

Positionsbasiertes Routing in mobilen Ad-hoc-Netzwerken

VIROR-Teleseminar WS 2001/2002

Ubiquitous Computing

Thomas Lang

Universität Mannheim

Februar 2002

Inhaltsverzeichnis

1	Einführung	2
2	Mobile Ad-hoc-Netzwerke	3
3	Vorgehensweise der Routingalgorithmen	4
4	Positionsdienste	4
4.1	Homezone	5
5	Die Greedy Forwarding Regel	5
5.1	Eigenschaften des Greedy Forwarding	6
5.1.1	Das Beaconing Protocol	6
5.1.2	Vorteile des Greedy Forwarding	7
5.1.3	Nachteil des Greedy Forwarding	7
5.2	Die Rechte-Hand-Regel	8
5.2.1	Die “No-Crossing”-Heuristik	9
5.2.2	Planare Graphen	10
6	Greedy Perimeter Stateless Routing (GPSR)	11
6.1	Simulationsergebnisse	13
6.1.1	Dynamic Source Routing	13
6.1.2	Simulationsumgebung	14
6.1.3	Erfolgsquote der Paketzustellung	14
7	Zukünftige Arbeit	15
8	Abschlußbemerkungen	16

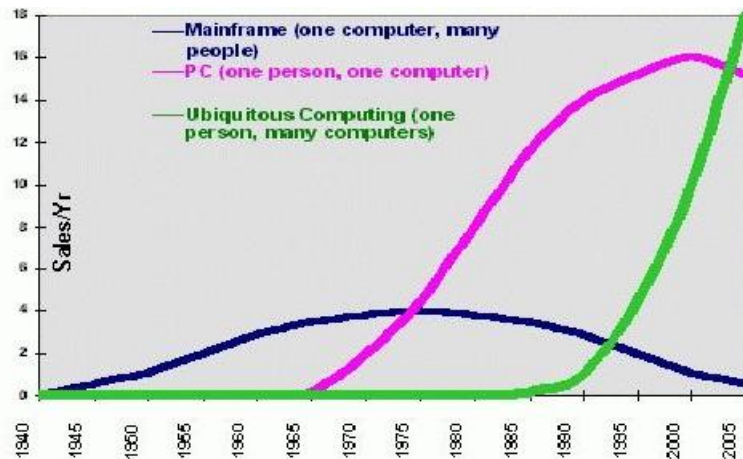


Abbildung 1: Trends der elektronischen Datenverarbeitung (Quelle: [7])

1 Einführung

Der Begriff des “Ubiquitous Computing” wurde 1988 von Marc Weiser, seinerzeit leitender Wissenschaftler am Forschungszentrum XEROX in Palo Alto, geprägt. Seine Vision der zukünftigen Entwicklung des Computers unterscheidet sich völlig vom bisherigen Verständnis. Das “Ubiquitous Computing” stellt nach Weiser die momentan beginnende dritte Generation der elektronischen Datenverarbeitung dar.

Wie in Abbildung 1 dargestellt existierten in der ersten Generation Großrechner, welche eine Vielzahl von Benutzern gleichzeitig bedienten. In der gegenwärtigen PC-Ära ist jedem Benutzer ein eigener persönlicher Computer zugeordnet. Dabei stellt sich das Problem, daß die Arbeitsweise am Desktop nicht der natürlichen Arbeitsweise des Menschen entspricht. So widerspricht die räumliche Fokussierung bei der Arbeit mit dem PC auf nur ein Gerät der räumlichen Ausdifferenzierung, die der Mensch von Natur aus beim Arbeiten vornimmt. Die Epoche des “Ubiquitous Computing” soll nach Weisers Vision ein Zeitalter der “ruhigen” Technologie werden, in der die Computergeräte in den Hintergrund treten.

Die Entwicklung der dritten Generation, wie Weiser sie erwartet, ist vergleichbar mit der Entwicklung der Schrift. Zuerst wurden Schriftzeichen festgelegt und die Technik des Schreibens wurde entwickelt und verbreitet. Die Technik des Schreibens muß erlernt werden, sobald man jedoch das Lesen und Schreiben beherrscht, tritt die Technik selbst in den Hintergrund. Beim Erkennen der Bedeutung der Zeichen muß der Mensch sich jedoch nicht erst

auf die Technik besinnen, sondern die Informationen werden sofort erkannt und verarbeitet.

Die Entwicklung der Schrift zeigt, daß erst die unbewußte Benutzung der Technik es ermöglicht, sich verstärkt auf die Inhalte zu konzentrieren.

Die meisten Computer der dritten Generation werden für den Menschen unsichtbar sein. Schon heute existieren in Lichtschaltern, Thermostaten und Öfen unterstützende Computerchips. Diese und andere Geräte sollen künftig mit einem allgegenwärtigen Netzwerk verbunden werden. Nahezu jedes Objekt wird demnach entweder einen Computer oder ein "Tab", eine elektronische "Post-it"-Nachricht, enthalten. Durch die Vernetzung der Objekte können Informationen schneller und unkomplizierter weitergeleitet und transparenter gemacht werden. Dies bringt die Möglichkeit der Automatisierung von Prozessen mit sich. Es wird dann einfach sein, Informationen wie beispielsweise "Wer hat dieses Kleid hergestellt? Gibt es weitere Kleider dieses Herstellers in diesem Geschäft?" zu erhalten.

