

Positionsbasiertes Routing in mobilen Ad-hoc Netzwerken

Carsten Busche

Abstract – Diese Arbeit liefert einen Überblick über Ad-hoc Routingverfahren, die die Entscheidung über die Weiterleitung der Daten aufgrund der geographischen Position des Empfängers fällen. In diesen hoch dynamischen Netzwerken profitieren sie vor allem von der Tatsache, dass sie keine festen Routen verwalten müssen und damit schnell auf die Topologieänderungen innerhalb des Ad-hoc Netzwerkes reagieren können. Neben der Zielposition benötigt ein Knoten nur seine eigene Position und die seiner *one-hop* Nachbarn. Die Grundvoraussetzung dieser Routingverfahren sind Positionsdienste, die für sie die Position des Empfängers ermitteln und ihnen zur Verfügung stellen. In dieser Arbeit werden sowohl Positionsdienste als auch Routingverfahren vorgestellt und qualitativ verglichen.

1. Einführung

In den letzten Jahren hat die immer größer werdende Verbreitung kabelloser Kommunikationsgeräte dazu geführt, dass man verstärkt in Richtung einer sich selbst organisierenden Netzwerkstruktur geforscht hat, die keine fest installierte Infrastruktur benötigt.

Diese Netzwerke werden Ad-hoc Netzwerke genannt und bestehen aus einzelnen Teilnehmern (Knoten), die sowohl als Endgerät als auch als Router fungieren.

Ad-hoc Netzwerke untergliedern sich in zwei große Klassen. In *statische Ad-hoc Netzwerke* und in *mobile Ad-hoc Netzwerke*. Während bei den statischen Ad-hoc Netzwerken ein Knoten seine Position nicht mehr verändert, sobald er Teil des Netzwerkes geworden ist, befindet sich ein Knoten bei einem mobilen Ad-hoc Netzwerk in Bewegung und verändert seine Position.

In dieser Arbeit werden die mobilen Ad-hoc Netzwerke betrachtet, die aufgrund der Veränderungen der Netztopologie, im Zuge der sich bewegenden Knoten, besondere Anforderungen an das Routing von Datenpaketen stellen.

Auch hier gibt es wieder zwei unterschiedliche Ansätze, um Daten zwischen den einzelnen Knoten zu versenden. Zum einen den Ansatz des *Topologie-basierenden* Routings, und das *Positions-basierende* Routing, das hier im Folgenden betrachtet wird.

Der größte Vorteil der mobilen Ad-hoc Netzwerke ist ihre Flexibilität und vor allem die Tatsache, dass sie spontan genutzt werden können, ohne vorher einen entsprechenden infrastrukturellen Aufbau getätigt zu haben. Im Folgenden wird Kapitel 1.1 einen Einblick in die Funktionsweise des positionsbasierten Routings geben. In Kapitel 2 werden dann verschiedene Positionsdienste vorgestellt, während Kapitel 3 sich anhand von ausgewählten Algorithmen mit drei verschiedenen Routingverfahren beschäftigt. Danach folgt in Kapitel 4 eine kurze Zusammenfassung der Daten und voraussichtliche Einsatzmöglichkeiten. In Kapitel 5 werden noch fehlende und wünschenswerte Eigenschaften sowohl der Positionsdienste als auch der Routingverfahren erwähnt.

1.1 Funktionsweise des Positionsbasierten Routings

Das Positionsbasierte Routing basiert auf der geographischen Position der Knoten zueinander. Es wird in einem Ad-hoc Netzwerk eingesetzt, in dem es keine feste Topologie und keine feste Infrastruktur gibt. Ein Knoten dieses Ad-hoc Netzwerkes erfährt seine eigene Position meist mit Hilfe eines Positionsdienstes wie z.B. GPS. Die geographische Position des gesuchten Knotens muss er erst durch einen anderen Positionsdienst in Erfahrung bringen, damit das Datenpaket durch ein Routingverfahren zugestellt werden kann.

Der Prozeß, dass ein Paket von einem Sender S zu einem Empfänger E gesendet werden kann, spaltet sich somit in zwei große Subprozesse auf:

- 1.) *Lokalisierung* der geographischen Position des Empfängers
- 2.) *Routen* der Daten zum Empfänger, basierend auf den aus 1. erhaltenen Informationen

Die existierenden Positionsdienste können in vier verschiedene Kategorien aufgeteilt werden, je nachdem, ob der Dienst von nur einigen oder von allen Knoten angeboten wird und für welche Anzahl von Knoten diese Positionsserver die geographischen Daten verwalten.

Somit ergeben sich folgende vier Varianten [1]:

- *All-for-All* (alle Knoten besitzen die Positionsdaten aller anderen Knoten)
- *All-for-Some* (alle Knoten haben Kenntnis von bestimmten anderen Knoten)
- *Some-for-All* (einige Knoten wissen die Positionsdaten aller Knoten)
- *Some-for-Some* (einige Knoten wissen die Positionen von einigen anderen Knoten)

Sobald ein Knoten die Zielposition des gesuchten Empfängers ermittelt hat, wird ein Datenpaket erstellt, in dem zusätzlich die letzte bekannte Position des Zielknotens und die eigene Position eingetragen werden. Dann wird das Datenpaket mit Hilfe eines Routingverfahrens, welches einen Pfad durch das sich ständig verändernde Netzwerk sucht, an den Empfänger zugestellt. Auf diesem Weg kann, in Abhängigkeit vom Routingverfahren, die Zieladresse im Datenpaket aktualisiert werden.

Bei den Routingverfahren wird die Art der Weitervermittlung meist von einem Knoten in Abhängigkeit von den Positionen seiner unmittelbar benachbarten Knoten getroffen. Diese Positionen erhält er durch periodische one-hop Broadcasts.

Es werden wiederum drei Routingverfahren unterschieden,

- Greedy Forwarding
- Restricted directional flooding
- Hierarchical Approaches

die auf verschiedenen Lösungsansätzen basieren und begrenzt mit den Positionsdiensten kombiniert werden können.

2. Positionsdienste

Damit die gesuchte Position eines Knotens bestimmt werden kann, ist die Hilfe eines Positionsdienstes notwendig. Mobile Knoten erfahren ihre eigene Position im allgemeinen durch z.B. den GPS-Dienst und wenn ein Knoten die aktuelle Position des Adressaten nicht kennt, muss er diesen Dienst in Anspruch nehmen.

