Universität Karlsruhe (TH)
Institut für Telematik

TeleMatics Technical Reports

Mobilkommunikation
Seminar WS04/05

Oliver Stanze, Christian Vogt, Kilian Weniger, Jidong Wu,
Martina Zitterbart, Hannes Hartenstein
{stanze,chvogt,weniger,wu,zit}@tm.uka.de, hannes.hartenstein@rz.uni-karlsruhe.de

August, 26th 2004

TM-2004-6

ISSN 1613-849X

http://doc.tm.uka.de/tr/

Institute of Telematics, University of Karlsruhe
Zirkel 2, D-76128 Karlsruhe, Germany

Vorwort

Das Seminar "Mobilkommunikation" wurde im Wintersemester 2003/2004 in Form eines Blockseminars am 09.02.2004 im Institut für Telematik durchgeführt. Die Themen des Seminars stammen aus den Bereichen "Mobile Ad-hoc Netze" und "Mobile IP/TCP". In diesem Seminarband werden nun die Ausarbeitungen der Studenten in Form eines Internen Berichts zusammengefasst. Folgende Themengebiete werden in diesem Internen Bericht behandelt:

Routing in mobilen Ad-hoc-Netzwerken

Mobile Ad-hoc-Netze sind infrastukturlose Netzwerke, die durch eine Gruppe von drahtlos miteinander kommunizierenden Geräten gebildet werden. Aufgrund der Mobilität der einzelnen Knoten unterscheidet sich das Routing in mobilen Ad-hoc-Netzen erheblich vom Routing im klassischen Internet. Die für MANETs vorgeschlagenen Protokolle lassen sich grob in *proaktive* und *reaktive* Routingverfahren einteilen. Proaktive Routingverfahren zeichnen sich dadurch aus, das jeder Knoten zu jedem Zeitpunkt aktuelle Topologieinformationen über das Ad-hoc-Netz besitzt. Die Verbreitung der Topologieinformation geschieht entweder mit Hilfe von Link-States (OLSR) oder mit Distanzvektoren (DSDV). Reaktive Routingverfahren besitzen dagegen keine aktuellen Topologieinformationen. Ein Knoten bestimmt erst dann eine Route zu einem Ziel, wenn diese auch benötigt wird. Das Finden einer Route besteht meistens aus einer *Request-Phase*, in der der entsprechende Knoten seine Anfrage in Netz sendet (meist durch Fluten), und einer *Reply-Phase*, in der Antworten (gültige Routen) an den Initiator des Requests gesendet werden. Die derzeit bekanntesten reaktiven Routingprotokolle verwenden entweder Distanzvektoren (AODV) oder Source-Routen (DSR). Es existieren allerding auch einige neuere Ansätze für *On-demand Link-State Routing* in MANETs. Vor allem proaktive Routingprotokolle haben Probleme bezüglich der Skalierbarkeit, da jeder Knoten Routen zu allen anderen Knoten bestimmen muss. Durch *hierachisches Routing* kann die Skalierbarkeit allerdings erhöht werden.

Spezielle Aspekte von MANETs

Mobile Ad-hoc-Netze unterscheiden sich aufgrund der Mobilität der Knoten und dem Verzicht auf eine vorhandene Infrastruktur erheblich vom klassischen Internet. Eine interessante Fragestellung bei der Betrachtung von MANETs stellt die *Kapazität* eines MANETs dar. Da sich alle Netzteilnehmer das Medium teilen und es keine zentrale Komponente für die Zugriffsteurung gibt, stellt sich die Frage: Wieviel Bandbreite steht einem einzelnen Teilnehmer zur Verfügung? Durch eine *Anbindung von MANETs an das Internet* könnten diese u.a. dazu verwendet werden, die Abdeckung von infrastrukturbasierten System, wie z.B. Hot Spots auf Basis von IEEE 802.11, zu erweitern. Die *Dienstsuche* in mobilen Ad-hoc-Netzwerken unterscheidet sich von Dienstsuche in infrastrukturbasierten Netzwerken, da i.A. keine zentralen Komponenten vorhanden sind, bei denen man nach einem speziellen Dienst nachfragen könnte. Bis heute wurden Protokolle (Routing, Service-Discovery, etc.) für mobile Ad-hoc-Netzwerke meist nur im Simulator evaluiert. Allerdings sind die dort verwendeten Bewegungsmodelle meist ziemlich unrealistisch. Um Protokolle für MANETs unter realistischen Bedigungen testen zu können, wurden in letzter Zeit einige *Testumgebungen für real world Ad-hoc-Netzwerke* entwickelt.

Spezielle Aspekte von Mobile IP

Protokolle zur Mobilitätsunterstützung, deren bekanntester Vertreter wohl Mobile IP ist, bieten Knoten die Möglichkeit sich in einem beliebigen Netz ans Internet anzuschließen und dabei

weiterhin über die IP-Adresse des Heimatnetzes erreichbar zu sein. Mobile IP besitzt jedoch einige Nachteile. Ein Nachteil besteht darin, dass ein mobiler Knoten bei jedem Netzwerkwechsel (Handover) seinem Heimatagenten mitteilen muss, wo er sich jetzt befindet. Deshalb wurden einige Verfahren zur Unterstützung von *Mikro Mobilität* für Mobile IP vorgeschlagen, die, falls möglich, die notwendige Signalisierung lokal begrenzen. Ein weiteres Problem von Mobile IP besteht darin, dass bei einem Wechsel des Verbindungspunktes zum Internet mitunter sehr lange Verzögerungen auftreten können. Es wurden deshalb Mechanismen zur Unterstützung *schneller Handover* entwickelt. Mobile IP ist sowohl in Version 4 als auch Version 6 nur für die Unicast-Kommunikation ausgelegt. Da jedoch Anwendung wie z.B. Konferenzsysteme, die zunehmend an Bedeutung gewinnen, auf einen Multicast-Dienst aufbauen, und damit auch mobile Nutzer die Systeme nutzen können, wurden bereits mehrere Ansätze für eine *Multicastunterstützung in Mobile IP* entwickelt. Die Mobilität der Geräte hat nicht nur Auswirkungen auf Protokolle der Vermittlungsschicht, sondern auch auf Protokolle der Transportschicht. Das liegt zum einen daran, das mobile Geräte meist drahtlos angebunden sind und dass aufgrund der höheren Bitfehlerraten auf dem drahtlosen Medium häufiger Paketverluste auftreten. Zum anderen können Handover auftreten, die ebenfalls zu Paketverlusten führen. Paketverluste bewirken bei TCP eine Drosselung der Senderate, wodurch die Performance leidet. Es wurden bereits eine Reihe von *TCP-Erweiterungen* vorgeschlagen, die die Performance von TCP in mobilen Netzwerken verbessert.

Inhaltsverzeichnis

Anbindung von MANETs an das Internet

Matthias Walliczek

Kurzfassung

Bei stationären Netzwerken mit statischen IP-Adressen gestaltet sich die Anbindung an das Internet vergleichsweise einfach - es muss lediglich einmal ein Default-Routers konfiguriert werden. Anders jedoch bei mobilen Ad-Hoc-Netzen: dort muss sich die Konfiguration in einem ständigen dynamischen Prozess den Veränderungen in der Netztopologie anpassen. Entsprechend den Bewegungen der Teilnehmer und den damit veränderten Empfangsreichweiten muss z.B. ein passendes Gateway gesucht und gefunden werden. Anschließend muss eine IP-Adresse konfiguriert werden, die ggf. in einem vom Gateway vorgegebenen Subnetz liegt, und schließlich müssen die Datenpakete auch auf dem richtigen Weg aus dem Ad-Hoc-Netz ins Internet gesendet und empfangen werden.

In diesem Text werden einige momentan diskutierte Protokollentwürfe vorgestellt und verglichen.

1 Einleitung

Bis 2005 werden mehr als 80% aller Notebooks im professionellen Bereich über eine Wireless Lan (WLAN)-Schnittstelle verfügen - so lautet eine Prognose des US-Analysten Gartner Dataquest [Keen03]. Selbst wenn es aber, wie ebenfalls prognostiziert, bis 2008 mehr als 167.000 Standorte durch öffentlichen WLAN-Hotspots versorgt werden, wird es immer noch zu viele Standorte ohne infrastrukturelle Netzanbindung per Access Point geben.

In solchen Situationen sind mobile Ad-hoc-Netze (MANET), die keinerlei feste Infrastruktur benötigen, eine kostengünstige Möglichkeit um ein Funknetz aufzubauen. Falls innerhalb dieses Funknetzes ein mobiles Endgerät (Mobile Node, MN) über einen Internet-Zugang (z.B. kabelgebunden per Ethernet oder drahtlos per GPRS) verfügt, wäre es sinnvoll, diesen Zugang auch anderen MNs innerhalb des Funknetzes zur Verfügung zu stellen.

Es könnte ebenfalls für Telefongesellschaften interessant sein, die Abdeckung ihres (UMTS-) Netzes dadurch zu vergrößern, indem MNs am Rand der Zellenreichweite als Relaisstation fungieren und dadurch anderen außerhalb der Zellenreichweite befindlichen MNs Zugang zum Netz verschaffen [CaPi03].

Doch selbst wenn ein MN innerhalb des WLANs über einen Internet-Zugang verfügt, reicht das allein nicht aus, um allen anderen MNs diesen Internet-Zugang zur Verfügung zu stellen: Es wird ein Protokoll benötigt, damit die anderen MNs dieses Gateway überhaupt benutzen können.

1.1 Anforderungen an das Protokoll

Dieses Protokoll muss definieren, wie ein MN mit Gateway-Funktionalität, im folgenden (Internet-)Gateway genannt, von anderen MNs gefunden werden kann und ob und wie der MN zusätzliche Adressen konfigurieren muss.

Die wichtigste Anforderung an dieses Protokoll ist dabei die Kompatibilität zu den bestehenden Routingprotokollen für Ad-Hoc-Netze und den Mobilitätsprotokollen des Internets (z.B. Mobile IP). Diese Protokolle sorgen dafür, dass überhaupt eine Kommunikation innerhalb eines solchen Netzes stattfinden kann und ein MN auch andere MNs erreichen kann, die außerhalb der Funkreichweite der eigenen Schnittstelle liegen. In diesem Fall müssen andere MNs als Router fungieren und diese Pakete weiterleiten [Schi03]. Die so entstandene Verbindung bezeichnet man auch als Multihopverbindung [XiBe02].

Die im Moment existierenden Protokollentwürfe lassen sich dabei in zwei Gruppen aufteilen. Einerseits existieren reaktive Routing-Protokolle, bei denen keinerlei Zustände gespeichert werden und eine Route nur bei Bedarf durch das Senden von Routing-Anfragen ermittelt wird.

Andererseits gibt es proaktive Protokolle, bei denen jeder MN versucht, durch den periodischen Austausch von Routinginformationen einen möglichst aktuellen Zustand des Netzes zu ermitteln und auf dieser Basis Routen zu errechnen, ohne irgendwelche Anfragen senden zu müssen.

Ebenfalls wichtig ist die Zusammenarbeit mit Mobile IP, einem anderen Protokoll in der Kategorie mobile Kommunikation. Mobile IP ermöglicht den Endgeräten weltweit unter derselben IP-Adresse erreichbar zu sein und so beim Wechsel von einem Netz in ein anderes Netz die Schicht 4-Verbindungen aufrecht erhalten zu können. Wichtig ist dieses Feature insbesondere dann, wenn der MN selber Dienste anbietet, die andere Hosts im Internet nutzen möchten und deshalb eine Verbindung vom Internet zum MN benötigen.

Eine wichtige Anforderung an jedes Protokoll für mobile Kommunikation ist die Berücksichtigung der begrenzten Ressourcen, die ein mobiles Endgerät auszeichnen. Insbesondere mit Strom, Bandbreite und CPU-Last sollte sparsam umgegangen werden. Das bedeutet zum Beispiel, dass ein Protokoll möglichst wenig Kontrollverkehr verursachen sollte.

Um zukunftsfähig zu sein, sollten die Protokolle auch IPv6 unterstützen. Dabei sollte jedoch beachtet werden, dass IPv6 bezüglich Mobile IP zahlreiche Änderungen bedeutet. Zum einen ist bei IPv6 der Adressraum deutlich größer geworden und es stehen genügend Adressen zur Verfügung, zum anderen ist bei diesem Protokoll ein erweiterbarer Routingheader vorgesehen. Deshalb werden keine *Foreign Agent Care-Of* Adressen (siehe 3.3) und keine *Foreign Agents* mehr benötigt. Stattdessen können die MNs *co-located Care-Of* Adressen (siehe 3.4) benutzen, die sie mittels automatischen Adresskonfiguration oder DHCPv6 lernen können. Auf die Tunnel-Technik kann ebenfalls verzichtet werden, da die beiden Adressen des MN, d.h. die *Care-Of-Address* und die Home-Adresse, in den Routing-Header aufgenommen werden können.

1.2 Aktueller Stand

Im Moment befinden sich im Wesentlichen zwei verschiedene Protokollentwürfe in der Diskussion: Zum einen liegt mit MIPMANET [JALJ+00] der Entwurf eines Protokolls vor, bei dem versucht wird, das Mobile IP-Protokoll so zu erweitern, dass es auch in Ad-Hoc-Netzen funktioniert. Zum anderen existiert ein Internet-Draft namens „Global connectivity for IPv4/IPv6 Mobile Ad Hoc Networks“, der ein eigenes und unabhängiges Verfahren vorstellt, das allerdings kompatibel zu Mobile IP ist. Dieses Protokoll existiert unter dem Namen Globalv4 [BRSP01] in einer Version für IPv4 bzw. unter dem Namen Globalv6 [WMPN+03] für IPv6.

Der folgende Text zeigt anhand eines chronologischen Ablaufes die Initialisierung und anschließend den laufenden Betrieb der verschiedenen Protokolle.

2 Auffinden und Auswahl eines Gateways

Bevor irgendwelche Daten ins Internet übertragen werden können, muss zunächst ermittelt werden, ob im Ad-Hoc-Netz ein Gateway vorhanden ist. Dabei gibt es zwei grundsätzlich verschiedene Vorgehensweisen: aktiv oder passiv [XiBe02].

Der Prozess des Auffindens eines Gateways ist nicht nur bei der Initialisierung eines MN nötig - er wird auch im laufenden Betrieb benötigt, um bei Standortwechseln den Wechsel in ein anderes Netz festzustellen. Bei diesem so genannten Handover muss der MN sämtliche Schritte vom Auffinden des Gateways bis zur Adresskonfiguration (siehe 3) erneut durchlaufen.

2.1 Passives Auffinden

Passives Auffinden bedeutet, dass das Gateway regelmäßig *Router Advertisement*-Nachrichten per Broadcast sendet, um seine Anwesenheit bekannt zu geben. Diese Nachrichten werden von allen MNs innerhalb der Funkreichweite empfangen und können von diesen anschließend weitergesendet werden, damit sie auch von MNs empfangen werden können, die außerhalb der direkten Funkreichweite liegen. Da ein Gateway jedoch typischerweise eine höhere Sendeleistung als ein MN hat, kann es theoretisch passieren, dass ein MN zwar das *Router Advertisement* empfängt, selber jedoch keine Verbindung zu diesem Gateway aufbauen kann [XiBe02].

Wenn ein MN mehrere *Router Response*- oder *Router Advertisement*-Nachrichten empfängt, soll er durch eine spezielle Metrik (z.B. Signalempfangsstärke, die Hopzahl, Belastung des Gateways oder eine Kombination dieser Kriterien) ein Gateway auswählen.

Beim Senden von Broadcast-Nachrichten sollte jedoch beachtet werden, dass solche Nachrichten extrem Ressourcen-intensiv sind. Da diese Nachrichten von jedem MN weitergeleitet werden müssen, verbrauchen sie bei jedem Host Batteriestrom und im ganzen Netz Bandbreite.

2.2 Aktives Auffinden

Aktives Auffinden bedeutet, dass ein MN bei Bedarf eine *Router Request*-Nachricht entweder als Broadcast-Nachricht oder an eine Multicast-Adresse sendet, um ein Gateway zu finden. Jedes Gateway, das diese Nachricht empfängt, antwortet anschließend mit einer *Router Response*-Nachricht. Dieser Prozess sollte einerseits bei der Initialisierung eines MN stattfinden, andererseits aber auch bei einer Verschlechterung der Multihopverbindung zum Gateway. Bei Bedarf kann dieser Prozess auch regelmäßig stattfinden.

2.3 MIPMANET

Bei MIPMANET wird das Gateway als *Foreign Agent* betrachtet, der analog zu Mobile IP regelmäßig *Agent Advertisements* per Broadcast versendet. Eine *Agent Advertisement*-Nachricht stellt keinen neuen Nachrichtentyp da, sondern erweitert die in RFC 1256 definierten ICMP-Nachrichten um mobilitätsspezifische Bestandteile [Schi03] und wird auch im Festnetz von Routern zur Bekanntgabe ihrer Dienste verwendet. Die obere Hälfte eines solchen Nachrichtenpakets besteht aus einem normalen ICMP-Paket, die untere Hälfte ist die notwendige Erweiterung. In dieser existieren Felder für eine Sequenznummer, die die Anzahl der bisherigen Ankündigungen anzeigt, für bestimmte Flags und mögliche *Foreign Agent Care-Of* Adressen. Über die Flags kann signalisiert werden, ob eine Registrierung beim *Foreign Agent* notwendig ist oder welche Kapselung verwendet werden soll (siehe 5.1).

Im Unterschied zu Mobile IP werden die *Agent Advertisement*-Nachrichten allerdings in größeren zeitlichen Abständen gesendet, um durch die dafür nötigen Broadcast-Nachrichten nicht zu viel Energie und Bandbreite zu verbrauchen: Während Mobile IP einen minimalen Abstand von eine Sekunde vorschreibt, wird bei MIPMANET mit einem Abstand von 5 Sekunden experimentiert [JALJ+00]. Dies bringt jedoch auch Nachteile mit sich: Das Registrieren und der Wechsel in ein neues Netz dauern länger bzw. werden instabiler.

Analog zu Mobile IP kann der MN bei MIPMANET den *Foreign Agent* auch aktiv durch *Agent Solicitations* suchen. Jedes Gateway, dass eine entsprechende Nachricht empfängt, antwortet mit einer *Agent Advertisement*-Nachricht.

2.4 Globalv4

Ebenso wie bei MIPMANET entspricht auch bei Globalv4 die Suche nach dem Gateway der Suche nach einem *Foreign Agent*, der seine Präsens durch *Agent Advertisements* anzeigen kann, die dem Mobile IP-Standard entsprechen [Perk01]. Diese Nachrichten werden von allen MNs weitergeleitet. Zusätzlich speichert der MN die IP-Adresse zusammen mit einer Sequenznummer. Falls er anschließend dieselbe Nachricht noch einmal erhält, braucht er dieses Duplikat nicht weiter zu verarbeiten, sondern kann es ignorieren. Nachdem er die Nachricht verarbeitet hat, sendet er sie über seine Schnittstelle zu anderen MNs. Um Kollisionen zu vermeiden, wartet er dabei eine zufällig bestimmte Zeit.

Wenn ein MN eine *Agent Advertisement*-Nachricht empfangen hat, in der das *R-Flag* gesetzt ist, muss er sich zwingend beim Gateway registrieren. Andernfalls ist die Registrierung nur notwendig, wenn der MN Internetzugriff erhalten möchte.

Um einen *Foreign Agent* aktiv zu suchen, kann ein MN eine *Route Request (RREQ)*-Nachricht senden. Diese Nachricht wird in dem entsprechenden Format und mit der „All Mobility Agents"-Multicastgruppenadresse `224.0.0.11` als Zieladresse per Broadcast an alle MNs in seiner Funkreichweite gesendet. Wenn ein anderer MN diese Nachricht empfängt, überprüft dieser zuerst, ob er gerade bei einem Foreign Agent registriert ist. Ist er bei keinem Agenten registriert bzw. unterstützt er kein Mobile IP, dann sendet er diese Nachricht erneut per Broadcast. Wenn der MN jedoch bei einem Agenten registriert ist und eine Verbindung zu diesem besteht, erzeugt er eine *Route Response (RREP)*-Nachricht, die im Prinzip dem AODV-Standard entspricht [PeRD01a], jedoch eine *Foreign Agent*-Erweiterung als Anhang hat. Diese Nachricht wird an den Sender der *RREQ*-Nachricht gesendet.

2.5 Globalv6

Bei Globalv6 hängt das für die Suche nach einem Gateway verwendete Verfahren von dem verwendeten Routingprotokoll ab. Bei proaktiven Routing-Protokollen wird ein aus Sicht des MNs passives Verfahren benutzt, bei dem das Gateway eine Nachricht namens *Internet-GateWay Advertisement (GWADV)* benutzt, die entweder als Teil des Routing-Protokolls oder des *Neighbor Discovery Protocols* versendet wird.

Sobald der MN eine *GWADV*-Nachricht empfangen hat, startet er die Adresskonfiguration und das Aufsetzen der Routen.

Da bei reaktiven Routing-Protokollen keine regelmäßigen Statusinformationen gesendet werden, muss hier der MN aktiv tätig werden und mit einer *Internet-GateWay Solicitation (GWSOL)*-Nachricht nach Gateways suchen, die mit einer *GWADV*-Nachricht antworten.

Als Absendeadresse für diese Operationen kann der MN eine beliebige routbare Adresse nehmen, z. B. die Mobile IPv6 Home Adresse. Wenn er keine derartige Adresse besitzt, muss er

eine temporäre Adresse mit dem *MANET_INITIAL_PREFIX* bilden, die er wieder löscht, sobald er Mithilfe des Gateways eine global routbare Adresse gebildet hat.

3 Adresskonfiguration

Ein wichtiger Unterschied zwischen den verschiedenen Mechanismen zur Adresskonfiguration betrifft eine mögliche *Care-Of* Adresse, an die bei Mobile IP der *Home Agent* die Datenpakete für den MN aus dem Internet weiterleitet. Der MN kann entweder eine eigene IP haben, die topologisch zum Netz des *Foreign Agent* passt. In diesem Fall spricht man von einer *co-located Care-Of* Adresse. Eine andere Möglichkeit ist eine *Foreign Agent Care-Of* Adresse. In diesem Fall benutzt der MN die Adresse des *Foreign Agent*, dieser leitet die Pakete für den MN dann an diesen weiter.

Um eine eigene *co-located Care-Of* Adresse zu bekommen, existieren zwei verschiedene Autokonfigurationsmechanismen: Ein zustandsbehafteter und ein zustandsloser Mechanismus.

3.1 Zustandslose Autokonfiguration

Die zustandslose Auto-Adresskonfiguration ist eine der Neuerungen, die das IPv6-Protokoll ermöglicht. Durch sie kann ein Host bereits nach Initialisierung der Schnittstelle und vor einem Auffinden eines Servers eine allerdings nur lokal gültige *Link Local*-Adresse konfigurieren. Durch einen Mechanismus namens *Duplicate Address Detection (DAD)* [PMWBR+01] wird bewirkt, dass diese Adresse innerhalb des lokalen Netzes einmalig ist und damit eine eindeutige Adressierung des MN innerhalb des Netzes ermöglicht. Für Ad-Hoc-Netze gibt es ein spezielles Ad-Hoc-Präfix (`fec0::ffff/64`) [PMWN+02], mit dem jeder MN eine so genannte *manet-local*-Adresse konfigurieren kann.

Um jedoch auch mit Stationen außerhalb des lokalen Netzwerkes kommunizieren zu können, braucht der MN eine global gültige Adresse. Dafür kann er entweder die Adresse aus seinem Heimatnetz weiterbenutzen oder er muss ggf. eine neue IP-Adresse konfigurieren, die zum Adressbereich des Gateways passt. In diesem Fall muss er nach dem Erhalt eines *Router Advertisements* eine neue Adresse konfigurieren, die auch einen Teil der per zustandslosen Autokonfiguration ermittelten lokalen Adresse enthalten kann.

3.2 Zustandsbehaftete Autokonfiguration

Die zustandsbehaftete Autokonfiguration unterscheidet sich dadurch von der zustandslosen, dass bei ihr eine zentrale Instanz eine Liste mit freien und belegten Adressen verwaltet und den MNs auf Anfrage freie Adressen zuweist.

Ein bereits aus IPv4 bekannter Mechanismus zur zustandsbehafteten Autokonfiguration ist DHCP [BCPD01]. Dazu benötigt man einen DHCP-Server auf dem Gateway, der die Adressen verwaltet und sie den MNs automatisch zuweist. Der MN kann diese Adresse z.B. bei der Suche nach einem Gateway lernen.

3.3 Foreign Agent Care-Of Adresse

Eine *Foreign Agent Care-Of* Adresse wird meistens dann verwendet, wenn entweder zu wenig Adressen zur Verfügung stehen (wie in vielen IPv4-Netzen) oder der MN aus anderen protokolltechnischen Gründen keine zusätzliche IP-Adresse benötigt (wie z.B. bei MIPMANET). Wird eine solche Adresse verwendet, endet damit der Tunnel vom *Home Agent* beim *Foreign Agent*. Dieser leitet die Pakete anschließend an den MN weiter.

3.4 Co-located Care-Of Adresse

Eine *co-located Care-Of* Adresse ist eine beim MN angesiedelte topologisch korrekte Adresse, die innerhalb des Subnetzes vom Gateway liegt und dadurch global routbar ist. In diesem Fall muss der MN also entweder mittels zustandsloser Adresskonfiguration eine neue Adresse mit dem passenden Präfix bilden oder eine solche vom Gateway anfordern. Der Tunnel endet damit beim MN direkt.[Schi03]

3.5 MIPMANET

Für MIPMANET werden keinerlei Adresskonfigurationsmechanismen benötigt. Bei diesem Protokoll wickelt der MN sämtliche Kommunikation über seine Home-Adresse ab, also die Adresse aus seinem Heimatnetz. Da die IP-Adressen der MNs innerhalb eines Ad-Hoc-Netzes in der Regel keinen topologischen Zusammenhang haben, muss ein Ad-Hoc-Routingprotokoll grundsätzlich topologisch nicht zusammenpassende Adresse routen können und kann folglich auch diesen MN erreichen [JALJ+00]. Um von außen erreichbar zu sein, wird eine *Foreign Agent Care-of* Adresse benutzt.

3.6 Globalv4

Um mit Rechnern außerhalb des lokalen Netzes kommunizieren zu können, benötigt ein MN eine global routbare Adresse. Falls der MN eine Adresse besitzt, die in seinem Heimatnetz gültig ist, muss er eine *Care-Of* Addresse anfordern, um anschließend durch Mobile IP-Verfahren seine Home Adresse dorthin umzuleiten.

Insgesamt gibt es drei verschiedene Arten, auf die ein MN bei Globalv4 eine *Care-Of* Adresse bekommen kann:

- 1. Er kann auf eine *Agent Advertisement*-Nachricht warten. Innerhalb dieser Nachricht können eine oder mehrere *Care-Of* Adressen enthalten sein. Wenn ein MN eine solche Nachricht erhält, kann er eine dieser Adresse aussuchen und als neue *Care-Of* Adresse konfigurieren.
- 2. Er kann eine solche Nachricht durch eine *Agent Solicitation*-Nachricht anfordern.
- 3. Er kann eine *co-located Care-Of Address* durch externe Mechanismen anfordern, z.B. durch die zustandsbehaftete Autokonfiguration per DHCP.

3.7 Globalv6

Bei Globalv6 muss der MN nach dem Erhalt der *GWADV*-Nachricht mittels zustandsloser Autokonfiguration eine global-gültige IPv6-Adresse als *co-located Care-Of* Adresse bilden, die aus dem vom Gateway vorgegebenen Präfix und der 64-bit langen Netzwerkkarten-ID als Host-Teil besteht. Dabei wird davon ausgegangen, dass der MN bereits eine *Link Local*-Adresse gebildet hat und diese auf Eindeutigkeit getestet hat. Deshalb muss die auf dem identischen Host-Teil basierende globale-gültige Adresse nicht erneut getestet werden; ansonsten muss der MN die *DAD* erneut durchführen.

Falls der MN eine mit dem *MANET_INITIAL_PREFIX* gebildete temporäre Adresse zum Suchen des Gateways benutzt hat, muss diese Adresse gelöscht werden, sobald er eine global-gültige Adresse konfiguriert hat. Außerdem müssen bei sämtlichen Durchgangs-Hosts und dem Gateway die Host-Routen für diese Adresse gelöscht werden. Dies wird durch entsprechende Routing-Protokoll-Nachrichten gemacht; bei AODV6 [PeRD01b] z.B. kann der MN eine *Route Error (RERR)*-Nachricht versenden.

4 Routing innerhalb eines MANETs

Beim Routing innerhalb eines Ad-Hoc-Netzes müssen zunächst einmal zwei unterschiedliche Netzstrukturen bedacht werden: Beim hierarchischen Routing, das z.B. bei Globalv6 verwendet wird, läßt sich das Netzwerk anhand der IP-Adressen in verschiedene Subnetze (*Cluster*) einteilen. Innerhalb dieses Subnetzes verfügen alle MNs über topologisch korrekte Adressen, z.B. über *co-located Care-Of* Adressen. Bei dem z.B. von MIPMANET und Globalv4 verwendeten flachen Routing sind dagegen keine weiteren Einteilungen vorhanden.

4.1 Hierarchisches Routing

Wenn ein MN innerhalb eines hierarchischen Ad-Hoc-Netzes Pakete an einen anderen Host senden möchte, muss er zuerst anhand der Ziel-Adresse (bzw. bei Mobile IP mit der *Care-Of-Address*) feststellen, ob dieser Host im selben Subnetz liegt [XiBe02]. Bei IPv6 muss er dazu das Netzwerk-Präfix der Zieladresse mit dem Präfix des lokalen Netzes vergleichen. Falls der Host in einem anderen Subnetz oder im Internet liegt, sendet er das Paket mit dem Ad-Hoc-Routingprotokoll an das konfigurierte Gateway.

4.2 Flaches Routing

Beim flachen Routing kann der MN nicht anhand der IP-Adresse feststellen, ob das Ziel im selben lokalen Ad-Hoc-Netz liegt und folglich direkt erreicht werden kann. Hier entscheidet das verwendeten Routing-Protokoll, ob er das Paket an das Gateway sendet oder direkt an das Ziel.

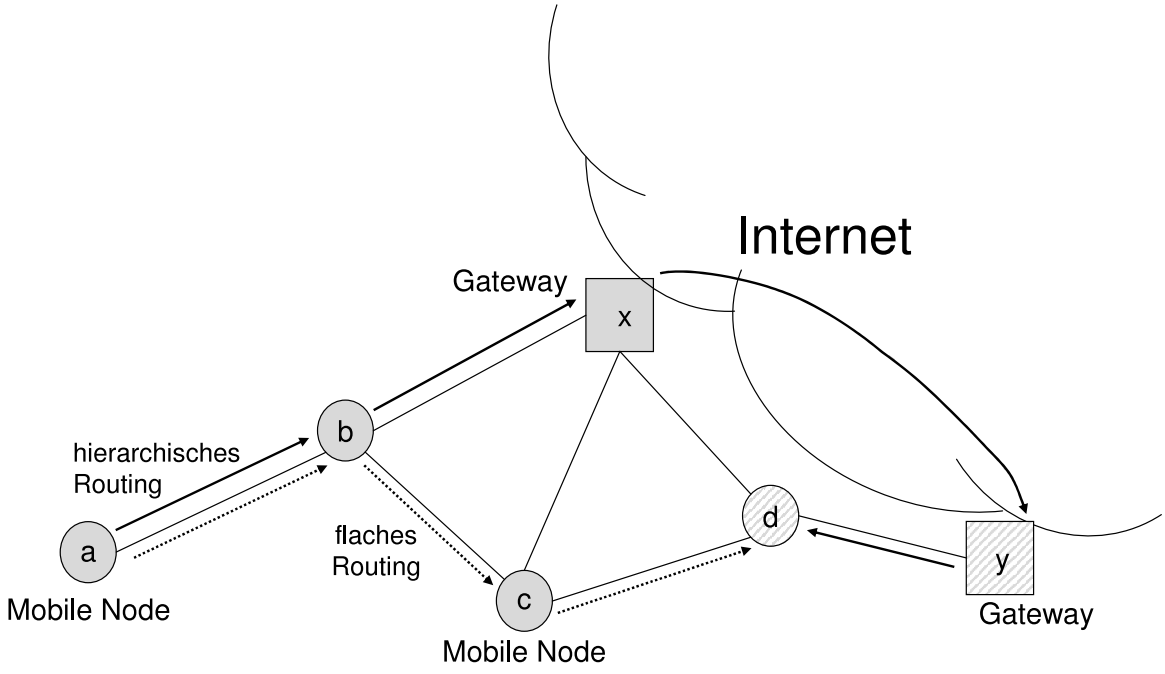

Abbildung 1: Beispiel für den Unterschied zwischen hierarchischem und flachem Routing

Bei dem abgebildeten Netz würde der MN a bei hierarchischem Routing den MN d, der beim Gateway y registriert ist, ausschließlich über den Weg $a \rightarrow b \rightarrow x \rightarrow y \rightarrow d$ erreichen können, bei flachem Routing wäre auch der Weg $a \rightarrow b \rightarrow c \rightarrow d$ möglich.

4.3 Proaktives Routing

Neben der Einteilung in hierarchisches und flaches Routing hat auch das verwendete Routing-Protokoll Einfluss auf den Routing-Prozess. Wenn ein proaktives Routing-Protokoll verwendet wird, hat der MN eine Routing-Tabelle mit allen Hosts, die innerhalb des Ad-Hoc-Netzes erreichbar ist. Um beim flachen Routing zu prüfen, ob ein Host lokal erreichbar ist, reicht ein Blick in diese Tabelle. Das eigentliche Senden eines Paketes beschränkt sich darauf, die zum Ziel-MN oder Gateway gehörende Host-Route zu ermitteln und das Paket entlang dieser Route zu senden.

4.4 Reaktives Routing

Bei einem reaktiven Routing-Protokoll werden keinerlei Host-Routen gespeichert. Sobald der MN also ein Paket versenden möchte, muss er die Route durch *Route Request*-Nachrichten neu bestimmen. Diese Nachricht wird von jedem MN weitergeleitet, bis sie schließlich ggf. beim Ziel ankommt. Dieses antwortet dann mit der *Route Response*-Nachricht, die über den Multihop-Pfad zurück gesendet wird und die genaue Route enthält. Falls es sich um flaches Routing handelt und ein Gateway, das einen Weg zum Home Agent des Ziels kennt, die *Route Request*-Nachricht empfängt, antwortet es ebenfalls.[XiBe02]

Der MN kann jetzt mit Hilfe spezieller Metriken den besten Pfad bestimmen. Beim flachen Routing ist es dabei irrelevant, ob Sender und Empfänger bei verschiedenen Gateways registriert sind. Der Weg muss nicht zwingend über die Gateways verlaufen - wenn der Weg durch das Ad-Hoc-Netz gemessen an bestimmten Metriken (siehe 2.1) „besser“ ist, wird dieser preferiert.

Beim hierarchischen Routing dagegen weiß der MN bereits, ob er die Daten über das Gateway versenden muss oder ob er sie normal über das Ad-Hoc-Netz senden kann.

4.5 MIPMANET

Da bei MIPMANET keinerlei topologisch zusammenhängende IP-Adressen verwendet werden, ist nur ein flaches Routing möglich. Ob ein Host im lokalen Netz ist oder nicht, ermittelt der MN über das Ad-Hoc-Routingprotokoll. Ist ein Host lokal erreichbar, wird das Paket lokal zugestellt.

4.6 Globalv4

Bei Globalv4 muss der MN zuerst prüfen, ob das Ziel über das Ad-Hoc-Netzwerk erreicht werden kann. Falls nicht, wird es über das Gateway gesendet. Da Globalv4 momentan hauptsächlich für reaktive Routing-Protokolle wie AODV konzipiert ist, wird für diese Prüfung das Senden von *Route Request*-Nachrichten gefordert. Diese Nachrichten werden durch das gesamte Netzwerk geleitet. Sobald sie von einem Gateway empfangen werden, prüft dieses zunächst, ob es eine Host-Route für die Zieladresse gespeichert hat. Eine solche Route hat es für jeden MN, der sich bei ihm registriert hat. Falls er einen passenden Eintrag findet, antwortet es mit einer normalen *Route Response*-Nachricht. Falls nicht, antwortet es mit einer *Route Response*-Nachricht, bei der durch das *F-Flag* signalisiert wird, dass das Ziel innerhalb eines anderen Netzwerkes liegt.

Wenn ein MN eine *Route Response*-Nachricht mit gesetztem *F-Flag* empfängt, sendet es seine Daten nicht unmittelbar über diese Route, sondern wartet erst noch eine bestimmte Zeit auf

das Eintreffen von anderen *Route Response*-Nachrichten, da der direkte Weg durch das Ad-Hoc-Netzwerk möglicherweise länger dauern kann. Falls ein direkter Weg existiert, wird dieser auf jeden Fall bevorzugt.

4.7 Globalv6

Nachdem der MN bei Globalv6 seine Adresse konfiguriert hat, muss er folgende Routen in seine Tabelle eintragen: Zum einen eine Default-Route, deren Ziel das Gateway ist und zum anderen eine Host-Route zum Gateway, dessen Ziel der nächsten MN ist, über den er das Gateway-Advertisement empfangen hat. Diese Einträge haben eine bestimmte Lebensdauer, die vom Gateway festgelegt wird. Die so genannte Default-Route darf von reaktiven Routing-Protokollen allerdings nicht als allgemeine Default-Route benutzt werden.