Diese Entwicklung wird nach Weiser letztlich dazu führen, daß die Maschinen sich in die Umwelt des Menschen einfügen, anstatt ihnen ihre Strukturen aufzuzwingen.

2 Mobile Ad-hoc-Netzwerke

Die dritte Generation der Datenverarbeitung führt zu einer zunehmenden Verbreitung von drahtlosen Endgeräten und sich selbst organisierenden Netzwerken. Diese mobilen Ad-hoc-Netzwerke benötigen keinerlei Infrastruktur um zu arbeiten. Die Endgeräte können miteinander kommunizieren, sobald sie sich im gegenseitigen Übertragungsbereich befinden, oder weitere Endgeräte zwischen ihnen die Daten weiterleiten. Dadurch wird eine große Flexibilität ermöglicht, wie sie für spontane Treffen, einen schnellen Ersatz einer zerstörten Infrastruktur oder einer Kommunikation fernab jeglicher Infrastruktur benötigt wird.

Diese Ad-hoc-Netzwerke lassen sich in *statische* und *mobile* Netzwerke unterteilen, welche sich in der Mobilität der einzelnen Knoten unterscheiden. Während sich bei statischen Ad-hoc-Netzwerken die Positionen eines Knoten, nachdem er Mitglied des Netzwerkes wurde, nicht mehr ändert, können sich die Knoten in einem mobilen Ad-hoc-Netzwerk beliebig bewegen.

Die häufigen Positionsänderungen der Knoten in diesen mobilen Netzwerken stellt das Weiterleiten von Datenpaketen zwischen einem Quell- und einem Zielknoten vor große Herausforderungen. In dieser Seminararbeit soll ein Lösungsansatz für dieses Problem vorgestellt werden.

3 Vorgehensweise der Routingalgorithmen

Für Ad-hoc-Netzwerke können die bisher existierenden Algorithmen in zwei Vorgehensweisen unterteilt werden. Es existieren *topologiebasierte* Ansätze, in denen die Knoten untereinander Informationen über den Aufbau des Netzes austauschen, um mit deren Hilfe Weiterleitungsentscheidungen zu treffen. Aus dem Festnetzbereich sind einige Algorithmen bekannt (Link State Routing, Distance Vector Routing), die dieses Verfahren erfolgreich anwenden. Wegen ihres *proaktiven* Vorgehens, welches die Wege durch das Netzwerk konstant aufrecht erhält, egal ob sie zur Zeit verwendet werden oder nicht, skalieren diese Algorithmen in mobilen Ad-hoc-Netzwerken aber nur sehr schlecht. Der hohe Aufwand für das Aufrechterhalten dieser unbenutzten Pfade erreicht leicht einen Umfang, welcher die Leistungsfähigkeit der mobilen Endgeräte bei weitem übersteigt.

Aufgrund dieser Beobachtungen wurden *reaktive* Vorgehensweisen entwickelt, welche nur Wege, die im Moment benutzt werden, speichern und aufrechterhalten. Obwohl diese Protokolle momentan die leistungsfähigste Variante darstellen, haben sie doch grundlegende Mängel. Da vor jeder Kommunikation zuerst ein Weg durch das Netz bestimmt werden muß, kommt es zu langen Verzögerungszeiten bis die ersten Daten fließen können. Zusätzlich müssen diese Wege für die Dauer der Kommunikation gepflegt werden, was bei Netzwerken mit hoher Änderungsrate der Topologie zu einer hohen Netzwerkbelastung führt. Nachteilig wirkt sich auch der häufige Verlust von Paketen, die sich bei Topologieänderungen, welche zu einer veränderten Wegewahl führen, auf dem "alten" Weg befinden, aus.

Um diese Schwächen zu beseitigen, wurde ein Ansatz entwickelt, bei dem die Informationen der geographischen Positionen der Knoten, benutzt werden. Dieses Vorgehen setzt zusätzlich die Existenz eines *Positionsdienstes* welchem die Positionen aller Knoten im Netzwerk bekannt sind, voraus.

4 Positionsdienste

Um jeden Knoten in einem mobilen Ad-hoc-Netzwerk identifizieren zu können, setzt sich die Adresse aus einem eindeutigen Identifizierer und der aktuellen geographischen Position zusammen. Diese geographische Position muß von jedem Teilnehmer selbst ermittelt werden, zum Beispiel mit GPS. Damit die Adresse von den anderen Netzwerkknoten ermittelt werden kann, registrieren die Knoten bei einem Positionsdienst und pflegen diesen Eintrag bei jeder Positionsänderung. Wenn ein Knoten mit einem anderen Knoten kommunizieren möchte und ihm die gewünschten Adresse des Kommuni-

kationspartners nicht bekannt ist, kann er diese von einem Positionsdienst anfordern. Beispiele für solche dezentral organisierten Positionsdienste sind:

- Homezone
- DREAM (Distance Routing Effect Algorithm for Mobility)
- Grid Location Service
- Quorum Based Location Service

Der nächste Abschnitt stellt einen kurzen Einblick in die Funktionsweise von *Homezone* dar.