In einem klassischen zellularen Netzwerk existieren Positions-Server mit fixen Adressen, die Positionsinformationen aller beteiligten Knoten innerhalb des Netzwerks verwalten (*Some-for-All*).

Bei einem mobilen Ad-hoc Netzwerk werden dezentralisierte Positionsdienste verwendet, da die Positionen der Knoten sich ständig verändern. Anhand eines Beispiels, des *Grid Location Services*, wird die Funktionsweise dieser Dienste erklärt und diverse weitere Dienste kurz vorgestellt.

2.1 Grid Location Service

Die Grundidee des *Grid Location Services* (GLS) [2][3] ist, dass die momentane Position eines Knoten auf einer im Netz möglichst gleichmäßig verstreuten Anzahl von Positionsservern hinterlegt wird. Diese Server zeichnen sich durch nichts weiter aus, sie sind ebenfalls nur Knotenpunkte und arbeiten im Auftrag anderer Knoten. Damit die Wege für distanznahe Knoten nicht unnötig lang werden, ist die Anzahl der Positionsserver nahe des Knotens relativ dicht und sie werden mit zunehmender Distanz immer spärlicher.

GLS stellt sicher, dass jedem Knoten eine zufällige, aber eindeutige ID zugewiesen wird, die mit Hilfe einer starken Hashfunktion aus dem Knotennamen ermittelt wird.

Am Anfang teilt der *Grid Location Service* das Gebiet, das das Ad-hoc Netzwerk aufspannt, in ein hierarchisches Gitternetz von Quadraten auf. Das kleinste Quadrat wird als Quadrat erster Ordnung betrachtet. Vier Quadrate erster Ordnung bilden ein Quadrat zweiter Ordnung und so weiter. Für die Funktionsweise ist es wichtig, dass sich die Quadrate nicht überlappen. Damit wird sichergestellt, dass sich ein Knoten jeweils nur in einem Quadrat beliebiger Ordnung befindet. In diesem Netz enthält ein Quadrat n -ter Ordnung vier Quadrate mit einer $(n-1)$ -ter Ordnung (Abb. 1) und bildet so einen sogenannten Quadratbaum. Dabei werden in der Abbildung ein Quadrat 1. Ordnung (A), ein Quadrat 2. Ordnung (B) und ein Quadrat 3. Ordnung (C) gezeigt. Kein gültiges Quadrat wäre z.B. D.

Jeder der Knoten besitzt eine Tabelle, in der alle Knoten erster Ordnung aufgeführt sind. Diese Tabelle wird durch periodische Broadcasts erzeugt und ein Knoten sendet seine Positionsänderungen in regelmäßigen Abständen an seine Positionsserver.

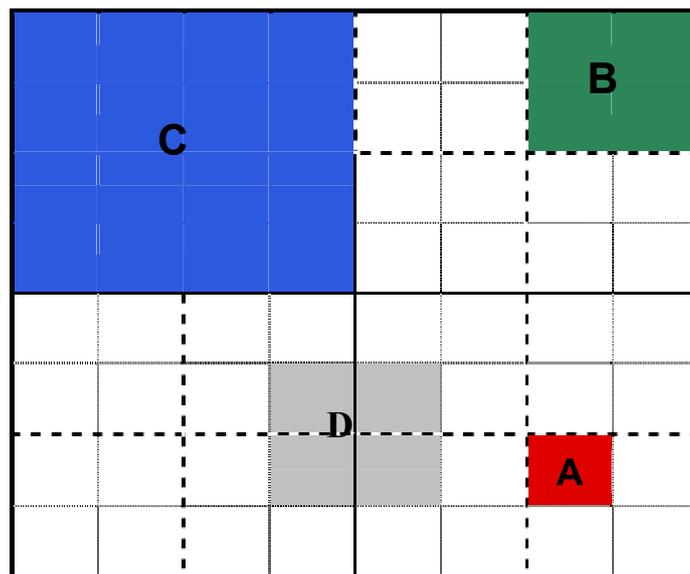


Abb. 1: Aufgespanntes Gitternetz des Grid Location Service.

Die Auswahl der Positionsserver eines Knoten E erfolgt dadurch, dass er sich Knoten aussucht, deren ID-Nummer *größer* als seine eigene ID-Nummer ist, und die seiner eigenen ID-Nummer *numerisch möglichst nah* sind. Dabei wird die Nummerierung als zirkular

angesehen, d.h. wenn ein Knoten die ID-Nummer 25 besitzt und in dem Bereich nur zwei Knoten mit den ID-Nummern 2 und 20 existieren, dann ist ihm ein Knoten mit der ID-Nummer 2 näher, als ein Knoten mit der ID-Nummer 20.

In jedem Quadrat einer Ordnungsklasse sucht sich der Knoten E dann drei Knoten aus, die als seine Positionsserver fungieren. Wenn nun ein Knoten S die Position des Knoten E ermitteln will, sendet er eine Anfrage an den nächsten Knoten, von dem ihm die Position bekannt ist und der eine ID-Nummer besitzt, die der von E am nächsten kommt oder gleich ist. Der Knoten wird die Anfrage in der gleichen Weise weitergeben oder wenn ihm die Position von Knoten E bekannt ist, direkt an Knoten E versenden. Da die Position von Knoten S in der Anfrage enthalten ist, kann Knoten E direkt antworten (Abb. 2).