Innerhalb reaktiv gerouteter Netze muss der MN zuerst in seiner Routing-Tabelle nach einem Eintrag für die Zieladresse suchen. Die Default-Route darf er zu diesem Zeitpunkt noch nicht benutzen. Wenn er keinen Eintrag gefunden hat, sendet er einen *Route Request*. Nur wenn er darauf innerhalb einer bestimmten Zeit keine Antwort bekommt, darf er das Paket über die Default-Route senden.

Wenn ihm vom Gateway signalisiert worden ist, dass er sein Ziel eigentlich auch direkt erreichen können müsste, er aber vermutet, dass sich das Gateway direkt zwischen ihm und dem Ziel befindet, der direkte Weg also zwangsläufig über das Gateway führt, kann er diese Kontrollmitteilung ignorieren und das Paket trotzdem zum Gateway senden.

Wird Globalv6 zusammen mit einem proaktiven Routing-Protokoll verwendet, kann der MN das Routing hauptsächlich diesem Protokoll überlassen. Wenn dieses eine Route zum Host gespeichert hat, wird das Paket über diese Route gesendet; ebenso bei einem Ziel, das topologisch ins lokale Netz gehört, also das passende Präfix hat. Die Default-Route wird nur für Ziele benutzt, bei denen beide Kriterien nicht zutreffen.

Bei einem reaktiven Routing-Protokoll hat das Gateway ebenso wie bei Globalv4 die Aufgabe, Host-Routen für alle MNs zu speichern, die sich bei ihm registriert haben oder per *GWSOL*-Nachricht nach einem Gateway gesucht haben.

Ebenso wie bei Globalv4 sendet es auch Kontrollmitteilungen, wenn es Pakete empfängt, für deren Zieladresse er einen Routing-Eintrag gespeichert hat. Wenn ein MN das Ad-Hoc-Netz verlässt, muss das Gateway dies bemerken und ggf. die Host-Routen löschen, falls diese nicht bereits aufgrund abgelaufener Lebensdauer gelöscht worden sind. Zum Feststellen dieser Topologiewechsel kann das Gateway *Route Error*-Nachrichten auswerten. Falls ein MN das Netz verlassen hat, der eine IP-Adresse mit dem Präfix des Gateways hatte, muss das Gateway anschließend alle Pakete für diese Adresse mit einer *ICMP Unreachable*-Nachricht beantworten. Hatte der MN eine Mobile IP *Home Address* konfiguriert, muss das Gateway die Pakete an den *Home Agent* weiterleiten und darf keine *ICMP Unreachable*-Nachrichten verschicken.

5 Routing zwischen MANETs und dem Internet

Um ein Paket Richtung Internet oder innerhalb des Netzes zu senden, gibt es unabhängig vom verwendeten Netzaufbau und Ad-Hoc-Routing-Protokoll auch noch zahlreiche Unterschiede zwischen den vorliegenden Entwürfen, die die Art und Weise des Sendens betreffen (verwendete IP-Adressen, mögliche Kapselung, Zieladresse).

Ein MN kann bei hierarchischem Routing anhand der IP-Adresse feststellen, dass das Ziel nicht lokal erreichbar ist. Bei flachem Routing kann er dazu das Routing-Protokoll befragen.

Wenn er festgestellt hat, dass er das Ziel nur über das Gateway erreichen kann, muss er zunächst eine Route dorthin bestimmen. Bei proaktiven Protokollen reicht dazu ein Blick in seine Routingtabelle, bei reaktiven Protokollen muss er einen *Route Request* für die Adresse des Gateways über das Netzwerk versenden. Wenn das Gateway über einen Multihop-Pfad mit dem MN verbunden ist, kann es die Nachricht empfangen und mit einer *Route Reply*-Nachricht antworten. Anschließend kann der MN die Daten über den etablierten Pfad zum Gateway senden. Dieses leitet sie anschließend an den Empfänger innerhalb des Internet weiter.

Für den Weg in die entgegengesetzte Richtung, also von einem Internet Host zu einem MN, wird bei Mobile IP folgender Mechanismus verwendet: Der Internet Host sendet seine Pakete an die Heimatadresse des MN. Dort wird es vom *Home Agent* abgefangen. Dieser sendet es über einen Tunnel zur *Care-Of* Adresse.

5.1 Tunnel

Um Pakete tunneln zu können, müssen sie innerhalb des IP-Protokolls gekapselt werden. Das bedeutet, dass ihnen ein neuer Header mit anderen IP-Adressen vorangestellt wird und das so erzeugte neue Paket anschließend wie ein normales Datenpaket weitergeschickt wird. Für diese Kapselung existieren verschiedene Protokolle [Schi03]: Es gibt die IP-in-IP Kapselung, die in RFC 2003 [Perk96] spezifiziert wird. Sie stellt die einfachste Form von Kapselung da und bedeutet, dass dem Paket ein vollständiger IP-Header vorangestellt wird. Da durch die zwei IP-Header allerdings manche Felder doppelt vorhanden sind, werden mehr Daten übertragen als notwendig sind und es entsteht unnötiger Overhead. Deshalb existiert unter dem Namen „Minimale Kapselung“ ein zweiter Standard, bei dem im zweiten (also inneren) IP-Header nicht benötigte Felder weggelassen werden. Um auch andere Protokolle tunneln zu können gibt es zusätzlich noch einen dritten Standard: *Generic Routing Encapsulation.* Dieses Protokoll erlaubt mit Hilfe von Sequenznummern die Sicherstellung der Reihenfolge, in der die Pakete gesendet worden sind.

5.2 MIPMANET

Alle Pakete, die ein MN zu Rechnern außerhalb des lokalen Netzes senden möchte, werden über ein Tunnel vom MN zum *Foreign Agent* gesendet. Dieser entpackt die Pakete wieder und sendet sie normal über das Internet weiter.

Wenn ein Internet-Host Pakete an einen MN senden möchte, werden diese vom *Home Agent* empfangen. Dieser sendet sie über einen Tunnel zum *Foreign Agent* und von dort werden sie als normales IP-Paket zum MN weitergeleitet.

5.3 Globalv4

Wenn der MN bei der Suche nach einer Route zu seinem Ziel eine Route findet, die über das Gateway führt, wird das Paket über normale IP-Paketweiterleitungsmechanismen an dieses Gateway gesendet. Ein Tunnel ist dabei nicht notwendig. Das Gateway leitet es anschließend weiter.

5.4 Globalv6

Wenn ein Gateway bei Globalv6 ein Paket aus dem Internet empfängt, dass an einen MN innerhalb des lokalen Netzes adressiert ist, sucht es ggf. eine Host-Route und leitet es dann normal weiter. Ein besonderes protokoll-spezifisches Verhalten ist dabei nicht notwendig.

6 Schlussbetrachtung

Nach genauerer Betrachtung der verschiedenen Protokolle stellt sich heraus, dass sie sich zwar ziemlich ähnlich sind, in einigen wichtige Punkte allerdings unterscheiden.

So bauen alle Entwürfe entweder auf Mobile IP auf oder arbeiten damit relativ eng zusammen. Insgesamt ist erkennbar, dass die Entwicklung in Richtung IPv6 geht und das auf IPv4-basierende Globalv4 offenbar weniger diskutiert bzw. weiterentwickelt wird. Von diesem Entwurf existiert nur eine Version vom November 2001; bislang sind keine neueren Versionen eingereicht worden.

Dieser Trend entspricht den aktuellen Tendenzen in der Forschung, in der die Kombination aus Mobile IP und IPv6 als Basis für ein Mobilkommunikationssystem der 4. Generation (auch 4G genannt) betrachtet wird [Kemp02].

Unterschiede gibt es allerdings bezüglich des empfohlenen Ad-Hoc-Routingprotokolls. MIPMANET empfiehlt ein proaktives Protokoll, da dieses besser zu Mobile IP passen würde und insbesondere Bewegungserkennung und die Suche nach Gateways besser funktionieren würde. Für diese Vorteile wäre aber ein hoher Preis in Form eines hohen Overheads durch Statusnachrichten notwendig. Deshalb wird in anderen Texten [PMWN+02] auch eher davon abgeraten.

Bezüglich der IP-Konfiguration wird von Globalv4/Globalv6 und anderen Texten eine *co-located Care-Of* Adresse empfohlen, um teilweise auch ein hierarchisches Routing zu ermöglichen. Grundsätzlich wird aber eher ein flaches Routing bevorzugt; insbesondere bei MIPMANET ist dies die einzig mögliche Variante, da dort nur topologisch nicht zusammenhängende Adressen verwendet werden..

Eine andere Gemeinsamkeit ist das Fehlen von einigen Protokollfunktionen:

- In keinem Entwurf wird beschrieben, wie Gruppenkommunikation (z.B. Multicast) aus dem Internet oder ins Internet geroutet werden kann.
- In keinem Entwurf werden Möglichkeiten der Zusammenarbeit von verschiedenen Gateways zur Lastverteilung aufgezeigt; Globalv4 und Globalv6 gehen nur von einem Gateway aus.

Literatur

[BCPD01] J. Bound, M. Carney, Charles E. Perkins und R. Droms. Dynamic host configuration protocol for IPv6 (DHCPv6). Internet Draft, Juni 2001.

[BRSP01] Elizabeth M. Belding-Royer, Yuan Sun und Charles E. Perkins. Global Connectivity for IPv4 Mobile Ad hoc Networks. Internet Draft draft-royer-manet-globalv4-00.txt, November 2001.

[CaPi03] Giovanni Camponovo und Yves Pigneur. Analyzing the m-business landscape. *Annals of Telecommunications* Band 58, 2003.

[JALJ+00] Ulf Jönsson, Fredrik Alriksson, Tony Larsson, Per Johansson und Gerald Q. Maguire Jr. MIPMANET - Mobile IP for Mobile Ad Hoc Networks. *Proc. ACM MobiHoc*, 2000, S. 75–85.

[Keen03] Ian Keene. Öffentliche WLAN-Hotspots weltweit, 2002-2008, Mai 2003. http://www.intel.com/deutsch/ebusiness/strategies/wireless/hotspot.htm.

[Kemp02] James Kempf. 4G Access Network Architectures, 2002. http://snrc.stanford.edu/events/industry-seminar/fall02/abstracts/kempf.pdf.

[PeRD01a] Charles E. Perkins, E. Royer und S. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. Internet Draft draft-ietf-manet-aodv-09.txt, November 2001.

[PeRD01b] Charles E. Perkins, E. Royer und S. Das. Ad Hoc On Demand Distance Vector (AODV) Routing for IP version 6. Internet Draft draft-perkins-manet-aodv6-01.txt, November 2001.

[Perk96] Charles E. Perkins. IP Encapsulation within IP. RFC 2003, Oktober 1996.

[Perk01] Charles E. Perkins. IP Mobility Support for IPv4. Internet Draft draft-ietf-mobileip-rfc2002-bis-08.txt, September 2001.

[PMWBR+01] Charles. E. Perkins, J. T. Malinen, R. Wakikawa, E. M. Belding-Royer und Y. Sun. IP address autoconfiguration for ad hoc networks. Internet Draft, November 2001.

[PMWN+02] Charles E. Perkins, Jari T. Malinen, Ryuji Wakikawa, Anders Nilssion und Antti J. Tuominen. Internet connectivity for mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 2002, S. 465–482.

[Schi03] Jochen Schiller. *Mobilkommunikation.* Pearson Education. 2003.

[WMPN+03] Ryuji Wakikawa, Jari T. Malinen, Charles E. Perkins, Anders Nilsson und Antti J. Tuominen. Global connectivity for IPv6 Mobile Ad Hoc Networks. Internet Draft draft-wakikawa-manet-globalv6-03.txt, Oktober 2003.

[XiBe02] Jin Xi und Christian Bettstetter. Wireless Multihop Internet Access: Gateway Discovery, Routing, and Addressing. *Proc. Intern. Conf. on 3G Wireless and Beyond (3Gwireless'02)*, Mai 2002.

Abbildungsverzeichnis

Testumgebungen für REAL WORLD Ad Hoc Netzwerke

Christian Matuschewski

Kurzfassung

In der Gesellschaft vermehrt sich zunehmend immer mehr der Wunsch, immer und überall mobil und dabei „online“ zu sein. Viele Anwendungen, die wir heute in Angriff nehmen wollen, sind jedoch noch lange nicht mit den derzeitigen Systemen so gut zu realisieren, dass wir sie auch ohne Probleme immer und überall in Echtzeit nutzen können. Bisher gibt es zwar schon einige Netzwerklösungen, wie WLAN, Bluetooth oder Infrarotschnittstellen (IR), diese können jedoch bei weitem nicht die Herausforderungen der Zukunft in vollem Umfang befriedigen. Ad Hoc Netzwerke sind bisher aber nur beim Militär oder als Simulation in z.B. Universitäten zu finden. In dieser Ausarbeitung soll nun ein Einblick in reale Testumgebungen für Ad Hoc Netzwerke gezeigt werden und welche Probleme es noch bei der Umsetzung sowie welche Lösungsmöglichkeiten es dafür gibt, damit die Tests möglichst real gestaltet werden können.

1 Einleitung: Was ist ein Ad Hoc Netzwerk und warum sollten wir es nutzen

Ein Ad Hoc Netzwerk ist ein drahtloses Netzwerk, welches aus vielen kleinen einzelnen Knoten (z.B. Notebooks, PDA, Handys) besteht. Diese einzelnen Knoten können sich spontan zu einem neuen dynamischen Netzwerk bilden, dass sich ständig in Bewegung befindet und sich immer wieder neu verändert. Diese Art von Netzwerk hat im Entwurf keine zentrale Verwaltung oder eine Infrastruktur, die im Hintergrund immer aufgebaut werden muss, wie etwa bei WLAN. Das Problem bei Ad Hoc Netzen ist, dass dadurch das es keine zentrale Verwaltung gibt, jeder einzelne Knoten für sich selbst arbeiten muss und Daten sowohl empfangen als auch senden, aber auch wie ein Router Datenpakete weiterleiten soll. Es gibt also keine Basisstationen so wie in Bluetooth Netzen, über die Daten versendet werden.

Die heutigen Standardnetze benötigen alle eine meistens sehr große Infrastruktur mit Basisstationen, Router, Leitungen usw., sowie viele Personen die diese Hardware ständig überwachen und bei einem Ausfall (der auch schon mal das ganze Netz betreffen kann) schnell diese Struktur wiederherstellen können. Diese sehr teuren Infrastrukturen würde man bei Ad Hoc Netzen überhaupt nicht benötigen, was dazu führt, dass man jederzeit und an jedem Ort ein sog. „spontanes Netzwerk“ bilden könnte. Dies wäre z.B. auch bei Rettungseinsätzen an denen Orten von Vorteil, wo es nicht immer ein Netzwerk oder Funksignal für Telefone gibt oder wo es einfach zu teuer wäre ein solches Netz aufzubauen, weil die Nachfrage dort einfach zu gering ist. Vor allem aber in Katastropheneinsätzen kann man heute nicht sagen, ob überhaupt solche Netzwerke wie Telefon, Handy und Internet aufrechterhalten bleiben können. Auf Ad Hoc Netze könnte man sich selbst in solchen Notsituationen verlassen, da sie von allen anderen Systemen unabhängig laufen und keinen Zugriff z.B. auf wichtige Datenbanken haben müssen. Die meisten Einsätze von Ad Hoc Netzen gibt es bisher im Militärbereich. Gerade dort werden Netzwerke benötigt, die jederzeit und schnell aufgebaut werden können, damit einzelne Einheiten ohne eine vorhandene Infrastruktur miteinander kommunizieren können.

Wenn es bereits existierende Ad Hoc Netze gäbe, hätten diese den Vorteil, dass sie mit allen anderen Systemen kompatibel sind und sich ohne Probleme z.B. an das Internet anschließen lassen. Ein weiterer Vorteil der Knoten, nämlich ihre Mobilität, ist zugleich auch ein großer Nachteil. Das Problem bei solchen Knoten ist, dass sie sich ihre Energie sehr gut einteilen müssen da sie meistens entweder mit Batterien oder Akkus betrieben werden und keine ständige Stromquelle besitzen. Da diese Knoten aber ständig Daten weiterleiten, auch wenn diese nicht direkt an sie gesendet wurden, kann diese Energie schnell zu Ende gehen. Ein anderes Problem ist die Bandbreite in Ad Hoc Netzen, die bisher meistens sowieso unter der Bandbreite von WLAN liegt. Diese Bandbreite kann sehr variieren. Selbst bei einer Verbindung von A nach B und der Rückrichtung von B nach A können hohe Unterschiede auftreten, da man nicht genau sagen kann, wie sich die Daten den Weg über welche anderen Knoten durchs Netz suchen. Im Festnetz z.B. kann man genau sagen, dass die Daten von A nach B und zurück immer den gleichen Weg nehmen. In Ad Hoc Netzen könnte es wenn das Routing Protokoll nicht optimal erstellt ist auch dazu kommen, dass die Datenpakete nicht immer den optimalen Weg von A nach B nehmen und so ein großer Zeitverlust entsteht. Hinzu kommt noch, dass durch die ständige Mobilität vieler Knoten und die damit verbundene Änderung der Netz-Topologie die Übertragungswege sich ständig ändern und ständig neu gesucht werden müssen. Dies ist die größte Herausforderung für die Protokolle, ansonsten würden solche Netze nur noch Protokolldaten senden und kaum zum übertragen von anderen Daten kommen.

Die Datenübertragungsreichweite wird in Ad Hoc Netzen nicht wie in WLAN in Metern, sondern sie wird in „Hops“ (ein Sprung von einem zum anderen Knoten) von der Basis zum Endknoten oder in Bitmetern angegeben. Im Ad Hoc Netz können Knoten miteinander direkt kommunizieren, hingegeben gibt es bei WLAN in der Regel nur Verbindungen von Knoten zu Basis-Stationen.

2 Die Testumgebungen von REAL WORLD Ad Hoc Netzen

In dieser Ausarbeitung stelle ich zwei Testumgebungen genauer vor. Die „Ad Hoc City“-Umgebung am Beispiel der Stadt Seattle im Staate Washington (USA) ist dabei eine Simulation eines großen Ad Hoc Netzes, während die APE-Tests auf real aufgebauten Testläufen beruhen.

2.1 Die „Ad Hoc City“ Architektur

Bisher bestanden gewöhnliche Netzwerke aus vielen kabelgebundenen Basisstationen, über die die verschiedenen Knoten miteinander kommuniziert haben. Diese Basis-Stationen mussten dabei so platziert werden, dass jeder Knoten direkten Kontakt zu mindestens einer Station hatte. Das heißt am Beispiel der Stadt Seattle, dass jeder Winkel der Stadt lückenlos von einer Basis abgedeckt werden muss, wobei jedoch die Überlappung der Signalbereiche so gering wie nur möglich gehalten werden sollen, damit keine Basis-Stationen verschwendet werden (Stationen sind teuer). Das „Rahmenwerk“ (Backbone) der mobilen Knoten, war also bisher immer kabelgebunden, hatte einen großen Hardwareaufwand und war recht teuer beim Aufbau. Dazu kam, dass es in so einer Stadt nicht so einfach war, für so viele Stationen eine Genehmigung zur Aufstellung an so vielen Orten zu bekommen.

An der Rice University hat man in der neuen „Ad Hoc City“ Architektur diese Basis-Stationen auf acht Stück reduziert. Die Abdeckung der restlichen weitaus größeren Fläche der Stadt wurde durch viele weitere kleine mobile Knoten vorgenommen. Diese Knoten befinden sich jeweils an den Bussen des städtischen Nahverkehrs sowie auf Kleintransportern der örtlichen Zulieferdiensten. Es wird angenommen, dass diese Fahrzeuge die Stadt Seattle vollständig und

zu jeder Zeit mit einem Signal bedecken können. Die acht Basis-Stationen müssen dadurch nun nicht auf den Meter genau in der Stadt platziert werden, denn die entstehenden Versorgungslücken können die mobilen Knoten auf den Bussen und Transporter sehr gut schließen. Lücken, die sich durch den Ausfall einer Station auftun können dann durch andere Stationen an anderen Bussen aufgefangen werden, die gerade in der Gegend sind. Ausfälle müssen nun nicht mehr zwingend zu einer Lähmung des Netzes führen und alle weiteren mobilen Knoten haben immer Kontakt zu einer Basis-Station, sei es direkt oder sei es über mehrere Hops (Sprünge) von Knoten zu Knoten bis zur Basis. Die mobilen Endnutzer können sich dann auch mit ihren mobilen Geräten (Laptops, PDA, Handy) über dieses mobile RRahmenwerkmmit dem Internet verbinden lassen. So konnte durch die mobilen Knoten auf den Bussen ein Netzwerk mit etwa 750 bis 850 Knoten simuliert werden.

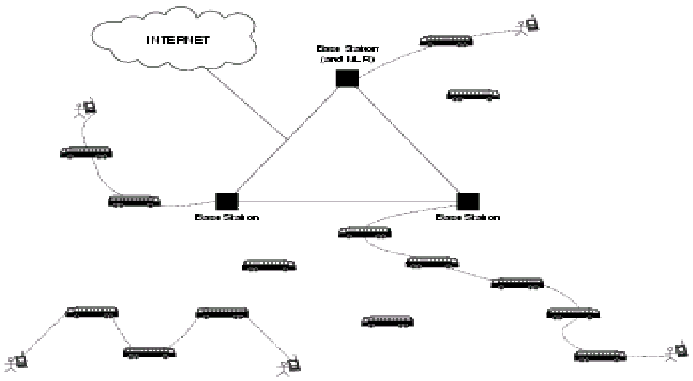

Abbildung 1: Architektur der „Ad Hoc City“ Seattle [JHPS+]

Jeder Knoten hat eine Basis, bei der er sich meldet und nach Wegen zu anderen Knoten fragt. Die Haupt-Basis im Ad Hoc ist aber immer diejenige, die am nächsten gelegen ist, was durch den Hop-Count bestimmt wird (Anzahl der Sprünge bzw. Hops zur Basis). Im Ad Hoc Netz kann die Verbreitung des Signals von einer Basis zu den noch Knoten, die noch von dieser Basis mit einem Signal bedient werden, vorher eingestellt werden. Der Abstand wird hier wiederum in Hops gezählt (h_b max).

Zusammengefasst besteht die Architektur der Simulation aus:

- acht Basis-Stationen, die an bestimmten Stellen in Seattle stehen sollen
- dem „mobilen Rahmenwerk“, also ca. 1200 mobilen Knoten, die an Bussen und Kleintransportern montiert wurden. Dies garantiert eine größere Ausbreitung bzw. eine bessere Abdeckung der Fläche in Raum und Zeit. Gleichzeitig kann eine größere Verlässlichkeit des Netzes garantiert werden, da ausgefallen Knoten leichter durch andere Knoten an anderen Bussen kompensiert werden können. Dieses „Rahmenwerk“ funktioniert wie ein Router
- am Ende der Kette steht der mobile Endbenutzer mit seinem Laptop, PDA, Handy. Dieser kann über mehrere Knoten hinweg Daten senden/ empfangen oder sich über die Basis mit dem Internet verbinden lassen (Verbindung zu einem kabelgebundenen System). Er selbst funktioniert nicht als Router, da er zu wenig Energie hat und die Sicherheit der Datenübertragung dann zu kompliziert wäre (dann kann jeder die über ihn laufenden Daten von anderen Knoten „mithören“).

Ein Internetzugang ist für diese Knoten auch möglich. Zur Vereinfachung der Verbindung an das Internet wird einfach angenommen, dass das zentrale „Mobile Location Register“ (MLR)

welches im Hintergrund arbeitet und alle Wege von der Basis zu den Knoten speichert, durch einen Präfix am Anfang der Daten erkennt, ob diese Daten für einen Knoten innerhalb des Ad Hoc Netzes oder für das Internet bestimmt sind.

Da es bisher keine allgemeine realistische Knotenbewegung gibt, die bei allen Simulationen verwendet wird, war die Realitätsnahe Knotenbewegung zunächst ein großes Problem für die Entwickler. Man musste sich überlegen, wie die mobilen Endnutzer sich später einmal in der Stadt Seattle bewegen würden und wie man diese Bereiche immer mit einem Signal abdecken kann. Sie kamen zu der Überlegung, dass man die Busflotte des städtischen Nahverkehrs nehmen und alle Busse mit einem mobilen Gerät bestücken könnte. Die Begründung lag in der Annahme, dass diese Busse zu jedem Zeitpunkt in einem vorher bestimmbaren Bereich der Stadt sich befinden sollten und dort die mobilen Endnutzer mit einem Dienst versorgen könnten. Daraufhin wurde die Bus-Flotte in Seattle mehrere Wochen lang 24 Stunden am Tag beobachtet. Dank einer Software der „Seattle Metro“ konnten alle Positionen der Busse zu bestimmten Zeitpunkten ohne großen Aufwand betrachtet werden. Dabei wurde allerdings vernachlässigt, dass die Daten in dieser Software nicht gleichmäßig aktualisiert werden, d.h. die einzelnen Busse werden in unterschiedlichen Zeitschritten aktualisiert. Trotzdem wurden alle Daten dieser Software in die Simulationsumgebung der „Ad hoc City“ übernommen, um damit die Knotenbewegung zu realisieren.

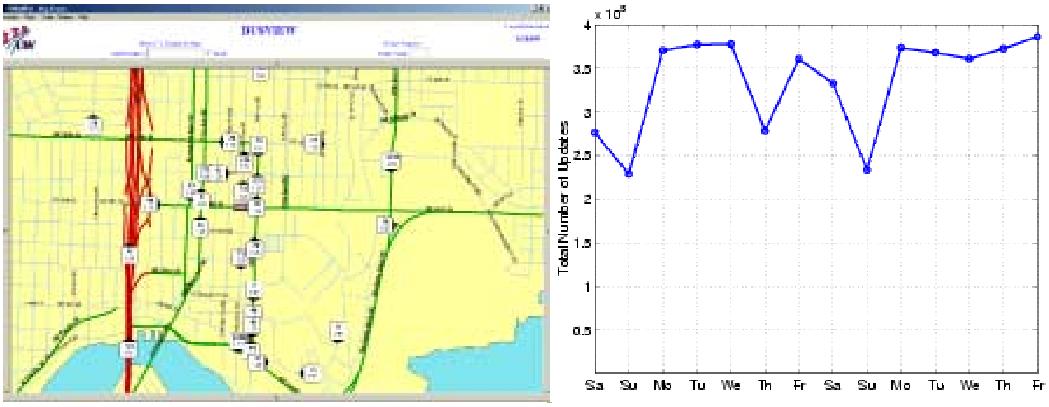

Abbildung 2: Online-Anzeige des Nahverkehrs von Seattle und Anzahl der Aktualisierungen der Bus-Anzeige pro Tag[bus]

2.1.1 Das DSR Protokoll

Als Routing Protokoll wird in der „Ad Hoc City“ das Protokoll DSR („Dynamic Source Routing Protocol“) benutzt. Die Beispiel-Daten im Test waren jeweils 4 x 64 Byte große Pakete.

DSR ist ein „on-demand“ Routing Protokoll für Netzwerke, d.h. es werden nur bei Anfrage Datenpakete gesendet. Dies spart Energie und vor allem auch Bandbreite, denn ansonsten werden ständig neue Wege gesucht, obwohl sie eventuell erst später oder sogar gar nicht gebraucht werden. Das Protokoll wurde schon oft in anderen kleineren Ad Hoc Netzen benutzt und hat dort bisher ganz gute Ergebnis geliefert. Für so große Netze wurde das Protokoll aber nicht entwickelt. Um es besser verwenden zu können wird es deswegen in einer etwas modifizierten Version als C-DSR Protokoll benutzt.

2.1.2 C-DSR (Cellular-DSR)

Folgende Änderungen wurden vorgenommen:

Unter anderem können bei einer Weg-Suche zu einem Ziel-Knoten die umliegenden Knoten abgefragt werden, ob sie der Ziel-Knoten sind. Falls der Knoten nicht gefunden wird, wird die Standard-Wegewahl-Suche („Route Discovery“) eingeleitet.

Die Daten müssen dabei nicht unbedingt über eine Basis-Station gesendet werden, damit sie am Ziel ankommen. Falls es mehrere Wege zum Ziel-Knoten gibt, wird der direkte Weg ohne Basis gewählt, falls dieser in einer bestimmten Hop-Zahl (h_b)erreichbar ist. So können Knoten auch direkt miteinander kommunizieren. Damit beim senden von Datenpaketen jedoch keine Schleifen durchlaufen werden, werden die ID-Nummern der Knoten mitgeschickt, über die die Daten gesendet werden.

2.1.3 Verbindungsaufbau und Wegewahl mit C-DSR

Beim Verbindungsaufbau muss immer beachtet werden, dass zu jedem Zeitpunkt ein Knoten z.B. zu einer anderen Basis wechseln oder sich stark bewegen kann und ein anderer Weg dadurch viel effizienter wäre, oder dass der Knoten kurzfristig sogar gar kein Signal mehr von sich gibt. Diese Fälle muss das Protokoll ständig überwachen und notfalls wieder einen neuen Weg suchen.

Wenn ein Weg durch das Netz gesucht wird, wird als erstes der eigene Cache des Quell-Knoten nach einem Eintrag durchsucht. Ist dort kein Eintrag über den Ziel-Knoten zu finden, wird ein „Route Discovery“ an die nächsten Knoten versendet, die sich in einem Abstand von h_b Hops befinden. Falls der Knoten gefunden wird, wird sofort nach der Antwort gesendet. Falls kein Knoten gefunden wurde, wird die nächste Basis gesucht die sich in einem maximalen Abstand von h_b Hops befindet (die sog. „Home-Basis“). An diese Basis wird ein „Route Request“ gesendet, damit die Basis-Station in seinem Cache nachschauen kann, ob dort ein Eintrag steht. Die Basis enthält alle Wege zu den Knoten, für die sie als naheste Basis erreichbar ist oder für die es zuletzt eine Weg-Anfrage gab. Falls dort ein Eintrag steht, geht ein „Route Reply“ mit samt dem Weg zurück an den Quell-Knoten (man muss beachten, dass der Cache der Basis nicht aktuell sein muss, da der Knoten sich ja bewegen kann und in der Zwischenzeit irgendwo anders ist, er sich aber nicht abmelden muss bei der Basis). Falls es keinen Eintrag gibt sendet die Basis-Station an die zentrale MLR („Mobile Location Register“) eine Anfrage, damit diese in ihrem Cache nach einem Eintrag sucht. Wird auch dort kein Eintrag gefunden sendet die MLR an alle Basis-Stationen ein „Request“ und jede Basis-Station sendet daraufhin ein „Local Route Request“ an alle seine Knoten. Falls eine Basis so den Knoten findet wird der Weg zurück an die MLR gemeldet. Diese aktualisiert ihren Register und meldet den Weg an die Quell-Basis. Die Quell-Basis kann nun dem Quell-Knoten den Weg melden. Die MLR und die Basis können jedoch jeweils immer nur eine Anfrage bearbeiten. Sobald eine weitere Anfrage kommt während sie gerade eine Weg suchen, werden diese in die Warteschleife gesetzt bis die letzte Anfrage erledigt wurde. Wichtig bei der Weiterleitung von Daten ist, dass sie am Anfang immer einen Overhead haben damit ein Knoten weiß, woher sie gerade kommen und welchen Weg sie genommen haben. Dabei werden immer alle Knoten ID´s über die die Daten wandern, an diesen Präfix drangehängt. So können Daten die verworfen werden (z.B. weil sie zu lange schon im Netz sind oder weil es den Ziel-Knoten gar nicht gib) beim Absender kenntlich gemacht werden.

Damit die Registrierungstabelle der MLR immer aktuell bleibt, werden dort immer neben der „Route Request ID“ des Knoten auch immer die Hop-Anzahl zum Knoten hin (damit erkennt man, ob der Knoten gewandert ist). Falls der Quell-Knoten noch nicht in der MLR registriert

ist, wird die Registrierungstabelle erst aktualisiert und danach wird der Ziel-Knoten gesucht. Wechselt ein Knoten zu einer neuen Basis, informiert dieser Knoten die neue Basis bei der ersten Weg-Anfrage, dass sie die neue Home-Basis ist (durch Abfrage der Hop-Anzahl zur Basis). Dies wird auch in der MLR aktualisiert.

Verbindungsaufbau:

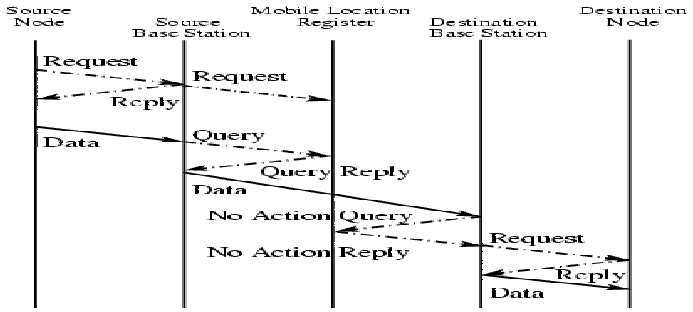

Abbildung 3: Wegefindung und Arbeitsprozess der Basis-Stationen[JHPS$^+$]

2.1.4 Ergebnisse („Ad Hoc City“ Architektur)

Insgesamt wurden bei dieser Simulation sechs Testdurchläufe vorgenommen.

Alle Testläufe waren genau 900 Sekunden lang.

Man beobachtete, dass ca. 92 bis 97 Prozent aller Datenpakete zum richtigen Empfänger geleitet wurde. Um ein Datenpaket zum Empfänger zu senden muss C-DSR im Schnitt allerdings 222 Routing-Pakete pro Knoten versenden. Das heißt es werden etwa 10 Mal so viele Routing-Pakete versendet als eigentliche Daten-Pakete. Die meisten Daten konnten jedoch trotzdem sehr schnell versendet werden, da nur 21 Prozent aller Pakete über eine Basis transportiert werden mussten. Es gab also viele direkte Verbindungen von Knoten zu Knoten. Die mittlere Latenz betrug 86 ms, wobei die mittlere Übertragungszeit bei nur 42,5 ms lag.

Insgesamt kann man die Performance von C-DSR als „gut“ beurteilen. Das Protokoll konnte die Datenpakete zufriedenstellend übertragen und es gab nur selten eine Versorgungslücke. Trotzdem muss noch einiges am Protokoll C-DSR verbessert werden wenn es auch wirklich einmal eingesetzt werden soll. Unter anderem müssen die Kontrollpakete weiter minimiert und die Wege zum Zielknoten schneller aufgebaut werden.

2.2 Die APE Testumgebung

Im Vergleich zu der „Ad Hoc City“ Simulation, wollte man mit der APE („Ad Hoc Protocol Evaluation“ Testbed) Testumgebung wirklich durchgeführte Tests darstellen. Wichtig war auch hier wieder, dass alle Netze leicht skalierbar sind. Ziel war es ein „Rahmenwerk“ zu modellieren, in dem die mobilen Knoten oder das Routing Protokoll leicht austauschbar sind. Dazu wurden drei Szenarien entworfen, welche Personen mit ihrem Laptop durch Anweisungen des Computers in einem Gebäude nachlaufen sollten. Diese Choreographien blieben immer die selben, da dadurch typische Situationen nachgebildet wurde.

APE sollte keine Simulation werden, weil sich dort nicht alles so realitätsnah darstellen lässt wie gewünscht. Die dynamischen Knotenbewegungen können dort ebenso wie die komplexe Signalausbreitung und Datenübertragung nur in einem gewissen Rahmen simuliert werden.

Im APE Testbett konnte man nun in realen Testumgebungen Tests durchführen und diese dann in Simulationen nachprüfen oder ggfs. die Simulationen anpassen.

Wie bei allen Tests ist es wichtig, dass sie auch reproduzierbar sind und vergleichbare Ergebnisse herauskommen. Um diese Ergebnisse genauer miteinander vergleichen zu können wurde eine Metrik eingeführt. Die sogenannte „Virtual Mobility Metric“ (vM) basiert statt auf der bei anderen Metriken üblichen geometrischen Distanz zwischen zwei Knoten, auf der tatsächlich gemessenen Signalqualität beim Empfänger.

2.2.1 Aufbau des Experiments

Die Tests finden statt innerhalb eines Gebäudekomplexes. Dabei geben die Buchstaben A bis H Punkte an, wo Knoten später platziert werden sollen. Die Wände der Gebäude sind dabei so dick, dass es kein Signal von einem zum anderen Ende des Testgeländes gibt.

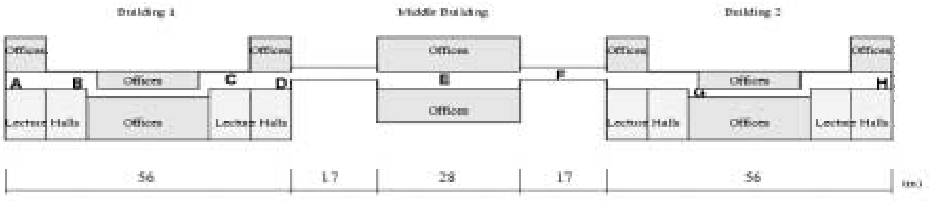

Abbildung 4: Gebäudeschema, wobei Punkt E immer der Ursprungspunkt für die Knoten ist.[sou]

Es werden drei verschiedene Szenarien getestet. Das Ziel ist es hier zu untersuchen, wie verschiedene Bewegungen und die dabei entstehenden Wegbrüche durch die sog. v_M-Metrik grafisch dargestellt werden.