4.1 Homezone

Homezone basiert auf der Idee der virtuellen Heimatregion eines jeden Knotens, in der die aktuelle Position gespeichert wird. Mit Hilfe einer wohlbekannten Funktion kann der eindeutige Identifizierer jedes Knotens auf eine Region abgebildet werden. Die in dieser Region befindlichen Knoten haben die Aufgabe, die Positionen der auf sie abgebildeten Knoten zu speichern und bei eventuellen Positionsänderungen zu pflegen. Wenn sich ein Knoten bewegt, flutet er seinen neuen Standort in der virtuellen Heimatregion.

Dem Problem, daß sich in einer Region nicht zwingend Knoten befinden müssen, kann begegnet werden, indem sich die Regionen dynamisch an die jeweilige Knotendichte anpassen oder die Adressen der Knoten auf mehrere Regionen abbildet werden.

5 Die Greedy Forwarding Regel

In diesem Kapitel sollen die Grundlagen der positionsbasierten Routingalgorithmen erläutert werden. Als Grundidee dient die *Greedy Forwarding* Regel. Dabei wird in die Datenpakete die geographische Position des Ziels eingetragen. Um Entscheidungen über den Weg eines Paketes treffen zu können, besitzt jeder Knoten eine Nachbarschaftstabelle, in der die Positionen der in Funkreichweite befindlichen Nachbarn abgelegt sind. Mit Hilfe dieser Tabelle und der Zielposition des Paketes, kann jeder Knoten lokal entscheiden, an welchen Nachbar das Datenpaket weitergeleitet werden muß, um den größten Fortschritt in Richtung Zielknoten zu erlangen. Das Datenpaket wird dann so lange von den Knoten im Netzwerk an diese Nachbarn weitergeleitet bis der Zielknoten erreicht ist.

Die Vorgehensweise dieser Regel ist in Abbildung 2 dargestellt. Hier erhält der

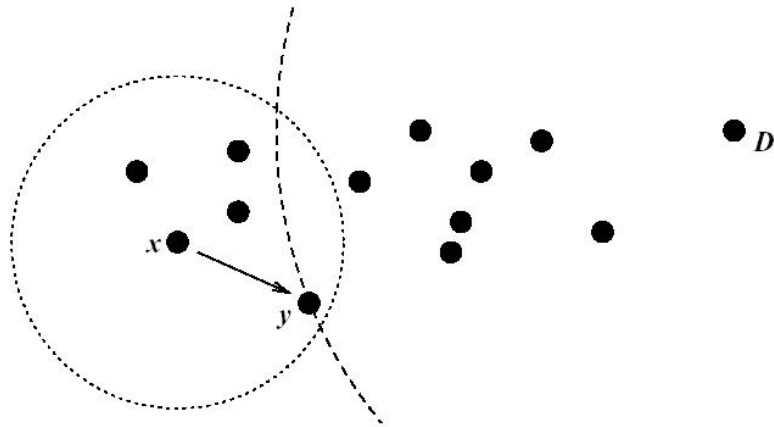


Abbildung 2: Beispiel für Greedy Forwarding (Quelle: [2])

Knoten x ein Datenpaket für den Zielknoten D . Der gepunktete Kreis um x stellt die maximale Funkreichweite von x dar, und die gestrichelte Linie durch den Knoten y entspricht der Entfernung des Knoten y vom Zielknoten. Da sich in diesem Beispiel in der Reichweite von x kein Knoten befindet, welcher eine kürzere Entfernung von D hat als y , leitet x das Datenpaket an y weiter.

5.1 Eigenschaften des Greedy Forwarding

5.1.1 Das Beaconing Protocol

Da jedem Knoten die Positionen seiner in Reichweite befindlichen Nachbarn bekannt sein muß, existiert ein *Beaconing Protocol*. Dabei übertragen die Netzwerkknoten periodisch mittels eines 1-Hop-Broadcasts ein *Beacon*, welches den eindeutigen Identifizierer und die aktuelle Position beinhaltet. Die richtige Wahl der Länge dieser Intervalle hängt dabei von der Mobilität der Knoten im Netzwerk ab. Um zu verhindern, daß alle Knoten gleichzeitig ihre Pakete senden, wird die Übertragung noch im Bereich der Länge des halben Sendeintervalls zufällig verzögert. Falls über die Länge eines *Timeout*-Intervalls T keine *Beacons* mehr von einem Nachbarn empfangen wurden, wird angenommen, daß sich der Nachbar nicht mehr innerhalb der Funkreichweite befindet, und er wird aus der Nachbarschaftstabelle entfernt.

Es ist auch ein *reaktives* Vorgehen des *Beaconing Protocol* denkbar. Dabei werden die Nachbarschaftstabellen nur aktualisiert, wenn ein Sendewunsch vorliegt. Da aber bei der *proaktiven* Übermittlung der Positionsinformationen nicht das gesamte Netzwerk geflutet wird, wird dies als nicht notwendig

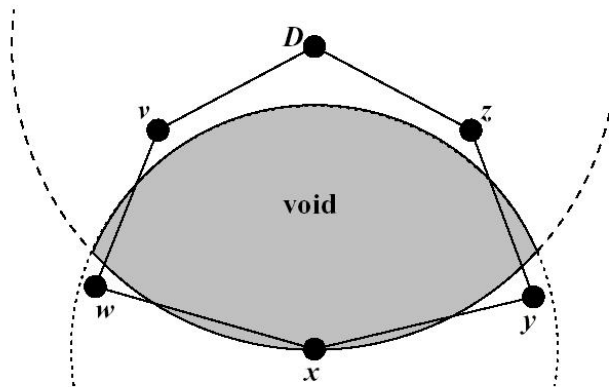


Abbildung 3: Fehlschlagen des Greedy Forwarding (Quelle: [2])

betrachtet.