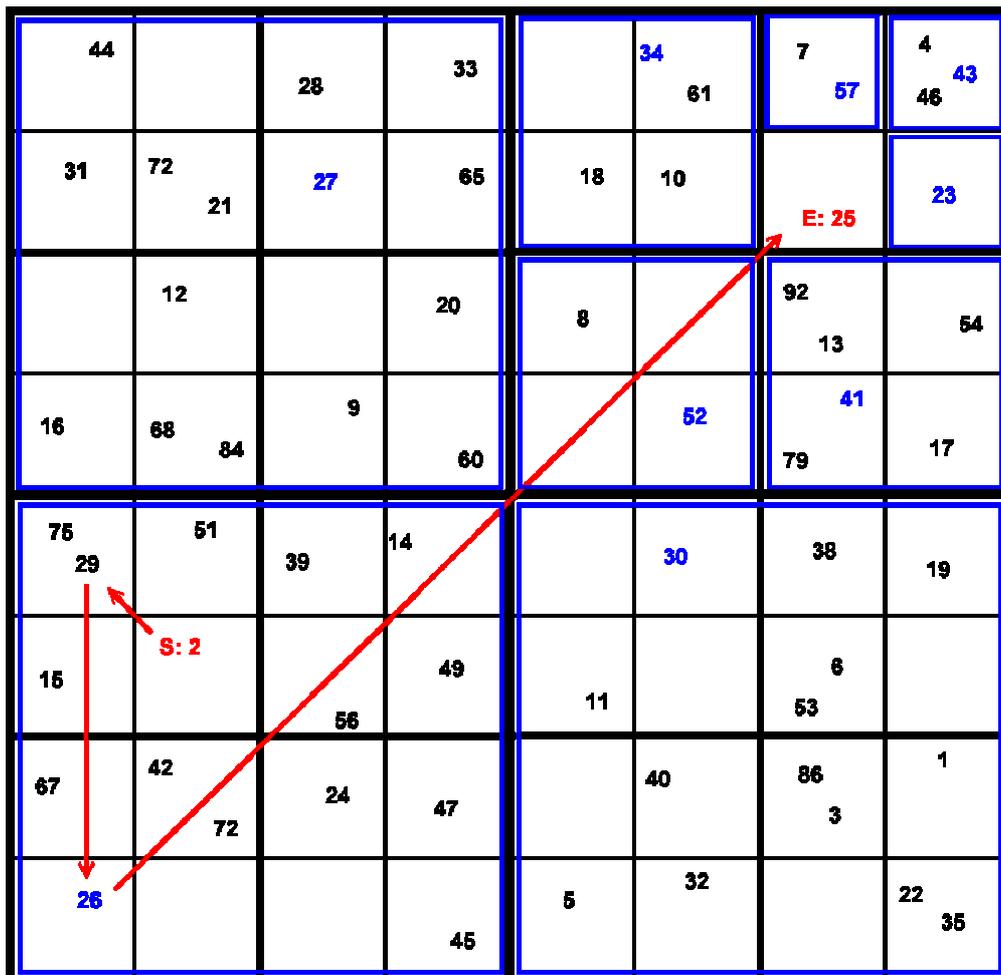


Abb. 2: Gitternetz des Grid Location Service

In Abbildung 2 stellen die abgehobenen blauen Quadrate die Quadrate der 1. – 3. Ordnung da. Die Positionsserver des Knotens E sind blau dargestellt (IDs = {57, 43, 23, 34, 52, 41, 27, 26, 30}). Sucht der Sender S (ID = 2) die Position des Empfänger E (ID = 25), kontaktiert er den Knoten mit der ID-29 (ID größer als die von E und möglichst nah an der ID-Nummer). Da der Knoten mit der ID 29 nun selbst als Positionsserver für den Knoten mit der ID 26 fungiert, kennt er auch dessen Position und kann eine Anfrage an den Knoten mit der ID 26 stellen. Hier trifft die Anfrage auf einen der Positionsserver von E und damit kann sie an den Empfänger E weitergeleitet werden.

Dadurch, dass ein Knoten die IDs der Positions-Server nicht kennen muss, wird die Implementierung eines Suchmechanismus unnötig. GLS ist ein *All-for-Some* Dienst.

2.2 DREAM

Der *Distance Routing Effect Algorithm for Mobility* [1] basiert darauf, dass jeder teilnehmende Knoten eine Positionsliste mit Knotenname, Richtung, Entfernung und einer Zeitmarke aller anderen Knoten besitzt. Jeder DREAM Knoten flutet dann proaktiv das gesamte Netz mit seinen Positionsaktualisierungen. Somit ist DREAM ein *All-for-All* Ansatz. Die Genauigkeit der Positionsangabe hängt von zwei Faktoren, der *temporalen* und der *räumlichen* Auflösung ab. Die temporale Auflösung gibt an, wie häufig eine Positionsänderung gesendet wird. Sie hängt von der Geschwindigkeit des Knotens ab, da bei höherer Mobilitätsrate auch die Positionsdaten des Knotens schneller veralten. Der Aspekt bei der räumlichen Auflösung geht davon aus, dass die Position eines Knotens E bei benachbarten Knoten genauer und bei den weit entfernten Knoten weniger genau bekannt sein muss. Dies basiert auf der Erkenntnis, dass weit entfernte Objekte sich langsamer zu bewegen scheinen, als näher liegende. Über die räumliche Auflösung kann auch festgelegt werden, wie weit ein Positionsupdate reisen darf, bevor es verworfen wird.

2.3 Quorum-basierender Positionsdienst

Das Konzept des *Quorum Systems* [1] verwendet den Ansatz eines sich selbst organisierenden virtuellem Backbones, das dynamisch über alle Netzwerkknoten verteilt arbeitet. Dazu wird die Menge aller Knoten in Untermengen, sogenannte Quoren aufgeteilt. Die Schnittmenge zweier Quoren ist niemals leer, so dass jedes Quorum mindestens mit einem weiteren Quorum verbunden ist. Abbildung 3 zeigt ein Netzwerk [1], in dem die Knoten 1 bis 6 ein virtuelles Backbone bilden, das durch die Verwendung eines nicht-positionsbasiertem Ad-hoc Routingverfahrens erzeugt wird. Diese 6 Backbone Knoten bilden drei Quoren A, B und C .

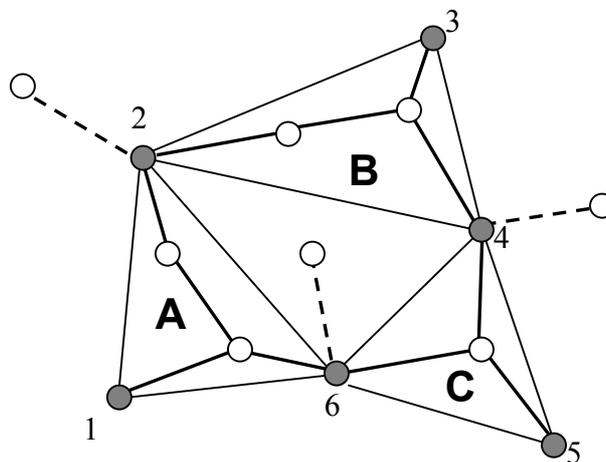


Abb. 3 : Quorum System

Die Knoten eines Quorums dienen als Positionsdatenbank und alle Knoten dieses Quorums besitzen ein Replikat, das in diesem Quorum zu verwaltenden Positionsdaten. Jeder Positionseintrag besitzt neben der geographischen Position eines Knotens auch noch eine Zeitmarke, damit ein anfragender Knoten immer die neueste Version des Positionseintrages eines gesuchten Knotens erhält. Für ein Positionsupdate kontaktiert ein Knoten den nächstgelegenen Backboneknoten. Dieser Backboneknoten wählt dann das Quorum aus, das die Positionsdaten verwalten soll. Benötigt ein Knoten S die Positionsinformationen eines

Knotens E, kontaktiert er ebenfalls seinen nächsten Backboneknoten. Dieser leitet die Anfrage an ein beliebiges Quorum weiter und sendet dann die Positionsinformationen an den Knoten S zurück. Dadurch, dass die Schnittmenge zweier Quoren niemals leer ist, wird sichergestellt, dass eine aktuelle Version der gesuchten Information zur Verfügung steht oder in Erfahrung gebracht werden kann. Der *Quorum-Based Location Service* kann als All-for-All, All-for-Some oder Some-for-Some Ansatz fungieren, je nach Wahl von Backbone- und Quorumsgröße. Er wird aber vorwiegend als *Some-for-Some* betrieben.