Szenario 1: „Lost´n´found“

In diesem Szenario teilt sich eine Zelle von Knoten in zwei Zellen und verliert dabei für kurze Zeit den Signalkontakt. Danach gehen alle Knoten wieder zum Ursprungspunkt zurück und die Verbindung wird wieder aufgebaut. Während des Versuchs senden alle Knoten einen Broadcast Ping an alle anderen Knoten.

Szenario 2: „Double lost´n´found“

Es trennen sich zwei Zellen von dem Ursprung ab und laufen in verschiedene Richtungen. Dabei verlieren diese zwei Zellen wiederum den Kontakt zur anderen Zelle. Danach kehrt zunächst Zelle1 zurück zum Ursprung und damit zu den noch dort verbliebenen Knoten und erst kurz darauf kehrt die Zelle2 wieder zurück zum Ursprung. Beide Zellen kriegen also wieder Kontakt zum Rest, jedoch passiert dies zu unterschiedlichen Zeitpunkten. Während des Versuchs senden die jeweiligen Zellen unicast Pings zu den anderen beiden Zellen.

Szenario 3: „Double Split“

Hier wird untersucht ob Datenpakete auch über mehrere Knoten hinweg zu einem anderen Knoten senden können, zu dem sie keinen direkten Kontakt haben. Es befinden sich zunächst zwei Zellen an zwei verschiedenen Punkten im Gebäude (D und G), an denen sie gerade noch Signalkontakt zur jeweils anderen Zelle haben. Im Ursprung sind keine Knoten. Danach trennen sich von den zwei Zellen jeweils noch mal die Hälfte und laufen weiter zum Rand des Gebäudes (A und H). Es entstehen vier verschiedene Zellen, die jedoch immer Kontakt zu den benachbarten Zellen haben. Hier werden immer unicast Pings von einer Zelle in die jeweils anderen drei Zellen gesendet.

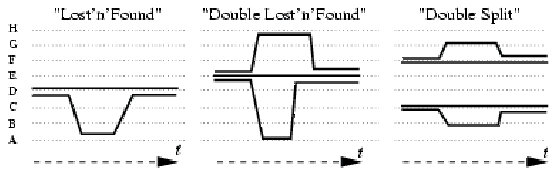

Abbildung 5: Weg-Zeit-Diagramm[sou]

2.2.2 Die „Virtuelle“-Mobilitäts-Metrik („Virtual“ Metric v_M)

Die v_M-Metrik berechnet die Stellungs-Änderungen der Knoten in einem virtuellen Abstand zwischen den Knoten. Diese Änderungen hängen unter anderem ab von der Signalqualität zwischen den Knoten ab. Diese kann durch verschiedene äußere Einwirkungen wie Störungen, Reflektionen oder Signalneutralisation bei zu nahe beieinander liegenden Knoten stark beeinflusst werden. Statt nun der echten Distanz zwischen zwei Knoten zu nehmen, berechnete man für das APE-Projekt die „virtuelle“ Distanz über die tatsächlich wahrgenommene Signalqualität des Daten empfangenden Knoten. So erhält man eine Metrik, die die reale wahrgenommene Dynamik aufzeigt, die zwischen den Knoten herrscht. Diese Metrik wird nun auch dazu benutzt, um die verschiedenen Testläufe miteinander zu vergleichen.

Alle wichtigen Daten und Messergebnisse der v_M-Metrik, sowie die Wegewahl der Datenpakete im Netzwerk, die Übertragungszeit der Daten oder verlorengegangene Pakete, werden in einer Log-File auf jedem einzelnen Laptop gespeichert und dann an einen zentralen Rechner gesendet, der alles auswertet.

Da Testpersonen die Szenarien nachlaufen mussten und wissen mussten, wie sie sich in den Räumen bewegen sollen, wurden ihnen klare Anweisungen über die Bildschirme der Laptops gegeben. Diese Anweisungen konnten von einer vorher geschriebenen Software Sekundengenau den Testpersonen mitgeteilt werden. So mussten die Personen nicht einmal wissen, worum es bei diesem Experiment genau geht und sie mussten keine großen Programmier-Kenntnisse haben, um das Programm zu bedienen. Das Programm, dass man zum Starten der Benutzeroberfläche haben musste konnte einfach durch die Testpersonen runtergeladen werden. Sobald man die Software gestartet hatte wurde der PC neu gebootet und automatisch (unter Linux) gelangte man zur Benutzeransicht. So konnte die Konfiguration sehr einfach gehalten werden, damit auch wirklich viele Knoten getestet werden konnten.

2.2.3 Analyse der Daten (APE)

Es gibt verschiedene Möglichkeiten, die Bewegungen in einer Testumgebung zu bewerten. Um zu sehen, wie stark die einzelnen Knoten in Bewegung waren bzw. wie stark und wie schnell sich ein Wegbruch bildete und wie schnell sich wiederum ein neuer Weg gebildet hat, kann man z.B. die Mobilitätsmetrik (die sog. Minimal-Wegänderungs-Metrik, in Englisch: „minimal route change metric“) von Johnson nehmen. Durch sie kann man sehen, dass eine hohe Knotenbewegung auch eine hohe Anzahl an Wegbrüchen mit sich bringt. Andersherum kann man dann aber auch sagen, dass je weniger Wegbruche, desto weniger Bewegung gibt es zwischen den Knoten.

Johannsson hingegeben beschrieb in einem seiner Werke die „Geometrische-Mobilitäts-Metrik“ („geometric mobility metric“). Diese Metrik basiert mehr auf den physikalischen Positionen der Knoten und wie sie ihre Positionen ändern. In einer Simulation kann man diese Positionen zu jedem Zeitpunkt ohne Probleme genau sagen, jedoch kann man dies in einem Test wie in

APE nur bedingt. Außerhalb von Gebäuden ist dies per GPS noch relativ einfach, jedoch innerhalb von den Testgebäuden ist dies kaum möglich (die Wände waren zu dick). Da man nicht unnötig noch große Arbeit in die Ortung der Knoten stecken wollte und durch die Metrik sowieso einer der wichtigsten Punkte, nämlich die Signalstärke und Signalausbreitung, nicht messbar gemacht wird, hat man von dieser Metrik Abstand genommen.

$$\frac{2}{N(N-1)T}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\int_{t=0}^{T}\left|\frac{d\|P_j(t)-P_i(t)\|_2}{dt}\right|dt$$

Abbildung 6: Formel der geometrischen Mobilitäts-Metrik

Wobei gilt:

P_k(t)=Position des Knoten k zum Zeitpunkt t, T = Zeitdauer des Szenarios, N = Anzahl der Beteiligten Knoten; Zusätzlich wird die Summe wird über alle Knoten-Paare während der gesamten Zeitdauer des Tests berechnet.

2.2.4 Berechnung von vM

Um die oben genannte „virtuelle“ Distanz nun mit der Signalqualität tatsächlich zu verbinden, wurde das Pfadverlust-Modell benutzt. In diesem Modell wird vorhergesagt, dass in Gebäuden ein Weg-Verlust Koeffizient von 3.3 existiert. Damit kommt man zu folgender Formel:

Q in dB = a - 33 * log(dist/ß)

Nach der Kalibrierung mit der Signalqualitätsspanne der WLAN-Karte der benutzten Laptops und den eigenen Messungen, konnte folgende Wege-Verlust-Formel aufgestellt werden:

$$D_j(node_i) = 4 * 10^{\frac{40-0.9*Q_j(node_i)}{33}}$$

Abbildung 7: Wege-Verlust-Formel

Wobei gilt: Q ist die WLAN Signalqualität (0 bis 75) für ein empfangenes Paket von Knoten j am Knoten i. D ist die Entfernung zwischen 0,5 und 65 Meter (max. Entfernung in den APE Tests).

„Virtuelle“ Mobilität zwischen Knoten(i) und Knoten(j) wird für Knoten(i) folgendermaßen berechnet:

$$D_j^k(node_i) = \frac{1}{N_j^k}\sum_{a=1}^{N_j^k} D_j^a$$

Abbildung 8: Berechnung der Mobiltät zwischen den Knoten

Für ein gegebenes Zeitintervall t_k werden nun von allen virtuellen Abständen der Mittelwert von allen Paketen genommen, die von einem bestimmten Knoten(j) mitgehört wurden. Es wurde dann D_k^j definiert, die mittlere virtuelle Distanz zum Knoten(j) für einen Zeitschlitz t_k, während N_k^j die Zahl der empfangenen Pakete von Knoten(j) während t_k ist. D_j^a ist

der virtuelle Abstand, den wir durch die Signalqualität eines Paketes a erhalten, dass von Knoten(j) während t_k empfangen wurde.

Die virtuelle Mobilität v_M von Knoten(i), unter Berücksichtigung von Knoten(j), ist für das Zeitintervall t_{k+1} einfach die Änderung des Durchschnittlichen virtuellen Abstands, nämlich:

$$vM_j^{k+1}(node_i) = |D_j^{k+1}(node_i) - D_j^k(node_i)|$$

Abbildung 9: Durchschnittlicher virtueller Abstand

Und die durchschnittliche virtuelle Mobilität, die von Knoten(i) zum Zeitpunkt t_k empfangen wird ergibt sich zu:

$$vM_{avg}^k(node_i) = \frac{1}{S}\sum_{l=1}^{S} vM_l^k(node_i)$$

Abbildung 10: Durchschnittlich empfangene virtuelle Mobiltät

Hierbei sei S die Zahl der Knoten, über die v_M berechnet wird.

Die virtuelle Netzwerk Mobilität zum Zeitpunkt t_k hingegen ist:

$$vM^k = \frac{1}{N}\sum_{i=1}^{N} vM_{avg}^k(node_i)$$

Abbildung 11: Virtuelle Netzwerk Mobiltät

Wobei hier N die Anzahl der Knoten im Netzwerk sind. Dadurch kommt man zur Ermittlung von Durchschnitts v_M -Netzwerk-Werten zu jedem Zeitpunkt. Dieser Wert gibt an, wie ein Durchschnitts-Knoten sich während des Test bewegt. Dazu kann man auch noch das obere und untere Quantil bestimmen, indem man einfach die 25 Prozent am stärksten (am wenigsten) sich bewegenden Knoten nimmt. Dadurch kann man zeigen, wie homogen die Bewegungen im Netz sind. Wenn sich alle Knoten ähnlich stark bewegen, müssen auch das obere und untere Quantil ähnlich zum Durchschnitt sein. Wenn hingegen sich nur wenige Knoten im Netzwerk bewegen weichen die Quantildarstellungen stark vom Durchschnitt ab. Am einfachsten werden der Durchschnittswert und die Quantile durch eine Grafik deutlich gemacht, die in Abhängigkeit von der Zeit (t in s) dargestellt wird.

2.2.5 Ergebnisse (APE Testumgebung)

Die drei oben genannten Szenarien wurden mehrmals mit einer immer unterschiedlichen Anzahl von Knoten durchgeführt. Durch die v_M-Metrik können wir diese Test nun miteinander vergleichen.

Zunächst mussten alle Metrik-Daten in den Log-Files erst mal grafisch dargestellt werden. Dabei mussten Log-Files untersucht werden, die manchmal 70 MB groß waren und bis zu 3 Mio. Einträge enthielten.

Die Grafiken zeigen nun die virtuelle Mobilität im Netzwerk (in m/s) in Abhängigkeit von der Zeit (in Sekunden).Zusätzlich wurden auch das obere und untere Quantil eingezeichnet

und mit v_M-high und v_M-low bezeichnet. Wie oben bereits beschrieben sind dadurch nur die Knotenbewegungen der 25 Prozent am meisten (wenigsten) sich bewegenden Knoten eingezeichnet. Beachtet werden muss, dass obwohl wenn alle Knoten am gleichen Punkt stehen bleiben, es trotzdem eine kleine Ausschwankung der Kurve geben kann (d.h. v_M zeigt eine Bewegung der Knoten an), weil es immer ein gewisses Signal zwischen sehr eng beieinander liegenden Knoten gibt.

2.2.6 Charakterisierung der Szenarien mit v_M

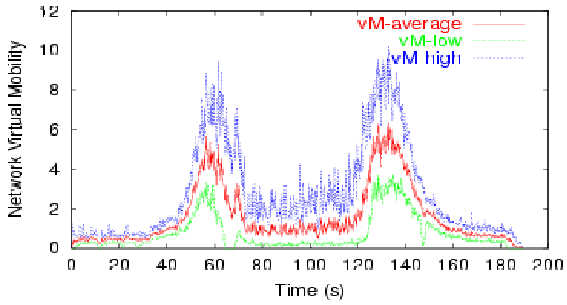

Abbildung 12: „Lost´n´Found“ mit 20 Knoten[sou]

Zuerkennen sind zwei ganz deutliche Spitzen, und zwar sind die genau zu dem Zeitpunkt, als die Zellen den Kontakt verlieren (bei 60s) und dann wiederfinden (bei 120s). Während der Zeit, als die beiden Zellen keinen Kontakt haben, kehrt die Kurve wieder fast in ihre Ausgangseben zurück, so als ob die beiden Zellen noch zusammen wären.

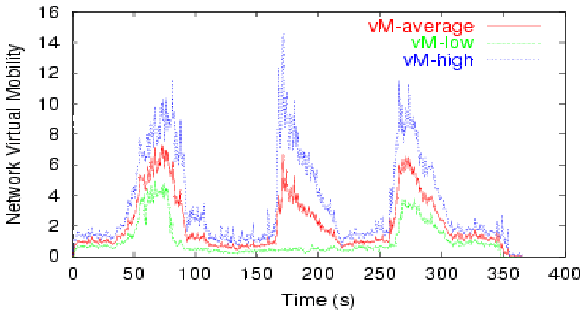

Abbildung 13: „Double Lost´n´Found“ mit 33 Knoten[sou]

Auch beim „Double Lost´n´Found“ Szenario sind folgerichtig drei Spitzen zu erkennen und auch diese befinden sich genau zu den Zeitpunkten, wenn die zwei Zellen sich trennen. Wobei die letzten beiden Spitzen die Zusammenführung zeigen. Zunächst kehrt die erste Zelle und kurze Zeit später auch die zweite Zelle zum Ursprung zurück.

Wenn man v_M sich genauer ansieht (nicht die Quantile) zeigt sich, dass die Trennung hier eine größere Mobilität in der Breite erzeigt, als es die Zusammenführung tut. Dies ergibt sich daraus, dass am Anfang alle Knoten zusammen den Ursprung verlassen, am Ende jedoch zuerst die Eine, danach die zweite Zelle der Knoten zurückkommt.

Interessant ist auch die v_M-low Kurve. Hier gibt es nämlich kaum eine richtige Spitze, d.h. es gibt einige Knoten, die sich kaum oder gar nicht bewegen. Dies liegt daran, dass ja eine

Zelle immer im Ursprung bleibt und sich nicht bewegt und für kurze Zeit keinen Kontakt mit den anderen beiden äußeren Zellen hat. Dadurch können wir mit v_M also sehen, ob sich eine Gruppe bzw. ein Netzwerk von Knoten heterogen oder eher homogen verhält, d.h. ob sich alle Knoten gleich stark bewegen.

2.2.7 Erfolgreiche Pings und Multi-Hopping über mehrere Knoten

Durch die Untersuchung, ob nun alle Datenpakete bzw. Pings bei den Zielknoten ankommen kann man erkennen, wie gut Daten durch das Netz gesendet werden können. Dies wird sehr gut durch das Verhältnis der tatsächlich empfangenen Pings angezeigt.

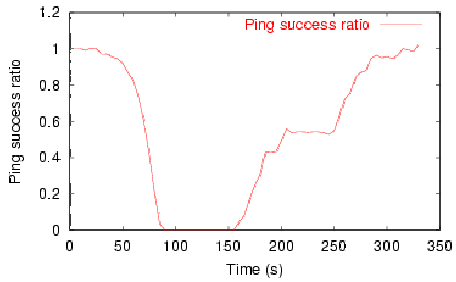

Abbildung 14: Erfolgreiche Pings zwischen den Knoten[sou]

Man sieht hier am Beispiel des „Double Lost´n´Found“ Szenarios, dass am Anfang als alle Knoten noch zusammen waren, die Pings zu fast 100 Prozent beim Zielknoten angekommen sind. Sobald jedoch die Knoten langsam wandern und irgendwann kein Signal mehr von den anderen Zellen empfangen (zwischen 90 und 160s) fällt die Rate. In der Grafik mit der v_M-Kurve (siehe oben) sieht man genau an diesem Zeitpunkt die großen Spitzen (d.h. Trennung und bei t=160 kommt die erste Zelle wieder langsam zurück). Das Verhältnis steigt weiter und bleibt dann zunächst auf einem Level (t=200s), bis bei t=250s die zweite Zelle mit den Knoten wieder zum Ursprung zurück kehrt.

Eigentlich wollte man wissen, ob die jeweiligen Zellen am Rand ihre Datenpakete über die mittlere Zelle zur anderen äußeren Zelle senden. Dies ist allerdings nicht so, wie man anhand des Hop-Zählers unten gut sehen kann. Sogar im Szenario3 „Double Split“, wo wir vier Zellen haben die jeweils zu ihren Nachbarzellen Kontakt haben, hat das Multi-Hopping (d.h. die Übertragung von Daten über mehrere Knoten) nicht funktioniert.

Travel distance:	1 hop	2 hops	3 hops
Request pings	1127	1	–
Reply pings	1127	1	–
Complete pings	n/a	1126	2

Abbildung 15: Anzahl der erfolgreichen Pings und die dazugehörige Anzahl der Hops zum Zielknoten (Szenario2 „Double Lost´n´Found“ mit 34 Knoten)[sou]

Travel distance:	1 hop	2 hops	3 hops
Request pings	405	–	–
Reply pings	405	–	–
Complete pings	n/a	405	–

Abbildung 16: Anzahl der erfolgreichen Pings und die dazugehörige Anzahl der Hops zum Zielknoten (Szenario3 „Double Split“ mit 34 Knoten)[sou]

Die gleichen Tests wurden jedoch mit einem weiteren Routing Protokoll gemacht (OLSR-Inria), dort wurde festgestellt, dass hier das Multi-Hopping ohne weiteres funktioniert. Hier war es sogar möglich bis zu vier Hops zu machen. Dies kann auf sieben Hops sogar erhöht werden, wenn man in beide Richtungen von einem Knoten aus geht.

2.2.8 Reproduzierbarkeit von Testdurchläufen

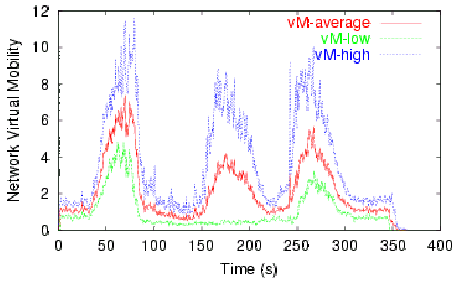

Abbildung 17: „Double Lost´n´Found“ mit nun 34 Knoten[sou]

Man siehtin der Abbildung, das obwohl hier mehr Knoten eingesetzt wurden (nun 34, vorher 33), die v_M-Metrik trotzdem wunderbar die gleichen Bewegungen anzeigt. Es gibt wieder drei Spitzen aus den gleichen Gründen wie vorhin (Trennung, Wiedervereinigung). Wenn man diese mit der letzten v_M-Grafik des gleichen Szenarios vergleicht stellt man fest, dass die v_M-Metrik Testläufe sehr gut miteinander vergleichbar macht, weil ja außer der Anzahl der Knoten nichts geändert wurde.

3 Zusammenfassung aller Testergebnisse

Zusammenfassend kann man sagen, dass z.B. die Reproduzierbarkeit von Tests möglich ist. Dies kann man wunderbar anhand der v_M Grafiken sehen, die zu den gleichen Choreografien jedoch mit einer unterschiedlichen Anzahl von Knoten gezeichnet wurden, jedoch immer sehr ähnlich aussehen. Genauso wurde anhand der verschiedenen Tests mit immer mehr Knoten gezeigt, dass die Ad Hoc Netze alle skalierbar sind (wobei sie aber trotzdem irgendwann an eine Grenze stoßen). Dabei hat uns die v_M-Metrik gut geholfen, diese Testdurchläufe miteinander vergleichbar zu machen und die Bewegungen der Knoten zu charakterisieren. Nebenbei hat man herausfinden können, dass die Zeit zum Auffinden eines Wegbruches stark variiert. Genauso wie die Zeit zum finden eines neuen Weges für die Datenpakete. Dies dauerte in den Test manchmal bis zu einer Minute. Dies wäre jedoch für alltägliche Anwendungen viel zu lange, deshalb muss hier das Routing wohl noch etwas verbessert werden. Gerade auch deswegen und weil es bisher nur kaum Protokolle für diese Art von Netzwerken gibt, wird in den APE-Test eher die Qualität der Protokolle (z.B. die Multi-Hop Fähigkeit) als die Richtigkeit der Protokolle und deren Implementierung getestet. Es ist also wichtiger die Routing Protokolle so zu verbessern, so dass die Wegefindung noch schneller funktioniert und Kontrollpakete die Bandbreite nicht noch mehr schmälern. Das APE Testbett kann dann später dazu benutzt werden die Implementierung zu Analysieren und dann auch um eine Art Benchmark für Ad Hoc Protokolle aufzustellen. Sobald es noch weitere Ad Hoc Routing Protokolle gibt, können auch diese in das APE Testbett implementiert und mit den gegebenen Testbedingungen und den bisherigen Protokollen verglichen werden. In naher Zukunft sollte dies auch mit jeweils 50 und mehr Knoten möglich sein.

Um diese Test auch anderen Institutionen zu ermöglichen, wird die APE-Software auch anderen Personen im Internet frei zugänglich gemacht. Diese sollten im Allgemeinen auch keine Probleme damit haben, da die Software so einfach wie möglich gehalten wurde. Ein weiteres Ziel ist es, die bisherigen Testergebnisse in Simulationen weiterzuverwenden und dort noch zu vertiefen. Dadurch können sich auf lange Sicht hin vielleicht die dann bis dahin existierenden und verbesserten Simulationen gegen die realen Experimente durchsetzen (was den großen Zeitaufwand für die Test verkleinern würde).

Die Simulation der „Ad hoc City“ Architektur hingegen hat viele Nachteile und ist bei weitem nicht so realistisch, wie das APE-Modell. Allein schon die Knotenbewegung ist nicht so optimal gelöst. Die Entwickler dachten sich dabei, dass sie durch die verschiedenen Untersuchungen (u.a. wochenlange Beobachtung der Fahrtwege der Busse in Seattle) sehr nah an die Realität ran kämen, aber äußere Einflüsse wie Wetter, Signalstörungen und Signalauslöschungen wurden völlig vernachlässigt und deshalb kann diese Knotenbewegung nicht als realistisch angesehen werden. Auch die Übertragung der Daten ist sehr unrealistisch, schließlich wurde hier einfach angenommen, dass alle Daten zu 100 Prozent richtig empfangen werden und absolut keine Daten verloren gehen, was in der Realität nie so passieren wird. Dazu kommt auch noch, dass die Knoten alle unendlich viel Energie hatten. Dadurch konnten zufällig auftretende Ausfälle von einzelnen Knoten erst gar nicht untersucht werden. In der realen Welt wird dies aber wohl leider (zumindest am Anfang) relativ häufig vorkommen.

Ein ebenso wichtiges Problem wie die Datenübertragung ist die Sicherheit in einem solchen Netz. Bisher gibt es weder in der Simulation noch in dem Test ausreichende Vorkehrungen gegen Eindringlinge von Außen. Dies wird aber auch erst dann wirklich interessant, wenn die Routing Protokolle verbessert wurden und das Verhältnis zwischen Kontroll- und Datenpakete sich weit verbessert hat.

Als Fazit kann man ziehen, dass noch weitere Tests gebraucht werden, um in der Zukunft große funktionstüchtige Netze laufen lassen zu können. Jedoch ist es einfach nicht so leicht 50, 100, 1000 oder mehr Knoten in der realen Welt zu testen. In naher Zukunft können Ad Hoc Netze aber schon bald die derzeitigen Netzwerke ablösen.

Literatur

[bus] im Internet unter: „busview.org“.

[JHPS+] Jetcheva, Hu, PalChaudhuri, Saha und David B. Johnson. Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture.

[sou] im Internet unter: „sourceforge.net/projects/apetestbed“.

Abbildungsverzeichnis

Mobile TCP-Varianten unter Beibehaltung der Ende-zu-Ende Semantik

Björn Zülch

Kurzfassung

Mit der zunehmenden Verbreitung von mobilen Geräten und Computern erwächst daraus auch die Forderung nach Nutzung des Internet zu jeder Zeit und an jedem Ort. Dies stellt nun vollkommen neue Anforderungen an die existierenden Protokolle im Internet. Diese Protokolle wurden unter dem Gesichtspunkt von leitungsgebundener Medien und dem Prinzip der Ende-zu-Ende Semantik entwickelt. Die leitungsgebundenen Medien haben eine vollkommen andere Charakteristik aufzuweisen. Um nun diese Protokolle weiterbenutzen zu können sind Erweiterungen und Verbesserungen notwendig, damit die Protokolle auch im drahtlosen Umfeld eingesetzt werden können. Im Rahmen dieser Arbeit sollen Varianten und Erweiterungen für das Transport-Control Protokoll (TCP) und deren Mechanismen vorgestellt werden. Diese versuchen mit unterschiedlichen Verfahren die Effizienz zu erhöhen ohne jedoch dabei das Prinzip der Ende-zu-Ende Semantik zu zerstören.

1 Einleitung

Die Kommunikation hat sich in den letzten Jahren verändert. War vor einigen Jahren überwiegend leitungsgebundene Kommunikation möglich, so hat sich dies stark gewandelt. Heute kommt kaum jemand ohne drahtlose oder mobile Kommunikation aus. Bestes Beispiel sind die öffentlichen Mobilkommunikationsnetze GSM und neuerdings UMTS. Im privaten Bereich existieren noch die Wireless-LAN Netzwerke (WLAN) mit diversen Standards. Im Zusammenhang mit der weiteren Verbreitung des Internets, besteht die Forderung der Nutzung der gleichen Protokolle auch in mobilen und drahtlosen Geräten. Dies erfordert eine Anpassung der vorhandenen Protokolle auf die neuen Umgebungsbedingungen, da die bisherigen Protokolle nur auf die Kommunikation im leitungsgebundenen Bereich hin ausgelegt sind. Diese sollen so modifiziert werden, dass sie auch in drahtlosen Umgebungen eingesetzt werden können.
In dieser Seminararbeit sollen Mechanismen und Protokolle vorgestellt werden, die helfen sollen, die Kommunikation über drahtlose Netze effizienter zu gestalten, indem Mängel bestehender Protokolle beseitigt werden.

2 Mobile Netzwerke

Ein Endsystem ist mobil, wenn es möglich ist sich mit diesem Endsystem frei zu bewegen. Dies erlaubt Benutzern die Nutzung aller Dienste, die auch an stationären Endsystemen verfügbar sind, überall und zu jeder Zeit. Mobile Netzwerke lassen sich in zwei Kategorien einteilen. Zum einen sind dies die infrastrukturbasierten Netzwerke wie GSM, UMTS und Wireless-LAN(z.B. IEEE 802.11) im Infrastruktur-Modus. Es existieren auch drahtlose

Strecken innerhalb der Kernnetze der Berteiber. Dabei handelt es meinst um Richtfunkübertragungsstrecken. Überwiegend handelt es sich aber um leitungsgebundene Kommunikation innerhalb der Kernnetze. Oft ist auch nur der letzte Hop einer Verbindung drahtlos. Für die Entwicklung und Erweiterung der Transportprotokolle wird meist ein Szenario entwickelt, in dem der Dienstanbieter (Server) im Kernnetz (leitungsgebunden) anzusiedeln ist und der Dienstnehmer (Client) ein drahtlos angebundenes Endsystem ist. Im weitern Verlauf wird dieses Szenario betrachtet. Dabei ist der Sender der Dienstanbieter und der Empfänger das drahtlose Endsystem.
Die zweite Kategorie von Netzwerken sind die mobilen Ad-hoc Netzwerke (MANET), bei denen keine Infrastruktur vorhanden ist, sondern alle Strecken über eine drahtlose Schnittstelle realisiert werden. In diesem Szenario befinden sich Dienstgeber und Dienstgeber im drahtlosen Bereich.

2.1 Probleme mobiler Netzwerke

Die drahtlosen Netzwerke haben nicht nur Vorteile. Diese Netze haben auch Nachteile, die sich auf die Kommunikation auswirken und durch die Mobilität entstehen können.

- *Hohe Bitfehlerraten:*
  Drahtlose Netzwerke haben bedingt durch ihre drahtlose Übertragung zwischen einer Basisstation und dem mobilen Endsystem im Vergleich zu drahtgebundenen Netzwerken oft eine höhere Bitfehlerraten. Jedoch ist dies je nach verwendeter Modulation und Codierung unterschiedlich. Diese können durch kurze oder lange Signaleinbrüche bedingt sein, z.B Interferenzen mit anderen Signalen.

- *Verbindungsabbrüche:*
  Verbindungsabbrüche können durch verschiedene Ereignisse ausgelöst werden :

  1. Zu hohe Bitfehlerraten
  2. Drahtloses Endsystem befindet sich ausserhalb der Reichweite der Funkbasisstation
  3. Zu viele Benutzer sind innerhalb eines Bereiches und keine weiteren Ressourcen stehen zur Verfügung
  4. Funksignale werden durch Hindernisse wie Gebäude oder ähnliches geblockt
  5. Ein drahtloses Endsystem bewegt sich von einer Funkzelle in eine andere Funkzelle und ein Handover zwischen den benachtbarten Zellen wird ausgelöst. Während dieses Handover ist die Verbindung kurzzeitig unterbrochen

- *Geringe Bandbreiten:*
  Die Bandbreiten drahtloser Netzwerke sind oftmals im Vergleich zu drahtgebundenen Systemen gering, so dass die Basisstation(z.B. GSM, UMTS) viele Pakete puffern muss. Dabei kann es zu Paketverlusten in der Basisstation kommen, wenn dort eine Warteschlange voll ist. Handelt es sich um WLAN (z.B. IEEE 802.11), so besteht die Basisstation aus einem sog. Access-Point, der keine Pakete zwischen speichert. Diese Zwischenspeicherung findet man nur im Basestationcontroler (BSC) der öffentlichen Mobilfunknetze. Diese Netze besitzen neben dem BSC noch den Basestationtransceiver (BTS), die zusammen die Basisstation bilden.

- *Link Latenz:*
  Die drahtlosen Netze zeigen oft lange Latenzen, die nicht unbedingt durch Engpässe ausgelöst werden. Stellen diese langen Verzögerungen einen großen Anteil der Round-Trip-Time (RTT) dar, so kann dies Einfluss auf die Protokolle der höheren Schichten wie TCP haben.

3 Klassisches TCP

TCP [Post81] ist ein verbindungsorientiertes Transportprotokoll auf Schicht 4 des ISO/OSI Referenzmodells. Entwickelt wurde es für Netzwerke mit drahtgebunden Verbindungen und stationären Endsystemen. Es ist ein zuverlässiges Protokoll, welches erfolgreich übertragene Dateneinheiten bestätigt und Übertragungswiderholungen auslöst, wenn ein Paketverlust aufgetreten ist. Auch wird der Gebrauch von kommulativen Bestätigungen gemacht. Weiterhin adaptiert sich TCP an Anforderungen des Netzwerkes. Die Anzahl der unquittierten Pakete wird durch das Vergrößern und Verkleinern des Staukontrollfensters (*Cwnd*) reguliert. Die Staukontrolle und Flusskontrolle von TCP lässt sich dabei in 3 Teile aufteilen.

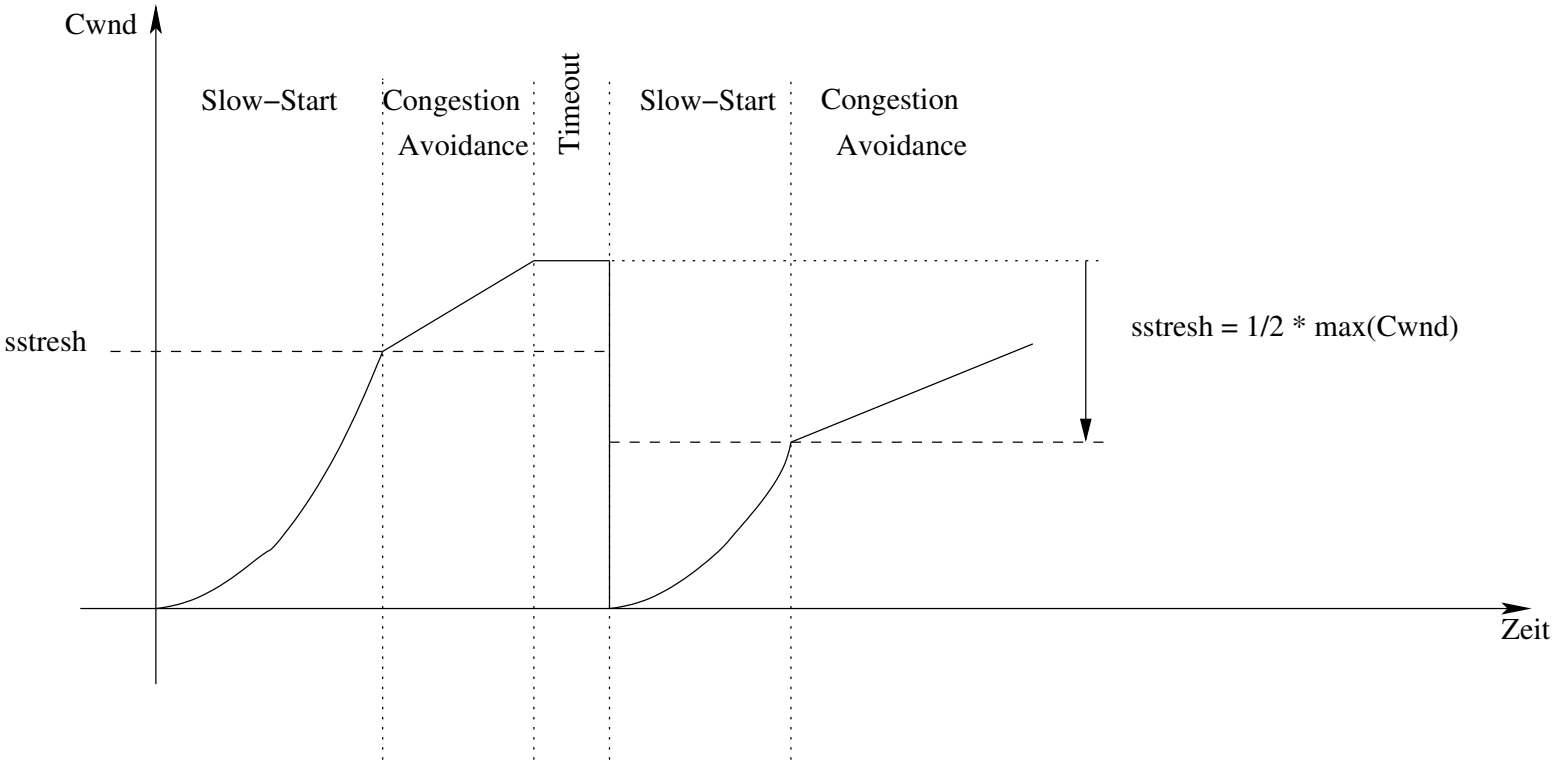

Abbildung 1: TCP Slow-Start Verhalten

Slow Start Phase 1.0 Als Slow-Start Phase bezeichnet man die Phase bei TCP, in der das Staukontrollfenster exponentiell wächst. Dies bedeutet, dass Staukontrollfenster (*Cwnd*) je empfangener Quittung verdoppelt wird. TCP bleibt so lang in der Slow-Start Phase, bis das Staukontrollfenster einen Schwellwert (*sstresh*) erreicht hat. Ab diesem Zeitpunkt, wechselt TCP in die Congestion-Avoidance Phase. Tritt jedoch ein Paketverlust auf, tritt TCP wieder in die Slow-Start Phase ein.

Congestion Avoidance Phase In der Congestion-Avoidance Phase [V. J92] vergrössert TCP das Staukontrollfenster (*Cwnd*) nur noch linear. Das heisst, die Variable Cwnd wird je empfangener Quittung nur noch um 1 erhöht. TCP bleibt so lang in dieser Phase, bis ein Paketverlust autftritt. Tritt ein Paketverlust, so wird der Schwellwert (*sstresh*) auf die Hälfte des aktuellen Staukontrollfensters (*Cwnd*) herabgesetzt. Weiterhin wird das Staukontrollfenster (*Cwnd*) auf 1 zurückgesetzt und in die Slow-Start Phase gewechselt. Dies ist in Abbildung 1 dargestellt..

Fast Retransmission/Fast Recovery Fast Retransmission [Stev97, S. F99] ist eine Erweiterung zu TCP, in der versucht wird durch das Senden mehrerer doppelter Bestätigungen den Sender darauf hinzuweisen, dass ein Paketverlust aufgetreten ist. Dies hat

beim Sender zur Folge, dass eine Sendewiederholung ausgelöst wird. Falls drei oder mehr (theoretisch sind so viele doppelte Quittungen erlaubt, wie der Empfänger aufgrund seines Staukontrollfensters senden darf) gleiche Quittungen beim Sender eintreffen, wird eine Sendewiederholung durchgeführt.
Fast Recovery besagt dass TCP nach einer Fast Retransmission nicht in die Slow-Start Phase wechselt. Jedoch wird das Staukontrollfenster für jede empfangene doppelte Quittung um 1 erhöht und *sstresh* auf die Hälfte des aktuellen *Cwnd* gesetzt. TCP wechselt erst wieder aus dem Fast-Recovery Modus, wenn dass erneut gesendete Paket korrekt bestätigt wurde. Danach wird *Cwnd* auf *sstresh* gesetzt und TCP wechselt in die Congestion-Avoidance Phase.