5.1.2 Vorteile des Greedy Forwarding

Der große Vorteil des *Greedy Forwarding* besteht darin, daß nur das lokale Wissen über die jeweils in direkter Nachbarschaft befindlichen Knoten benötigt wird, um Weiterleitungsentscheidungen zu treffen. Dies hat zur Folge, daß nur eine geringe Anzahl von Informationen, welche nicht von der Gesamtzahl der Zielknoten abhängig ist, in den Netzwerkknoten zwischengespeichert werden muß. Da die Bewegungen von weiter entfernten Knoten nicht beachtet werden, ist der Algorithmus robust gegenüber entfernten Topologieänderungen. Zusätzlich profitiert der Algorithmus auch von dicht besetzten Netzwerken, da sich der gewählte Weg bei höherer Knotendichte der direkten Linie zwischen dem Start- und dem Zielknoten annähert.

5.1.3 Nachteil des Greedy Forwarding

Die Vorteile, die das “Greedy Forwarding” aufgrund der *lokalen* Sichtweise bietet, verursachen aber das folgende Problem:

Es existieren Topologien, in denen der einzige Weg zum Ziel es erfordert, daß Datenpakete an Knoten, die vom Ziel weiter entfernt liegen als der aktuelle Knoten, weitergeleitet werden müssen. Ein einfaches Beispiel für solch eine Topologie zeigt die Abbildung 3.

In diesem Schaubild entspricht der gestrichelte Kreis um D der Entfernung von x zu D. Der gepunktete Kreis kennzeichnet die Funkreichweite von x. Zwischen x und D existiert im grau schattierten Bereich ein “Funkloch”, da x

näher an D ist als alle seine Nachbarn. Obwohl die Wege $(x \rightarrow w \rightarrow v \rightarrow D)$ und $(x \rightarrow y \rightarrow z \rightarrow D)$ existieren, wird x diese nicht wählen, da x ein lokales Extremum der Nähe zu D darstellt.

5.2 Die Rechte-Hand-Regel

Um die Datenpakete von diesen lokalen Extrema zu retten, muß ein Weg um das “Funkloch” gefunden werden. Aus der Graphentheorie ist die Rechte-

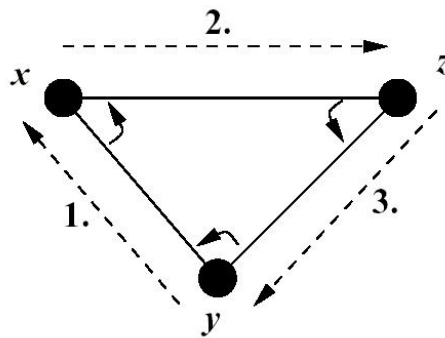


Abbildung 4: Vorgehensweise der Rechte-Hand-Regel (Quelle: [2])

Hand-Regel bekannt, welche es ermöglicht, einen Graphen, wie in Abbildung 4 dargestellt, zu durchqueren. Die Regel ist folgendermaßen definiert:

Wenn man von einem Knoten y aus den Knoten x erreicht hat, so führt der weitere Weg über die Kante, die sich gegen den Uhrzeigersinn zu der Kante (x,y) als nächstes befindet.

Falls zwei Nachbarn sich auf derselben Kanten befinden, so wird der geographisch nähere Nachbar ausgewählt. Dieser Weg durchquert die inneren Kanten einer **polygonalen Fläche** im Uhrzeigersinn. Der nach dieser Regel definierte Pfad wird als *Perimeter* bezeichnet. Wir sehen im Schaubild 3, daß die Rechte-Hand-Regel zu dem Weg $(x \rightarrow w \rightarrow v \rightarrow D \rightarrow z \rightarrow y \rightarrow x)$ um das “Funkloch” herum führt.

Dieses gewünschte Verhalten wird von dieser Regel aber leider nicht immer erreicht. In Graphen mit sich kreuzenden Kanten, wird ein Weg gewählt, der keiner Umrandung einer abgeschlossenen **polygonalen Fläche** entspricht. Abbildung 4 zeigt einen solchen Graphen. Wir sehen in diesem Schaubild, daß wenn x ein Datenpaket an u schickt, es nach der Rechten-Hand-Regel

den Weg $(x \rightarrow u \rightarrow z \rightarrow w \rightarrow u \rightarrow x)$ nehmen würde. Falls die Kante (w,z) aus dem Graphen entfernt werden würde, würde der Weg die gewünschten Knoten $(x \rightarrow u \rightarrow z \rightarrow v \rightarrow x)$ besuchen. Da die Auswahl der richtigen zu entfernenden Kante nicht trivial ist, werden in den nächsten beiden Kapiteln zwei mögliche Ansätze vorgestellt.

