ist ein sogenanntes *Entity Mobility Model*, was bedeutet, dass lediglich die Bewegungen eines einzelnen Knotens in Betracht gezogen werden. Die Entfernung, die der Knoten zurücklegt, ist entweder durch eine feste Zeitdauer oder durch eine feste Entfernung festgelegt. Dieser Prozess findet an jeder erreichten Position erneut statt. Sollten Knoten an das Ende der Simulationsfläche kommen, werden sie wieder zurück reflektiert. Regulieren lässt sich das Modell in der Anzahl seiner Knoten, der Größe der Fläche, auf der sich die Knoten bewegen, dem Geschwindigkeitsbereich und in den zugelassenen Winkeln.

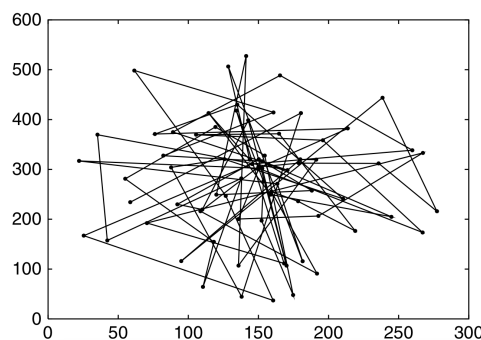


Abbildung 3: Zeigt das Random Walk Bewegungsmuster eines Knotens. **Quelle:** [8]

In Abbildung 3 ist das Random Walk Bewegungsmuster eines Knotens dargestellt. Nach jeder Geschwindigkeits- und Richtungsneuberechnung bewegt sich der Knoten für 60s lang in eine Richtung. Es ist zu erkennen, dass die Bewegungsrichtungen teilweise extrem spitze Winkel aufweisen. Dies liegt an der zufälligen Auswahl der Richtung, welche keinen Bezug zu den bereits zurückgelegten Wegen haben.

4.2 Pursue Mobility Model

Wie in Abschnitt 3.3 zu sehen ist, arbeiten Protokolle wie SimBet auf Basis des Gruppenverhaltens der Knoten. Dies muss in den Simulationen berücksichtigt werden und erfordert somit Bewegungsmodelle, die Gruppendynamik beachten.

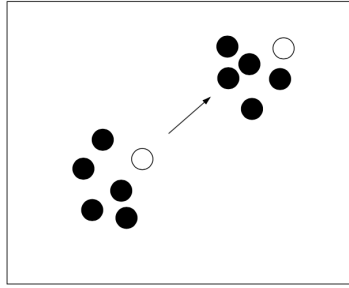


Abbildung 4: Zeigt die Gruppenbewegung des Pursue Mobility Model. **Quelle:** [9]

Das *Pursue Mobility Model* [9] ist ein *Group Mobility Model*, da es Gruppenbewegungen modelliert. Die Idee hinter diesem Modell ist, dass eine Gruppe von Knoten ein Ziel verfolgt. Dabei stellt das Ziel meist ein anderer Knoten dar. In Abbildung 4 ist zu sehen, wie eine Gruppe von schwarzen Knoten einen weißen Knoten verfolgt. Neben dem Verfolgen des weißen Knotens haben die schwarzen Knoten eine zufällige Komponente mit der sie sich in einem gewissen Radius selber bewegen dürfen. In Gleichung (6) wird diese Komponente durch *random_vector* ausgedrückt, welcher z.B. mithilfe des Random Walk Mobility Models erstellt werden kann.

$$\text{newPosition} = \text{oldPosition} + \text{acceleration}(\text{target} - \text{oldPosition}) + \text{randomVector} \quad (6)$$

Die Funktion $\text{acceleration}(\text{target} - \text{oldPosition})$ berechnet aus dem verfolgten Ziel und der aktuellen Position die neue Richtung, in die gegangen werden soll.

5 Zusammenfassung

Im Rahmen dieser Ausarbeitung sollen die Grundlagen des Bereichs der opportunistischen Netze vermittelt werden. Zunächst wurde der Begriff der opportunistischen Netze erläutert und in einen Kontext gebracht. Anschließend wurden Beispiele gegeben in denen OppNets verwendet vorkommen. Danach wurde ein Einblick in verschiedene Forwarding-Strategien gegeben. Daraufhin wurden Bewegungsmodelle vorgestellt, welche zur Messung der Performance von OppNets verwendet werden können.

Literatur

- [1] K. Zeng, W. Lou, and M. Li, *Multihop Wireless Networks: Opportunistic Routing*, vol. 21. John Wiley & Sons, 2011.
- [2] V. F. Mota, F. D. Cunha, D. F. Macedo, J. M. Nogueira, and A. A. Loureiro, “Protocols, mobility models and tools in opportunistic networks: A survey,” *Computer Communications*, vol. 48, no. 0, pp. 5 – 19, 2014. Opportunistic networks.
- [3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” in *ACM Sigplan Notices*, vol. 37, pp. 96–107, ACM, 2002.
- [4] “Haggle project.” <https://code.google.com/p/haggle/>. Accessed: 2014-11-13.
- [5] A. Vahdat, D. Becker, *et al.*, “Epidemic routing for partially connected ad hoc networks,” tech. rep., Technical Report CS-200006, Duke University, 2000.

- [6] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.
- [7] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 32–40, ACM, 2007.
- [8] R. R. Roy, “Random walk mobility,” in *Handbook of Mobile Ad Hoc Networks for Mobility Models*, pp. 35–63, Springer, 2011.
- [9] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.