3.1 TCP in mobilen Umgebungen

Betrachtet man die Erweiterungen und Verbesserungen an TCP, so ist festzustellen, dass diese Erweiterungen überwiegend Verbesserungen hinsichtlich drahtgebundener Netze sind. Diese haben die Eigenschaft, dass ein Paketverlust meist durch den Überlauf von Router-Queues hervorgerufen wird und weniger bedingt durch das Medium.
In mobilen Umgebungen liegt der Hauptgrund für Paketverlust in der Mobilität der Endsysteme begründet. Hinzu kommen Paketverluste durch Engpässe in Routern. Weitere Probleme im Zusammenhang mit mobilen und drahtlosen Umgebungen findet sich in Abschnitt 2.1.
Analysiert man klassisches TCP in einer mobilen Umgebung. So lässt sich zeigen, dass TCP merkliche Performanceschwankungen hat, wenn ein Paketverlust auftritt und dass TCP, bedingt durch den Slow-Start, recht träge auf einen Paketverlust reagiert. Da TCP bei einem Paketverlust von einem Stau in einem Router ausgeht, passt es sein Staukontrollfenster und den sstresh-Schwellwert an. Dies hat zur Folge, dass TCP die Situation sehr pessimistisch einschätzt und *Cwnd* und *sstresh* neu berechnet.

4 TCP Erweiterungen für MANETs

Um die Probleme, die sich aus der Mobilität der Endsysteme ergeben, sind viele Vorschläge für TCP Erweiterungen zur Verbesserung der Performance in mobilen Umgebungen erarbeitet worden.
In dieser Arbeit wird vertieft auf die Arbeiten der Ende-zu-Ende Semantikerhaltung bei TCP-Erweiterungen eingegangen. Die anderen Klassen werden genannt und kurz die Erweiterungen in diesen Klassen erwähnt.

4.1 Klassifizierung

Die verschiedenen Vorschläge für TCP Erweiterungen zur Verbesserung der TCP Performance lassen sich grundsätzlich in drei Kategorien klassifizieren. Die drei Kategorien sind Split-Connections, Local Recovery und Ende-zu-Ende Semantikerhaltung. Diesen Kategorien lassen sich dann die verschiedene Ansätze zu TCP-Erweiterungen zuordnen. Diese Klassifizierungen sind auch in Abbildung 2 ersichtlich.

4.1.1 Split-Connections

In dieser Klasse von TCP-Erweiterungen, wird die TCP Verbindung an der Basisstation aufgetrennt. Das klassische TCP wird nur auf der Verbindung zwischen Sender und Basisstation genutzt. Der TCP Sender bekommt nicht die Charakteristik des mobilen Abschnittes mit, er

sieht nur die Paketverluste, die durch Stausituationen im Festnetz entstehen. Diese Situation hat den Vorteil, dass die Basisstation Nutzen aus dem Wissen über den mobilen Abschnitt ziehen kann und gezielt auf Applikationsanforderungen eingehen kann. Probleme dieser Klasse von TCP Erweiterungen sind die Effizienz, Robustheit und die Anforderungen, die bei einem Handover nötig sind. Zu dieser Klasse der Erweiterungen zählen: I(ndirect)-TCP [BaBa95], Wireless Application Protocol (WAP) [WAPF] und Mobile End Transport Protokoll (METP) [WaTr98].

4.1.2 Local Recovery

Hinter lokaler Wiederholung (Local Recovery) steckt die Absicht einen Kompromiss zwischen Split-Connections und Ende-zu-Ende Semantikerhaltung zufinden. Diese Kategorie der TCP Erweiterungen geht davon aus, dass Paketverluste in drahtlosen Netzen lokal begrenzt sind und damit die Paketverluste lokal, d.h. an der Basisstation eine Sendewiederholung ausgelöst wird. Dies beinhaltet, dass diese Ansätze Paketverluste erkennen müssen und Pakete aus einem lokal angelegten Cache erneut senden. Zu dieser Kategorie von Erweiterungen gehören Transport Unaware Link Improvement Protocol (TULIP) [PaGLA99a], AIRMAIL [APLS+95] and Performance Enhanced Proxies (PEP) [BKGM+01].

4.1.3 Ende-zu-Ende Semantikerhaltung

TCP Erweiterungen mit Ende-zu-Ende Semantikerhaltung versuchen die Paketverluste, verursacht durch mobile Übertragungsabschnitte innerhalb von TCP zu beheben. Dies bedeutet, sowohl Empfänger als auch Sender sind sich der mobilen Übertragungsstrecke bewusst. Sie versuchen durch verschiedene Mechanismen die Paketverluste so zu behandelt, dass TCP nicht in die Slow-Start Phase eintritt und die Datenübertragung ohne signifikante Performanceeinbrüche fortgeführt werden kann. In diese Kategorie fallen die folgenden Vorschläge: Selective Acknowledgements (SACK) [MMFR96, FMMP00], TCP Santa Cruz (TCP SC) [PaGLA99b], ACK Pacing [AgSA00],und Explicit Bad State Notification (EBSN) [BKVP97]. Diese Verfahren benötigen jedoch Änderungen in den Protokollstacks der Endsysteme.
1.0 In den weiteren Abschnitten wird näher auf diese Klasse der TCP Erweiterung für mobile Umgebungen eingegangen. Die beiden anderen Klassen werden hier nicht weiter betrachtet.

5 Erweiterungen zur Erhaltung der Ende-zu-Ende Semantik

5.1 TCP Santa Cruz

TCP Santa Cruz (TCP SC) [PaGLA99b] schlägt Verbesserungen im Bereich des Staukontrolle und Fehlerbehebung vor. TCP SC ermittelt, ob ein Stau existiert, oder sich aufbaut, in dem versucht wird die Anzahl der Pakete in der Bottleneckqueue zu errechnen und auf Grund dieser Daten das Staukontrollfenster zu vergrößern, oder zu verkleinern. Um dies berechnen zu können, werden erweiterte und veränderte Berechnungen zur Ermittlung des Round-Trip-Time (RTT) vorgeschlagen. Die Veränderungen werden in den folgenden Punkten näher beschrieben.

Congestion Control

Relative Verzögerungen Zur Entscheidung, ob ein Stau auf dem Datenpfad vorhanden ist, oder nicht reicht es nicht aus die RTTs zu berechnen. Die RTTs berücksichtigen nur Verzögerungen, die während des ganzen Paketumlaufes geschehen.

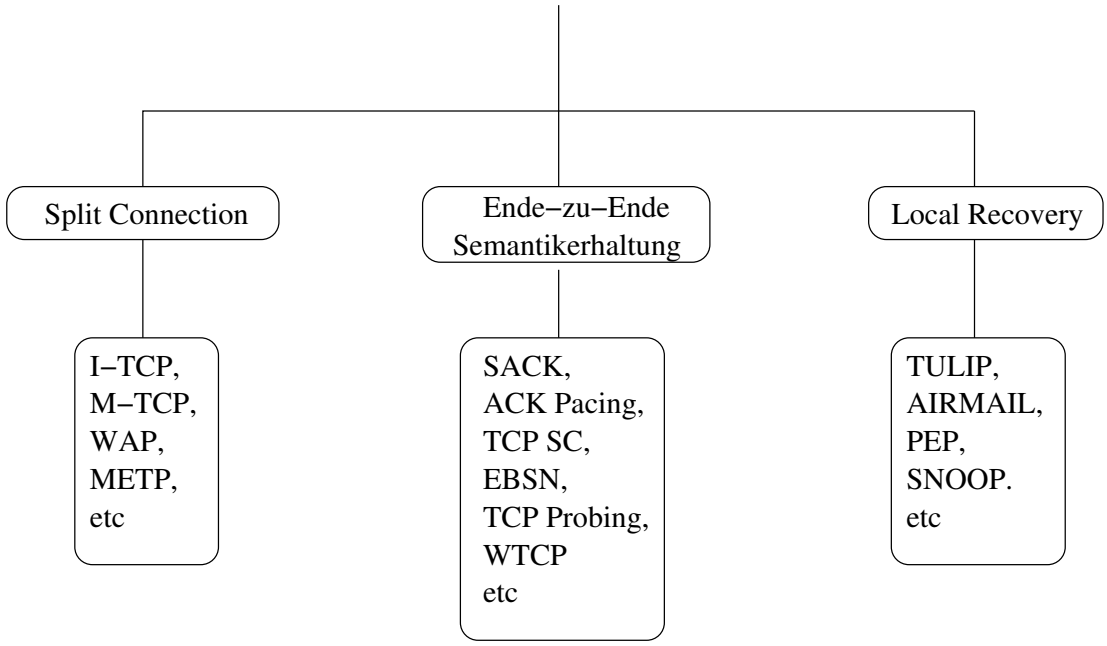

Abbildung 2: Einteilung der TCP-Erweiterungen für mobile Umgebungen

So bleibt etwa Stau in Vorwärtsrichtung, bzw. Rückrichtung unbeachtet. TCP SC führt im Gegensatz zu klassischem TCP relative Verzögerungen ein. Diese lassen sich recht einfach berechnen. Zum einen soeichert der Sender die Zeit eines Paketes, zu der er das Paket versendet hat und die Zeit, zu dem er das Acknowledgement-Paket (ACK) empfängt. Zum anderen empfängt der Sender mit jedem ACK einen Zeitstempel des Empfängers, der die Zeit enthält, zu dem das Paket eingetroffen ist. Aus diesen drei Zeitwerten pro Daten-ACk Paketpaar kann der Sender die relativen Verzögerungen je Richtung berechnen. Ebenfalls kann er damit die Interpacket-Delay Zeit zwischen zwei aufeinanderfolgenden Paketen berechnen. 0.5 Die relative Verzögerung läßt sich somit schreiben als.

$$D^F_{j,i} = R_{j,i} - S_{j,i} \tag{1}$$

Dabei ist $S_{j,i}$ die Zeit zwischen der Übertragung zweier Datenpakete beim Sender und $R_{j,i}$ die Zeit zwischen dem Empfang zweier Datenpakete beim Empfänger. Aus dieser Formel lassen sich dann vier Fälle ableiten, wie Pakete beim Empfänger ankommen. Die Zeiten sind in Abb. 3 dargestellt. Diese sind (a) Pakete haben gleiche Verzögerung, (b) erstes Paket verzögert, (c) zweites Paket verzögert und (d) Out-of-Order Empfang. Die Werte $D^F_{j,i}$ sind Ausgangspunkt für den Staukontroll Algorithmus. Die Granualität der Messungen hängt im wesentlichen von der verwendeten ACK-Policy, d.h. werden kommulative ACK verwendet oder nicht, ab.

Staukontroll-Algorithmus Modelliert man die Verbindung aus einer Reihe von Queues, so läßt sich daraus ableiten, dass sich die Queues verändern und sich in verschiedenen Zuständen befinden können. Diese sind (a) Queue wird länger (b) Queue wird kürzer (c) Queue bleibt gleichlang. Bildet man nun die verschiedenen möglichen Werte von $D^F_{j,i}$ auf diese Zustände ab, so erhält man einen Zustandsübergangsgraphen, der die Verbindung charakterisiert. Dieser Graph wird in TCP SC durch einfaches aufaddieren der $D^F_{j,i}$ realisiert. Ändert sich sich nun diese Summe innerhalb eines Intervalls um einen gewissen Schwellwert nach unten oder oben, so wird das Staukontrollfenster (*Cwnd*) linear nach oben bzw. unten angepasst.

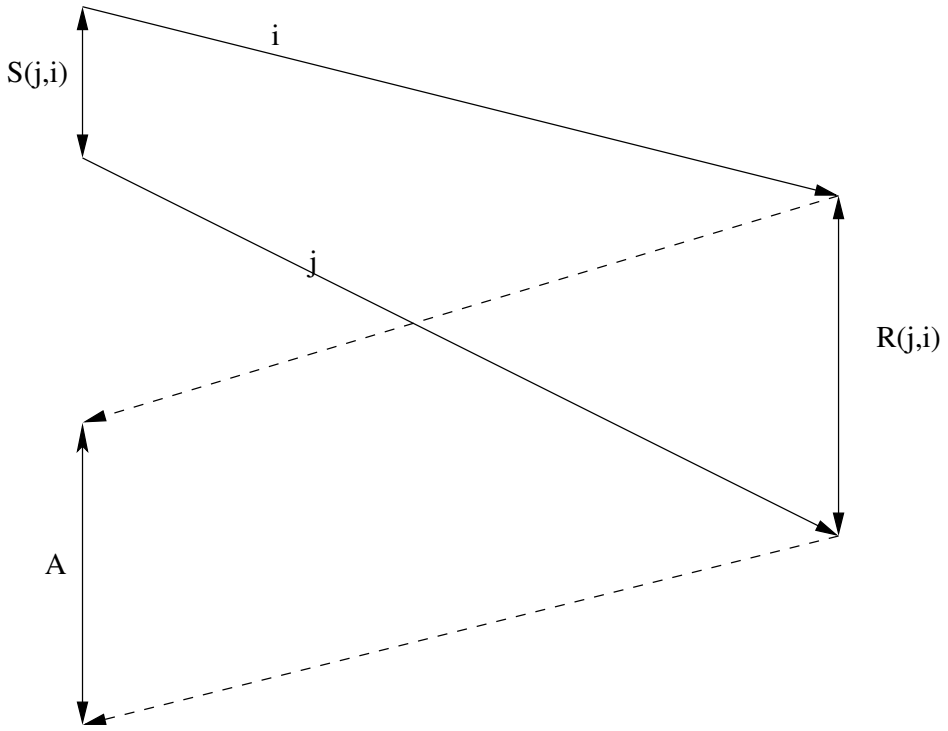

Abbildung 3: Interpaket Zeiten bei TCP-SC

Zu Beginn einer Verbindung wird in TCP SC wie im klassischen TCP die Slow-Start Phase benutzt. Jedoch hat TCP SC hier eine weitere Abbruchbedingung. Die Slow-Start Phase kann abgebrochen werden, wenn eine deutlichen Satturierung der Warteschlangen insbesondere der Bottleneck-Warteschlange ermittelt wird.

Error Recovery Zur Behandlung von Paketfehlern generiert der Empfänger von Daten ein sogenanntes Acknowlegement-Window (ACK-Window). In diesem ACK-Window wird über ein Bitmuster angezeigt, welche Teile des Datenstroms fehlen. In einem weiteren Feld wird definiert wie viele Bytes ein Bit repräsentiert. Dies kann von 50 Byte bis 65 KByte variieren. Dieses Bitfeld kann bis zu 18 Byte groß sein. Es orientiert sich nicht unbedingt an den TCP-Segmenten. Wird ein solches ACK-Window vom Sender empfangen, so kann dieser direkt die fehlenden Daten übertragen, ohne dass auf einen Timeout gewartet werden muss. Jedoch werden nicht sofort Pakete übertragen, sondern erst, wenn $t_{current} - t_i > EstRTT$ ist, d.h. das Paket wird frühestens wiederholt, wenn $EstRTT$ vergangen ist.

Vorgeschlagene Implementierung 0.6 Zur Implementierung von TCP SC wird in [PaGLA99b] vorgeschlagen, die zusätzlich benötigten Felder in einem TCP-Option Feld unterzubringen. Dazu wird eine neue TCP-Option definiert. Diese kann zwischen 11 und 40 Byte groß sein. Der Aufbau der TCP-Option ist dargestellt in Abb. 4.

5.2 ACK Pacing

ACK Pacing [AgSA00] - ACK Schrittsteuerung ist, ein Congestion Control Algorithmus. Er kann als hybrides Verfahren zwischen reinen ratenbasierten (WTCP) und fensterbasierten Verfahren (klass. TCP) verstanden werden. Klassisches TCP nutzt sein Fenster zur Entscheidung wie viele Pakete gesendet werden dürfen und Acknowledgements zur Entscheidung wann Pakete gesendet werden. Reine ratenbasierte Algorithmen benutzen Senderaten mit denen sie entscheiden, wann und wie viele Pakete versendet werden. ACK Pacing hingegen nutzt das

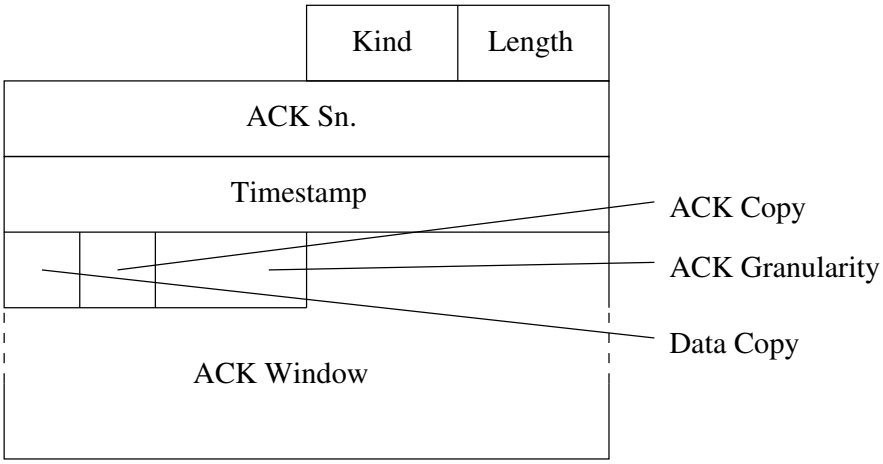

Abbildung 4: Aufbau der TCP-SC Option

TCP Fenster zur Entscheidung wie viele Pakete gesendet werden, und Senderaten dazu wann diese Pakete gesendet werden. Dazu wird bei ACK Pacing das Fenster durch die gemessene RTT geteilt und somit die Senderate festgelegt. Pacing im Allgemeinen kann auf Senderseite wie auch Empfängerseite durchgeführt werden. Dies hat zur Folge, dass die Datenpakete nicht in einem Burst übertragen werden. Auf der Seite des Empfänger kann ein Pacing durchgeführt werden. Jedoch ist dies anfällig gegenüber ACK-Komprimierung, so dass Datenpakete beim Sender wieder als Burst gesendet werden. Betrachtet man nun das senderseitige Pacing, so kann hier ein modifizierter Leaky-Bucket eingesetzt werden, dessen Zeitintervalle die Senderate durch $RTT/Window$ bestimmt werden. Diese Zeitintervalle zwischen zwei folgenden Paketen werden nach jedem Empfang eines ACK-Paketes erneuert . Zur besseren Bestimmung der RTT wird die TCP-Timestampoption benutzt. Jedoch wird die RTT Berechnung seperat von der der Flusskontrolle durchgeführt, da hier feingranularer gerechnet werden muss.
Betrachtet man ACK Pacing unter dem Gesichtspunkt der Queueing Theorie, so läßt sich feststellen, dass vom theoretischen Standpunkt aus ACK Pacing besser gegenüber klassischem TCP ist. Dies liegt in der Verminderung der burstaritgen Übertragungen, so dass Queues nicht so schnell überlaufen. Jedoch muss beachtet werden, befindet sich ein Netzwerk in einer Stausituation, so wird durch ACK Pacing die Stausituation verzögert und somit kommt es zu einer negativen Auswirkung auf die Performance des Verfahrens. Ebenfalls wird durch Pacing die Latenzzeit einer Übertragung erhöht. Hinzu kommen synchronisations- und desynchronisations-Effekte. Synchronisations-Effekte treten dadurch auf, dass alle ACK-Pacing Verbindungen zur gleichen Zeit Stau vermuten und somit gleichzeitig reagieren. Aber es existieren auch desynchroisations Effekte. Diese beziehen sich auf das auseinander Laufen der Fenstergrößsen bei Pacing, da durch Pacing Pakete verzögert werden und somit ein Flow mehrere Paketeverluste erfährt, andere Verbindungen jedoch nicht.

5.3 W-TCP

WTCP [SNVS+02] arbeitet anders gegenüber den bisher geschilderten Protokollen. WTCP arbeitet ratenbasiert. Der Grund für die Nutzung eines ratenbasierenden Algorithmus liegt nach [SNVS+02] darin begründet, dass Paketverluste auf Grund der “burstiness” von Paketfolgen entstehen. Von TCP werden das Verbindungsmanagement und die Flusskontrolle übernommen. Das Hauptaugenmerk bei der Berechnung der Paketrate liegt auf der Beobachtung und Analyse der Zeitabstände zwischen zwei aufeinanderfolgenden Paketen. Diese Berechnungen zur Anpassung der Paketrate erfolgt beim Empfänger. Der Empfänger teilt das Ergebnis der Berechnung dem Sender in regelmäßigen Abständen mit. Ebenfalls wird die

Granularität der Ratenvergrößerung bzw. -verkleinerung mitgeteilt.
WTCP benutzt einen ratenbasierten Algorithmus zur Verminderung von Paketverlusten durch Stau- und Paketfehler. Dieses Verfahren wird deshalb gewählt, da oft ACK-Pakete in einer Basisstation auf Grund ihrer geringen Größe zusammengefasst werden. Dadurch entsteht ein burstartiger Empfang beim Sender, der daraufhin burstartig seine Daten versendet. Dies hat wiederum die Folge, dass Paketqueues überlaufen und es so zu Paketverlusten kommen kann. WTCP verhindert dies durch seinen Algorithmus.
Als Grundlage zur Berechnung der Paketrate wird bei WTCP die Zeit zwischen zwei aufeinander folgenden Paketen gemessen und ausgewertet. Dies geschieht beim Empfänger und nicht beim Sender. Die Inter-Packet Zeiten werden hauptsächlich zur Ratenkontrolle und zur Vermeidung von Paketverlusten genutzt, da die Senderate sofort herabgesetzt wird, wenn Anzeichen für einen Stau vorhanden sind. Weiterhin wird versucht den Grund der Paketverluste zu ermitteln. Dazu führt der Empfänger eine Liste von Gründen warum in einem Zeitabschnitt Pakete verlorengegangen sind. Zu dem werden hiervon die Mittelwerte und die mittlere Abweichung berechnet. Diese Berechnungen werden immer dann durchgeführt, wenn das Netz nicht in einer Stausituation ist. Tritt nun ein Paketverlust auf wird vorrausgesagt, ob dieser Paketverlust auf Grund eines Staus verursacht wurde, oder nicht. Handelt es sich um einen Paketverlust während eines Staus, so wird die Senderate herabgesetzt, sonst wird die Senderate beibehalten. Empfängt der Empfänger nicht innerhalb einer Wartezeit (*threshold*) ein Paket, so geht er davon aus, dass es sich um einen Blackout handelt und keine Pakete gesendet werden können. Unter einem Blackout wird hier die Nichterreichbarkeit der drahtlosen Endsystems durch Unterbrechnung der Funkverbindung verstanden. Nach dem der Blackout-Modus verlassen wurde, wird die Senderate wieder auf den Wert gesetzt, den er vor einem Blackout/Stau hatte.
Die Anpassung der Senderate wird nicht wie bei klassischem TCP mit *additive Increase multiplicative Decrease* (AIMD) durchgeführt, sondern läßt sich variieren. Zum einen wird die Senderate reduziert um Fairness zu erreichen. Schneller wird die Rate nach unten angepasst, wenn es sich um einen abzeichnenden Stau handelt und sehr schnell (aggressiv), wenn es sich um einen Stau handelt, um Paketverluste zu minimieren.
Zu Beginn einer Übertragung startet WTCP nicht mit Slow-Start Phase, sondern benutzt einen Mechanismus namens "Packet-Pair" [Kesh91], der zu Beginn einer Verbindung zwei Pakete mit voller Maximum Segment Size (MSS) sendet und dann deren Inter-Paket Zeit berechnet. Dieser Wert wird dann als Startwert für die Senderate genommen. Dieser Machanismus kann ebenfalls zum Verlassen des Blackout Zustandes genutzt werden. Ebenfalls kann dazu ein Probing genutzt werden, dabei werden "probe"-Pakete versendet und berechnet, ob sich der Stau aufgelöst hat oder nicht.
Für die Zuverlässigkeit bei WTCP wird der Selective Acknowledgment Algorithmus [MMFR96] genutzt, jedoch ist dieser so modifiziert, dass keine Retransmission Timeouts benutzt werden. Ein anderer Aspekt der Zuverlässigkeit ist, dass eine Senderate für Acknowledge Pakete festgelegt wird, da die ACK's Übertragungsinformationen und Zuverlässigkeitsinformationen übertragen. Diese ACK-Senderate wird so festegelegt, dass der Sender mindestens ein ACK-Paket innerhalb einer treshold-Zeit empfängt, da er sonst in den Blackout-Zustand wechselt.

5.4 EBSN, ECN und ELN

Ein Grund für die Performanceeinbrüche bei TCP über drahtlose Links ist auch darin zu sehen, dass Paketverluste auf dem drahtlosen Teil der Verbindung zu Timeouts beim Sender führen können. Dies hat zur Folge, dass es zu einer doppelten Sendewiederholung von Paketen kommt, wenn die Basisstation Sendewiederholungen durchführt. Aus diesem Grund kann man auch den folgenden Ansatz: Explicit Bad State Notification nicht komplett als

Ende-zu-Ende Semantikerhaltung bezeichnen, jedoch wird der Sender darüber informiert, dass ein Paketverlust aufgetreten ist. Somit ist auch dem Sender bewusst, dass ein drahtloser Abschnitt existiert. Hinzu kommen zwei weitere Vorschläge, die geringügig anders funktionieren: Explicit Congestion Notification und Explicit Loss Notification.

Explicit Bad State Notification bei Explicit Bad State Notification (EBSN) [BKVP97] wird eine Meldung sofort nach dem Erkennen eines Paketverlustes zum Sender gesendet. Die Meldung kann als ICMP Message gesendet werden. Dies erfordert jedoch die Einführung eines neuen ICMP-Message Typs. Als Reaktion auf diese Meldung der Basisstation/Bridge löscht der Sender vorhandene TCP-Timouts und initialisiert diese neu. Dies bedeutet, dass solange der drahtlose Link im "Bad-State" ist, der Sender regelmäßig neue Meldungen bekommt und den Timout zurücksetzt. Setzt man nun vorraus, dass sich ein solcher drahtloser Link nur kurz im "Bad-State" befindet, so hat dies zur Folge, dass Paketwiederholungen verringert werden und dass das TCP-Verhalten verbessert werden kann. Jedoch muss durch dieses Verfahren der Durchsatz nicht verbessert werden, da ja evtl. durch größere Timouts länger auf ACK-Pakete gewartet werden muss und somit unter Beachtung des *Cwnd* keine weiteren Pakete versendet werden können.

Explicit Congestion Notification EBSN kann als Weiterentwicklung von Explicit Congestion Notification (ECN) (RFC 3168) [K. R01] gesehen werden. Bei ECN wird zunächst die Anzeige von Stau bzw. Paketverlust in Vorwärtsrichtung, d.h. Richtung Empfänger vorgenommen. Erst danach wird eine Meldung an den Empfänger gesendet, der darauf reagiert. Dies hat jedoch den Nachteil bei drahtlosen Links, dass selbst die Meldung über Congestion über den drahtlosen Link geschickt werden muss, was jedoch eventuelle Timeouts beim Sender nicht verhindern kann. ECN operiert innerhalb der Protokolle von IP und TCP.

Explicit Loss Notification Explicit Loss Notification (ELN) ist ein änliches Verfahren wie EBSN, jedoch mit dem Unterschied, dass die Sendewiederholungen durch den Sender durchgeführt werden und nicht durch die Basisstation. Ebenfalls wird dem Sender von der Basisstation signalisiert, dass ein Paketverlust vorliegt und der Sender eine Sendewiederholung initieren muss. Bei ELN exisitiert eine Snoop-Instanz in der Basisstation, die jedoch keine Pakete zwischenspeichert. Sie ist nur für die Erkennung von fehlenden Sequenznummern zuständig und signalisiert diese fehlenden Sequenznummern innerhalb von TCP zum Sender. Dieser kann nun eine Trennung von Staukontrolle und Sendewiederholungen vornehmen und die verlorengegangenen TCP-Segmente erneut übertragen. In dem vorliegenden Vorschlag wird diese Signalisierung innerhalb des TCP-Headers transportiert. Eine andere Möglichkeit besteht darin dies per ICMP zu realisieren.

5.5 SACK

Die Selective Acknowledgement (SACK)-Erweiterung für TCP ist in RFC 2018 [MMFR96] standardisiert. TCP hat oft eine schlechte Perfomance, wenn mehrere Segmente innerhalb eines Übertragungsfensters verloren gehen. Der SACK Mechanismus kombiniert mit einer selektiven Retransmission-Policy kann dies beheben. Der empfangende Client sendet SACK Packete zurück zum Sender. Damit informiert er ihn, dass ein Teil der gesendeten Pakete

korrekt empfangen wurden. Der Sender braucht nur noch die fehlenden Pakete erneut zu senden.
Die TCP-SACK Erweiterung benutzt zwei TCP-Optionen. Die erste Option signalisiert das Vorhandensein von SACK bei Sender bzw. Empfänger (SACK-Permitted). Diese Option wird während der Verbindungsaufbauphase ausgehandelt. Die andere ist die SACK-Option selbst, welche in einer Verbindung gesendet werden kann, bei der es durch "SACK-Permitted" erlaubt wurde. Diese beiden Optionen werden innerhalb des TCP-Optionen Feldes transportiert.
Die SACK-Permitted Option ist 2 Oktette lang. Zum einen enthält sie Art der Option ($Kind = 4$) und die Länge der Option ($Length = 4$). 0.4 Das Format der SACK Option ist

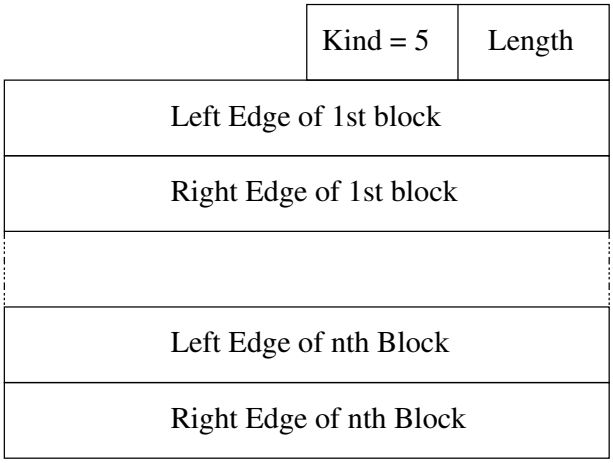

Abbildung 5: Aufbau der TCP-SACK Option

wie folgt aufgebaut. Zuerst wird die Art der TCP-Option gekennzeichnet ($Kind = 5$) und anschließend die Länge der gesamten TCP-Option. Die Länge setzt sich aus der Typangabe, Längenindikator und der Byteanzahl der Attribute zusammen. Die Länge beträgt $2 + n * 8$ Byte, wobei sich die 8 Byte aus der Angabe zweier TCP-Sequenznummern zu je 32 Bit (4 Byte) zusammensetzt. Die Abbildung 5 verdeutlicht den Ausbau der SACK-Option.
Die TCP-SACK Option wird durch den Empfänger von Datenpaketen an den Sender zurück gesendet. Der Empfänger teilt dem Sender dadurch mit, dass nicht-reihenfolgetreue Pakete empfangen wurden. Der Empfänger wartet auf die Übertragung der SACK-Option um die fehlenden Sequenznummern erneut zu übertragen.. Ein nicht-reihenfolgetreues Segment wird in der SACK Option durch seine kleinste empfangene Sequenznummer (Left Edge of Block) und durch seine größte Sequenznummer (Right Edge of Block) gekennzeichnet. Dieser Block ist in sich zusammenhängend und reihenfolgetreu.
Diese SACK Signalisierung verändert nicht die Interpretation der gesendetet Achnowledge-Sequenznummer. Solang die Lücke zwischen zwei korrekt empfangenen Blöcken noch nicht durch eine Retransmission geschlossen wurde schickt der Empfänger doppelte ACK-Pakete, je mit der gleiche Acknowledge-Sequenznummer und der SACK-Option im Anhang. Die obere angegebene Sequenznummer innerhalb der SACK Option wird jeweils auf die Sequenznummer gesetzt, die der Empfänger innerhalb des Blocks korrekt empfangen hat. Da die Länge der TCP-Optionen auf 40 Byte begrenzt ist, können maximal 4 Out-of-Order Blöcke gleichzeitig signalisiert werden. In Verbindung mit der Timestamp-Option sind es nur 3 Blöcke.
Empfängt der Sender von Datenpaketen Acknowledgement-Pakete mit integrierter SACK Option, so schaut dieser in der Liste der übertragenen Segmente, die noch nicht bestätigt wurden. Daraufhin werden die Pakete, die selektiv bestätigt wurden, mit einem Flag versehen. Pakete, die unterhalb der Left Edge Sequenznummer liegen, werden erneut übertragen. Jedoch haben die Flags, in der Retransmission-Queue keine Bedeutung, wenn dennoch ein Timeout auftritt. Nach einem Timeout muss der Sender die Pakete wiederholen, egal ob

diese schon per SACK bestätigt wurden oder nicht.

6 Vergleich der TCP-Varianten

Wie schon in Abschnitt 3.1 beschrieben, sind die bisherigen Erweiterungen und Erweiterungsvorschläge überwiegend nur für drahtgebundene Medien entworfen worden. Diese weisen jedoch gegenüber dem drahtlosen Medium eine andere Charakterisitk auf. So enstehen in drahtgebundenen Netzen Paketverluste meist nur dadurch, dass Routerqueues überlaufen und weniger Paketverluste aufgrund von Medienfehlern. Dies wird näher in Abschnitt 2.1 geschildert. Aufgrund dieser Überlegungen und der Tatsache, dass sich mobile Geräte immer größerer Beliebtheit erfreuen, müssen Möglichkeiten geschaffen werden die bekannten Protokolle aus dem Internet in die mobilen Endgeräte zu integrieren, ohne dass größere Änderungen an den Protokollstacks vorgenommen werden sollten. Dies spricht eigentlich gegen die Ende-zu-Ende Protokolle, sondern mehr für die Klasse der Split-Connections (siehe 4.1.1). Jedoch zerstören diese die Ende-zu-Ende Eigenschaften der Protokolle im Internet, sowie das Vertrauen der Anwender in den angebotenen Dienst. Somit ist klar, dass man Erweiterungen für TCP finden muss, die die Ende-zu-Ende Eigenschaften erhalten, trotz geänderter Mediencharakteristiken. Jedoch sollte man bei der grundstätzlichen Diskusion über die Art der eingesetzten Protokolle nicht vergessen, TCP so zu erweitern, dass ein Performancegewinn im Vordergrund steht.
Schaut man sich nun TCP im Detail an, aus welchen Gründen es zu großen Perfomanceeinbrüchen kommt, so kristallisiert sich schnell heraus, dass die Art und Weise wie TCP einen Fehler behebt eine Ursache ist. Diese ist die sogenannte Slow-Start Phase von TCP, der dazu führt, dass nach einem Paketverlust TCP die Senderate erst langsam wieder ansteigen läßt. Desweiteren geht TCP bei einem Paketverlust immer von einer Stausituation aus, die es erfordert längerfristig die Senderate herunterzusetzen und die eigentliche Erkennung von Paketverlusten dauert lang, da erst auf einen Timeout gewartet wird, bevor eine Sendewiederholung angestoßen wird. Betrachtet man dies unter Maßgabe der drahtlosen Medien, ist dies eine fatale Annahme, da es hier oft nur zu einzelnen Fehlern kommt und kein Stau dem Paketverlust zugrunde liegt. Hinzu kommt noch, dass TCP seine Staukontrolle an die Flusskontrolle koppelt. Aus diesen Tatsachen lassen sich Forderungen ableiten, die an eine Erweiterung gestellt werden müssen, wenn Performanceverbesserungen erziehlt werden sollen.