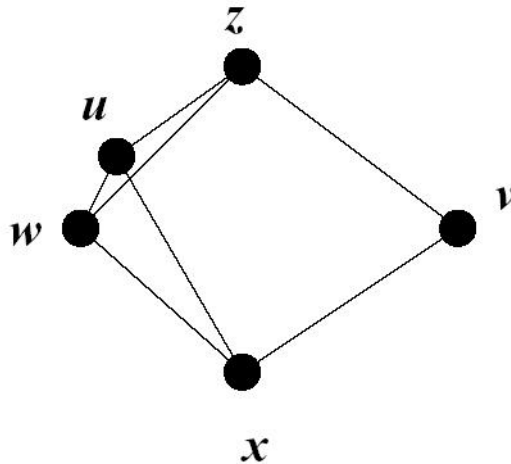


Abbildung 5: Ein Netzwerkgraph mit sich kreuzenden Kanten (Quelle: [2])

5.2.1 Die “No-Crossing”-Heuristik

Bei der “No-Crossing”-Heuristik werden in den Datenpaketen die Kanten des Netzwerkgraphen abgespeichert, welche schon einmal besucht wurden. Falls die nächste, nach der Rechten-Hand-Regel ausgewählte Kante eine dieser abgespeicherten Kanten schneidet, wird die gegen den Uhrzeigersinn nächste Kante ausgewählt.

Dieses Vorgehen führt in Abbildung 5 dazu, daß im Knoten z die Kante (z,w) übergangen wird, und das Paket über die Kante (z,v) weitergeleitet wird. Für dieses Schaubild führt die Heuristik zu dem gewünschten Weg $(x \rightarrow u \rightarrow z \rightarrow v \rightarrow x)$. Da aber bei diesem Vorgehen immer die Kante entfernt wird, die als Zweite erreicht wird, kann es zu Partitionierungen des Netzwerks kommen, bei denen der Zielknoten nicht mehr erreichbar ist. Um dieses Problem zu beseitigen, hat man sich den aus der Graphentheorie bekannten planaren Graphen zugewandt.

5.2.2 Planare Graphen

Graphen, in denen sich keine kreuzenden Kanten befinden, werden als planare Graphen bezeichnet. Es existieren verschiedene Algorithmen, um nicht-planare Graphen zu planarisieren. Ein Beispiel dafür ist der “Relative Neighborhood Graph” der folgendermaßen ermittelt wird:

Eine Kante (u,v) existiert zwischen den beiden Knoten u und v , falls es keinen Knoten w gibt, dessen Kanten (u,w) und (w,v) einen Weg zwischen u und v darstellen, dessen Kanten jeweils kürzer sind als \overline{uv} .

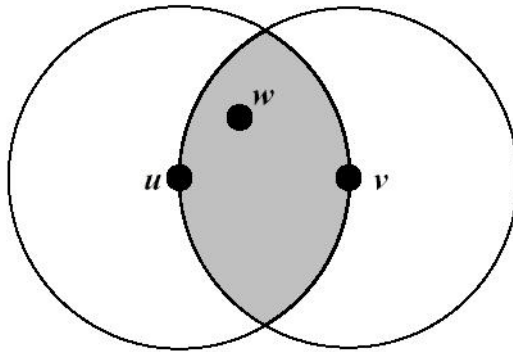


Abbildung 6: Bestimmung des Relative Neighborhood Graph. (Quelle: [2])

In Abbildung 6 darf der grau schattierte Bereich zwischen den Knoten u und v keine weiteren Knoten enthalten. Aus diesem Schaubild ist auch ersichtlich, daß jeder Knoten lokal, nur mit Hilfe des Wissens über seine direkten Nachbarn, entscheiden kann, welche Kanten im planaren Graphen enthalten sind und welche nicht.

Da eine Kante (u,v) nur aus dem Graphen entfernt wird, wenn ein “Zeugenknoten” w innerhalb der Reichweite von u und v existiert, kann es durch diesen Algorithmus nicht zu Partitionierungen des Netzwerkes kommen. Es besteht immer ein alternativer Pfad für entfernte Kanten, und die ursprüngliche Konnektivität bleibt erhalten.

Die Abbildung 7 zeigt einen vollständigen Graphen eines Netzwerkes und den daraus resultierenden planarisierten Relative Neighborhood Graph.

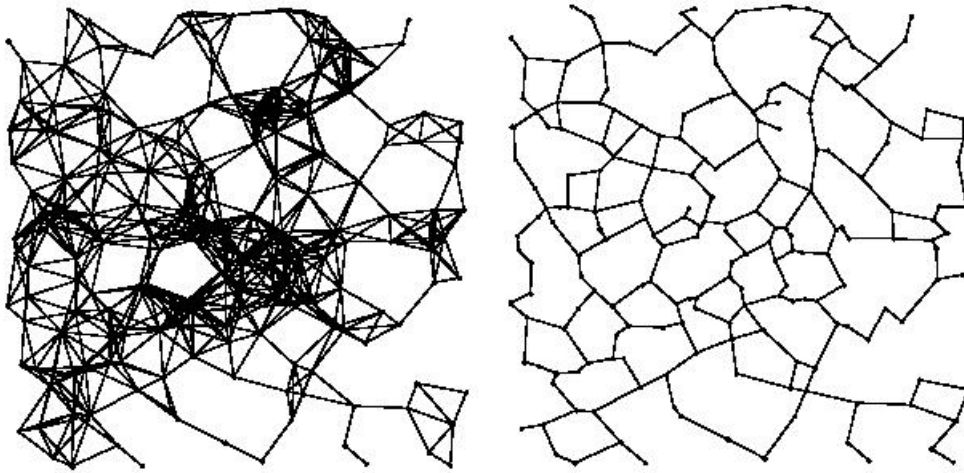


Abbildung 7: Vollständiger Graph eines Funknetzwerkes mit dem resultierende Relative Neighborhood Graph (Quelle: [2])

6 Greedy Perimeter Stateless Routing (GPSR)

Der in diesem Kapitel vorgestellte Algorithmus des *Greedy Perimeter Stateless Routing* wurde von Brad Nelson Karp im Rahmen seiner Doktorarbeit mit dem Titel “Geographic Routing for Wireless Networks” im Oktober 2000 an der Harvard University vorgestellt [2].