2.4 Homezone

Dieser Dienst verwendet die Idee einer virtuellen *Homezone* [4], in der die Positionsinformationen eines Knoten gespeichert und gepflegt werden. Mit Hilfe einer wohlbekannten, auf die Knoten-ID angewandten Hashfunktion, kann die geografische Position der *Homezone* H eines Knotens ermittelt werden. Alle Knoten innerhalb eines Radius R um H erhalten die Positionsinformationen des Knotens. In regelmäßigen Abständen sendet der Knoten seine aktuelle Position zu seiner eigenen *Homezone*. So kann, ähnlich wie beim GLS, die Positionsdatenbank zu einem jeden Knoten ohne Informationsaustausch gefunden werden.

Sucht ein Knoten S, die Position eines Knotens E, so schickt er eine Suchanfrage an die ihm mit Hilfe der Hashfunktion bekannten Homezone H von E. Da der Knoten E in regelmäßigen Abständen seine Position an seine eigene Homezone meldet, erhält Knoten S die gesuchten Informationen (Abb. 4). *Homezone* ist ein *All-for-Some* Ansatz.

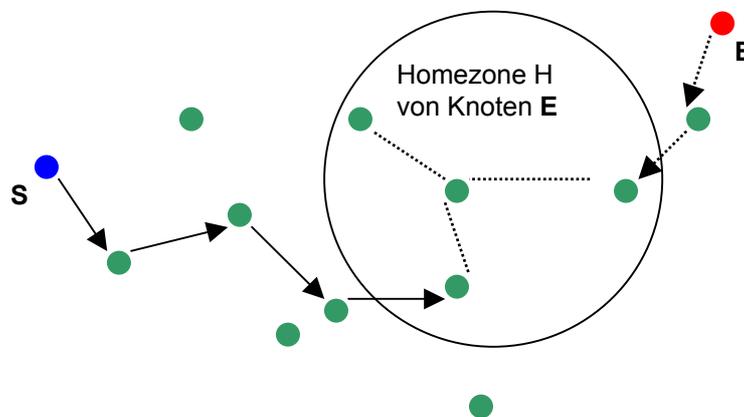


Abb. 4: Homezone

2.5 Vergleich der verschiedenen Positionsdienste

Bei dem Vergleich der verschiedenen Positionsdienste soll gezeigt werden, wie sie sich bei einer Erhöhung der Knotenanzahl in dem Ad-hoc Netzwerk verhalten. Hierfür ist wichtig, dass die Knotendichte in dem Netzwerk immer konstant bleiben soll. Wird die Knotenanzahl erhöht, vergrößert sich auch die Fläche des Ad-hoc Netzwerkes. Hierbei ist zu beachten, dass wenn der Abstand zweier Punkte in einem Quadrat der Größe $N \times N$ um den Faktor N skaliert, sich die Anzahl der nötigen Übertragungen proportional zur Wurzel der Knotenerhöhung verhält. Da es bei dem Einsatz von Positionsdiensten vor allem um die Frage einer möglichst effizienten und vor allem aktuellen Lokalisierung einer Knotenposition geht, werden die oben behandelten Dienste tabellarisch unter folgenden Kriterien betrachtet.

	GLS	DREAM	Quorum	Homezone
Art des Dienst	All-for-Some	All-for-All	Some-for-Some	All-for-Some
Kommunikationskomplexität (Update / Suche)	$O(\sqrt{n}) / O(\sqrt{n})$	$O(n) / O(c)$	$O(\sqrt{n}) / O(\sqrt{n})$	$O(\sqrt{n}) / O(\sqrt{n})$
Zeitkomplexität (Update / Suche)	$O(\sqrt{n}) / O(\sqrt{n})$	$O(\sqrt{n}) / O(c)$	$O(\sqrt{n}) / O(\sqrt{n})$	$O(\sqrt{n}) / O(\sqrt{n})$
Anzahl Lokaler Informationen	$O(\log(n))$	$O(n)$	$O(c)$	$O(c)$
Lokale Informationsverwaltung	JA	JA	NEIN	NEIN
Robustheit	Mittelmäßig	Hoch	Mittelmäßig	Mittelmäßig
Implementationskompl.	Mittel	Niedrig	Hoch	Mittel

Tabelle 1: Übersichtstabelle der Positionsdienste und ihren Leistungen. n =Anzahl der Knoten, c =Konstante

Als erster Punkt wird die *Art* des Dienstes angegeben. Wichtig für den Einsatz ist die Frage der *Kommunikationskomplexität*, in der die durchschnittliche Anzahl an *one-hop* Übertragungen ermittelt wird, die für die Lokalisierung einer Knotenposition oder ein Positionsupdate eines Knotens benötigt werden. Bei DREAM skaliert sie linear zu der Anzahl der Netzwerkknoten, was ein Positionsupdate angeht, da alle Knoten des Netzwerkes mit der Information versehen werden. Eine Positionsanfrage erfordert hingegen nur eine lokale Suche in der Datenbank des Knotens. GLS und Homezone skalieren in Hinsicht auf die Kommunikationskomplexität beide linear mit dem Durchmesser des Netzwerkes. Dies kann man in Abb. 5 erkennen, wo ersichtlich wird, dass wenn sich der Durchmesser D eines Netzwerkes um einen konstanten Faktor vergrößert, auch die Anzahl der notwendigen Übertragungen von A nach B dazu proportional steigt. Der Abstand zwischen den Knoten bleibt laut Definition konstant und die Knotenanzahl steigt quadratisch an.