- Entkopplung von Staukontrolle und Flusskontrolle
- Verhinderung des Slow-Start im Fall eines Paketverlustes
- Schnelles Recovery nach Paketverlust
- Signalisierung von Paketverlusten, d.h. Verhinderung von Timeouts

Sicher lassen sich noch weitere Forderungen finden, damit die Performance von TCP über drahtlose Medien gesteigert werden kann. Um nun diese Ziele zu erreichen, können verschiedene Mechanismen eingesetzt werden. Diese lassen sich grob in drei Klassen trennen

- ratenbasierte Verfahren
- fensterbasierende Verfahren
- hybride Verfahren

Zur ersten Klasse zählt klar WTCP (Kap. 5.3). Zur zweiten Klasse gehöhren SACK (Kap. 5.5), TCP-SC (Kap. 5.1) sowie EBSN, ELN, ECN (Kap. 5.4). In die dritte Klasse läßt sich ACK-Pacing (Kap. 5.2 integrieren. Jedes Protokoll für sich erfüllt eine oder mehrere der oben gestellten Forderungen.
Die Signalisierung, dass ein Paketverlust aufgetreten ist, wird in den vorgestellten Protokollen durch zwei verschiedene Verfahren erbracht. Zum einen ist hier die Signalisierung durch Senden von ICMP Nachrichten vom Empfänger, oder von Zwischensystemen aus zu nennen. Zum anderen die Signalisierung von Paketverlusten bzw. selektiver Bestätigungen durch den Empfänger. Die Signalisierung vom Empfänger aus werden meist als TCP-Option realisiert. So übertragen SACK und TCP-SC selektive Bestätigungen zum Sender, der schnell reagieren kann. Jedoch werden die Pakete bei SACK in die Flusskontrolle von klass. TCP integriert. Somit findet hier keine Trennung von Fluss- und Staukontrolle statt. Ebenfalls findet man die auch bei EBSN, ELN und ECN, jedoch findet hier die Erkennung von Stau- bzw. Paketverlust verteilt statt, d.h. nicht nur beim Sender, sondern auch in den Zwischensystemen. Dies hat aber zur Folge, dass EBSN, ELN und ECN nicht vollständig die Ende-zu-Ende Semantik erhalten, da ein Teil der Aufgaben nicht lokal durchgeführt werden. Dies kann aber wiederum ein Vorteil sein, um die Zeit zu minimieren um Pakete erneut zu übertragen, wenn z.B. ACK Pakete verloren gehen, aber Datenpakete nicht. So kann ein Zwischensystem dies erkennen und das dem Sender signalisieren. Dies hat zur Folge, dass entweder der Timer zur Sendewiederholdung zurückgesetzt wird, oder schneller eine Paketwiederholung durchgeführt werden kann. Daraus kann man drei Orte ableiten, wo Signalisierung von Paketverlust bzw. Stau vorgenommen werden kann:

- indirekt beim Sender
- direkt beim Empfänger
- indirekt und dirket in Zwischensystemen (indirekt über Timer, direkt über Informationen von unteren Layern)

Ein weiterer Punkt zur Verbesserung der Performance ist, dass Interpaketzeiten statt Round Trip Zeiten berechnet werden, da diese besser den Zustand einer Verbindung charakterisieren können wie RTTs. Bei RTTs sieht man nur die Gesamt-Round-Trip-Time, nicht jedoch die einzelnen Anteile von Hin- bzw. Rückweg einer Verbindung, da diese nicht notwendigerweise gleich sein müssen. Welche Eigenschaften man aus diesen sogenannten relativen Verzögerungen ablesen kann, ist in Abschnitt 5.1 und in [PaGLA99b] erläutert. Diese relativen Verzögerungen sind dann Ausganspunkt von Berechnungen der Flusskontrolle und der Staukontrolle zur Anpassung Senderate. Zum anderen wird zur Anpassung der Senderate bei den ratenbasierten Verfahren Traffic-Shaping-Algorithmen wie der Leaky-Bucket verwendet.

7 Zusammenfassung

Fazit der Untersuchung von Transportprotokollen mit Erhaltung der Ende-zu-Ende Semantik ist, dass nicht jedes vorgesellte Protokoll alle Probleme im Zusammenhang mit mobilen Endgeräten lösen kann. Jedoch erläutern diese Protokolle Werkzeuge und Verfahren , mit denen man je nach vorgesehenem Einsatzgebiet ein Protokoll designen kann. Durch geschickte Kombination dieser Machnanismen kann ein Protokoll implementiert werden, dass nahezu alle Anforderungen erfüllt und somit Einzug in die Internet-Protokolle finden kann.

Literatur

[AgSA00] Amit Aggarwal, Stefan Savage und Thomas Anderson. Understanding the Performance of TCP Pacing. In *INFOCOM (3)*, 2000, S. 1157–1165.

[APLS+95] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani und R. D. Gitlin. AIRMAIL: A Link-Layer Protocol for Wireless Networks. *ACM Wireless Networks* 1(1), 1995, S. 47–60.

[BaBa95] Ajay Bakre und B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. *15th International Conference on Distributed Computing Systems*, 1995.

[BKGM+01] J. Border, M. Kojo, J. Griner, G. Montenegro und Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135 (Informational), Jun. 2001.

[BKVP97] Bikram S. Bakshi, P. Krishna, Nitin H. Vaidya und Dhiraj K. Pradhan. Improving Performance of TCP over Wireless Networks. In *International Conference on Distributed Computing Systems*, 1997.

[FMMP00] S. Floyd, J. Mahdavi, M. Mathis und M. Podolsky. An Extension to the Selective Acknowledgement (SACK) Option for TCP. RFC 2883 (Proposed Standard), Jul. 2000.

[K. R01] D. Black K. Ramakrishnan, S. Floyd. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Proposed Standard), Sept. 2001.

[Kesh91] S. Keshav. Congestion Control in Computer Networks, Sept. 1991.

[MMFR96] M. Mathis, J. Mahdavi, S. Floyd und A. Romanow. TCP Selective Acknowledgement Options. RFC 2018 (Proposed Standard), Oct. 1996.

[PaGLA99a] C. Parsa und J. Garcia-Luna-Aceves. TULIP: A Link-Level Protocol for Improving TCP over Wireless Links, 1999.

[PaGLA99b] Christina Parsa und J. J. Garcia-Luna-Aceves. Improving TCP Congestion Control Over Internets with Heterogeneous Transmission Media. In *Proceedings of the 7th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 1999.

[Post81] J. Postel. Transmission Control Protocol. RFC 793 (STANDARD), Sep. 1981.

[S. F99] T. Henderson S. Floyd. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582 (Experimental), Apr. 1999.

[SNVS+02] Prasun Sinha, Thyagarajan Nandagopal, Narayanan Venkitaraman, Raghupathy Sivakumar und Vaduvur Bharghavan. WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. *Wireless Networks* 8(2-3), 2002, S. 301–316.

[Stev97] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001 (Proposed Standard), Jan. 1997.

[V. J92] D. Borman V. Jacobson, R. Braden. TCP Extensions for High Performance. RFC 2581 (Proposed Standard), May 1992.

[WAPF] WAPFORUM. WAP 2.0 Spezification. http://www.wapforum.org.

[WaTr98] Kuang-Yeh Wang und Satish K. Tripathi. Mobile-End Transport Protocol: An Alternative to TCP/IP Over Wireless Links. In *INFOCOM (3)*, 1998, S. 1046–1053.

Abbildungsverzeichnis

Hierarchische Ad-hoc-Routingprotokolle

Timo Schönwald

Kurzfassung

In tragbare Geräte wie Notebooks, PDAs oder Mobiltelefone werden immer häufiger Schnittstellen zur drahtlosen Kommunikation z.B. nach dem IEEE 802.11 Standard oder Bluetooth, eingebaut. Diese Funktechniken müssen nicht auf eine vorhandene Infrastruktur zurückgreifen und können somit infrastrukturlose spontane Netze bilden. Solche spontan gebildeten Netze mit ihrem zugehörigen Routingprotokoll nennt man mobile Ad-hoc-Netzwerke (MANETs). Um der Mobilität dieser Teilnehmer in einem Mobilen Ad-hoc-Netzwerk gerecht zu werden, wurden für diesen Zweck neue Routingprotokolle entwickelt. Diese Routingprotokolle müssen sich sehr schnell an die häufigen Änderungen der Netztopologie anpassen und hierbei so wenig Protokoll-Overhead wie möglich produzieren. Ziel ist, die Routingtabellen so klein wie möglich zu halten und das Versenden von Routingprotokollnachrichten auf ein Minimum zu reduzieren, um eine gute Ausnutzung des Funkmediums zu garantieren. Eine Möglichkeit, um die Skalierbarkeit von Routingprotokollen zu verbessern, ist die Einführung einer Hierarchie. In dieser Ausarbeitung werden verschiedene Routingprotokolle vorgestellt, die auf unterschiedliche Weisen eine Hierarchie im Netzwerk einführen.

1 Einleitung

In den vergangenen Jahren haben IEEE 802.11 und Bluetooth an Bedeutung gewonnen, da sie immer häufiger in tragbare Geräte wie Notebooks, PDAs und Mobiltelefone eingebaut werden. Durch diese Schnittstellen zur drahtlosen Vernetzung ist es möglich, mobile Ad-hoc Netzwerke (MANETs) zu bilden. Netze solcher Art haben besondere Eigenschaften: die Teilnehmer können mobil sein, wodurch sich die Netztopologie ständig ändert. Da keine feste Infrastruktur vorhanden ist, muss sich das Netz selbst organisieren. Hierdurch treten in MANETs neue Anforderungen auf, die es in Netzwerken mit fester Infrastruktur nicht gibt. Ein Knoten in einem MANET muss gegenüber einem Knoten in einem Netz mit fester Infrastruktur mehr Funktionalität zur Verfügung stellen. So muss ein Knoten in einem MANET sowohl die Funktion eines Endgerätes wie auch die eines Routers bereitstellen.

Für Festnetze entwickelte Routingprotokolle sind für relativ seltene Topologieänderungen und symmetrische Verbindungen ausgelegt. Da durch die Mobilität der Knoten häufige Topologieänderungen auftreten, kann man diese Routingprotokolle nicht ohne weiteres in drahtlosen Netzen ohne Infrastruktur einsetzen. Die Verbindung zwischen zwei Knoten in einem MANET ist asymmetrisch, da die Verbindungsqualität zwischen den zwei Knoten in beiden Richtungen unterschiedlich sein kann. Dies kann z.B. auftreten wenn ein Knoten einen stärkeren Sender hat als der andere, oder wenn Hindernisse die Ausbreitung der Funksignale stören. Aus diesen Gründen sind spezielle Routingprotokolle für MANETs entwickelt worden.

Innerhalb der Internet Engineering Task Force (ITEF) gibt es die Arbeitsgruppe Mobile Ad-hoc Networks (manet) [mWor], die sich mit der Standardisierung von Routingprotokollen für mobile Ad-hoc Netze beschäftigt.

Ziel solcher Routingprotokolle für MANETs ist die Bereitstellung einer geeigneten selbst organisierenden Kontrollstruktur für große dynamische Netze. Solche Routingprotokolle sollen möglichst schnell konvergieren und möglichst wenig Protokollnachrichten versenden, da die Bandbreite des Übertragungsmediums in drahtlosen Netzwerken begrenzt ist. Ein Ansatz zur Verbesserung der Skalierbarkeit der Routingprotokolle in MANETs sind hierarchische Ad-hoc-Routingprotokolle.

Ziel dieser Arbeit ist es einige dieser Routingprotokolle vorzustellen und deren Vor- und Nachteile herauszuarbeiten.

2 Eigenschaften hierarchischer Ad-hoc Routingprotokolle

In einem hierarchisch organisiertem MANET werden Knoten, die Gemeinsamkeiten besitzen oder gewisse Eigenschaften erfüllen zu einer Gruppe zusammengeschlossen. Jede Gruppe innerhalb des MANET bildet somit ein logisches Subnetzwerk mit einer eindeutigen Subnetz-ID.

Durch die Mobilität der Knoten in einem MANET können sie von einem Subnetz in ein anderes Subnetz wechseln. Es wäre vorteilhaft wenn die Knoten beim diesem Wechsel des Subnetzes auch ihre Subnetz-ID wechseln würden. Ein Adresswechsel beim Wechsel des Subnetzes hat den Vorteil, dass Pakte wieder auf dem kürzesten Weg zu einem Knoten geroutet werden können, und nicht den Umweg über das ursprüngliche Subnetz gehen. Wenn Knoten die Subnetz-ID des neuen Subnetzes annehmen, so würden bestehende Verbindungen abbrechen, da der Adresswechsel für höhere Schichten (TCP) nicht transparent ist. Pakete, die für den Knoten bestimmt sind, der das Subnetz gewechselt hat, würden das Ziel nicht mehr erreichen. Kommunikationspartner könnten diesem Knoten keine weiteren Pakete mehr senden, da ihnen die neue Adresse des Knotens nicht bekannt ist. Um den Adresswechsel eines Knotens in einem MANET zu ermöglichen und die neue Adresse im MANET bekannt zu machen ist ein Location Management, oder auch Mobilitäts Management genannt, nötig.

Mit der hierarchischen Einteilung von MANETs in Subnetze wird jedem Subnetz genau ein Leader zugeordnet. Der Leader übernimmt die Verwaltung des zugehörigen Subnetzes. Der Leader eines Subnetzes kann auf verschiedene Arten bestimmt werden. Gruppen können wieder zu einer Gruppe zusammengefasst werden, dadurch erhält man eine mehrstufige Hierarchie in einem MANET.

Im Folgenden werden hierarchische Ad-hoc Routingprotokolle einer groben Einteilung unterzogen und in den folgenden Kapiteln einige von ihnen näher betrachtet.

2.1 Klassifizierung hierarchischer Ad-hoc Routingprotokolle

Routingprotokolle lassen sich aufgrund verschiedener Eigenschaften in unterschiedliche Gruppen einteilen (Abbildung 1). Zunächst lassen sie sich in proaktive, reaktive und hybride Routingprotokolle unterteilen [Perk01].

Proaktive Routingprotokolle aktualisieren ihre Routinginformationen periodisch durch Updatenachrichten. Hierdurch entsteht eine konstante Belastung des Netzes durch Routingprotokollnachrichten. Ein Beispiel für ein proaktives Routingprotokoll ist das Fisheye State Routing Protocol (FSR) [PeGC00].

Reaktive Routingprotokolle unterscheiden das Auffinden und Aufrechterhalten eines Weges. Ein Weg wird nur dann gesucht, wenn er benötigt wird und noch nicht vorhanden ist. Wege werden nur solange aufrecht erhalten, solange sie benötigt werden. Dies bedeutet, dass bei

wenigen Verbindungen weniger Routingprotokollnachrichten, als bei proaktiven Routingprotokollen, versendet werden. Dadurch dass ein Weg erst dann gesucht wird, wenn er benötigt wird, dauert das Zustellen eines Paketes länger als bei proaktiven Routingprotokollen. Als Beispiele für reaktive Routingprotokolle, sind das Dynamic Source Routing Protocol (DSR) [JoMa96] und das Ad-hoc On-Demand Distance Vector Routing Protocol (AODV) [PeBRD03] [Perk97] zu nennen.

Hybride Routingprotokolle vereinigen die Vorteile von proaktiven und reaktiven Routingprotokollen. Für Knoten innerhalb eines gewissen Bereichs wird ein proaktives Routingprotokoll verwendet. Für Knoten, die sich außerhalb dieses Bereichs befinden wird ein reaktives Routingprotokoll eingesetzt. Ein Beispiel für ein hybrides Routingprotokoll ist das Zone Routing Protocol (ZRP) [Beij].

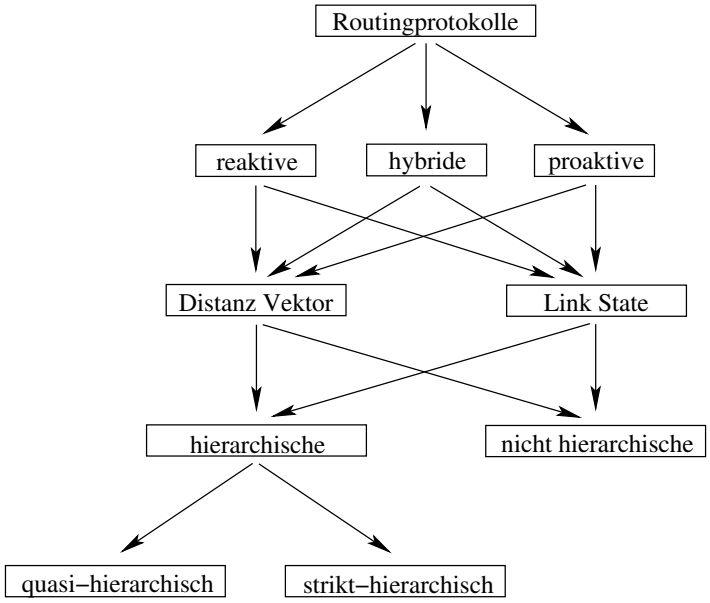

Abbildung 1: Unterteilung von Routingprotokollen

Des weiteren lassen sich Routingprotokolle in Distanz-Vektor beziehungsweise Link-State-Routingprotokolle aufteilen [KuRo02] [Tann02] [Perk01].

Distanz-Vektor-Routingprotokolle zeichnen sich dadurch aus, dass jeder Router eine Distanztabelle besitzt. Diese Tabelle besteht aus je einer Zeile für jedes Ziel und je einer Spalte für jeden direkt verbundenen Nachbarn. So stellt der Eintrag $D^X(Y,Z)$ zum Beispiel dar, dass der Router X über seinen Nachbarn Z zum Ziel Y will. Dieser Eintrag besteht aus den Kosten von X nach Z plus den bisher bekannten minimalen Kosten von Z nach Y.

$$D^X(Y,Z) = c(X,Z) + min_w\{D^Z(Y,w)\} \tag{2}$$

Dabei stellt $c(X,Z)$ die Kosten von X nach Z dar, w sind alle mit Z direkt verbundenen Nachbarn inklusive X und $min_w\{D^Z(Y,w)\}$ die bisher minimalen Kosten von Z nach Y. Hierbei wird davon ausgegangen, dass jeder Router die Kosten der Verbindungen zu allen seinen direkt angeschlossenen Nachbarn kennt. Als Kostenfunktion kann man zum Beispiel die Anzahl an Hops nehmen. Aus der Distanztabelle kann man den Next Hop auf dem kürzesten Pfad zu allen Zielen bestimmen. Ein Router kennt nie den ganzen Weg von der Quelle zum Ziel, sondern nur den Next Hop auf dem kürzestem Pfad zum Ziel. Er sendet seine Routinginformationen an alle direkt angeschlossene Nachbarn. So kann er Routinginformationen von einem oder mehreren Nachbarn empfangen und benutzt diese zur Berechnung des kürzesten Pfades. Dies geschieht mit dem Bellman-Ford Algorithmus, der nach seinem Erfinder benannt

wird. Wenn ein neuer kürzester Pfad bestimmt wurde, so wird dies den direkt angeschlossenen Nachbarn mitgeteilt. Dies geschieht so lange bis keine neuen Routinginformationen mehr entstehen. Als Beispiel für ein Distanz-Vektor-Routingprotokoll, für Festnetze, ist das Routing Information Protocol (RIP) [Malk98] zu nennen. Destination Sequenced Distance Vector Protocol (DSDV) [PeBh94] ist ein Beispiel für ein Distanz-Vektor-Routingprotokoll für Ad-hoc-Netzwerke.

Bei einem *Link-State-Routingprotokoll*, wie z.B. Open Shortest Path First (OSPF) [Moy98], verfügt jeder Router über vollständige Kenntnis des gesamten Netzwerkes. Bei OSPF handelt es sich um ein Link-State-Routingprotokoll für Festnetze. Optimized Link State Routing Protocol (OLSR) [ClJa03] ist ein Beispiel für ein Link-State-Routingprotokoll für Ad-Hoc-Netzwerke. Ein Router muss zunächst seine direkt angeschlossenen Nachbarn kennen, diese entdeckt er durch das Versenden eines HELLO-Paketes. Empfängt ein Router ein HELLO-Paket, so antwortet er darauf mit einem Paket, in dem seine Adresse steht. Der Router der das HELLO-Paket versendet hat, speichert die empfangenen Adressen in einer Tabelle mit den zugehörigen Verbindungskosten und dem Status dieser Verbindung. Jeder Router flutet seine Routinginformationen in das gesamte Netzwerk. Diese Nachrichten werden Link-State-Nachrichten genannt. Durch den Empfang von Link-State-Nachrichten bekommt jeder Router Informationen über den Teil des Netzes, der nicht direkt mit ihm verbunden ist. Nach dem Austausch von Link-State-Nachrichten haben alle Router eine vollständige und identische Sicht des gesamten Netzwerkes. Der kürzeste Pfad zu einem Ziel kann nun z.B. mit dem Dijkstra-Algorithmus bestimmt werden.

Mit wachsender Größe des Netzwerkes wird es sehr aufwändig für jeden Router einen Eintrag in einer Tabelle zu speichern. Deshalb kann man eine Hierarchie einführen. Hier wird ein Netzwerk in Gruppen unterteilt, und es werden dann nur noch Einträge für die Knoten innerhalb der gleichen Gruppe und wie man in die andere Gruppen gelangt, gespeichert. Hierarchische Routingprotokolle kann man in quasi- und strikt-hierarchische Routingprotokolle unterteilen [Perk01].

Bei *quasi-hierarchischen Protokollen* werden Pakete auf dem kürzesten Weg in Richtung Zielsubnetz geroutet. Pakete werden in Richtung des Leaders des Zielsubnetzes geschickt. Sobald ein Weg zum Ziel bekannt ist, werden die Pakete direkt dorthin geroutet. Der Leader des Zielsubnetzes dient hierbei als grobe Richtungsangabe dafür, in welcher Richtung sich der Zielknoten befindet. Ein Beispiel für ein quasi-hierarchisches Routingprotokoll ist das Landmark Routing Protocol (LANMAR), dessen Funktionsweise in Kapitel 3.1 erläutert wird.

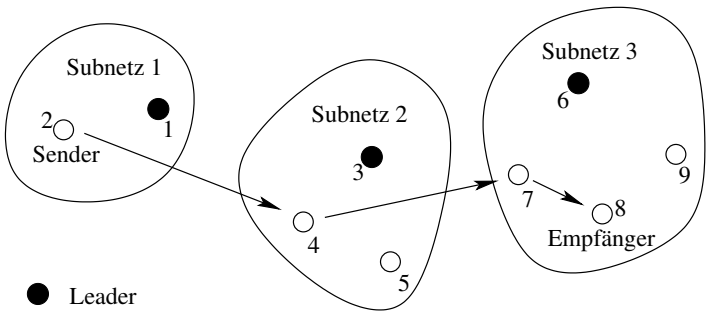

Abbildung 2: Beispieltopologie mit quasi-hierarchischem Routing

Abbildung 2 zeigt ein Beispiel, an dem die Funktion eines quasi-hierarchischen Routingprotokolls anschaulich dargestellt werden soll. In diesem Beispiel will Knoten 2 ein Paket an Knoten 8 senden. Knoten 2 sendet das Paket zunächst an Knoten 4, da dieser in der groben Richtung des Zielsubnetzes liegt. Im nächsten Schritt routet Knoten 4 das Paket in Richtung

des Knoten 7, welcher wiederum in der groben Richtung des Zielsubnetzes liegt. Von Knoten 7 aus wird das Paket direkt an den Knoten 8 weitergeleitet.

Strikt-hierarchische Routingprotokollen routen Pakete erst zum Leader des Subnetzes, in dem sich der Sender befindet. Nachdem das Paket beim Leader des Senders angekommen ist, wird das Paket über die Leader anderer Subnetze weiter zum Leader des Empfängers geroutet, von dort aus dann zum Empfänger. Hierarchical State Routing Protocol (HSR) ist ein Beispiel für ein strikt-hierarchisches Routingprotokoll. Die Funktionsweise von HSR wird in Kapitel 3.2 erläutert.

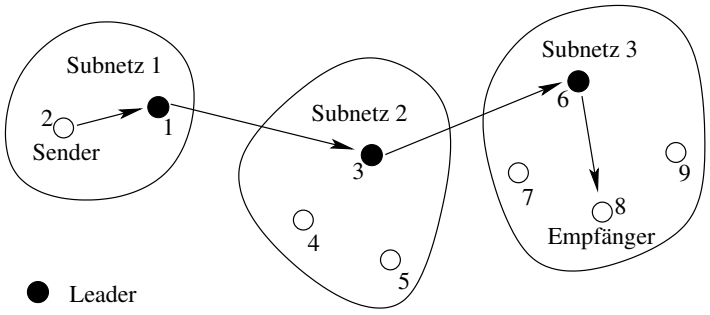

Abbildung 3: Beispieltopologie mit strikt-hierarchischem Routing

In Abbildung 3 will Knoten 2 ein Paket an Knoten 8 senden. Hierzu sendet Knoten 2 das Paket an den Leader seines Subnetzes den Knoten 1. Knoten 1 sendet das Paket an den Leader des Subnetzes 2 weiter. Von dort wird das Paket an den Leader des Zielsubnetzes gesendet. Der Leader des Zielsubnetzes kennt den direkten Weg zu Knoten 8 und sendet das Paket dorthin.

Es gibt verschiedene hierarchische Ad-hoc Routingprotokolle, z.B.:

- Landmark Routing Protocol (LANMAR) (siehe Kapitel 3.1)
- Hierarchical State Routing Protocol (HSR) (siehe Kapitel 3.2)
- Clusterhead Gateway Switch Routing Protocol (CGSR) (siehe Kapitel 3.3)
- Zone-based Hierarchical Link-State Routing Protocol (ZHLS) [JNLu99] [Misr99] [Sun00]
- Wireless Hierarchical Routing Protocol (WHIRL) [PGHC99b]
- Multimedia Support for Mobile Wireless Networks (MMWN) [RaSt98]

3 Beispiele hierarchischer Ad-Hoc Routingprotokolle

3.1 Landmark Routing Protocol (LANMAR)

Das Landmark Routing Protocol [PeGH00] [GeHP00] ist ein proaktives, quasi-hierarchisches Distanz-Vektor-Routingprotokoll. In LANMAR werden Merkmale des Fisheye State Routing Protocol (FSR) [PeGC00] verwendet. Zusätzlich wird in LANMAR das Konzept des Landmark Routing [Tsuc88] verwendet, das für große Festnetze entwickelt wurde. Hierbei wurde in den Festnetzen eine vordefinierte mehrstufige hierarchische Adressierung benötigt.

Das MANET wird hier in Gruppen eingeteilt. In einer Gruppe werden Knoten zusammengefasst, die sich im Verbund bewegen. Die Gruppen bilden logische Subnetze mit einer eigenen

Subnetz-ID. Innerhalb der Subnetze hat jeder Knoten eine eindeutige Host-ID. Subnetz-ID und Host-ID bilden zusammen die logische Adresse eines Knotens. Jeder Knoten besitzt neben der logischen Adresse eine physikalische Adresse (z.B. MAC-Adresse). Bei LANMAR besitzt jedes Subnetz im MANET einen ausgezeichneten Knoten, den Landmark-Knoten. Der Landmark-Knoten ist Repräsentant für sein Subnetz und kennt alle Knoten in seinem Subnetz. Der Bereich um einen Knoten herum, in dem Knoten liegen die nicht mehr als n Hops entfernt sind, wird als Scope bezeichnet.

Innerhalb der Subnetze wird ein proaktives Routingprotokoll verwendet, so können z.B. Destination Sequenced Distance Vector Protocol (DSDV) [PeBh94] oder FSR mit geringen Veränderungen eingesetzt werden. Falls FSR verwendet wird, so werden nur die Routingtabelleneinträge für Knoten, die sich innerhalb des Scopes befinden, periodisch aktualisiert und die Routingtabelleneinträge für Knoten außerhalb des Scopes werden nicht aktualisiert.

Jeder Knoten besitzt ein Flag, das aussagt, ob er Landmark-Knoten ist oder nicht, zusätzlich hat jeder Knoten eine Liste seiner Nachbarn, eine Topologietabelle, eine Next Hop Tabelle und eine Distanztabelle. In der Topologietabelle gibt es für jedes Ziel, das sich innerhalb des gleichen Subnetzes befindet, einen Eintrag mit Link-State-Informationen und einem Zeitstempel wie alt diese Informationen sind. Für jeden Knoten innerhalb des gleichen Subnetzes und jeden Landmark-Knoten wird der Next Hop auf dem kürzesten Pfad in der Next Hop-Tabelle gespeichert. Die Distanz des kürzesten Pfades von einem Knoten A zu einem Knoten B wird in der Distanztabelle gespeichert. Die Einträge in der Next Hop-Tabelle, die auf Landmarks weisen, bilden eine zusätzliche Tabelle, die Landmark-Distance-Vector Tabelle (LMDV). Diese drei Tabellen werden periodisch mit den Nachbarn ausgetauscht.

Der Landmark-Knoten innerhalb eines Subnetzes wird dynamisch von allen Knoten im Subnetz gewählt. Zu Beginn wenn noch keine Landmark-Knoten vorhanden sind, erklärt sich jeder Knoten, der feststellt, dass er eine gewisse Anzahl m an direkt verbundenen Nachbarn in seinem Scope hat zum Landmark-Knoten. Nach mehrmaligem Austausch von Routinginformationen, haben die Knoten, die sich in der Mitte des Subnetzes befinden genügend Nachbarn und erklären sich zum Landmark-Knoten. Jeder Knoten der sich zum Landmark-Knoten erklärt hat, versendet mit dem LMDV und den anderen Routinginformationen ein Landmark-Status-Tupel. Das Landmark-Status-Tupel besteht aus der Adresse des Landmarks und der Anzahl an Knoten, die sich in seinem Scope befinden. Empfängt ein Knoten, der nicht Landmark ist eine solche Nachricht, so aktualisiert er seine Routinginformationen. Wenn ein Landmark-Knoten der sich im gleichen Subnetz befindet eine solche Nachricht empfängt, so wird ein Wettbewerb zwischen den beiden Landmark-Knoten ausgetragen. Der Knoten, der die meisten direkt verbundenen Nachbarn in seinem Scope hat, wird dann zum Landmark und der andere wird zu einem normalen Knoten im Subnetz. Falls sie gleich viele Nachbarn in ihrem Scope haben, so wird der Knoten mit der kleineren Host-ID Landmark. Wenn die Nachricht des letzten Landmarkwechsels alle Knoten erreicht hat, so befindet sich in jedem Subnetz nur noch ein einzelner Landmark-Knoten. Für den Fall, dass ein Nachbar eines Landmarks eine Zeit lang nichts mehr von diesem Landmark gehört hat, geht er davon aus, dass der Landmark nicht mehr vorhanden ist und erklärt sich selbst zum Landmark. Nun beginnt das Verfahren von neuem, bis alle Knoten den neuen Landmark im Subnetz bestimmt haben.

Soll nun ein Datenpaket an einen Knoten innerhalb des gleichen Subnetzes gesendet werden, so kann das Datenpaket direkt zum Zielknoten geroutet werden, denn die Adresse des Zielknotens befindet sich in der Routingtabelle. Wenn ein Datenpaket zu einem Knoten außerhalb des Subnetzes geschickt werden soll, so wird die Subnetz-ID aus der logischen Adresse extrahiert. Anhand der Subnetz-ID findet der Sender den Eintrag des Landmark-Knoten des Zielsubnetzes in seiner Routingtabelle. Das Datenpaket wird in Richtung des Zielsubnetzes geroutet. Sobald ein Knoten, in dessen Scope sich der Zielknoten befindet, das Datenpaket empfängt,

routet dieser das Paket direkt zum Zielknoten. Daher kann es sein, dass das Datenpaket nicht über den Landmark-Knoten des Zielsubnetzes geroutet wird.

Im Normalfall sind alle Knoten des Subnetzes im Scope des Landmark-Knotens, so dass er zu jedem Knoten eine Route kennt. Es kann aber passieren, dass sich ein Knoten aus dem Scope des Landmark-Knotens bewegt. Solche Knoten werden Drifter genannt. Damit der Landmark-Knoten trotzdem eine Route zu jedem Drifter kennt, werden zusätzliche Routinginformationen benötigt. Die Knoten vergleichen den vorgegebenen Scope-Wert mit der Distanz zu ihrem Landmark-Knoten. Ist die Distanz zum Landmark-Knoten größer als der Scope-Wert, so trägt sich der Drifter in eine Drifter-Liste ein. Diese Liste wird als Drifter-Distanz-Vektor (DFDV) an den Landmark-Knoten weitergegeben. Der DFDV enthält die Adresse des Drifters, den Next Hop auf dem Weg zum Drifter, die Distanz des kürzesten Pfades zum Drifter und einen Zeitstempel, wann zuletzt etwas vom Drifter gehört wurde. Ein Knoten A, der auf dem kürzesten Pfad zwischen dem Landmark-Knoten L und dem Drifter D liegt, speichert einen Drifter-Distanz-Vektor. Falls der Drifter sich im Scope des Knotens A befindet, so enthält die Routingtabelle des Knotens A bereits einen Eintrag für den Drifter. Ein Knoten B übernimmt beim periodischen Austausch der Routinginformationen mit Knoten A den Drifter-Distanz-Vektor wenn entweder $d(B, D) < scope$ oder $d(B, L) < d(A, L)$ gilt. Wobei $d(x, y)$ die Distanz des kürzesten Pfades zwischen zwei Knoten ist. Hierbei wird die Distanz zum Drifter um eins erhöht, wenn die Distanz in Hops gemessen wird. Die zweite Bedingung ist genau dann erfüllt, wenn Knoten B auf dem kürzesten Pfad zwischen dem Landmark-Knoten und dem Drifter liegt. Hierdurch wird eine Route zwischen dem Landmark-Knoten und Drifter geschaffen.

In einem MANET können auch isolierte Knoten auftreten, z.B. wenn die Gruppe nur aus einem einzelnen Knoten besteht. Wenn es nicht viele Knoten dieser Art gibt, so kann man sie als Landmark-Knoten behandeln. Jedoch wenn solche Knoten einen beträchtlichen Teil des MANET ausmachen, so ist es besser die Routen zu diesen Knoten auf andere Art und Weise zu speichern. Eine Möglichkeit ist es die Routen zu isolierten Knoten auf Bedarf zu berechnen. Eine andere ist die Einführung von Home Agents bei denen sich die isolierte Knoten registrieren und regelmäßig melden. So wird die Route vom Home Agent zu den isolierten Knoten gespeichert. Die Methode, um eine Route zwischen Home Agent und isoliertem Knoten aufrecht zu erhalten entspricht der Methode für Landmark-Knoten und Drifter. Die effiziente Verwaltung von isolierten Knoten in LANMAR stellt ein noch ungelöstes Problem dar. Die oben genannten Möglichkeiten sind nur einige mögliche Lösungsansätze.

3.2 Hierarchical State Routing Protocol (HSR)

Das Hierarchical State Routing Protocol [PGHC99a] [PeGe99] [ICPG+99] [Misr99] ist ein proaktives, strikt-hierarchisches Link-State-Routingprotokoll. Bei HSR wird eine mehrstufige Hierarchie im MANET eingeführt.

Das MANET wird in Cluster unterteilt und in jedem dieser Cluster wird ein Knoten zum Cluster-Head gewählt. Dieser Cluster-Head ist Repräsentant für sein Cluster. Die Cluster-Heads bilden selbst wieder Cluster auf der nächst höheren Hierarchieebene. Die Bildung der Cluster beruht auf geographischen Beziehungen zwischen Knoten, deshalb nennt man es physikalische Clusterbildung. Es gibt verschiedene Algorithmen [GeTs95] [CWLG97] [ChGe97] für das dynamische Bilden der Cluster und das Wählen der Cluster-Heads.

Ein physikalisches Cluster besteht aus drei unterschiedlichen Arten von Knoten. Einem Cluster-Head, der Repräsentant für das jeweilige Cluster ist. In einem Cluster gibt es einen oder mehrere Gateway-Knoten. Ein Gateway-Knoten ist ein Knoten, der zwei Clustern angehört und somit die Verbindung zwischen diesen Clustern herstellt. Des weiteren gehören jedem Cluster keine oder mehrere innere Knoten an. So ist in Abbildung 4 der Knoten 1

Cluster-Head des Clusters C11 und die Knoten 6 und 7 Gateway-Knoten, die die Verbindung zu den Clustern C12 und C14 herstellen. Der Knoten 5 ist ein innerer Knoten.

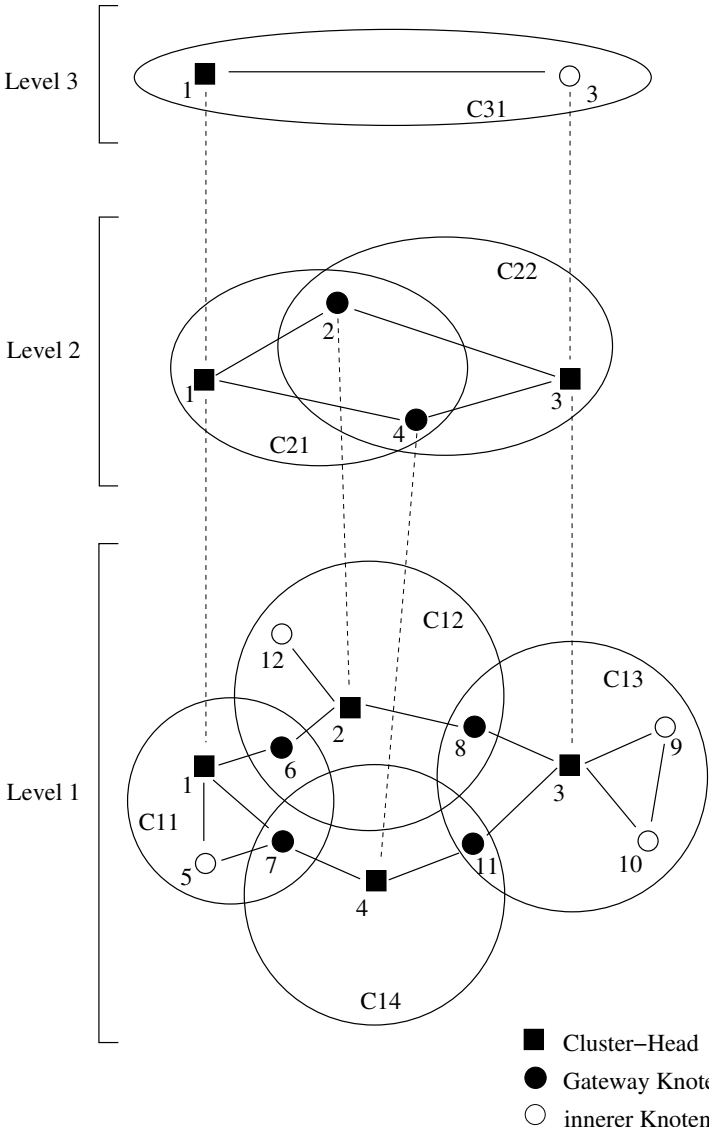

Abbildung 4: Beispiel der physikalischen Clusterbildung mit HSR

Jeder Knoten eines physikalischen Clusters tauscht mit allen Knoten des gleichen Clusters seine Link-State-Informationen aus. Der Cluster-Head eines Clusters sammelt alle Link-State-Informationen seines Clusters und tauscht sie mit den benachbarten Cluster-Heads der anderen Cluster aus. Der Cluster-Head sendet seine Link-State-Informationen zu den Gateway-Knoten in seinem Cluster und diese übermitteln diese an die benachbarten Cluster-Heads. Benachbarte Cluster-Heads sind Mitglieder des gleichen Clusters auf der nächst höheren Hierarchieebene. Die Knoten der Cluster auf den höheren Hierarchieebnen tauschen ihre Link-State-Informationen genauso wie die Knoten auf der untersten Hierarchieebene aus. Ein Knoten jeder Hierarchieebene flutet seine Informationen zur nächst tieferen Hierarchieebene. So erhalten die Knoten in der tieferen Hierarchieebene eine hierarchische Topologie des MANET.