GPSR ist eine Kombination von *Greedy Forwarding* auf dem vollständigen Netzwerkgraphen und, falls dies nicht möglich ist, *Perimeter Forwarding* auf dem planarisierten Graphen. Jeder Knoten benötigt dazu eine Nachbarschaftstabelle mit den Positionen seiner Nachbarn. Die zusätzlich benötigten Informationen, die für den *Perimeter Modus* benötigt werden, sind im Paket selbst enthalten. Der jeweilige Modus, in dem ein Datenpaket weitergeleitet werden muß, wird im GPSR-Paketkopf abgespeichert. Jedes Datenpaket wird im *Greedy Modus* erzeugt und, soweit möglich, in diesem Modus weitergeleitet. Falls kein Weiterleiten mit Fortschritt möglich ist, wird das Paket in den *Perimeter Modus* übernommen.

Im *Perimeter Modus* wird das Paket nach der Rechten-Hand-Regel auf dem planarisierten Netzwerkgraphen weitergeleitet. Diese Vorgehensweise von GPSR ist in Abbildung 8 veranschaulicht. Hierbei ist anzumerken, daß dieses Schaubild nur zur Illustration des Konzepts des *Perimeter Routing* dient, da das Ziel im *Greedy Modus* erreichbar wäre. In dieser Abbildung ist x der Knoten, an dem das Datenpaket für das Ziel D in den *Perimeter Modus* wechselt.

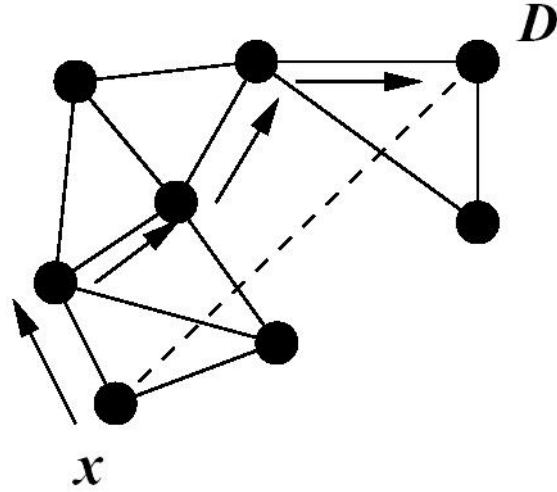


Abbildung 8: Beispiel für Perimeter Forwarding. (Quelle: [2])

Die *Perimeter*-Pakete werden bei GPSR an fortschreitend nähere polygonale Flächen des planaren Graphen, welche von der Linie \overline{xD} geschnitten werden, weitergeleitet.

Ein planarer Graph besteht aus zwei Arten von Flächen. Die *inneren* Flächen sind die abgeschlossenen Regionen, die von den Kanten des Graphen begrenzt werden. Bei der *äußeren* Fläche handelt es sich um die äußere Region, welche die außenliegenden Kanten des Graphen umschließt.

Im *Perimeter Modus* wird auf jeder besuchten Fläche mit Hilfe der Rechten-Hand-Regel eine Kante gesucht, welche die Linie \overline{xD} schneidet. An dieser Kante wechselt das Paket auf die nächste angrenzende Fläche, die auch von der Linie \overline{xD} geschnitten wird. Dieser Wechsel erfolgt dadurch, daß nicht die kreuzende Kante, sondern die nach der Rechten-Hand-Regel nächste Kante ausgewählt wird. Um zu verhindern, daß ein Paket ewig um eine Fläche kreist, wird beim Erreichen einer neuen Fläche die Kante, auf der diese Fläche erreicht wurde, im Paketkopf gespeichert. Dadurch wird verhindert, daß ein Paket dieselbe Kante ein zweites Mal in derselben Richtung benutzt. Sobald ein Knoten erreicht wurde, der dem Ziel näher ist als der Ausgangsknoten des *Perimeter Modus*, wechselt das Paket wieder in den *Greedy Modus*.

6.1 Simulationsergebnisse

Um die Performance von GPSR zu messen, wurde der Algorithmus in einer Vielzahl von statischen und dynamischen Netzwerktopologien simuliert. Das Hauptaugenmerk richtete sich dabei, aufgrund der höheren Anforderungen, auf mobile Netzwerke. Hierbei wurde die Leistungsfähigkeit von GPSR mit Dynamic Source Routing (DSR [5][6]) verglichen, welches sich anderen Routingalgorithmen für mobilen Ad-hoc-Netzwerke überlegen zeigte.