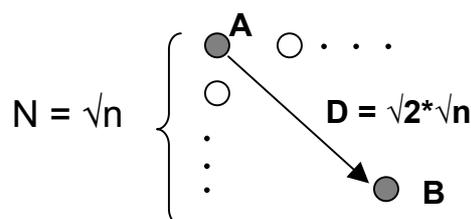


Abb. 5 : gleichverteiltes $N \times N$ Netzwerkquadrat bestehend aus n Knoten

GLS arbeitet aufgrund der lokalen Informationsverwaltung um einen konstanten Faktor schlechter als Homezone. Das Quorum System verwendet für ein Update oder eine Suche dieselben Operationen. In beiden Fällen muss ein Quorum kontaktiert werden, was eine konstante Anzahl an Knoten enthält. Da die Anzahl der Quoren und somit auch ihre Verteilung über das Netzwerk von der Menge aller Knoten abhängt, skaliert die Anzahl der Übertragungen, wie auch bei GLS und Homezone linear mit dem Durchmesser des Ad-hoc Netzwerkes.

In der *Zeitkomplexität* wird die durchschnittliche Dauer dieser Such- oder Positionsupdates angegeben. Hierbei hängt auch die zeitliche Dauer eines Updates für DREAM Knoten linear vom Durchmesser des Netzwerkes ab, da sich mit einer Erhöhung des Durchmessers die Anzahl der Knoten und somit auch die Anzahl der minimalen Übertragungen erhöht. Eine Suche ist wiederum ein konstanter Zeitaufwand und erfordert nur eine Anfrage an die lokale Positionsdatenbank des Knotens. Die Zeitkomplexitäten von GLS, Homezone und dem Quorum System gleichen denen ihrer Kommunikationskomplexitäten, da sie die gleichen Operationen erfordern. Ein weiterer Aspekt ist die *Anzahl* der *lokal* gespeicherten *Informationen* in jedem Knoten. Ein DREAM-Knoten speichert die Positionen eines jeden anderen Knotens des Netzwerkes. Bei dem Quorum System wird nur eine bestimmte Anzahl von Knoteninformationen in einem Backbone Knoten abgelegt. Ebenso ist die Anzahl der Informationen bei Homezone konstant, da in der Homezone eines Knotens nur eine begrenzte Anzahl von Knoten die Positionsserver bilden. Da die Fläche der Ordnungsquadrate bei GLS mit zunehmender Entfernung exponentiell größer wird, aber die Anzahl der Positionsserver eines Knotens für ein Quadrat konstant bleibt, steigt die Anzahl der Positionsserver logarithmisch zur Anzahl der Knoten.

In der Sparte *Lokale Informationen* ist kurz erwähnt, ob einem Knoten ein Mechanismus zur Verfügung steht, um die Positionen seiner ihm unmittelbar benachbarten Knoten zu erkennen und diese Informationen regelmäßig zu erneuern.

Bei der *Robustheit* eines Dienstes geht es darum, ob die Position eines Knotens bei dem Ausfall nur eines Knotens, einer Gruppe von Knoten oder bei einem Ausfall aller Knoten nicht mehr ermittelt werden kann.

Die Implementationskomplexität gibt an, wie schwierig es ist, den Dienst zu implementieren und zu testen.

3. Routingverfahren

In diesem Kapitel werden verschiedene Prinzipien und Algorithmen vorgestellt, die ein Datenpaket von einem Sender zu dem gewünschten Empfänger transportieren. Bei allen Algorithmen wird davon ausgegangen, dass die geografische Zielposition des Empfängers, eventuell durch den Gebrauch einer der oben angeführten Positionsdienste schon bekannt ist.

3.1 Greedy Packet forwarding

Das Grundprinzip dieser Methode basiert darauf, dass ein Sender die geographische Position eines Zielknotens mit in das Datenpaket integriert. Dann wird aufgrund einer lokalen Tabelle, in der die Positionen aller in Funkreichweite liegenden benachbarten Knoten eingetragen sind, entschieden, welcher den größtmöglichen Fortschritt in Richtung Zielposition bringt und das Datenpaket dann an den Knoten weitergeleitet. Wenn ein dazwischenliegender Knoten das Paket empfängt, entscheidet dieser wieder aufgrund seiner lokalen Informationen über den nächsten Knoten und sendet es weiter, bis das Datenpaket den Empfänger erreicht.

Die Auswahlstrategien, ob ein Knoten aufgrund möglichst großer Distanz vom Sender innerhalb der Sendereichweite oder auf einer verlängerten Linie zwischen Sender und Empfänger liegt, sind dabei recht unterschiedlich, wichtig ist nur, dass hier die Datenpakete immer mit maximaler Effizienz in Richtung Empfänger gesendet werden. Am nun folgenden Beispiel des *Greedy Perimeter Stateless Routing* (GPSR) [5] werden auch die dabei auftretenden Probleme und ihre Lösungsansätze behandelt.

Bei dem GPSR handelt es sich um einen Algorithmus, der zwei Methoden des Datenpakettroutings kombiniert. Der eine Teil verwendet das oben genannte Prinzip des Greedy Packet forwarding auf dem gesamten Netzwerkgraphen, das solange verwendet wird, bis er auf eine für ihn nicht lösbare Konstellation trifft. Danach tritt ein Algorithmus auf der Basis eines Perimeter forwardings auf dem planaren Netzwerkgraphen in Kraft.

Alle Knoten des Netzwerks besitzen bei diesem Algorithmus eine Tabelle, in der ihre unmittelbar benachbarten Knoten mit ID und Position verwaltet werden. Diese Daten werden in regelmäßigen Abständen durch einen proaktiven Broadcast aktualisiert.

Der Sender versieht das Datenpaket mit der Zielposition, die während des Routings von keinem Knoten verändert wird. Alle GPSR Pakete besitzen außerdem noch eine Reihe von Datenfeldern, die im Header verankert sind (Abb. 6).

Feld	Funktion
D	Zielkoordinaten
L _p	Position, an der das Datenpaket in den Perimeter Modus gewechselt ist
L _r	Schnittpunkt der Kante mit dem Vektor xD (gestrichelte Linie Abb. 7)
e ₀	Erste Kante, beim Erreichen einer neuen Fläche (Anfangs- u. Endpunkt)
M	Datenpaket Modus: Greedy oder Perimeter

Abb. 6: GPSR Header Datenfelder

Vom Sender wird der Modus auf Greedy gesetzt und dann das Datenpaket an einen in der Richtung liegenden Knoten gesendet, der sich geografisch möglichst nah an den Zielkoordinaten befindet. Aufgrund der lokalen Informationen über alle benachbarten Knoten in Reichweite, entscheidet dann der Knoten, wohin das Datenpaket als nächstes gesendet wird. Existiert kein Knoten innerhalb seines Senderradius, der näher an der Zielposition liegt, als er selbst (lokales Maximum in Knoten x), wird das Datenpaket in den Perimeter Modus versetzt und der zweite Algorithmus tritt in Kraft (Abb. 7).