Die Knoten im MANET besitzen eine eindeutige Knoten-ID, z.B. die MAC-Adresse. Zusätzlich zur Knoten-ID hat bei HSR jeder Knoten im MANET eine hierarchische Adresse. Verknüpft man die Knoten-IDs der Knoten, die auf dem Weg von der obersten Hierarchieebene zum Knoten selbst liegen, so erhält man die hierarchische Adresse dieses Knotens. Ein

Gateway-Knoten besitzt mehrere hierarchische Adresse, denn er gehört mehreren Clustern an und kann somit auf unterschiedlichen Wegen von der obersten Hierarchieebene erreicht werden.

So ist zum Beispiel die hierarchische Adresse des Knoten 12 <3.2.12> (Abbildung 4). Die hierarchische Adresse des Knotens 12 ist die Verknüpfung der Knoten-IDs der Knoten, die auf dem Pfad von der obersten Hierarchieebene zum Knoten 12 selbst liegen. Knoten 3 ist Mitglied der obersten Hierarchieeben und ebenfalls Cluster-Head des Clusters C22 in dem der Knoten 2 liegt. Der Knoten 2 ist Cluster-Head des Clusters C12, in welchem auch der Knoten 12 liegt. So kann der Knoten 12 direkt vom Knoten 2 aus erreicht werden. Für Knoten 2 ist der Knoten 3 der Repräsentant auf der obersten Hierarchieebene und für den Knoten 12 der Knoten 2 auf der nächst höheren Hierarchieebene.

Die hierarchische Adresse genügt, um den Weg zu jedem Knoten aus dem gesamten MANET zu kennen. Wenn Knoten 12 nun ein Paket an den Knoten 10 senden möchte, so sendet der Knoten 12 das Paket zuerst an den Cluster-Head des Clusters C12, den Knoten 2. Der Knoten 2 ist Mitglied des Clusters C22 auf der nächst höheren Hierarchieebene. Knoten 2 tauscht auf dieser Hierarchieebene die Routinginformationen mit Knoten 3 und 4 aus und weiß daher dass Knoten 10 im Cluster C13 liegt. Er reicht das Paket an Knoten 3, den Cluster-Head des Clusters C22 weiter. Knoten 3 ist Repräsentant von Knoten 10 auf der obersten Hierarchieebene und kennt den Weg zum Knoten 10. Knoten 3 routet das Paket direkt zum Knoten 10.

Neben der physikalischen Clusterbildung gibt es noch eine Einteilung der Knoten in logische Subnetze (Abbildung 5). Knoten die eine logische Beziehung untereinander haben werden zu einem Subnetz zusammengefasst. Jedes Subnetz hat eine eindeutige Subnetz-ID. Somit hat jeder Knoten eine logische Adresse, die aus Subnetz-ID und Knoten-ID besteht. Ein Subnetz kann Knoten aus verschiedenen physikalischen Clustern umfassen. Diese logische Partitionierung ist für das Lokations Management in HSR wichtig, da hierdurch eine Trennung zwischen Lokation Management und der physikalischen Hierarchie stattfindet. Durch die Trennung von Lokation Management und physikalischer Hierarchie ändert sich die Struktur des Lokation Managements nicht bei jeder Änderung der physikalischen Cluster.

Jedem logischen Subnetz ist ein Location Management Server (LMS) zugeordnet. Der LMS verschickt seine hierarchische Adresse an die höheren Hierarchieebenen und an alle anderen LMS. Somit ist die hierarchische Adresse des LMS allen Knoten im Subnetz bekannt. Alle Knoten eines Subnetzes registrieren sich mit ihrer logischen und der zugehörigen hierarchischen Adresse beim LMS des Subnetzes. Die Knoten registrieren sich periodisch bei ihrem LMS neu. Wenn ein LMS von einem Knoten nach einer gewissen Zeit keine erneute Registrierung empfängt, so entfernt er diesen aus seiner Tabelle. Sobald ein Knoten von einem Cluster in ein anderes wechselt, wird der Eintrag in der Tabelle des LMS durch eine erneute Registrierung aktualisiert. Anstatt sich jeder Knoten eines Clusters, der dem gleichen Subnetz angehört, selbst beim zugehörigen LMS registriert, kann der Cluster-Head dieses Clusters eine zusammengesetzte Registrierungs-Nachricht für alle Knoten, die diesem Subnetz angehören, an den LMS schicken.

Möchte ein Knoten A ein Paket an einen Knoten B senden, so verwendet er die logische Adresse von Knoten B. Aus der logischen Adresse wird die Subnetz-ID extrahiert, mit dieser bekommt man die hierarchische Adresse des LMS der im Zielsubnetz liegt. Die hierarchische Adresse des LMS im Zielsubnetz ist entweder in der Routingtabelle von Knoten A oder man bekommt sie von einer höheren Hierarchieebene. Das Paket wird anhand der hierarchischen Adresse an den LMS im Zielsubnetz gesendet und von dort aus direkt an Knoten B, da hier die hierarchische Adresse von B bekannt ist. Der LMS sendet die hierarchische Adresse von B an den Knoten A zurück. Jeder Knoten besitzt einen Cache, in dem die hierarchischen Adressen gespeichert werden. Sobald Knoten A und Knoten B gegenseitig ihre hierarchische

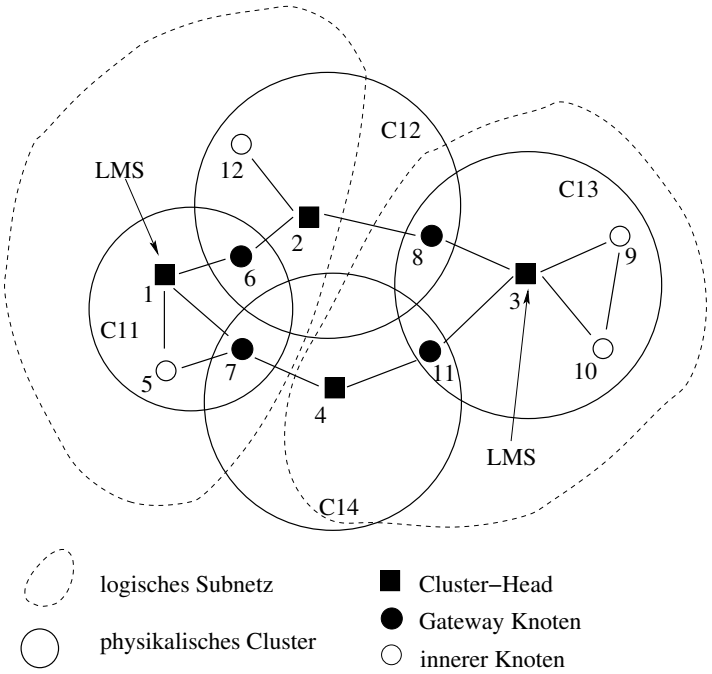

Abbildung 5: Beispiel der logischen Subnetzbildung mit HSR

Adresse kennen, können sie miteinander kommunizieren, ohne den LMS mit einzubeziehen. Falls die hierarchische Adresse des Knotens, an den das Paket gesendet werden soll, schon im Cache des Senders vorliegt, so wird das Paket direkt an das Ziel geroutet.

3.3 Clusterhead Gateway Switch Routing Protocol (CGSR)

Das Clusterhead Gateway Switch Routing Protocol [CWLG97] [Sun00] [Misr99] ist ein proaktives, strikt-hierarchisches Distanz-Vektor-Routingprotokoll. Bei CGSR dient der DSDV-Routing-Algorithmus [PeBh94] als Grundlage. CGSR unterscheidet sich von DSDV durch die Art der Adressierung und der Organisation des Netzwerkes. Bei CGSR wird im Gegensatz zu DSDV ein hierarchisches Adressierungsschema verwendet.

Innerhalb des MANET werden Knoten zu Clustern zusammengefasst. In jedem Cluster wird dynamisch ein Cluster-Head gewählt. Hierzu wird der Least Cluster Change Algorithmus (LCC) [CWLG97] eingesetzt. Der LCC Algorithmus unterscheidet fünf Zustände:

1. Zu Beginn, wenn noch keine Cluster-Heads bestimmt sind, werden diese mit dem Lowest-ID oder Highest-Connectivity Cluster-Algorithmus bestimmt. Beim Lowest-ID Cluster-Algorithmus [GeTs95] wird der Knoten im Cluster mit der niedrigsten Host-ID zum Cluster-Head. Beim Highest-Connectivity Cluster-Algorithmus [GeTs95] wird der Knoten mit den meisten direkt verbundenen Nachbarn zum Cluster-Head.

2. Falls ein Knoten aus dem Cluster A in das Cluster B wechselt, und der Knoten kein Cluster-Head ist, so ändert sich nichts an den Cluster-Heads der Cluster A und B.

3. Wenn ein Knoten sich aus einem Cluster bewegt, aber nicht in ein vorhandenes Cluster wechselt, so wird dieser Knoten Cluster-Head eines neuen Clusters.

4. Der Cluster-Head des Clusters A wechselt in das Cluster B, so wird ein Wettbewerb zwischen den beiden Cluster-Heads ausgetragen, entweder der Cluster-Head mit der

niedrigsten Host-ID oder der mit den meisten direkt verbundenen Nachbarn wird der neue Cluster-Head im Cluster B. Im Cluster A wird mit dem Lowest-ID oder dem Highest-Connectivity Cluster-Algorithmus ein neuer Cluster-Head bestimmt.

5. Wenn mehrere Knoten ein Cluster verlassen und nicht in ein vorhandenes Cluster wechseln, so bestimmen sie mit dem Lowest-ID oder Highest-Connectivity Cluster-Algorithmus einen Cluster-Head und bilden ein neues Cluster.

Beim LCC Algorithmus bewirken also nur zwei Gegebenheiten eine Änderung der Cluster-Heads. Dies stellt eine Verbesserung gegenüber anderen Cluster-Algorithmen dar, bei denen der Cluster-Head jedes Mal wenn ein Knoten das Cluster wechselt neu bestimmt wird.

Zu einem Cluster gehören alle benachbarten Knoten des Cluster-Heads. Gateway Knoten, die sich in mehreren Clustern befinden, stellen die Verbindung zwischen den einzelnen Clustern her.

Jeder Knoten im Cluster speichert eine Cluster Member-Tabelle, in der zu jedem Knoten im MANET der Cluster-Head des zugehörigen Clusters gespeichert wird. Diese Tabelle wird periodisch durch den DSDV-Algorithmus in das MANET geflutet. Wenn ein Knoten eine solche Nachricht empfängt, so aktualisiert er seine Routing-Informationen. Zusätzlich zur Cluster Member-Tabelle hat jeder Knoten eine Next Hop-Tabelle, in der der Next Hop zu jedem Ziel gespeichert wird.

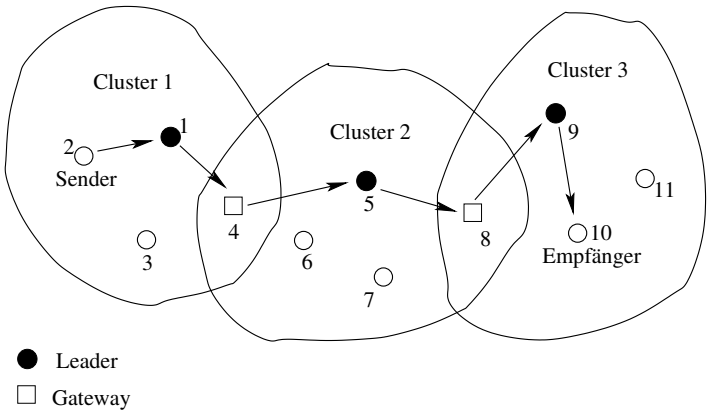

Abbildung 6: Beispiel eines Routing mit CGSR

Will nun Knoten 2 ein Paket an Knoten 10 senden, so schickt der Knoten 2 das Paket an den Knoten 1, den Cluster-Head seines Clusters (Abbildung 6). Der Cluster-Head findet in seiner Cluster Member-Tabelle und der Next Hop-Tabelle den Gateway Knoten 4 zum nächsten Cluster-Head auf dem Weg zum Zielknoten. Der Knoten 1 sendet das Paket an den Gateway-Knoten 4 und dieser an den nächsten Cluster-Head (Konten 5) auf dem Weg zum Zielknoten. Knoten 5 sendet das Paket zum nächsten Gateway-Knoten auf dem Weg zu Ziel. Kommt das Paket beim Cluster-Head des Clusters an, in dem sich der Zielknoten befindet, so wird das Paket direkt an den Knoten 10 weitergeleitet.

4 Vor- und Nachteile der vorgestellten Protokolle

Bei LANMAR ist ein Ansatz eines Mobility Managements in Form der Verwaltung der Drifter zu finden. Diese Verwaltung ist nicht optimal. Durch die Verwaltung von Driftern wird versucht, die Mobilität der Knoten im MANET zu unterstützen. Jedoch können die Drifter

beim Wechsel des Subnetzes ihre Subnetz-ID nicht ändern. Somit muss der Landmark des ursprünglichen Subnetzes dafür sorgen, dass die Pakete für diesen Knoten in das neue Subnetz weitergeleitet werden. Dadurch steigt der Verwaltungsaufwand für den Landmark. Weiterhin können durch häufige Subnetzwechsel sehr lange Wege entstehen. Es ist möglich, dass sehr viele Knoten ihr Subnetz wechseln, hierdurch wächst die Drifter Liste stark an. Dadurch, dass die Drifter Liste (DFDV) periodisch mit allen Nachbarn ausgetauscht wird, wird das MANET sehr stark beansprucht und die vorhandene Bandbreite stark eingeschränkt.

Dennoch hat das Mobility Management von LANMAR den Vorteil, dass es sehr einfach ist und für wenige Knoten, die ihr Subnetz wechseln, ausreicht. Dagegen ist durch die Verwendung von Location Management Servern bei HSR ein besseres Mobility Management vorhanden. Trotz des besseren Mobility Management bei HSR treten noch Probleme auf:

Wenn ein Location Management Server (LMS) aus einem Subnetz in ein anderes Subnetz wechselt, müsste ein neuer LMS bestimmt werden. Dieser müsste zuerst die ganze Hierarchie des Subnetzes in seinen Routingtabellen aufbauen. Dies dauert einige Zeit, in dieser Zeit sind die Knoten die sich in diesem Subnetz befinden unerreichbar für andere Knoten außerhalb des Subnetzes. Weiterhin muss die hierarchische Adresse des neuen LMS erst im ganzen Netz bekannt gemacht werden.

Bei CGSR ist das Mobility Management sehr einfach gehalten. Es wird dadurch realisiert, dass jeder Konten seine Cluster Member-Tabelle in das Netzwerk flutet. Es dauert lange bis alle Knoten von dem Wechsel eines Knotens aus einem Cluster in ein anderes Cluster erfahren. Das Fluten der Cluster Member-Tabelle stellt eine hohe Netzbelastung dar und schränkt somit die verfügbare Bandbreite ein.

5 Fazit

Die hier vorgestellten Routingprotokolle unterscheiden sich in vielen Eigenschaften und sie sind für verschiedene Netzwerke unterschiedlich gut geeignet. Deshalb muss man in jedem Fall das Routingprotokoll auswählen das am besten geeignet ist. LANMAR ist ein proaktives, quasi-hierarchisches Distanz-Vektor-Routingprotokoll. Bei LANMAR werden die Routing Tabellen reduziert und dadurch wird eine gute Skalierbarkeit für große MANETs erreicht. LANMAR ist gut geeignet für MANETs, in denen wenige Knoten ihr Subnetz wechseln, denn das Mobility Management in LANMAR ist nicht dafür ausgelegt, dass viele Knoten ihr Subnetz wechseln. Bei HSR handelt es sich um ein proaktives, strikt-hierarchisches Link-State-Routingprotokoll. Durch das Mobility Management das in HSR verwendet wird, bietet es sich an HSR in MANETs zu verwenden, in denen viele Knoten ihr Subnetz wechseln. HSR bietet eine gute Skalierbarkeit für große MANETs durch die Verwendung einer mehrstufigen Hierarchie. CGSR ist ein proaktives, strikt-hierarchisches Distanz-Vektor-Routingprotokoll. Dadurch, dass in CGSR nur ein sehr einfaches Mobility Management vorhanden ist, eignet sich CGSR nur für MANETs, in denen sich die Knoten sehr selten oder fast gar nicht aus ihrem Subnetz bewegen.

Literatur

[Beij] Nicklas Beijar. Zone Routing Protocol (ZRP).

[ChGe97] Ching-Chuan Chiang und Mario Gerla. Routing in Clustered Multihop, Mobile Wireless Networks. In *IEEE International Conference on Universal Personal Communications (ICUPC'97), San Diego, California, Oct. 1997.*, 1997, S. 546–551.

[ClJa03] Thomas Heide Clausen und Philippe Jacquet. IETF RFC 3626: Optimized Link State Routing Protocol (OLSR). http://www.ietf.org/rfc/rfc3626.txt?number=3626, Oktober 2003.

[CWLG97] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu und Mario Gerla. Routing in Clustered Multihop, Mobile Wireless Networks With Fading Channel. In *The IEEE Singapore International Conference on Networks, 1997*, 1997, S. 197–211.

[GeHP00] Mario Gerla, Xiaoyan Hong und Guangyu Pei. Landmark routing for large ad hoc wireless networks. In *Proceedings of IEEE GLOBECOM 2000, San Francisco, CA, Nov. 2000*, 2000.

[GeTs95] Mario Gerla und Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. In *ACM/Baltzer Journal of Wireless Networks. vol. 1, (no. 3), 1995*, 1995, S. 255–265.

[ICPG+99] Atsushi Iwata, Ching-Chuan Chiang, Guangyu Pei, Mario Gerla und Tsu wei Chen. Scalable Routing Strategies for Ad-hoc Wireless Networks. In *IEEE JSAC, August 1999*, 1999.

[JNLu99] Mario Joa-Ng und I-Tai Lu. A Peer-to-Peer zone-based two-level link state routing for mobile Ad Hoc Networks. In *IEEE JSAC, August 1999*, 1999, S. 1415–1425.

[JoMa96] David B. Johnson und David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, 1996.

[KuRo02] James F. Kurose und Keith W. Ross. Routing-Prinzipien. In *Computernetzwerke – Ein Top-Down-Ansatz mit Schwerpunkt Internet*, Kapitel 4.2., S. 281–301. Pearson Studium, 2002.

[Malk98] Gary Scott Malkin. IETF RFC 2453: RIP Version 2. http://www.ietf.org/rfc/rfc2453.txt?number=2453, November 1998.

[Misr99] Padmini Misra. Routing Protocols for Ad-hoc Mobile Wireless Networks. http://cis.ohio-state.edu/~jain/cis788-99/ftp/adhoc_routing/, 1999.

[Moy98] John Moy. IETF RFC 2328: OSPF Version 2. http://www.ietf.org/rfc/rfc2328.txt?number=2328, April 1998.

[mWor] IETF manet Working Group. Mobile Ad Hoc Networks (manet). http://www.ietf.org/html.charters/manet-charter.html.

[PeBh94] Charles Perkins und Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994, S. 234–244.

[PeBRD03] Charles E. Perkins, Elizabeth M. Belding-Royer und Samir R. Das. IETF RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing. http://www.ietf.org/rfc/rfc3561.txt?number=3561, Juli 2003.

[PeGC00] Guangyu Pei, Mario Gerla und Tsu-Wei Chen. Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks. In *Proceedings of the IEEE International Conference on Communications (ICC), New Orleans, LA, June 2000*, 2000, S. 70–74.

[PeGe99] Guangyu Pei und Mario Gerla. Mobility Management in Hierarchical Multi-hop Mobile Wireless Networks. In *Proceedings of IEEE ICCCN'99, Boston, MA, Oct. 1999*, 1999, S. 324–329.

[PeGH00] Guangyu Pei, Mario Gerla und Xiaoyan Hong. LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility. In *Proceedings of IEEE/ACM MobiHOC 2000, Boston, MA, Aug. 2000*, 2000, S. 11–18.

[Perk97] Charles E. Perkins. Ad-hoc on-demand distance vector routing. In *Proceedings of MILCOM '97 panel on Ad Hoc Networks, Nov. 1997.*, 1997.

[Perk01] Charles E. Perkins. *Ad Hoc Networking.* Addison-Wesley. 2001.

[PGHC99a] Guangyu Pei, Mario Gerla, Xiaoyan Hong und Ching-Chuan Chiang. A Wireless Hierarchical Routing Protocol with Group Mobility. In *Proceedings of IEEE WCNC'99, New Orleans, LA, Sep. 1999*, 1999, S. 18.

[PGHC99b] Guangyu Pei, Mario Gerla, Xiaoyan Hong und Ching-Chuan Chiang. Wireless Hierarchical Routing Protocol with Group Mobility (WHIRL). In *Proceedings of IEEE WCNC'99, New Orleans, LA, Sep. 1999*, 1999.

[RaSt98] Ram Ramanathan und Martha Steenstrup. Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support. In *ACM/Baltzer Mobile Networks and Applications, vol. 3, no. 1, Jun. 1998*, 1998, S. 101–119.

[Sun00] Allen C. Sun. Design and Implementation of Fisheye Routing Protocol for Mobile Ad Hoc Networks. Diplomarbeit, Massachusetts Institute of Technology, 2000.

[Tann02] Andrew S. Tannenbaum. Routing Algorithms. In *Computer Networks*, Kapitel 5.2., S. 350–380. Prentice Hall PTR, 2002.

[Tsuc88] Paul F. Tsuchiya. The landmark hierarchy: A new hierarchy for routing in very large networks. In *ACM Computer Communication Review, vol. 18, no. 4, August 1988*, 1988, S. 35–42.

Abbildungsverzeichnis

Mobile IP Erweiterungen zur Unterstützung von Mikro Mobilität

Pei Niu

Kurzfassung

Die IETF Mobile IP Working Group hat verschiedene Protokolle über Mikro-Mobilität diskutiert, um die Latenz, den Paketverlust und den Overhead für die Signalisierung zu reduzieren, welche bei normalen Mobile IP existieren. In diesem Dokument werden wir verschiedene Mikro-Mobilitäts-Protokolle mit der Fähigkeit für Fast Handoff und Paging vergleichen. Diese Dokument bietet daneben eine Überblick über Hierarchische Mobile IP.

1 Einleitung

1.1 Mobile IP

Das traditionelle Internet Protokoll, welches von Internet Engineering Task Force(IETF) standardisiert wurde, basiert auf IP-Adresse und Netzwerk-Präfix. Die IP-Adresse legt deshalb das Subnetz fest. Falls sich ein Teilnehmer von einem Subnetz zu einem anderen bewegt, muss die IP-Adresse und das Netzwerk-Präfix geändert werden. Momentan tauchen immer neue Rechner oder Elektronik-Geräte auf, die mobilitätsfähig sind. Damit können die Teilnehmer zu einer beliebigen Zeit, an einem beliebigen Ort auf das Internet zugreifen. Deswegen gründet die IETF eine neue Working Group, die Mobile IP Working Group, um Mobilität für Endgeräte zu unterstützen.

1.1.1 Einführung in Mobile IP

In diesem Abschnitt diskutieren wir das Prinzip von Mobile IP. Auf den Ablauf gehen wir später ausführlich ein.

Damit ein Mobile-IP-Node den Access Point wechseln kann, werden zwei Typen von Mobility Agent definiert: der Home Agent und der Foreign Agent. Der Home Agent liegt innerhalb des Heimat-Netzwerks von des Mobile Nodes, dagegen steht der Foreign Agent in Fremd-Netzwerk, wo sich der Mobile Node momentan befindet.

Jedes Mal, wenn der Mobile Node einen anderen Access Point wählt, erhält er eine neue, zeitweilige Adresse, die so genannte Care-of-Adresse (COA), von einem Foreign Agent zugewiesen, mit dem der Mobile Node verbunden ist. Der Foreign Agent sendet Advertisement Messages, die dem Mobile Node eine CoA zuteilen. Die Router Advertisement Messages werden immer periodisch gesendet. Der Mobile Node kann eine Router Solicitation Message senden, um den Foreign Agent aufzufordern, eine Router Advertisement Message zu senden. Der Mobile Node meldet sich dann einem Foreign Agent an. Wenn sich der Mobile Node anmeldet, trägt der Foreign Agent die Verbindungsinformation in einer Tabelle ein, der so genannten Visitor List.

Erst wenn der Mobile Node eine neue Care of Adresse erhält, muss er die neue Adresse zu dem Home Agent in einem Registrierungsprozess übermitteln. Wenn der Home Agent die neue Care of Adresse von Mobile Node bekommt, fängt er alle zu dem Mobile Node schickende Packet ab, danach tunnelt er die Pakete zu dem Foreign Agent. Der Foreign Agent die auspacket die getunnelten Pakete und leitet die Pakete weiter zu dem Mobile Node.

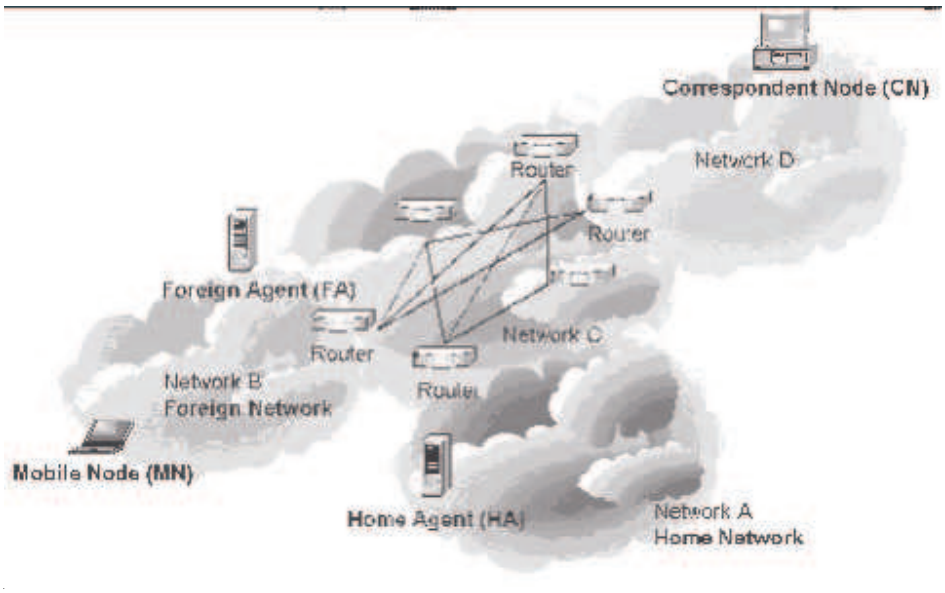

Abbildung 1: Mobile IP

1.1.2 Terminologie

- Mobile Node (MN): Knoten, der den Ort des Netzanschlusses wechseln kann, ohne seine IP-Adresse ändern zu müssen.
- Home Agent (HA): Verwaltet Aufenthaltsort des MN, tunnelt IP-Datagramme zum Mobile Node, typischerweise ist der Router im Heimatnetz des MN.
- Foreign Agent (FA): Einheit im momentanen "Fremdnetz" der MNs, ist typischerweise ein Router. Der leitet die getunnelte Datagramme zum mobilen Node weiter. Meist stellt er auch den Default-Router für den MN dar und stellt er die Care-of-Adresse zur Verfügung.
- Care-of-Adresse (COA): die Adresse des für den Mobile Node aktuell gültigen Tunnelendpunkt. Die aktuelle Lokation von MN wird von COA dargestellt.
- Correspondent Node (CN): der Kommunikationspartner von MN

1.1.3 Anforderungen an Mobile IP

Ein mobiles Endgerät hat genau eine IP Adresse, trotz es zu einem fremden Netz kommt. Beim Wechseln eines Anschlusspunkts oder Abtrennung von Netz wird die Verbindung sehr selten beeinflusst. Für den Korrespondent Node ist der Mobile IP transparent.

Mobile IP sollte mit normalem IP kompatibel sein. Ohne Änderungen von bisherigen Rechnern und Routern soll das mobile Endgerät mit Endgeräten im Festnetz kommunizieren können.

Für das Kommunizieren über die Luft wird hohe Sicherheit gebraucht. Alle Registrierungsnachrichten müssen authentifiziert werden.

Möglichst wenige zusätzliche Daten sollen mit dem mobilen Endgerät ausgetauscht werden zur Unterstützung einer großen Anzahl mobiler Endgeräte wird Effizienz und Skalierbarkeit benötigt.

1.2 Makro-Mobilität und Mikro-Mobilität

Mobile IP hat einige offensichtliche Schwächen, im Bereich der Mikro-Mobilität.

In Mobile IP enthält die grundlegende Mobilitätsverwaltung zwei Teile: die Entdeckung des Zugangspunkts von dem MN und die Registrierung zu dem HA. Jedes Mal wenn der MN den Zugangspunkt wechselt, muss er die beiden Verfahren durchführen. Aber der MN sendet eine Registrierungsanforderung erst wenn er den neuen Zugangspunkt entdeckt hat und eine neue COA erhalten hat. Dies bringt zwei Latenzen mit sich: die Latenz von der Entdeckung des Zugangspunkts und die Latenz der Registrierung. Die Latenz von der Entdeckung des Zugangspunkts kann größer werden, weil sie auf dem Intervall von Router Advertisement Messages und dem Vergleich der Präfixe verschiedener Router Advertisement Messages basiert. Die Latenz der Registrierung ist die Zeit, die für die Registrierung der neuen CoA bei dem HA gebraucht wird, weil sich der HA über all im Internet befinden kann, kann dieses Verfahren sehr lange daueren werden. Manchmal kann dieses Verfahren auch nicht erfolgreich beendet werden.

Wegen dieser Latenzen werden Pakete zum alten FA gesendet, wenn sich der Mobile Node schon im Zugangsbereich des neuen FA befindet. Dies verursacht den Verlust von diesen Paketen. Falls sich der Mobile Node mit einer hohen Geschwindigkeit bewegt, wird der Verlust von Paketen immer größer. Im Internet und im lokalen Netzwerk wird auch viel Verkehr von Kontrollinformation erzeugt.

Deshalb unterscheidet man die Mikro-Mobilität und die Makro-Mobilität. Das heißt, für die Mobilitäts-Unterstutzung werden zwei unterschiedliche Typen von Protokoll benutzt. Mobile IP wird für die Verwaltung der Bewegung von MN zwischen unterschiedliche Domänen benutzt (Makro-Mobilität). Das Mikro-Mobilitäts-Protokoll wird für die Mobilität innerhalb einer Domäne gebraucht.

Ein Mikro-Mobilitäts-Protokoll läuft wie folgt. Der MN ermittelt eine lokale COA wenn er an einer Domain anmeldet. Die COA bleibt wenn der MN in dieser Domäne steht. Deshalb registriert der MN seine CoA beim HA nur ein Mal, nämlich wenn er in der Domäne ankommt. Wenn sich der MN innerhalb der Domäne bewegt, greift das Mikro-Mobilitäts-Protokoll. Dies ist transparent für den HA. Die Latenz sowie der Kontrollverkehr innerhalb des Netzwerks und des Internets werden stark reduziert.

1.3 Motivation

Mit Mikro-Mobilitäts-Protokolle lässt sich der Verlust von Packet und die Minimierung der Signalisierung minimieren. Diese Mechanismen sind erforderlich um Handover schnell und effizient durchführen zu können. Paging ist ein weiteres Feature von Mikro-Mobilitäts-Protokollen. Nun werden wir über zwei Techniken sprechen, welche für Mikro-Mobilität entworfen wurden.

1.3.1 Fast Handoff

Die Unterstützt von Fast Handoff ist eine wichtige Attribut von Mikro-Mobilität, welche die Packet Verlust reduzieren kann. Momentan gibt es viele Kandidaten von Protokollen, die Fast

Handoff Technik unterstützen. Viele Vorschläge davon diskutieren die so genanten Seamless Handoff, welche die Packet Verlust bei der Weiterleitung von Daten zwischen altere Zugangspunkt und neue Zugangspunkt zu null reduzieren können. Und viele Vorschläge unterstützen auch die gerechte komplexe Signaling, Puffer Technik und Synchrone Prozeduren. Die Entdeckung von neuen Zugangspunkt in Schichte drei spielt eine wichtige Rolle für die Handoff Leistung. Die Verzögerung von dem Handoff zwischen der Erkennung und der Registration von einer neuen Zugangspunkt hat ein starker Einfluss für die Mobilität und Datenverkehr. Um die Verzögerung zu minimieren, wird die Handoff, die auf die Signalstärke beruht, eine bessere Lösung geben. In diesem Fall wird die Kontrolle von Handoff in der Schicht 3 von Schichte 2 aufgerufen.

1.3.2 Paging

Normalerweise bleiben die Computer, die mit dem Internet verbinden, immer die Status "Online", auch wenn sie nicht kommunizieren. Aber mobile Teilnehmer nutzen meist kleine Endgeräte. Diese Geräte arbeiten auf Batterie mit begrenzter Laufzeit. Diese Endgeräte gehen in Zeiten längerer Inaktivität in den Idle-Modus, um Energie zu sparen. Sie können durch so genanntes Paging aufgeweckt werden.

Das mobile Gerät, das für längere Zeit keine Pakete sendet und empfängt schaltet in den Energiesparmodus. In dieser Zeit wird keine Signalisierung für Mobilität benötigt. Das mobile Gerät braucht nicht zu registrieren, falls es in dem gleichen Paging Gebiet bleibt. Es registriert sich nur, wenn es das Paging Gebiet wechselt.

2 Vergleich unterschiedlicher Protokolle

2.1 Protokolle für Mikro-Mobilität

Bei der IETF gibt es momentan unterschiedliche Vorschläge für ein Mikro-Mobilität-Protokoll. Beispiele sind Cellular IP, Hawaii und Hierarchische Mobile IP. Unter diesen basiert nur Hierarchische Mobile IP auf Mobile IP. Im Folgenden wird auf die Vorschläge eingegangen.

2.1.1 Cellular IP

Cellular IP wurde in 1998 und 1999 bei Ericsson und der Columbia University von Campbell, Valko entwickelt. Neben der Protokoll Mobil IP, muss der Mobile Node ein spezielles Cellular-IP-Protokoll implementieren.

Eine Cellular-IP-Domain besteht aus die so genannten Mobilität Agents (MA). Einer von diesen funktioniert wie einem Gateway zu dem Internet. Eine anderer funktioniert als einen Mobile-IP-FA für Makro-Mobilität. Jeder MA enthält einen Routing Cache, der den nächsten Router auf dem Weg zu MN und den nächsten Router auf dem Weg zum Gateway enthält. Dieser Routing-Cache wird beim Weiterleiten der Pakete von dem MN zu dem Gateway oder von dem Gateway zu dem MN benutzt. Der Weg wird beim Transportieren von zwei speziellen Nachrichten erstellt. Eine Beacon wird jedes mal nach einem bestimmten Intervall von dem Gateway in das Netzwerk geflutet. Dieser Mechanismus ermöglicht es jeden Router zu wissen, zu welchem Router er das Paket weiterschicken soll. Eine Route Update Nachricht wird von dem MN gesendet, wenn er erstmal mit dem Domain verbindet. Der Packet wird von Router zu Router bis hin zum Gateway weitergeleitet. so wissen alle Router auf dem Pfad, dass der nächste Router zu dem MN derjenige ist, von dem er daP Paket bekommt.