6.1.1 Dynamic Source Routing

Die Grundidee von DSR besteht darin, daß jedes Paket vom Absender eine vollständige Beschreibung des Pfades zum Empfänger erhält (Source Routing). DSR vermeidet dabei jegliche periodischen Aktualisierungen von Wegewahltabellen und arbeitet wie folgt:

Muß ein Knoten einen Weg ausfindig machen, so sendet er ein "Request"-Paket mit einer eindeutigen Kennung und der Zieladresse als Parameter an alle Nachbarknoten im Netzwerk. Jeder Knoten, der diese Nachricht empfängt, hat die folgenden Möglichkeiten:

- Hat der Knoten dieses Paket bereits einmal empfangen, was an der eindeutigen Kennung bemerkt wird, so verwirft er das Paket.
- Erkennt ein Knoten seine eigene Adresse als Zieladresse, so hat die Anfrage nach einem Weg ihr Ziel erreicht.
- In allen anderen Fällen hängt der Knoten seine eigene Adresse an die Liste der bereits durchlaufenen Knoten an und sendet dieses aktualisierte Paket wieder aus.

Bei diesem Algorithmus sammelt eine Anfrage nach und nach die Adressen der Knoten auf einem möglichen Pfad zum Ziel. Wenn ein solches "Request"-Paket den Zielknoten erreicht, erhält dieser den vom Datenpaket gewählten Weg. Diese erhaltene Pfadliste kann dann in umgekehrter Reihenfolge als Wegvorgabe für Antwortpakete benutzt werden.

Während der Kommunikation zwischen den Knoten muß der Pfad von allen Knoten, die sich auf diesem Weg befinden, gepflegt werden. Sobald es einem Knoten auf dem Pfad nicht mehr möglich ist, die Datenpakete an den gewünschten Nachbarknoten weiterzuleiten, muß dieser per "Request"-Paket versuchen, einen anderen Weg zu finden.

6.1.2 Simulationsumgebung

Als Simulationsumgebung diente die Software ns-2 [3], in welcher die in Carnegie Mellon [4] entwickelten Erweiterungen für drahtlose Netzwerke benutzt wurden.

Die Bewegungen innerhalb der Topologie wurden mit dem *random waypoint*-Modell simuliert: Ein Knoten wählt eine zufällige Zielposition im Simulationsbereich aus und bewegt sich mit einer aus einem gegebenen Intervall, zufällig ausgewählten Geschwindigkeit auf diesen Punkt zu. Wenn der Knoten an dieser Position angekommen ist, wird dieser Prozeß nach einer Pause wiederholt. Das Ändern der Länge dieser Pause bietet die Möglichkeit, den Grad der Mobilität der Knoten zu verändern.

In der Simulation variierte die Geschwindigkeit der Knoten zwischen 1 m/s und 20 m/s. Als Sendepause wurde 0, 30, 60 oder 120 Sekunden gewählt. Die Netzwerke beinhalteten 50, 112 und 200 Knoten mit einer Funkreichweite von 250 Metern, welche sich alle auf derselbe Ebene befinden. Als Vereinfachung wurde die Verzögerung für die Benutzung eines Positionsdienst ignoriert, und es wurde angenommen, daß alle Funkverbindungen zwischen den Knoten synchron sind. Die Datenübermittlung wurde durch 30 Datenströme simuliert, die mit einer konstanten Datenrate von 2Kbps und einer Paketgröße von 64 Byte, von 20 Knoten übertragen wurden.

6.1.3 Erfolgsquote der Paketzustellung

Die Abbildung 9 zeigt den Anteil der erfolgreich zugestellten Pakete im Verhältnis zur Länge der Pause, die die Knoten nach Erreichen ihrer Zielposition einlegen. Die drei Varianten des GPSR unterscheiden sich nur in der Länge B des "Beaconing"-Intervalls, welches mit 1, 1,5 und 3 Sekunden angegeben wurde. Das Schaubild bezieht sich außerdem nur auf die Pakete, für welche auch ein Weg zum Ziel existierte. Der Grund dafür ist, daß das Versagen des Algorithmus in diesen nicht-zusammenhängenden Graphen keine Aussage über dessen Qualität macht.

Wie erwartet, verursacht das "Beaconing"-Intervall der Länge 3 Sekunden eine leicht geringere Erfolgsquote. Es zeigt sich, daß das weitere Senken unterhalb von 1,5 Sekunden nur geringen Nutzen hinzufügt. Die Simulationen zeigten, daß GPSR mit allen getesteten Pausenlängen eine höhere Erfolgsquote liefert als DSR.