Im Perimeter Modus wird zunächst versucht, das Datenpaket um den leeren Bereich herum, zu einem Knoten zu versenden, der näher an der Zielposition liegt, als der momentane Knoten. In diesem Falle wäre das der Weg $w \rightarrow v \rightarrow D$. Der gestrichelte Kreis kennzeichnet den Senderradius von Knoten x .

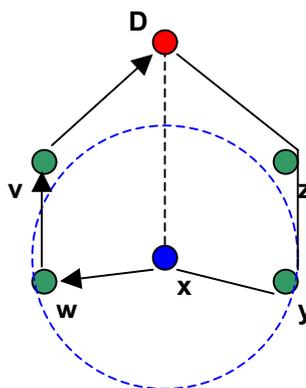


Abb. 7: Greedy forwarding Fehler

Erreicht das Paket einen Knoten, wird überprüft, ob eine Kante dieses Knotens, den Vektor xD schneidet. Ist dies der Fall, wird die Position des Schnittpunktes in L_f hinterlegt. Mit Hilfe dieser Information sucht sich der Algorithmus eine neue Kante und Fläche, die er anlaufen kann. Um zu verhindern, dass sich ein Paket in eine Endlosschleife um eine Fläche begibt, wird das Headerflag e_0 verwendet, das beim Erreichen einer neuen Fläche die Kante abspeichert, auf der die Fläche angelaufen wurde. Dazu werden die Position des Startknotens und die Position des Endknotens in das Datenpaket eingetragen. Erreicht das Datenpaket schließlich einen Knoten, der näher an der Zielposition liegt, als die im Header abgespeicherte Perimeter Modus Position, wechselt das Paket wieder in den Greedy Modus.

Die Entscheidung, ob eine Kante zu einem planaren Untergraphen gehört, kann von jedem Knoten lokal getroffen werden. Da zudem in dem Dateihader alle für die Bewertungsfindung wichtigen Daten abgespeichert wurden, kann ein Knoten aufgrund der lokalen Informationen über seine nächsten Nachbarn, alle Routing Entscheidungen selbstständig treffen, was auch das Erkennen einer unerreichbaren Zielposition mit einschließt.

3.2 Restricted Directional Flooding

Am Beispiel des DREAM Algorithmus [1] soll das Prinzip dieses Routingverfahrens verdeutlicht werden. Bei dem DREAM Routingverfahren wird das Paket eines Senders S, das zu einer Zielposition Z geschickt werden soll, an alle benachbarten Knoten des Senders S gesendet, die sich in der Richtung von Z befinden.

Das vom Senders S erwartete Gebiet um Z, berechnet der Knoten anhand der ihm bekannten Position von Z, mit einem Radius r , der auf einen Wert von $r = (t_1 - t_0) \cdot v_{\max}$ gesetzt wird. Wobei t_1 die aktuelle, t_0 die Zeitmarke der letzten Positionsinformation und v_{\max} die maximale Geschwindigkeit des Knoten innerhalb des Ad-hoc Netzwerkes ist (Abb. 8). Alle Knoten, die sich in dem Gebiet, das aus der Geraden durch S und Z, sowie dem Winkel φ bestimmt wird, erhalten somit das Datenpaket.

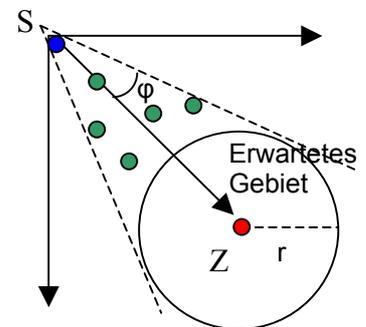


Abb. 8: Funktionsweise des DREAM Algorithmus

Die benachbarten Knoten, die dieses Paket empfangen, wiederholen die Prozedur mit ihren eventuell neueren Informationen über den Zielpunkt Z, bis das Paket den gesuchten Knoten erreicht. Sollte ein Knoten keinen Nachbarn in der benötigten Richtung besitzen, muss eine Recovery-Strategie angewandt werden, die aber nicht Teil der DREAM Spezifikation ist.

3.3. Hierarchisches Routing

Die Funktionsweise des hierarchischen Routingverfahrens soll anhand des *Terminoderoutings* [6][7] erklärt werden. Ein Knoten innerhalb eines mobilen Ad-hoc Netzwerkes wird deshalb Terminode genannt, da er sowohl als Netzwerkknoten als auch als Terminal arbeitet.

Beim Terminoderouting wird eine zweistufige Hierarchie (TLR und TRR) verwendet. Das *Terminode Local Routing* (TLR) wird zum Versenden im Nahbereich der Zielposition verwendet, wobei ein Nahbereich als zwei Hops angesehen wird. TLR erlaubt einem Terminode die Identität und Position aller Nachbarn innerhalb dieses Bereichs durch

periodisches, proaktives Senden einer Broadcast Nachricht zu erhalten. Mit Hilfe des *Terminode Remote Routing* (TRR) werden Datenpakete an alle nicht-TLR erreichbaren Knoten versendet. Das TRR besteht aus folgenden Elementen:

- Geodesic Packet Forwarding (GPF)
- Anchored Geodesic Packet Forwarding (AGPF)
- Friend Assisted Path Discovery (FAPD) / Geographical Map-based Path Discovery (GMPD)
- Pfad Verwaltung

Das GPF ist, wie auch das GPSR ein Greedy-Forwarding Ansatz, um die Datenpakete immer näher an die Zielposition zu versenden. Um lokale Maxima zu vermeiden, wird das GPF durch das AGPF unterstützt. Hierbei handelt es sich um eine Liste von relativ *festen* Routen, die über das FAPD oder in diesem Beispiel das GMPD erzeugt werden und nur Anlaufpunkte bilden. GMPD geht davon aus, dass jeder Terminode einen groben geographischen Überblick über das Ad-hoc Netzwerk besitzt. Ferner wird hierbei angenommen, dass sich die Knoten nicht gleichmäßig über das Netzwerk verteilen, sondern dass es Gebiete mit einer höheren Knotendichte gibt. Diese Gebiete werden im weiteren *Städte* genannt (Abb. 9). Die Städte sind untereinander durch alle dazwischen liegenden Knoten verbunden (*Datenstraßen*). Jeder Terminode besitzt nun eine *Karte* von diesen Städten. Eine *Karte* definiert die Städte als ein Quadrat mit einem geographischen Mittelpunkt und beinhaltet auch Informationen über die *Datenstraßen*. Um nun einen *festen* Weg zu dem Zielknoten zu finden, erstellt ein Terminode anhand dieser *Karte* eine Liste von Stadtzentren, die das Datenpaket anlaufen muss, um seine Zielposition zu erreichen. Aufgrund der teilweise *festen* Routen, besitzt das TRR reaktive Routingansätze.