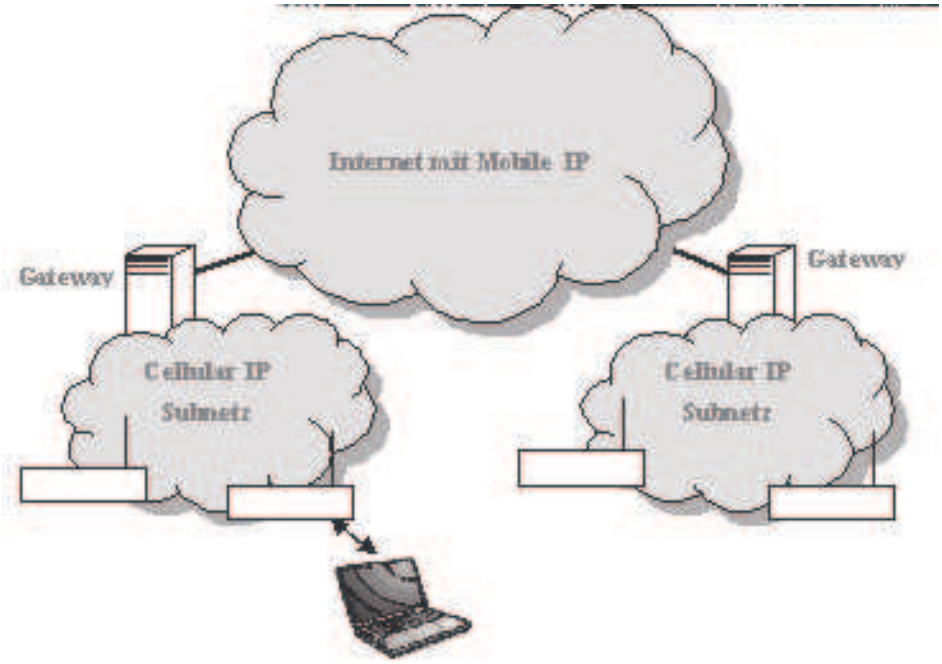

Abbildung 2: Cellular IP

Der grundlegende Handover-Mechanismus in Cellular IP ist der so genannt Hard Handoff. Der MN sendet einen Route Update Packet zu dem Gateway erst nach der Erstellung einer neuen Schicht-2-Verbindung. Um diesen Mechanismus zu verbessern wird ein neuer Handover, der so genannte Semisoft-Handoff wird definiert. Mit dem Semisoft-Handoff sendet der MN vor dem Schicht-2-Handoff ein besonders Paket, um eine Verbindung von dem neuen Zugangspunkt und dem alten Zugangspunkt aufzubauen. Dies reduziert den Verlust von Paketen, wenn der MN von einem Zugangspunkt zu einem anderen wechselt.

Cellular IP unterstützt auch passive Verbindungen mit einem Paging Mechanismus. Die Stationen werden in Paging Gebiet gruppieren. Router pro Paging-Gebiet enthält einen Paging Cache, um passive MNs in dem Gebiet zu verwalten.

2.1.2 Hawaii

In diesem Kapitel wird ein weiteres Mikro-Mobilitäts-Protokoll, Handoff-Aware Wireless Access Internet Infrastruktur(Hawaii) vorgestellt. Hawaii wurde 1999 von den Mitarbeitern des Labor Lucent Bell empfohlen. Ähnlich wie Cellular IP, benutzt Hawaii die Intra-Domain-Mobilität zur Verwaltung von Zugangspunkten. Daneben wird für die Inter-Domain Mobilität normales Mobile IP benutzt.

Anders als Cellular IP, setzt Hawaii auf Mobile IP auf. Jede Station innerhalb des Netzwerks muss nicht nur als Router arbeiten, sonder unterstützt auch spezielle Mobilitäts-Funktionen. Die grundlegende Arbeit von Hawaii ist gleich wie Cellular IP. Jede Station behält einen Routing Cache, um die Mobilität zu verwalten. Die Stationen schicken spezielle Pakete, um die Routing Caches zu aktualisieren. Das Netzwerk wird als einen Baum organisiert. In dem Baum ist das Gateway die Wurzel. Hawaii definiert zwei unterschiedliche Handover-Mechanismen für unterschiedliche Funk-Technologien. Die Mechanismen hängen davon ab, ob der MN gleichzeitig mit mehreren Base Stations kommunizieren kann.

Ähnlich wie Cellular IP, unterstützt Hawaii auch passive Verbindungen mit einem Paging Mechanismus. In Hawaii entspricht jedes Paging-Gebiet einer IP-Multicast-Gruppe. Die Sta-

tionen, die sich in diesem Paging-Gebiet aufhalten, sind Teilnehmer in der Multicast-Gruppe. Das Paging Verfahren ist ähnlich wie das von Cellular IP.

2.1.3 Hierarchische Mobile IP

Das Hierarchische Mobile IP, das von Ericsson und Nokia vorgeschlagen wurde, ist eine natürliche Erweiterung von Mobile IP. Es benutzt eine hierarchische Struktur von Foreign Agents, um die Mobile-IP-Registrierung lokal zu behandeln.

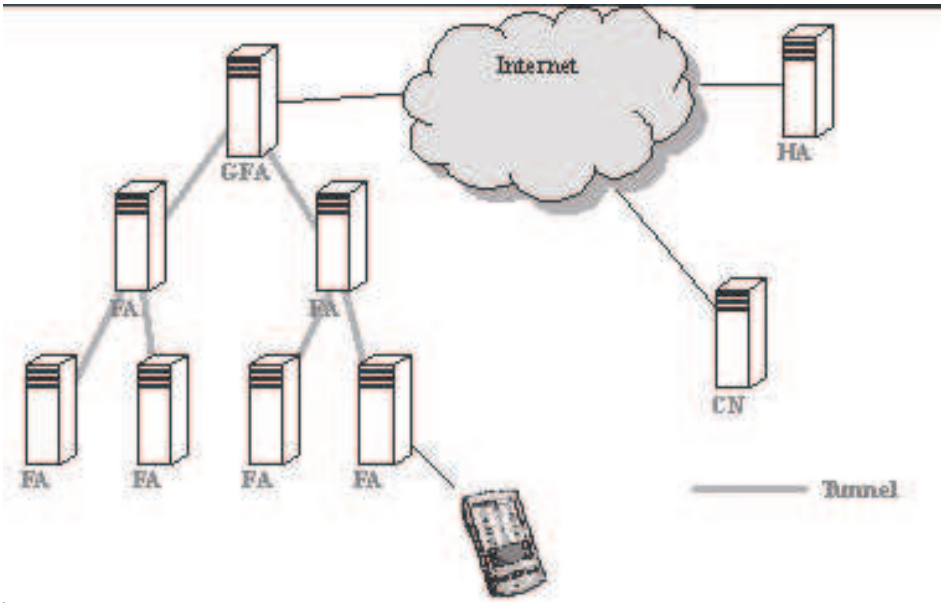

Abbildung 3: Hierarchische Mobile IP

In dem Protokoll sendet der Mobile Node Registrierungsnachrichten, um seinen Standort zu aktualisieren. Diese Nachrichten baut die Tunnel zwischen benachbarten Foreign Agents entlang dem Pfad von Mobile Node zum Gateway Foreign Agent auf. Die Pakete werden in diesem Tunnelnetzwerk transportiert, welches wir als getrenntes Overlay-Netz gesehen werden kann. Typischweise benutzt man eine one-level Hierarchie, bei der alle Foreign Agents an den Gateway Foreign Agent anschließen. In diesem Fall verbinden direkte Tunnel den GFA mit den FAs am Zugangspunkt. Paging-Erweiterungen ermöglichen es einem Mobile Node, in den Energiesparmodus zu wechseln, solange der Mobile Node in einem Paging Gebiet bleibt.

3 Hierarchische Mobile IP

Als auf Mobile IP erweitertes Mikro-Mobilitäts-Protokoll wird Hierarchisches Mobile IP in diesem Dokument ausführlich als Beispiel vorgestellt. Wegen des Unterschieds von IPv4 und IPv6 unterscheiden sich das Hierarchische Mobile IPv4 und das Hierarchische Mobile IPv6.

3.1 Mobile IPv4 Regional Registration

3.1.1 Architektur

- Gateway Foreign Agent (GFA): Ein Foreign Agent mit einer globalen IP Adresse.

- Regional Foreign Agent (RFA): Ein Foreign Agent, der für Regionale Registrierung entworfen ist.
- Network Access Identifier (NAI): Jeder Node besitzt ein NAI, um sich zu identifizieren
- Crossover Foreign Agent: Der niedrigste Foreign Agent in dem gemeinsamen Teil von beiden Pfaden, die von dem neuen FA und dem alten FA zu dem GFA wenn MN den Zugangspunkt wechselt.

Hierarchisches Mobile IP bietet die Möglichkeit, mehr als eine Schicht zu implementieren. Im Folgenden betrachten wir eine Hierarchie von FAs, die mehrere Schichten umfasst. Dabei bezeichnen wir den obersten FA wieder als GFA, und alle anderen FAs außer den untersten als Regional FAs

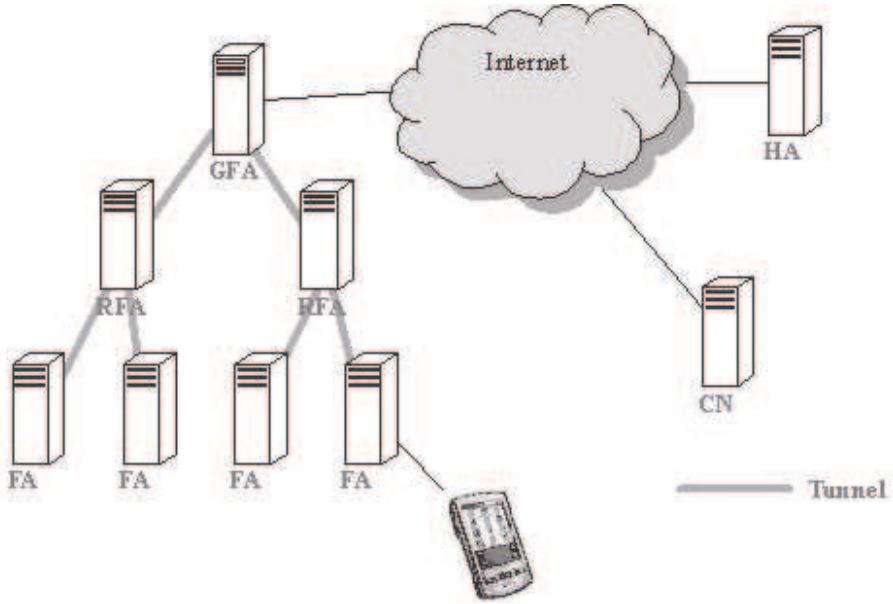

Abbildung 4: Mobile IPv4 Regional Registration

Wenn ein Mobile Node in der Domäne ankommt, führt er eine Registration mit seinem HA durch. Für diese Registration registriert der HA die care-of Adresse (COA) vom Mobile Node. Falls die Domäne die Regional Registration unterstützt, ist die eingetragene COA die Adresse vom GFA. Der GFA behält eine Besucher-Liste von allen in dieser Domäne registrierten Mobile Nodes. Weil die eingetragene COA die Adresse von GFA ist, verändert sich die COA nicht, falls sich der Mobile Node in der Domäne des GFAs bewegt. Deshalb ist es unnötig, die Lokalität des Mobile Nodes dem HA mitzuteilen. In diesem Fall wird nur eine regionale Registrierung ausgeführt.

3.1.2 Protokollablauf

- Aufbau des Pfades

  FA schickt periodisch Agent Advertisement, um die IP Adresse des GFAs und seine eigene IP Adresse dem MN mitzuteilen.

  Wenn ein MN in einer Domäne ankommt, entdeckt er den Zugangspunkt gemäß dem erhaltenen Agent Advertisement. MN vergleicht den Foreign Agent Network-Access-Identifier (FA-NAI) mit dem älteren. Wenn der FA-NAI nicht gleich ist wie früher,

sendet der MN eine Registrierungsanforderung an die Adresse des GFAs, die gerade in der Agent Advertisement bekommen wird. Die Registrierungsnachricht benutzt die Adresse des GFAs als CoA.

Wenn der nächste FA die Registrierungsanforderung erhält, liest er das CoA-Feld in der Nachricht aus. Dann kann der FA herausfinden, zu welchem GFA er die Nachricht weiterleiten soll. Danach fügt er eine Hierarchische FA-Erweiterung in die Nachricht hinzu, welche die Adresse des Fas enthält und leitet die bearbeitete Nachricht weiter an den nächsten RFA in Richtung zum GFA. Wenn der FA aus der Nachricht die Information des älteren FAs erfährt, sendet er auch eine Binding Update an die Adresse des älteren FAs.

Jeder RFA und GFA besitzt eine Besucher-Liste und verwaltet alle MNs, die sich im Bereich unter dem RFA bewegen. Wenn der RFA eine Registrierungsanforderung eines MNs empfängt, erneuert er die Information in der Liste für den MN. Mit dieser Information weiß der RFA, an welchen RFA oder FA er die an MN geschickten Pakete weiterleiten soll. Der RFA verarbeitet die neu erhaltene Registrierungsnachricht beim Wechsel der Hierarchischen FA-Erweiterung. Er entfernt die ältere, die vom vorhergehenden Agenten hinzugefügt wurde, und fügt eine neue FA-Erweiterung mit der Information von sich hinzu. Nach dieser Verarbeitung leitet er die Nachricht weiter zum nächsten RFA oder zum GFA. Diese Verfahren werden von jedem RFA unter dem GFA durchgeführt.

Wenn der GFA die Registrierungsanforderung erhält, merkt er sich die Routing Information für den MN in der Besucher-Liste. Dann leitet er die Nachricht an HA weiter.

Nach dem Empfang der Registrierungsanforderung, merkt sich der HA die GFA Adresse als die COA des MNs. Er erzeugt danach eine Registrierungsrückmeldung und sendet die Nachricht an den GFA zurück.

Wenn der GFA die Registrierungsrückmeldung erhält, sucht er die Information in der Liste der noch nicht erledigten Registrierungsanforderungen, damit er feststellen kann, zu welchem RFA oder FA die Rückmeldung weiterzuleiten ist. Vor dem Senden der Registrierungsrückmeldung sollte er eine FA-FA Schlüssel-Erweiterung hinzufügen. Die neue FA-FA Schlüssel Erweiterung enthält die Registrierungsschlüssel, die mit einem zwischen dem GFA und dem nächstem RFA vereinbartem Code verschlüsselt wird.

Jeder RFA arbeitet wie der GFA wenn er die Registrierungsrückmeldung empfängt und leitet die Nachricht weiter. Diese Rückmeldung kommt nun schließlich beim MN an. Mit dieser Nachricht weiß der MN, dass die Registrierung beim HA erfolgreich oder fehlerhaft war.

Sobald der HA die Adresse von dem GFA als COA Adresse des MNs registriert, kann der MN die Regionale Registrierung ausführen, wenn er sich in der gleichen Domäne bewegt. Der MN schickt eine Regionale Registrierungsnachricht an den nächsten FA, wenn er die Zellgrenze durchschreitet.

Wenn die Regionale Registrierungsnachricht zum nächsten FA ankommt, kontrolliert der FA die Besucher-Liste, ob der MN vorher schon bei ihm registriert hat. Falls noch nicht registriert, arbeitet der FA ähnlich wie die Registrierung beim HA und leitet die Nachricht an den nächsten RFA auf dem Weg zum GFA weiter. Wenn eine Registrierungsinformation vom MN in Besucher-Liste einer RFA oder GFA gefunden wird, stellt der entsprechende RFA oder GFA eine Rückmeldung für die Regionale Registrierungsnachricht her und leitet diese an den MN.

Im Mechanismus der Regionalen Registrierung spiegelt sich die Idee von Hierarchie wieder. Die Regionale Registrierungsnachricht muss nicht zum GFA ankommen, sondern zu dem so genannten Crossover-FA. Der Crossover-FA spielt dann die Rolle von FA in

normalem Mobile IP. Der Crossover-FA kann sehr niedrig sein, dabei wird nur sehr wenige Kontrollnachrichten geschickt. Die Latenz von Registrierung ist stark reduziert.

- Routing von Datenverkehr

  Datenpaket aus CN wird erst an den HA geschickt. Der HA tunnelt das Paket an den GFA weiter. Nach den Informationen der Besucher-Liste leitet der GFA herunter an den nächsten RFA. Jeder RFA auf dem Weg zu MN arbeitet gleich wie der GFA. Schließlich bekommt MN das Paket.

  Es ist nicht immer nötig, dass das Paket auf jedem Schritt in der Hierarchie entschlüsselt und verschlüsselt wird. Stattdessen verändert der GFA oder RFA nur die Quelladresse und die Zieladresse des IP Kopfes.

  Im Fall von Routing-Optimierung schickt HA einen Binding Update an den CN wenn MN mit CN kommuniziert. Der Binding Update enthält die Adresse vom GFA. Binding Update kann auch von MN geschickt werden. Der Binding Update soll auch die Adresse vom GFA haben. Danach kann CN direkt die Datenverkehr an den MN schicken.

- Handover

  Wie in dem letzten Abschnitt bereits erwähnt wird ein Binding Update vom neuen FA (nFA) an ältere FA (oFA) geschickt, wenn er eine neue Registrierungsanforderung des MNs erhält. Wenn dieser Binding Update zu dem oFA ankommt, weiß der oFA, dass der MN schon weggegangen ist und sendet einen Binding Update Acknowlage an den nFA zurück. Außerdem sendet er einen Binding Update zum nächsten RFA oder GFA auf dem Weg zu GFA. Dieses informiert jeden RFA, dass der MN zu einem anderen FA Gebiet gegangen sind. Jeder RFA in dem Pfad muss eine Rückmeldung zurückschicken, wenn er den Binding Update erhält. Danach aktualisiert der RFA die Binding Cache für den MN.

  Der Crossover-FA kann zwei Kontrollnachrichten erhalten. Eine Registrierungsanforderung des MNs vom neueren Pfad und der Binding Update für den MN vom älteren Pfad. Wenn der Crossover-FA den Binding Update erhält, leitet er die Nachricht nicht nach oben weiter, sondern schickt einen Binding Acknowledgement an den MN durch den Pfad zum oFA. Wenn der oFA diesen Binding-Acknowledgement erhält, tunnelt er den Binding Acknowledgement am nFA.

  Der Crossover-FA schickt die Datenpakete an den MN mit dem neuen Pfad erst wenn die Registrierungsanforderung, die aus den neuen Pfad kommt, erfolgreich beendet ist.

3.2 Hierarchical Mobile IPv6 mobility managment

3.2.1 Architektur und Terminologie

- Access Router (AR): Default-Router von MN
- Mobility Anchor Point (MAP): Ein Router in der Fremd Domäne. Als lokaler HA des MN angesehen. Eine oder mehrere MAPs können vorhanden sein.
- Regionale Care-of- Adresse(RCoA): eine Adresse von MAP Subnetz, wird automatisch konfiguriert wenn der MN die MAP Option erhält.
- On-link CoA (LCoA): normale Care of address wie in IPv4. Konfiguriert beim MN basiert auf dem Präfix im Router-Advertisement..
- Local Binding Update: sendet der MN an den MAP, um eine Verbindung zwischen RCoA und LCoA aufzubauen.

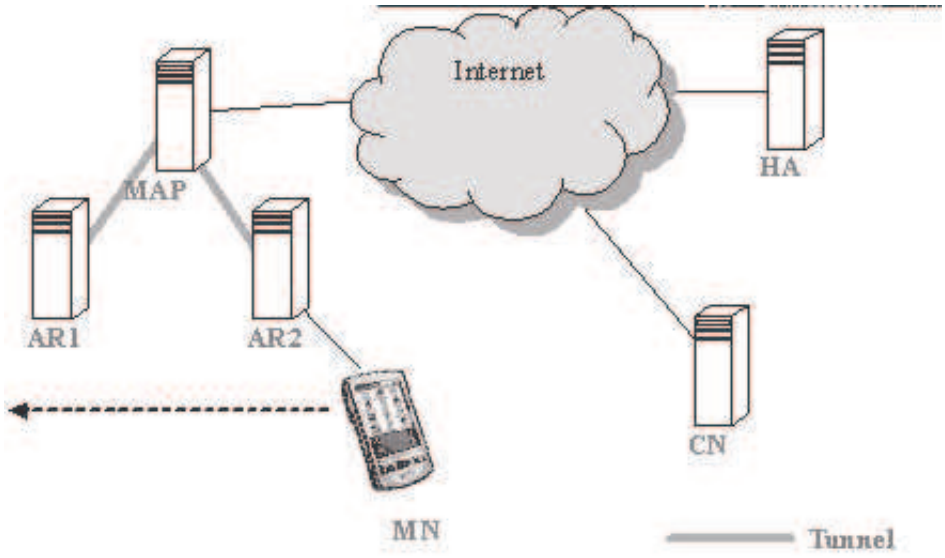

Abbildung 5: Hierarchische Mobile IPv6

Die Hierarchische Mobile IPv6 Schema stellt eine neue Funktion vor, den Mobility Anchor Point(MAP). Der MAP bietet die Möglichkeit, eine seamless Mobilität beim Wechsel des Access Routers haben zu können. In der Abbildung ist eine Beispiel-Architektur für von Hierarchisches Mobile IPv6 zu sehen. Eine mehrstufige Hierarchie von HMIPv6 ist standardmäßig nicht vorgesehen. Es könnte aber mehrere MAPs im Netzwerk geben.

Wenn ein MN in der Domäne ankommt, entdeckt er die globale Adresse des MAPs, die in dem AR gespeichert ist und durch ein Router Advertisements (RAs) zu dem MN geschickt wird. Eine neue Option wird benötigt für AR, um den MNs die Anwesenheit des MAPs mitzuteilen (MAP-Discovery).

Um die Bandbreite des Netzwerks optional auszunutzen, kann der MN mehr als einen MAP gleichzeitig benutzen.

3.2.2 Protokollablauf

In diesem Abschnitt diskutieren wir das Protokoll von HMIPv6. In HMIPv6 hat der MN zwei Adressen, eine Regionale Care of Adresse (RCoA) und eine on-Link CoA(LCoA).

- Aufbau des Pfades

  Der Access-Router schickt periodisch Router-Advertisement in die Luft. Der Router-Advertisement enthält die Information von einem oder mehreren MAPs. Wenn eine MN erst zu einem MAP ankommt, erhält er einen Router-Advertisement. Der MN wählt der MN einen MAP davon aus und konfiguriert die RCoA auf dem Link von MAP und die LCoA. Die beiden Adressen sind zustandslos aufgebaut. Die RCoA basiert auf dem Präfix von MAP Option in dem Router-Advertisement. Stattdessen basiert die LCoA auf dem Präfix von dem Access-Router, der die Router-Advertisement schickt.

  Wenn diese Verfahren fertig durchgeführt werden, schickt MN eines Lokales-Binding-Update (LBU) an den ausgewählten MAP. Das LBU enthält die gerade konfigurierte RCoA und benutzt die LCoA als Quelladresse. Der LBU dient dazu, die RCoA mit LCoA verbinden zu können.

Nach dem Erhalten des LBUs führt der MAP eine Duplicate Address Detection (DAD) durch für die RCoA des MNs und schickt einen Binding Acknowledgement (BA) zu dem MN zurück. Dieser BA bedeutet eine erfolgreiche Registrierung oder einen Fehler mit einem Fehler Code.

Der MAP kann eine OCOT Option am BA anhängen, dann muss der MN einen BA mit dieser OCOT Option an den MAP zurücksenden, wenn er diesen BA erhält. Mit dieser Option sichert der MAP, dass der MN wirklich auf diesem Link liegt.

Wenn LBU erfolgreich beendet ist, registriert der MN mit dem HA beim Senden einem Binding-Update (BU), um die neue RCoA dem HA mitzuteilen. Die RCoA bleibt, wenn sich der MN in der Domäne des MAPs bewegt. Falls der MN mit einem CN beim Wechsel der Domäne kommuniziert, kann der MN auch einen BU an den CN schicken. Der CN erneuert die Binding-Cache und schickt die Datenpakete an die RCoA weiter.

- Routing von Datenverkehr

  Es gibt drei unterschiedliche Möglichkeiten, um die Datenpakete im Hierarchischen Mobile IP zu routen.

  MN kann durch HA mit CN kommunizieren, dann tunnelt er alle Out-going-Pakete mit der Information des HAs an MAP. Wenn der MAP die Pakete erhält, leitet er die Pakete direkt an den CN. Wenn der CN ein Paket an den MN schicken möchte, schickt er das Paket zuerst an den HA, danach tunnelt der HA das Paket an RCoA weiter. Beim Empfangen des Pakets von HA leitet der MAP weiter an MN. Diese Verfahren ist ähnlich wie der Routing-Mechanismus in Mobile IPv4.

  MN kann auch direkt mit CN kommunizieren. In diesem Fall behält der CN einen Binding-Cache für MN. In dem Out-going Paket setzt der MN die Quelladresse mit der RCoA und tunnelt das Paket an den MAP. Nach dem Erhalten des Pakets leitet der MAP das Paket an den CN weiter. Der CN erneut den Binding-Cache für den MN. Wenn der CN Paket an den MN schicken möchte, schickt er alle Pakete direkt an die RCoA des MNs, ohne Umweg zum HA. Beim Empfangen des an den MN geschickten Pakets leitet es dieses weiter an LCoA des MNs.

  Im dritten Fall benutzt MN RCoA als Quelladresse und schickt die Pakete direkt an den CN ohne Tunnel zum MAP. Der CN schickt Datenpaket direkt an RCoA wie im zweite Verfahren. Alle an RCoA gesendeten Pakete werden von MAP abgefangen, danach leitet der MAP die Pakete an die LoA des MNs weiter.

- Handover

  Wenn eine MN von einer MAP-Domäne zu einer anderen MAP-Domäne kommt, kann er einen Inter-MAP Handover durchführen. Der MN schickt eines LBU an dem älteren MAP, um den MAP über die neue LCoA zu informieren. Nach dem Erhalten des LBUs erneuert er die Binding-Cache für den MN und leitet alle zum MN geschickten Datenpakete an die LCoA weiter.

  Wenn sich der MN in einer MAP-Domäne bewegt, aber den AR wechselt, kann er einen Handover durchführen bei der Unterstützung von Fast Mobile IPv6 Handover. In diesem Fall, schickt er eine Anforderung von Handover an den alten AR (oAR) nach Entdeckung vom neuen AR (nAR), um Handover zu initiieren. Der oAR schickt eine Rückmeldung zurück, wenn die Anforderung ankommt. Danach kann der MN einen Fast-Binding-Update an oAR schicken. Nach dem Erhalten des Fast-Binding-Update baut oAR einen bidirektionalen Tunnel zwischen oAR und nAR auf, dann schickt er eine Fast- Binding-Acknowledgement an MN zurück. Jetzt kann MN den AR wechseln. Später werden alle an den oAR gesendeten Pakete an den nAR weitergeleitet.

4 Zusammenfassung

In diesem Dokument haben wir Mikro-Mobilität-Protokolle diskutiert, mit denen Handover-Latenzen und Paketverlust bei Handover vermieden werden können. Wir haben in dieser Dokumentation drei wichtige Protokolle für Mikro-Mobilität vorgestellt. Als natürliche Erweiterung von Mobile IP ist das Hierarchische Mobile IP Protokolle anzusehen.

Basierend auf unserer Diskussion können wir folgendes feststellen: Für Mikro-Mobilitäts-Protokolle spielt der Handover-Mechanismus und die Paging-Technologie eine große Rolle. Um den Verlust von Paketen zu minimieren, erfordert es einen schnellen Handover. Mit einer Paging-Technologie lässt sich die Betriebszeit des Mobile Node in der Domäne verlängern.

In der Zukunft werden mehrere Mobile Nodes wie Laptop, PDA und andere Mobile Endgeräte benutzt. Die zeitkritische Arbeit mit der so genannten seamless Datenübertragung ist immer erforderlich. Deshalb wird die Forschung von Mikro-Mobilität eine große Rolle spielen.

Literatur

[CGKW+02] A.T. Campbell, J. Gomez, S. Kim, C. Wan, Z. Turanyi und A. G. Valko. Comparison of IP Micromobility Protocols. *IEEE Wireless Communications*, 2002.

[Gome] Andrew T. Campbell Javier Gomez-Castellanos. IP Micro Mobility Protocols.

[GuJP02] E. Gustafsson, A. Jonsson und Charles E. Perkins. Mobile IPv4 Regional Registration. IETF Mobile IP Working Group, October 2002.

[Pier] Olivier Bonaventure Pierre Reinbold. IP Micro Mobility Protocols.

[SCEMB03] H. Soliman, C. Castelluccia, K. El-Malki und L. Bellier. Hierarchical Mobile IPv6 mobility management (HMIPv6). IETF Mobile IP Working Group, June 2003.

[ZiWe03] M. Zitterbart und K. Weniger. Vorlesung Mobilkommunikation. http://www.tm.uka.de/lehre/aktuell/vorlesung/V_MK_Unterlagen/mk08-1.pdf, 2003.

Abbildungsverzeichnis

Mobile IP & Multicast

Patrick Freudenstein

Kurzfassung

IP Multicast war ursprünglich nur auf den Einsatz in Netzwerken mit stationären Geräten zugeschnitten. Durch die Mobilität der Geräte - sei es nun in der Rolle des Senders oder des Empfängers von Multicast-Paketen - muss auch der ursprüngliche Ansatz von IP Multicast für den Einsatz in Umgebungen mit mobilen Geräten neu überdacht werden. Die vorliegende Ausarbeitung stellt die Probleme von Multicast in mobilen Umgebungen sowie Lösungsansätze vor. Namentlich sind dies Remote Subscription, Bi-directional Tunneling, das MoM-Protokoll sowie MobiCast.

1 Einleitung

Mobile Geräte mit (drahtlosem) Anschluss an das Internet sind heute schon nicht mehr wegzudenken und werden immer mehr an Bedeutung gewinnen. Ein weiteres, zukunftsträchtiges Gebiet, dessen Potential in der Vergangenheit jedoch vernachlässigt wurde, ist Multicasting, das eine effiziente und Netzwerk-Ressourcen schonende 1:n Kommunikation ermöglicht. Insbesondere für Anwendungen wie Video-Konferenzen, computergestützte Zusammenarbeit (CSCW), Software-Verteilung oder verteilte Spiele ist Multicast als effiziente Kommunikationsbasis unersetzbar.

IP Multicast war ursprünglich nur auf den Einsatz in Netzwerken mit stationären Geräten zugeschnitten. Durch die Mobilität der Geräte- sei es nun in der Rolle des Senders oder des Empfängers von Multicast-Paketen - muss auch der ursprüngliche Ansatz von IP Multicast für den Einsatz in Umgebungen mit mobilen Geräten neu überdacht werden.

Diese Ausarbeitung führt in Abschnitt 2 zunächst in die Grundlagen von IP Multicast, wie es für Netzwerke mit stationären Geräten entwickelt wurde, ein. Anschließend wird in Abschnitt 3 die effiziente Eingliederung mobiler Geräte in das Internet anhand des Protokolls Mobile IP vorgestellt. In Abschnitt 4 wird auf die Probleme, die bei der Verwendung von IP Multicast in Umgebungen mit mobilen Geräten entstehen, eingegangen. Abschnitt 5 stellt die von der Internet Engineering Task Force (IETF) vorgeschlagenen Alternativen zur Erweiterung von Mobile IP zur Unterstützung von Multicast und deren Schwächen vor. Schließlich werden in den Abschnitten 6 und 7 zwei weitere Ansätze - namentlich das MoM Protokoll sowie das MobiCast Protokoll - ausführlich dargestellt, die auf je einem der beiden IETF Vorschläge basieren und diesen unter Beseitigung der bekannten Schwächen erweitern. Eine Bewertung jedes Ansatzes am Ende der jeweiligen Abschnitte rundet die Arbeit ab.

2 Einführung in IP Multicast

IP Multicast bietet eine einfache und effiziente Methode zur 1:n-Kommunikation im Internet unter Annahme stationärer Geräte. Zu den Haupteinsatzgebieten zählen Multimedia-Anwendungen wie zum Beispiel Videokonferenzen, Tele-Kooperation, Video On Demand , Tele-Teaching oder Spiele.

Diese Anwendungen benötigen oft eine hohe Bandbreite, so dass die triviale Abbildung der 1:n Kommunikation auf n Unicast-Verbindungen, d.h. die Quelle vervielfältigt das zu sendende Paket und schickt jedem Empfänger eine separate Kopie, zu einer Überlastung des Netzwerkes führen kann. Abbildung 1 veranschaulicht diese Problematik sowie den Lösungsansatz von IP Multicast:

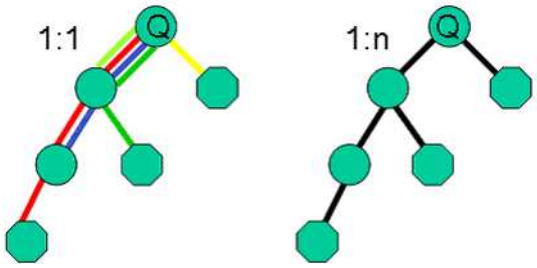

Abbildung 1: Versendung eines Pakets über fünf Unicast-Verbindungen (links) vs. Versendung via Multicast (rechts)

Die Abbildung stellt den Versand eines Pakets an fünf Kommunikationspartner via fünf separater Unicast-Verbindungen (links in der Abbildung) und via IP Multicast (rechts in der Abbildung) gegenüber. Es ist offensichtlich, dass die Netzwerkbelastung - vor allem auf der ersten Teilstrecke von der Quelle zu ihrem linken Nachbar - viermal so hoch ist wie bei IP Multicast, wo jedes Paket nur einmal pro Teilstrecke versendet wird.

Bei der Verwendung von IP Multicast und z.B. Multicast Open Shortest Path First [Moy94] als Routing Algorithmus wird bei der Ankunft eines Multicast Pakets in jedem Router - basierend auf den gesammelten Informationen hinsichtlich der Netzwerk-Topologie - ein Baum kürzester Wege berechnet, der als Wurzel die Quelle und als Blätter die Empfänger enthält. Dieser Baum wird als „Multicast Delivery Tree“ bezeichnet. Alle Router verwenden den gleichen Algorithmus zur Erstellung des Baums und verfügen über die gleichen Informationen bzgl. der Netzwerktopologie und des Netzwerkstatus und generieren somit identische Bäume. Multicast-Pakete werden dann von allen Routern konsistent anhand des Baums auf jedem Pfad nur einfach (d.h. keine Duplikate auf derselben Verbindung) gesendet.

Um Daten an eine bestimmte, dynamische Gruppe von Geräten zu adressieren, sieht IP die sog. „Class D-Adressen“ vor. Um ein Paket an alle zur Multicast-Gruppe gehörenden Geräte zu senden, genügt es, das Paket an die Multicast-Adresse dieser Gruppe zu adressieren. Router erkennen am Aufbau der Empfängeradresse, dass es sich um ein Multicast-Paket handelt und routen es dementsprechend. Geräte, die einer bestimmten Multicast-Gruppe als Empfänger beitreten möchten, teilen dies den Multicast-Routern über das Internet Group Management Protocol [Cain02] mit. Die Gruppenmitgliedschaft ist dynamisch, ein Gerät kann jederzeit einer Truppe beitreten bzw. sie wieder verlassen.

Zur eigentlichen Abwicklung des Multicast Routings, d.h. zur Konstruktion des Multicast Delivery Trees und die Wegewahl existieren verschiedene Algorithmen:

- Multicast Open Shortest Path First [Moy94]
- Distance-Vector Multicast Routing Protocol [D. W88]
- Core Based Trees [Ball97]
- Protocol Independent Multicast [eal94]

Auf die Funktionsweise dieser Algorithmen wird hier nicht weiter eingegangen, da sie für das behandelte Thema nicht von Bedeutung ist.

3 Einführung in Mobile IP

Ziel von Mobile IP ist es, mobile Geräte unabhängig vom Netzwerk, in dem sie sich gerade befinden, d.h. unabhängig vom Netzanschlusspunkt, unter einer konstanten IP-Adresse erreichbar zu machen. Bewegt sich ein mobiles Gerät („Mobile Host (MH)“) aus seinem Heimat-Netzwerk heraus, wird ihm normalerweise per DHCP [Drom93] oder durch manuelle Konfiguration eine neue IP-Adresse zugewiesen. Mit dem Wechsel der IP-Adresse würde - ohne Mobile IP - eine Trennung aller bestehenden logischen Kommunikationskanäle einhergehen. Wie Mobile IP diese Problematik löst und eine unterbrechungsfreie, lokationstransparente und damit auch mobilitätsunabhängige Erreichbarkeit mobiler Geräte ermöglicht, ist Inhalt dieses Kapitels.

Bei Mobile IP wird jedem MH eine IP-Adresse in seinem Heimat-Netzwerk zugeordnet, die sog. „Home Address“. Unter dieser Adresse ist der MH dank Mobile IP unabhängig von seinem aktuellen Aufenthaltsort für Kommunikationspartner immer erreichbar.

Im Heimat-Netzwerk des MH ist auch der sog. „Home Agent“ (HA) angesiedelt. Typischerweise übernimmt der Default-Router des Heimat-Netzwerkes diese Rolle. Zu seinen Hauptaufgaben zählen die Verwaltung der Aufenthaltsorte der MHs aus seinem Netzwerk sowie die Weiterleitung von IP-Paketen für MHs, die sich gerade in einem fremden Netzwerk aufhalten.

Wird ein MH mit einem fremden Netzwerk verbunden - sei es nun drahtlos oder drahtgebunden, so registriert er sich beim dort ansässigen „Foreign Agent“ (FA), der gleichzeitig auch der Default-Router des Netzwerkes ist. Der FA kann dem MH bei der Registrierung eine sog. „Care-Of Address“ (COA) zu, die dem aktuellen Aufenthaltsort des MH entspricht. Man spricht dann von einer „Foreign Agent COA“, da die COA die Adresse des FA ist. Mobile IP sieht alternativ auch die sog. „Co-located COA“ vor. In diesem Fall befindet sich die IP-Adresse direkt beim MH und wird beispielsweise via DHCP zugewiesen. Ein FA wird hier zum Routing nicht benötigt; der MH teilt seine COA direkt dem HA mit und dieser leitet Pakete für den MH per IP-in-IP-Encapsulation [Perk96] direkt an diesen weiter.