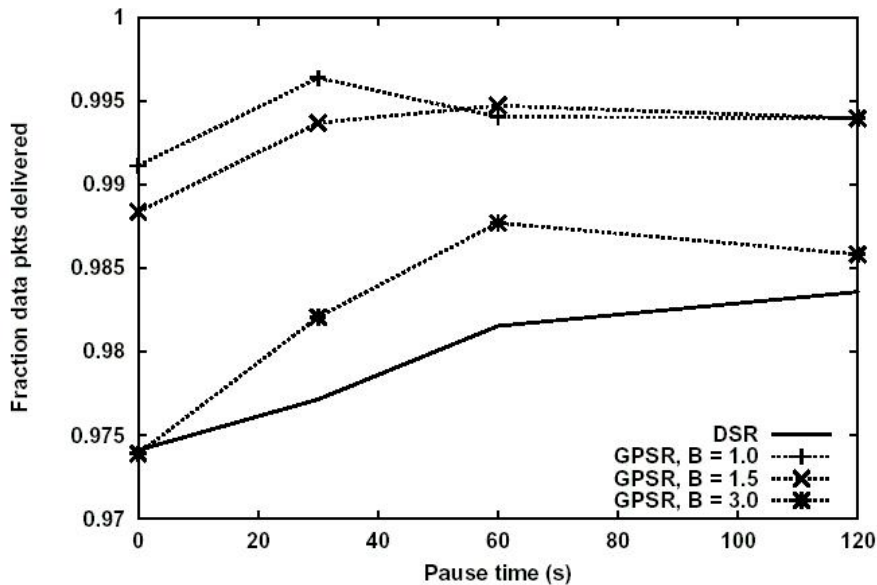


Abbildung 9: Erfolgsrate der Paketzustellung (Quelle: [2])

7 Zukünftige Arbeit

Die Versorgung mit Bandbreite ist in Netzwerken, in denen geographisches Routing benutzt wird, ein offenes Problem. In Netzwerken, in denen alle Netzwerkknoten ähnliche Sendeleistungen haben, kann es durch die Ballung des Verkehrs zu Stauungen kommen. Solche “hot spots” treten auf, falls Verbindungen durch dieselbe geographische Region geleitet werden. Um den Verkehr zu streuen, wird versucht, eine Menge an geographischen Wegpunkten zu definieren, die zufällig ausgewählt werden, und dann den Verkehr über ihre Positionen “zwingen”.

Zusätzlich ist geplant, das Protokoll um die Möglichkeit, die Knoten im dreidimensionalen Raum zu positionieren, zu erweitern. Das “Greedy Forwarding” lässt sich einfach auf drei Dimensionen umstellen. Ein Problem bildet dabei aber noch das planarisieren des dreidimensionalen Graphen. Es existiert zwar ein Algorithmus, um den Relative Neighborhood Graph zu ermitteln, aber dieser arbeitet nicht verteilt. Jeder Knoten benötigt also noch die vollständige Topologie zur Umwandlung.

8 Abschlußbemerkungen

Das hier vorgestellte Greedy Perimeter Stateless Routing ist ein Routingalgorithmus welcher die geographischen Positionen der Netzwerkknoten benutzt und die folgenden Eigenschaften besitzt:

- Es wird nur eine geringe Anzahl von Zuständen, welche nicht von der Anzahl der Ziele abhängig ist, in den Knoten zwischengespeichert.
- Die Komplexität des Algorithmus ist sehr gering.
- Die Paketübertragung ist in dicht besetzten drahtlosen Netzwerken sehr robust.

GPSR benutzt, wenn es die Topologie erlaubt, *Greedy Forwarding* und profitiert von dessen Robustheit und seiner Tendenz zum kürzesten Pfad. An Stellen, an denen *Greedy Forwarding* nicht möglich ist, benutzt GPSR *Perimeter Forwarding*, um das Paket von dem lokalen Maximum auf einen dem Ziel näheren Knoten zu retten. Das *Perimeter Forwarding* weicht vom kürzesten Weg ab und ist manchmal weniger robust gegenüber sich verändernden Perimetern. GPSR skaliert gut, wenn die Knotenanzahl und die Mobilitätsrate erhöht wird. Zusätzlich erzeugt GPSR Protokollverkehr, der von der Menge her nicht abhängig ist von der Länge der Pfade durch das Netzwerk. GPSR profitiert davon, daß es seine Entscheidungen nur mit den Informationen über die unmittelbar in geographischer Nähe liegenden Knoten trifft. Andere Protokolle, die darauf aufbauen, den Weg zum Ziel mit Hilfe von Zustandsinformationen in den Zwischenknoten zu ermöglichen, werden mit großen Herausforderungen konfrontiert, wenn sich der Durchmesser des Netzwerks (gemessen in Hops) und die Mobilität der Knoten vergrößert. Der geographische Ansatz, wie hier bei GPSR, stellt also einen weiteren leistungsfähigen Schritt betreffend der Skalierbarkeit von Routingalgorithmen für mobile Ad-hoc-Netzwerke dar.

Literatur

- [1] M. MAUVE, J. WIDMER, H. HARTENSTEIN. *A Survey on Position-Based Routing in Mobile Ad-Hoc Networks*. IEEE Network, 2001.
- [2] B. N. KARP. *Geographic Routing for Wireless Networks*. PhD thesis, Harvard University, 2000.
- [3] THE VINT PROJECT. *The UCB/LBNL/VINT Network Simulator-ns(version 2)*. <http://mash.cs.berkeley.edu/ns>.
- [4] THE CMU MONARCH GROUP. *Wireless and Mobility Extensions to ns-2*. <http://www.monarch.cs.cmu.edu/cmu-ns.html>, 1999.
- [5] D. B. JOHNSON, D.B. MALTZ. *Dynamic Source Routing in ad hoc wireless networks*. In *Mobile Computing* , T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 1996, ch. 5, pp. 153 - 181.
- [6] D. MALTZ, J. BROCH, J. JETCHEVA, D. JOHNSON. *The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks*. *IEEE Journal on Selected Areas on Communications* 17,8 (1999), 1439 -1453.
- [7] M. WEISER. *The Computer for the 21st Century*. *Scientific American* pp. 66-75, September 1991.