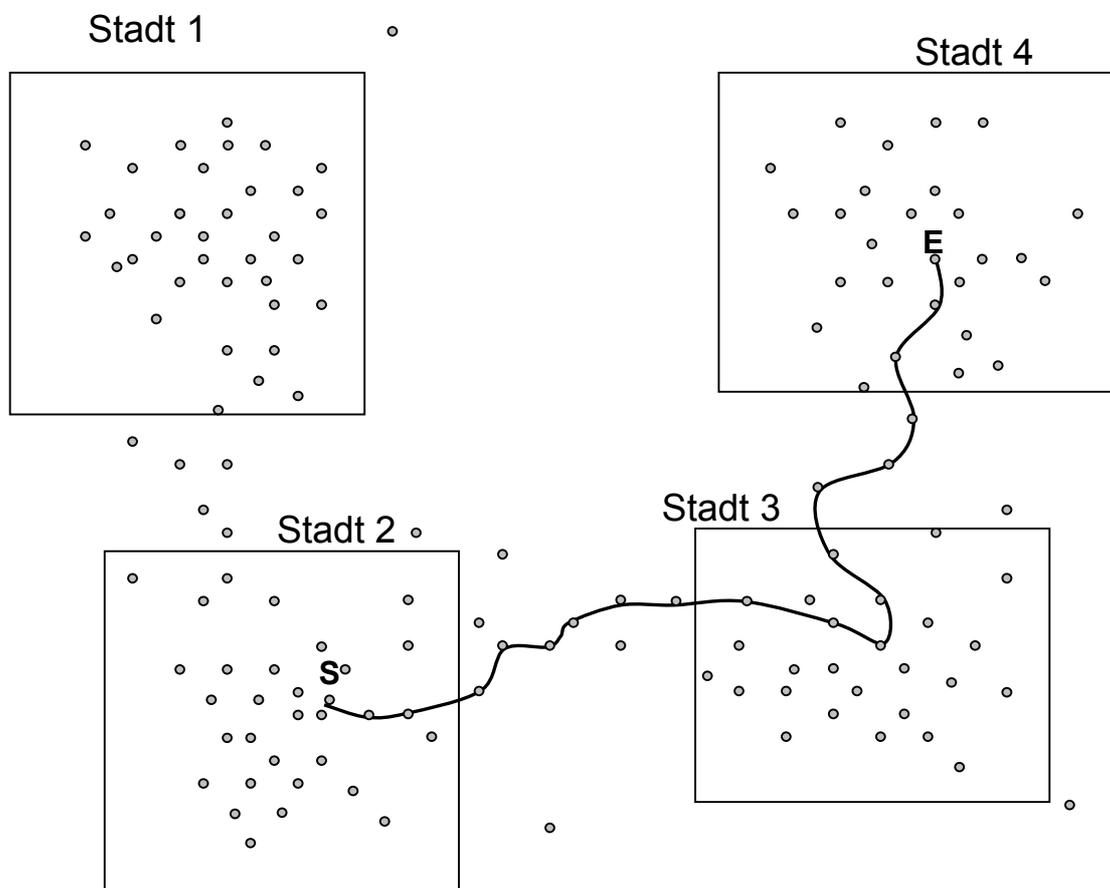


Abb. 9: Übersichtskarte des Netzwerks und die AGPF-Route

Wenn ein Knoten S Daten an einen Empfänger E senden will, wird zunächst überprüft, ob sich der Empfänger in TRR Reichweite befindet. Wenn ja, wird das Paket ausgeliefert, ansonsten geht der Algorithmus in den TLR Modus über. Hier wird die Zielstadt lokalisiert, in deren Nähe sich der Knoten E befindet. Existiert eine direkte Verbindung über die *Datenstraßen* zwischen den Städten oder befinden sich Sender und Empfänger in der gleichen Stadt, wird das Paket via *Geodesic Packet Forwarding* zugestellt. Wird aber für die Empfängerposition festgestellt, dass keine direkte Verbindung, entlang einer der *Datenstraßen* existiert, so erhält das Datenpaket zusätzlich zu der Zielposition eine Liste mit Anlaufpunkten. Diese durch GMPD gefundenen Anlaufpunkte bringen das Datenpaket bis in die Nähe der Zielposition. Sobald das Datenpaket in die festgelegte TLR Reichweite, der im Dateiheder verankerten Zielposition des Datenpaketes gelangt, geht der Algorithmus in den TLR Modus über.

3.4 Vergleich der verschiedenen Routingverfahren

Tabelle 2 zeigt einen Vergleich der Routingverfahren, mit ähnlichen Kriterien wie bei den Positionsdiensten.

	GPSR	DREAM	Terminode
Art	Greedy Forwarding	Restricted Directional Flooding	Hierarchisches Routing
Kommunikationskomplexität	$O(\sqrt{n})$	$O(n)$	$O(\sqrt{n})$
Toleranzgrenze	Funkreichweite	Erwartetes Gebiet des Ziels	Nah Distanz
Vorausgesetzte Positionsdienste	Kein bestimmter	All-for-All	Kein bestimmter
Robustheit	Mittel	Hoch	Mittel
Implementierungskmpl.	Mittel	Niedrig	Hoch

Tabelle 2: Tabelle zur Übersicht der Merkmale der verschiedenen Routingstrategien. Die Kommunikationskomplexität ist in Abhängigkeit von n = Anzahl der beteiligten Knoten gegeben.

Bei der *Kommunikationskomplexität* werden die durchschnittliche Anzahl der Sprünge zu benachbarten Knoten gewertet, bis das Paket vom Sender bis zum Empfänger gelangt ist, unter der Annahme, dass die Position des Empfängers bekannt ist.

Die *Toleranzgrenze* gibt an, wie hoch die geografische Abweichung der Empfängerposition sein darf, damit das Routingverfahren fehlerfrei arbeiten kann. Der Punkt der *Vorausgesetzten Positionsdienste* soll einen Überblick über die benötigten Positionsdienste geben, da bei der Kommunikationskomplexität davon ausgegangen wird, dass die aktuelle Position des Empfängers schon bekannt ist.

Die *Robustheit* eines Verfahrens wird, wie auch bei den Positionsdiensten nach Ausfallanfälligkeit ermittelt. Sie ist hoch, wenn der Ausfall eines dazwischenliegenden Knotens das Paket nicht am Erreichen des Zielknotens hindert. Mittelmäßig, wenn der Ausfall eines dazwischenliegenden Knotens zwar den Verlust des Paketes, aber kein Neuberechnen einer alternativen Route erfordert. Da aber alle hier vorgestellten Strategien über keine festgelegten Routen verfügen, ist dies auch die niedrigste Robustheit. Der

Implementationsaufwand soll ein Maß dafür sein, wie komplex ein Routingverfahren in Bezug auf Implementierung und Test ist.

4. Zusammenfassung

Aufgrund der oben vorliegenden Daten lässt sich sagen, dass es bei der Auswahl der Positionsdienste und bei der Wahl der Routingstrategien darauf ankommt, in welchem Umfeld sie angewandt werden.

DREAM eignet sich zum Beispiel für kleine Netze, und für Anwendungsgebiete, in denen hohe Zuverlässigkeit erforderlich ist. In großen Netzwerken mit einem großen Datenaufkommen skaliert es hingegen sehr viel schlechter als die übrigen Routingverfahren. Zudem benötigt das DREAM Routingverfahren einen All-for-All Dienst, weswegen in diesem Fall nur eine Betrachtungskombination zwischen dem DREAM Positionsdienst und dem DREAM Routingverfahren in Frage kommt.

Als Nachteil des *Quorum Location Dienstes* sollte erwähnt werden, dass er von einem nicht-positions basierenden Routingprotokoll abhängt, Dieses wird für den virtuellen Backbone verwendet, was sich zu Lasten der Komplexität der Implementierung dieses Dienstes und eventuell auch negativ auf die Skalierbarkeit auswirkt.

Homezone ist ebenfalls, wie auch der *Grid Location Service* in Bezug auf Skalierbarkeit und Effizienz ein hoffnungsvoller Ansatz. Homezone unterscheidet sich von GLS nur dadurch, dass beim GLS noch Informationen über unmittelbar benachbarte Knoten vorhanden sind, was die Kommunikation im Nahbereich erheblich einfacher gestaltet. Dafür ist Homezone implementierungsfreundlicher und die Mechanismen zur Reaktion auf auftretende Fehler oder die Anpassung an die sich verändernden Topologieverhältnisse sind einfacher zu handhaben. Eine Kombination zwischen dem GLS Positionsdienst und einem Greedy Forwarding (GPSR) scheint am besten geeignet zu sein, wenn man gute Skalierbarkeit, Bandbreitennutzung und zeiteffiziente Ergebnisse haben will. GLS benötigt die Positionstabelle für benachbarte Knoten ebenso, wie GPSR sie für das Routing verwendet.

5. Absehbare weitere Entwicklungen

Ein wichtiger Gesichtspunkt in der weiteren Entwicklung positionsbasierter mobiler Ad-hoc Netzwerke ist die Sicherheit und der Schutz der Privatsphäre. Wenn es möglich ist, dass die Position eines bestimmten Knotens leicht lokalisiert werden kann, besteht theoretisch auch die Möglichkeit, dass die gewünschte Anonymität eines Anwenders dadurch verletzt und eventuell missbraucht werden kann.

Wenn sich die Verwendung von Ad-hoc Netzwerken weiter durchsetzen wird, könnte ein weiterer Versuch sein, mehrere unterschiedliche Ad-hoc Netzwerken miteinander zu verbinden, was eventuell eine Hierarchie ähnlich des Terminodeansatzes erfordern könnte.

Auch der Ansatz des Greedy-Forwarding müsste noch ausgebaut werden. Immerhin ist er äußerst effizient und damit für hochdynamische Ad-hoc Netzwerke geeignet. Gegebenfalls könnten andere Auswahlstrategien bezüglich der Wahl des nächsten Knoten und bessere Recovery-Strategien, falls das Greedy Forwarding fehlschlägt, entwickelt werden.

Auch die Entwicklung anderer Methoden der ID Zuweisung, bei Knoten der *All-for-Some* Ansätze von Homezone und GLS, könnten in Angriff genommen werden. Ein weiterer Punkt, der in Betracht gezogen werden sollte, ist der einer Anpassung der Sendereichweite an die zu überbrückende Entfernung zwischen zwei Knoten. Wenn alle Knoten ungefähr die gleiche Sendeleistung haben, kann es zu Stauungen der Pakete kommen. Vor allem beim Terminode-Ansatz könnte, bei einer schlechten Verteilung der Knoten, der Einsatz der Datenstraßen solche Probleme fördern.

6. Quellenangaben

- [1] Martin Mauve, Jörg Widmer, Hannes Hartenstein. A Survey on Position-Based Routing in Mobile Ad-Hoc Networks. IEEE Network, 2001.
- [2] Jinyang Li, John Jannotti, Douglas S.J. De Couto, David R. Karger, Robert Morris. A Scalable Location Service for Geographic Ad Hoc Routing. In Proc. Of the 6th Annual ACM/IEEE Int. Conf. On Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000.
- [3] The grid project homepage. <http://www.pdos.lcs.mit.edu/grid/>.
- [4] Ivan Stojmenovic. Home agent based location update and destination search schemes in ad hoc wireless networks. Technical Report TR-99-10, Computer Science, SITE, University of Ottawa, September 1999.
- [5] Brad Karp, H.T. Kung . GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In Proc. Of the 6th Annual ACM/IEEE Int. Conf. On Mobile Computing and Networking (MobiCom 2000), Boston, MA, USA, August 2000.
- [6] Ljubica Blazevic, Silvia Giordano, Jean-Yves Le Boudec, Self Organized Terminode Routing. Technical report, DSC/2001/024, Swiss Federal Institute of Technologie, Lausanne.
- [7] Ljubica Blazevic, Silvia Giordano, Jean-Yves Le Boudec. Self Organized Routing in Wide Area Mobile Ad-Hoc Networks. Swiss Federal Institute of Technologie, Lausanne.