Das Paar, bestehend aus Home Address und Care-Of Address eines MH, wird als „Binding“ bezeichnet und ist mit einem vom MH ausgestellten Zeitstempel versehen, um dessen Gültigkeit zu beschränken.

Die Alternative mit Foreign-Agent COA soll im Folgenden noch etwas näher vorgestellt werden: Jeder FA verwaltet eine sog. „Visitor List“. Dabei handelt es sich um eine Liste aller derzeit registrierten MHs, deren Heimat-Netzwerk nicht das Netzwerk des FAs ist, die also „zu Besuch“ sind. Jeder Eintrag besteht aus der Home Adress, der Adresse des zugehörigen HAs sowie der MAC-Adresse des MHs und ist mit dem oben erwähnten Zeitstempel und einer im Registrierungsprozess ausgehandelten Gültigkeitsdauer versehen. Der MH muss sich vor Ablauf der Gültigkeitsdauer beim FA erneut registrieren, damit ein unterbrechungsfreier Service vom FA gewährleistet werden kann. Im Gegenzug muss der MH sich beim FA nicht abmelden, wenn er das Netzwerk wieder verlässt, da seine Registrierung dann automatisch mit dem Ablauf der Gültigkeitsdauer erlischt. Nach der Registrierung beim FA meldet sich der MH beim HA, um diesem seinen neuen Aufenthaltsort mitzuteilen. Der HA verwaltet eine sog. „Mobility Binding Table“, die die Home Address, die COA sowie die Gültigkeitsdauer des Bindings für jeden MH enthält.

Abbildung 2 veranschaulicht dieses Szenario.

Befindet sich ein MH nun nicht in seinem Heimat-Netzwerk, so fängt der HA in seiner Funktion als Default-Router des Heimat-Netzwerkes an den MH adressierte Pakete ab und leitet diese durch IP-in-IP encapsulating („Tunneling“) an die COA des MH weiter. Je nachdem, ob es sich um eine Co-located COA oder eine Foreign Agent COA handelt, entkapselt der MH oder

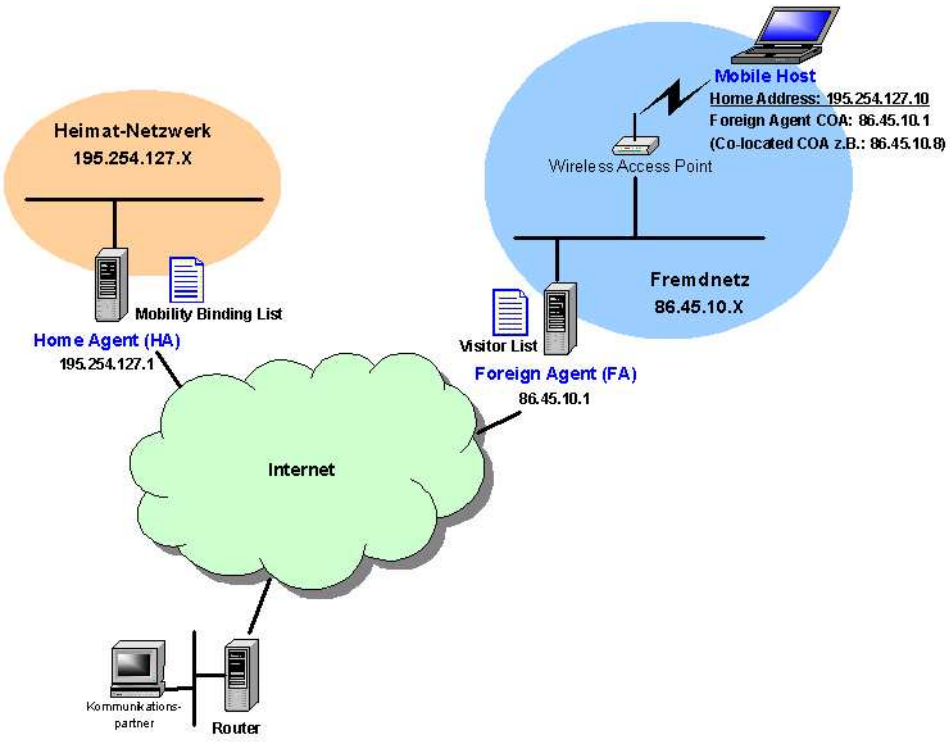

Abbildung 2: Mobile IP Szenario

der FA die Pakete. Im Falle einer Foreign Agent COA leitet der FA die entkapselten Pakete an den MH lokal weiter.

Vom MH ausgehende Kommunikation kann entweder direkt oder via „Reverse Tunneling“ [Mont98] erfolgen. Ersteres ist ’ ’jedoch meist problematisch, da der MH als Absender-Adresse seine Home Address verwendet und diese im Fremdnetz vom Router bzw. der Firewall als topologisch inkorrekt erkannt und somit nicht akzeptiert wird. Beim Reverse Tunneling sendet der MH das Paket an den HA - im Falle einer Foreign Agent COA via Kapselung durch den FA, im Falle einer Co-located COA unter Verwendung dieser COA als Absender-Adresse direkt an den HA. Der HA leitet dann das Paket normal an den Empfänger weiter.

Die obige Beschreibung bezieht sich auf Mobile IPv4. Mittlerweile steht auch Mobile IPv6 [D. J03] kurz vor der Standardisierung durch die IETF. Die in den folgenden Kapiteln beschriebenen Protokolle zur Unterstützung von Multicast in Verbindung mit mobilen Geräten beziehen sich jedoch auf Mobile IPv4, weshalb auf die Unterschiede zwischen den beiden Versionen hier nicht weiter eingegangen wird.

Mobile IP wurde von der IETF zur Unterstützung von IP Unicast Routing für mobile Geräte im Internet entwickelt. Die IETF schlägt auch zwei Methoden zur Behandlung von Multicast Paketen vor, die in Kapitel 5 vorgestellt werden.

4 Besonderheiten von Multicast in mobilen Umgebungen

Bei der Verwendung vom herkömmlichen IP Multicast in Umgebungen mit mobilen Geräten können vielfältige Probleme auftreten, die aus der Mobilität der Geräte resultieren. Einige der Wichtigsten werden in diesem Kapitel vorgestellt.

Ein Problem beruht auf der Tatsache, dass die im vorigen Kapitel genannten Multicast Routing Algorithmen (DVRM, MOSPF, ...) beim Berechnen des Multicast Delivery Trees von

statischen Geräten ausgehen. Die Neuberechnung des Baums bei jeder Bewegung eines Geräts in ein anderes Netzwerk ist keine effiziente Option. Andererseits ist die Beibehaltung des ursprünglichen Baums trotz der Bewegung eines Gerätes in ein anderes Netzwerk keine Alternative, da die Auslieferung der Multicast Pakete sehr ineffizient werden und es auch zu Ausfällen kommen kann.

Ein weiteres, aus der Mobilität der Geräte resultierendes Problem kann sich ergeben, wenn das Fremdnetzwerk, in dem sich ein Gerät gerade aufhält, keinen Multicast-Router besitzt.

Ferner können in der Zeit, wo das mobile Gerät das Netz wechselt, Paketverluste auftreten, da Multicast-Router nicht die Bewegung der Geräte verfolgen.

Schließlich besteht die Gefahr, dass Multicast-Router unter den Gruppenbeitritts-Anfragen einer großen Anzahl sich schnell bewegender Geräte zusammenberechen.

5 IETF Mobile IP Multicast

Die IETF schlägt in ihrer Mobile IP Spezifikation zwei Methoden zur Behandlung von Multicasting mit mobilen Geräten vor: Remote Subscription und Bi-directional Tunneling.

5.1 Remote Subscription

Bei der Verwendung von Remote Subscription registriert sich der MH bei jedem Wechsel in ein neues Netz beim lokalen Multicast-Router für die gewünschten Multicast-Gruppen.

Dies ist die einfachste Lösungsmöglichkeit, da kein Tunneling vom bzw. zum mobilen Gerät erforderlich ist und ausschließlich bereits existierende Protokolle verwendet werden. Da der Multicast Delivery Tree immer den tatsächlichen Aufenthaltsorten der Geräte angepasst wird, ermöglicht dieses Verfahren eine sehr gute Dienstgüte und liefert ein effizientes Routing.

Die Anwendbarkeit dieser Methode ist abhängig von der Existenz von Multicast-Routern in allen Fremdnetzwerken und der Bewegungsintensität des MH:

- Zur Registrierung bei einem lokalen Multicast-Router bei jedem Eintritt in ein neues Fremdnetz ist es erforderlich, dass alle Fremdnetze auch über einen solchen verfügen.
- Ferner sollten die mobilen Geräte sich nicht zu schnell und zu häufig zwischen Netzen bewegen, damit die Multicast-Router von den häufigen Neuregistrierungen nicht überlastet werden. Außerdem könnten bei einer hohen Mobilität des MHs Pakete in der Zeit, die zum Neuregistrieren beim Multicast-Router in einem neuen Netz benötigt wird, verloren gehen.

Für mobile Geräte, die sehr viel in Bewegung sind oder die garantierte, (fast) unterbrechungsfreie 2-Wege Kommunikation mit der Multicast-Gruppe benötigen und keine Co-located COA erhalten können, ist Remote Subscription keine Alternative.

5.2 Bi-directional Tunneling

Bei diesem Verfahren vertritt der HA den MH als Sender und Empfänger von Multicast-Paketen und leitet diese (über den FA) vom / an das mobile Gerät weiter. Der Beitritt des MH zu Multicast Gruppen wird über den HA als Stellvertreter abgewickelt. Zwischen dem

mobilen Gerät und dem HA wird ein bidirektionaler Tunnel aufgebaut, wodurch das Senden und Empfangen von Multicast-Paketen wie unter statischen Bedingungen ermöglicht wird.

Bei diesem Verfahren treten vor allem die folgenden vier Probleme auf:

- **Ineffiziente Routing Pfade** durch Dreiecks-Routing
- **Skalierbarkeit:** Befinden sich mehrere mobile Geräte aus dem gleichen Heimatnetzwerk in einem Fremdnetzwerk, sendet der HA für jedes der mobilen Geräte eine Kopie des Multicast-Pakets über den jeweiligen Tunnel (über den FA) an das mobile Gerät, was zu einer zusätzlichen (unnötigen) Netzwerkbelastung führt und somit die Skalierbarkeit einschränkt.
- **Das Tunnelkonvergenz-Problem:** Dadurch, dass sich MHs der gleichen Multicast-Gruppe aus verschiedenen Heimatnetzen in einem Fremdnetz aufhalten können und somit mehrere Mobile IP Tunnel von verschiedenen HAs bei einem FA enden, wird erneut eine unnötig hohe Netzwerklast generiert und somit die Skalierbarkeit des Bi-directional Tunnelings stark beeinträchtigt.
- **Header-Overhead** durch ständige Kapselung der Multicast Pakete aufgrund des bidirektionalen Tunnels. Beim Weiterleiten von Multicast-Paketen vom HA zum mobilen Gerät ist sogar eine doppelte Kapselung erforderlich, da das an die Multicast-Adresse adressierte Paket zuerst in ein an die Home Adress des MH adressiertes Paket gekapselt werden muss, bevor es dann in ein weiteres, an die COA adressiertes Paket gekapselt wird.

In Anbetracht dieser Probleme ist Bi-directional ' 'Tunneling keine gute Alternative, um Multicast-Services für mobile Geräte effizient zu ermöglichen. Es degeneriert vielmehr Multicasting zu MHs zu einer Vielzahl von Unicasts.

Ein weiteres Problem, das bei beiden Verfahren auftritt, ist das „Scoping Problem“. Über das Time To Live - Flag von IP können Multicast-Pakete neben der Eingrenzung der Empfänger auf eine bestimmte Multicast-Gruppe auch noch hinsichtlich der „Nähe“ zum Sender eingegrenzt werden. Man spricht dann von „Scope Groups“. Das Hauptproblem besteht hier in der Verwischung des Begriffs „lokal“ bzw. „nah“ aufgrund der Mobilität der Geräte. Diese Problematik soll durch das folgende Beispiel verdeutlicht werden.

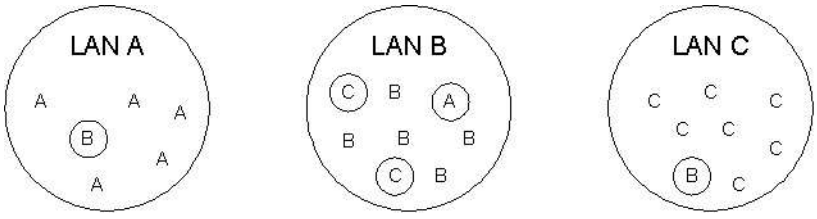

Abbildung 3: LAN Broadcast Szenario mit drei LANs und mobilen Geräten - Scoping Problem

In dem in Abbildung 3 dargestellten Szenario stellt sich im Falle eines Multicasts im LAN C an die Adresse 224.0.0.1, die von IP v4 Multicast gleichbedeutend mit „alle Geräte in DIESEM LAN“ festgelegt wurde, die Frage, wie dies nun ausgelegt werden soll:

1. Auslieferung an alle Geräte, die sich momentan im LAN C befinden, unabhängig von deren Heimatnetz oder
2. Auslieferung an alle Geräte, deren Heimatnetz LAN C ist, und die sich auch momentan dort befinden oder

3. Auslieferung an alle Geräte, deren Heimatnetz LAN C ist, unabhängig von ihrem derzeitigen Aufenthaltsort?

Die IETF hat im April 1995 entschieden, dass LAN Broadcasts an entfernte Geräte weitergeleitet werden müssen. Diese Entscheidung wurde für Multicasts an „Scope Groups“ aufgrund der Ähnlichkeit zu Broadcasts analog übernommen. Das Weiterleiten der Pakete als Unicast-Pakete per Kapselung an mobile Geräte in Fremdnetzen wäre beispielsweise eine einfache, jedoch ineffiziente Lösung. Eine bessere Lösung wird im Rahmen des MoM-Protokolls im nächsten Kapitel vorgestellt.

6 Mobile Multicast Protocol: MoM

Das Mobile Multicast Protocol [TGHBu97][VCMa98] wurde erstmal 1997 von Tim G. Harrison, Carey L. Wiliamson, Richard B. Bunt und Wayne Mackrell von der Universität in Saskatchewan in Kanada vorgestellt. Ihr Vorschlag basiert auf dem Ansatz des Bi-directional Tunnelings, behebt jedoch die dabei auftretenden Probleme. Das MoM Protokoll bietet Skalierbarkeit, Robustheit, d.h. die Unterbrechung des Multicast Services aufgrund Netzwechsels ist minimal, und Einfachheit, d.h. es kooperiert mit bestehenden Internet-Protokollen und Routing-Algorithmen und erfordert nur sehr wenige Änderungen an der bereits existierenden Infrastruktur. Für das im Rahmen des Protokolls erforderliche Unicast-Routing zu mobilen Geräten werden die Mechanismen von Mobile IP verwendet.

Im Folgenden wird die dem MoM Protokoll zugrunde liegende Idee, die dem bereits vorgestellten „Bi-directional Tunneling“ sehr ähnelt, kurz beschrieben. Dann wird auf die daraus resultierenden Probleme eingegangen und erläutert, wie das MoM Protokoll diese löst. Schließlich wird der daraus abgeleitete Protokollablauf für das Senden und Empfangen von Multicast-Paketen durch mobile Geräte nochmals im Überblick vorgestellt.

6.1 Die Grundidee des MoM Protokolls

In der Rolle des Senders von Multicast-Paketen verschickt der MH, wenn er sich im Heimatnetz befindet, die Pakete via Link-Level Multicast. Der Multicast-Home Agent (MHA) verbreitet diese dann wie gewohnt entlang des Baumes in Abwärtsrichtung weiter. Befindet sich der MH hingegen in einem Fremdnetz, so sendet er die Pakete über den bidirektionalen Tunnel von Mobile IP zum MHA. Dieser leitet das Paket dann entlang des Baumes in Abwärtsrichtung an alle angrenzenden Interfaces weiter, inklusive dem Schicht 2-Interface des eigenen Netzes. In beiden Fällen ist die Absenderadresse in den Multicast Paketen die Home Adress des MH. Die Mobilität des Senders ist für die Empfänger somit transparent.

Multicast-Pakete, die an den MH gerichtet sind, werden vom MHA empfangen und durch den bidirektionalen Tunnel über den FA an den MH weitergeleitet. Dadurch muss weder der FA sich als Vertreter für die MHs in seinem Netzwerk in Multicast-Gruppen registrieren, noch müssen sich MHs nach jedem Netzwechsel in den Multicast-Gruppen neu anmelden. Die Informationen zum Weiterleiten von Multicast-Paketen an den MH können aus dem Registrierungsprozess von Mobile IP übernommen werden.

Die Zeit, die zum Weiterleiten von Multicast-Paketen benötigt wird (im Millisekundenbereich), ist, verglichen mit der benötigten Zeit zur Neuberechnung eines Multicast Delivery Trees nach jedem Netzwechsel (unter Umständen sogar im Sekundenbereich), wie es die Remote Subscription Methode vorschlägt, relativ kurz, so dass diese Lösung eine durchaus effiziente Alternative darstellt.

Soweit ähnelt das MoM Protokoll sehr stark dem Bi-directional Tunneling Vorschlag der IETF. Im Folgenden wird nun näher erläutert, wie das MoM Protokoll die aus Kapitel 5 bekannten Probleme des IETF Vorschlages löst.

6.2 Die damit verbundenen Probleme und deren Lösung

Das im vorigen Abschnitt beschriebene Problem hinsichtlich der **Skalierbarkeit**, ausgelöst durch die Tatsache, dass ein HA für jeden MH in einem Fremdnetz eine *separate* Kopie des gleichen Multicast Pakets *als Unicast* Paket an den FA versendet, wird beim MoM Protokoll dadurch gelöst, dass ein HA *nur ein Multicast*-Paket pro Fremdnetz an den dortigen FA verschickt. Der FA verteilt dieses dann an alle Mitglieder der Multicast-Gruppe per Schicht 2-Multicast.

Das Tunnelkonvergenz-Problem wird mit obigem Lösungsansatz sogar noch verschärft, da nun aufgrund des Schicht 2-Multicasts im Fremdnetz bei jedem MH Multicast-Pakete mehrfach ankommen würden. Die Anzahl der Duplikate und damit die Netzlast im Fremdnetz erhöhen sich mit jedem neuen mobilen Gruppenmitglied, das aus einem bisher nicht vertretenen Heimatnetzwerk stammt.

Zur Lösung dieses Problems sieht das MoM Protokoll die Berufung eines HAs zum Designated Multicast Service Provider (DMSP) für eine bestimmte Multicast-Gruppe durch den FA vor. Nur der HA, der DMSP für eine bestimmte Multicast-Gruppe ist, leitet bei ihm ankommende Multicast-Pakete für diese Gruppe in einfacher Ausführung - unabhängig von der Anzahl der MHs im Fremdnetz - an den FA weiter. Dadurch erhält jeder MH jedes Paket in den meisten Fällen nur einmal. Eine Ausnahme davon wird weiter unten beschrieben.

Durch die Auswahl nur eines DMSPs entsteht ein Übergabe-Problem, wenn der letzte MH aus dem Heimatnetz des DMSPs das Fremdnetz verlässt und somit der DMSP-HA seine Weiterleitung an den FA des Netzes einstellt, wodurch auch MHs aus anderen Heimatnetzen vom Multicast-Service abgeschnitten werden. Der Grund hierfür ist, dass der MH nur seinen HA über den Netzwechsel explizit informiert, nicht jedoch den FA. Dieser erfährt davon erst viel später über den Timeout des zugehörigen Bindings. Der FA ist daher nicht in der Lage, rechtzeitig einen neuen HA zum DMSP zu ernennen, so dass es zeitweise zu einem Ausfall des Multicast-Services kommt. Darüber hinaus stellt der DMSP einen Single Point of Failure dar, wodurch die Robustheit dieses Systems gefährdet ist. Zur Behebung dieser beiden Schwächen schlägt das MoM Protokoll den Einsatz von maximal 3 redundanten DMSPs vor. Der FA filtert dann Duplikate heraus und leitet jedes Multicast Paket nur einfach in sein Netz weiter.

Trotz all dieser Vorkehrungen kann es immer noch zur ' '**Mehrfach-Versendung von Multicast-Paketen innerhalb des Fremdnetzes** kommen, wenn neben den MHs auch noch stationäre Geräte des Fremdnetzes zur gleichen Multicast Gruppe gehören. Wenn der FA davon keine Kenntnis hat, leitet er die vom DMSP erhaltenen Multicast-Pakete per lokalem Schicht 2-Multicast weiter. Zusätzlich werden dieselben Pakete auch vom lokalen Multicast-Router des Fremdnetzes verschickt, der das / die stationären Gerät(e) im Fremdnetz bedient. Dies kann beispielsweise durch die Zusammenlegung von FA und Multicast-Router umgangen werden. Eine weitere Möglichkeit wäre, dass der FA die Multicast-Pakete, die in seinem Netz versendet werden, beobachtet und bei der Entdeckung von Duplikaten für dieselbe Multicast-Gruppe die Weiterleitung von Multicast-Paketen vom DMSP einstellt.

Das sog. „**Scoping Problem**“ wurde bereits in Kapitel 5 vorgestellt. Das Problem liegt in der Frage, wie Pakete von lokal begrenzten Multicasts (bzw. Broadcasts) am effizientesten an MHs, die zu dieser lokalen Zielgruppe gehören, weitergeleitet werden, ohne einen lokalen Multicast im Fremdnetzwerk, in dem sie sich gerade aufhalten, auszulösen.

Das MoM Protokoll schlägt hierfür vor, dass der FA-Multicast-Router MHs, die zum selben HA gehören, in einer neuen Multicast-Gruppe auf Schicht 2 zusammenfasst, wobei die Adresse dieser Gruppe aus der IP-Adresse des HA abgeleitet wird. Empfängt der HA nun Pakete, die an eine lokale Multicast-Gruppe bzw. Broadcast-Adresse adressiert sind, so leitet er diese zum FA weiter. Der FA ordnet der IP-Adresse des HAs die lokale Schicht 2-Multicast-Adresse zu und leitet die entkapselten Pakete an diese Gruppe auf Schicht 2 weiter. So können auf effiziente Art und Weise auch mobile Geräte in „lokale“ Multicasts / Broadcasts einbezogen werden.

6.3 Das MoM Protokoll im Überblick

6.3.1 Mobile Geräte als Sender von Multicast-Paketen

Alle Multicast-Pakete, die vom MH ausgehen, werden direkt zum MHA getunnelt, der diese dann in seiner Rolle als Multicast-Router weiterleitet.

6.3.2 Mobile Geräte als Empfänger von Multicast-Paketen

1. **Die Registrierung in einer Multicast-Gruppe**
   Befindet sich der MH in einem Fremdnetz, so registriert er sich in einer Multicast-Gruppe, indem er einen IGMP Membership Report direkt an den FA schickt. Befindet sich der MH hingegen in seinem Heimatnetzwerk, sendet er - wie für statische Geräte üblich - einen IGMP Membership Report an die Adresse der Multicast-Gruppe, der er beitreten möchte. Möchte der MH einer „Scope Group“ beitreten, so setzt er das Broadcast Flag in der Registrierungsanfrage. Der FA erstellt dann eine Schicht 2-Multicast-Gruppe für alle MHs dieses HAs, deren Adresse er aus der IP-Adresse des HA ableitet.

2. **Verarbeitung der Registrierung durch den FA und den HA**
   Der FA leitet den IGMP Membership Report an den MHA weiter. Wenn der MH das erste mobile Mitglied der gewünschten Multicast-Gruppe im Netz des FA ist, so teilt der FA dem MHA mit, dass er der DMSP für diese Gruppe ist. Ansonsten ist die Auswahl des DMSPs vom dafür verwendeten Algorithmus abhängig. Im Hinblick auf Übergabe-Häufigkeit und optimale Routen haben sich besonders Algorithmen, die die Nähe des DMSPs zum FA berücksichtigen, als effizient erwiesen. Ein Beispiel wäre der in [TGHBu97] vorgeschlagene „Closest-to-FA“-Algorithmus, nach dem derjenige HA zum DMSP bestimmt wird, der dem FA am nächsten gelegen ist.

   Der MHA fügt sich selbst zum Multicast Delivery Tree für diese Gruppe hinzu und empfängt somit in Vertretung für den MH alle Multicast-Pakete für diese Gruppe. Der MHA leitet Multicast-Pakete jedoch nur dann an entfernte MHs weiter, wenn er der DMSP für die jeweilige Multicast-Gruppe in deren aktuellem Netz ist. Pakete, die an „Scope Groups“ adressiert sind, werden vom MHA an den FA des MHs weitergeleitet, der diese dann der zugehörigen Multicast-Gruppe auf Schicht 2 zuordnet und an diese Gruppe per Schicht 2-Multicast weiterleitet.

3. **Die Übergabe-Prozedur zwischen DMSPs**
   Um Paketverluste durch den Netzwechsel des letzten MHs eines HAs, der DMSP ist, vorzubeugen, ist ein Benachrichtigungssystem vorgesehen. In allen Fällen, in denen der HA seine Rolle als DMSP für eine Multicast-Gruppe im Netz des FAs nicht mehr weiter fortführen kann - physische Umstände wie z.B. Stromausfall ausgenommen - informiert er den FA darüber. Daraufhin bestimmt der FA einen neuen DMSP für diese Gruppe,

sofern sich noch weitere MHs aus anderen Heimatnetzen im Netz des FAs aufhalten, die diese Gruppe abonniert haben.

Darüber hinaus besteht noch die weiter oben bereits erwähnte Möglichkeit der redundanten DMSPs. Auch eine Kombination von beidem ist denkbar und sinnvoll, da dadurch auch Paketverluste im Fall eines Stromausfalls sowie in der Zeit bis zur Bestimmung eines neuen DMSPs überbrückt werden können.

6.4 Bewertung

Das MoM Protokoll bietet - verglichen mit dem Bi-directional Tunneling-Ansatz der IETF - eine bessere Performance im Hinblick auf die verursache Netzwerbelastung. Umfangreiche Simulationen, die in [TGHBu97] vorgestellt werden, haben gezeigt, dass das MoM Protokoll dem Bi-directional Tunneling im Hinblick auf die Netzwerklast und damit die Skalierbarkeit deutlich überlegen ist.

Gegenüber dem Remote Subscription Ansatz kann das MoM Protokoll hinsichtlich der Routenoptimalität und der Netzwerklast nicht konkurrieren. Das in Abschnitt 5 beschriebene Problem des „Triangular Routings“ bleibt bestehen, was jedoch für die meisten Anwendungen vernachlässigbar ist, da weder IP noch Mobile IP optimale Routen garantieren. Der Zusatzaufwand zur Optimierung des Routings ist nur für Anwendungen mit hohen Quality of Service-Ansprüchen gerechtfertigt. Das MoM Protokoll bietet gegenüber Remote Subscription folgende Vorteile:

- Transparenz gegenüber den Protokollen und Anwendungen der höheren Schichten
- Schnelle Weiterleitung von Multicast-Paketen an entfernte Geräte, verglichen mit der bei Remote Subscription zur Neuberechnung des Multicast Delivery Trees benötigten Zeitspanne
- Unterstützung der Auslieferung von Multicast-Paketen an Scope Groups
- Kaum Änderungen an Mobile IP nötig

Ein bisher nicht gelöstes Problem des MoM Ansatzes ist der mehrfache Unicast-Versand eines Multicast-Pakets durch den MHA, wenn dieser als DMSP für MHs der gleichen Multicast-Gruppe in verschiedenen Fremdnetzen zuständig ist. Diese Ineffizienz kann jedoch nur durch eine auf Multicast basierende Methode behoben werden, was jedoch wieder eine Neuberechnung des Multicast Delivery Trees bei jedem Netzwechsel eines MHs nach sich ziehen würde. Diese aufwändige Neuberechnung zu umgehen, war jedoch gerade eines der Ziele des MoM Protokolls. Die in [TGHBu97] vorgestellten Simulationsergebnisse zeigen jedoch, dass diese Problematik nicht als kritischer Erfolgsfaktor zu sehen ist, da das MoM Protokoll dennoch eine gute Skalierbarkeit bietet.

7 MobiCast

MobiCast wurde 1999 von Cheng Lin Tan und Stephen Pink von der Universität Luelå, Schweden vorgestellt [TaPi00]. MobiCast ist hauptsächlich für den Einsatz in an das Internet angeschlossenen LANs, die mobile Geräte mittels Wireless LAN unter Verwendung kleiner Zellen anbinden - im Folgenden als „Domain“ bezeichnet -, ausgerichtet. Ein Beispiel dafür wäre das Netzwerk der Universität Karlsruhe.

Im Gegensatz zum MoM Protokoll ähnelt MobiCast eher dem Remote Subscription Ansatz der IETF. Dadurch werden das Tunnelkonvergenz-Problem sowie das suboptimale Routing des Bi-directional Tunneling-Ansatzes umgangen. MobiCast verwendet jedoch im Gegensatz zu Remote Subscription einen hierarchischen Mobilitätsmanagement-Ansatz, um die Mobilität der Geräte innerhalb der Domain vor den anderen Gruppenmitgliedern zu verdecken und somit die Neuberechnung des Multicast Delivery Trees zu verhindern.

Zur Minimierung von Paketverlusten beim Wechsel eines mobilen Multicast-Empfängers zu einer anderen Base Station werden Multicast Pakete auch an Base Stations in räumlich benachbarten Subnets weitergeleitet, so dass nach einem Wechsel die Pakete bereits im Puffer der neuen Base Station vorhanden sind und auf Anfrage sofort an den MH weitergeleitet werden können. Im Falle eines Multicast-Senders teilt die neue Base Station dem MH sofort nach dem Wechsel die ID des letzten Multicast-Pakets bei der vorigen Base Station mit, so dass dieser (bzw. die Anwendung) darauf geeignet reagieren und ggf. Pakete erneut senden kann.

7.1 Der hierarchische Mobilitätsmanagement-Ansatz

Abbildung 4 zeigt eine schematische Darstellung eines klassischen Campus-Netzwerkes mit einem MH sowie seinem HA im Heimat-Netzwerk und einem Multicast-Sender, die beide über das Internet mit dem Campus-Netzwerk verbunden sind. Innerhalb des Campus-Netzes existieren mehrere Subnetze, die mittels WLAN-Basisstationen die MHs an das Netz anschließen.

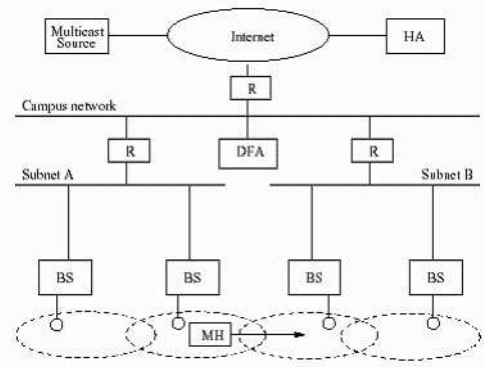

Abbildung 4: Klassisches Campus-(WLAN-)Netzwerk mit Domain Foreign Agent

Neu hinzugekommen ist der Domain Foreign Agent (DFA), der für alle fremden MHs zuständig ist. Wenn ein MH neu in das Campus-Netz kommt, registriert er sich beim DFA und sendet die IP-Adresse des DFA als seine COA an seinen HA. Von nun an fungiert der DFA bei der Registrierung in Multicast-Gruppen sowie beim Senden und Empfangen von Multicast-Paketen als Vertreter des MHs. Wechsel des MH von einer Wireless-Zelle in eine andere innerhalb der Domain bleiben so vor der Multicast Gruppe verborgen und eine Neuberechnung des Multicast Delivery Trees ist daher nicht notwendig.

7.2 Der Übergabe-Mechanismus

Zur Realisierung eines schnellen Übergangs des MHs zwischen zwei BSs und damit zur Minimierung von Paketverlusten während der Übergangszeit, werden - wie in Abbildung 5 schematisch dargestellt - physikalisch angrenzende Zellen in Dynamic Virtual Macrocells (DVM) zusammengefasst. Dabei gibt es innerhalb einer DVM einen Kern - bei DVM A ist dies BS2, bei DVM B ist es BS3 - und mehrere Mitglieder. Ein Übergang kann nur von der Kern-BS

zu einer Mitglied-BS stattfinden, z.B. also von BS2 nach BS1 oder BS3 oder von BS3 nach BS2 oder BS4. Außerdem kann auch nur der Kern Nachrichten an die übrigen Mitglieder der DVM übermitteln. Dieser Nachrichtenaustausch wird ebenfalls über Multicast abgewickelt, wobei jede DVM eine eigene Multicast-Adresse besitzt.

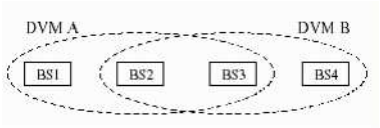

Abbildung 5: Dynamic Virtual Macrocells für schnelle Übergänge eines MHs zwischen benachbarten WLAN-Basisstationen

Multicast-Pakete, die vom DFA für einen MH empfangen werden, werden nun nicht nur an die BS, die momentan den MH bedient, weitergeleitet, sondern auch an die übrigen BSs innerhalb der gleichen DVM. Die Pakete werden dann allerdings nur von derjenigen BS aktiv weitergeleitet, die den MH momentan bedient. Die übrigen BS speichern die Pakete in ihrem Puffer und können sie im Falle eines Übergangs des MHs in ihren Bereich sofort an den MH weiterleiten. Abbildung 6 verdeutlicht diese Methodik nochmals.

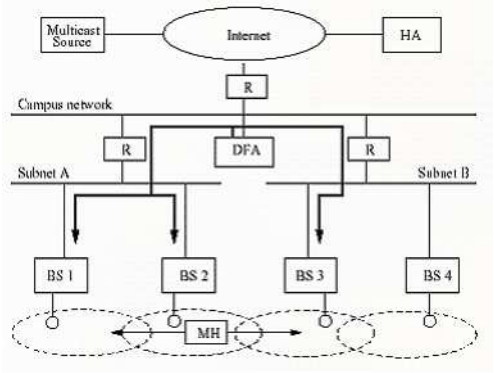

Abbildung 6: Multicast-Pakete werden vom DFA via lokalem Multicast an alle Mitglieder einer DVM weitergeleitet

Multicast-Pakete werden sowohl an BS2 als auch an BS1 und BS3 weitergeleitet, da sich BS1, 2 und 3 alle in der DVM A befinden. Wechselt der MH nun zu BS1 oder BS3, so stehen die letzten Multicast-Pakete bereits in deren Puffern bereit und können, nachdem der MH die ID des zuletzt empfangen IP-Pakets bekannt gegeben hat, sofort an diesen weitergeleitet werden. Dadurch wird ein Übergang des MHs zwischen verschiedenen Zellen mit minimalem Paketverlust und sehr schneller Übergangszeit möglich.

Bisher wurde nur die Übergangsprozedur für den Fall, dass der MH Empfänger von Multicast-Paketen ist, beschrieben. Fungiert der MH hingegen als Sender von Multicast-Paketen, so gehen alle vom MH gesendeten Pakete zwischen dem Zeitpunkt, wo der MH den Abdeckungsbereich der alten BS verlässt bis zu dem Zeitpunkt, wo der Übergang zur neuen BS abgeschlossen ist, verloren. Da es meist stark von der Anwendung abhängt, wie mit dem Datenverlust umgegangen werden soll, stellt MobiCast dem MH lediglich Informationen bereit, aus denen er entnehmen kann, welche Pakete ggf. neu gesendet werden müssen. Konkret bedeutet dies, dass die neue BS dem MH sofort nach dem Übergang die ID des letzten von ihm gesendeten Pakets, das die vorige BS noch empfangen hat, mitteilt.

Durch umfangreiche Simulationen konnten die Entwickler von MobiCast belegen, dass die Schwankungen im Multicast-Datenstrom beim Übergang eines MHs - sei es in der Rolle des Senders oder der des Empfängers von Multicast-Paketen - von einer BS zu einer anderen un-

terhalb der maximal tolerierbaren Grenze für eine interaktive Audio-Konferenz via Multicast liegen.

7.3 Mobile Geräte als Sender von Multicast-Paketen

Zum Senden von Multicast-Paketen an eine Multicast-Gruppe schickt der MH die zu sendenden Pakete gekapselt per Unicast an den DFA, der dann die Pakete wieder entkapselt und mit seiner eigenen Adresse als Absender-Adresse an die Multicast-Gruppe weiterleitet.

Falls die Anwendung jedoch erfordert, dass die Absenderadresse der Multicast-Pakete die des MHs ist, so kann der MH die zu sendenden Pakete auch an seinen HA tunneln, der diese dann für ihn an die Multicast-Gruppe weiterleitet. Bei dieser Alternative kommt jedoch wieder das Tunnelkonvergenz-Problem zum Tragen.

7.4 Mobile Geräte als Empfänger von Multicast-Paketen

Zunächst sendet der MH - wie vom normalen IP Multicast gewohnt - seinen Registrierungswunsch für eine Multicast-Gruppe X mittels IGMP Report an die für ihn zuständige BS. Die darauf folgenden Schritte von MobiCast bleiben ihm verborgen:

- Die BS leitet einen Sucscription Request, der die Adresse der gewünschten Multicast-Gruppe enthält, an den zuständigen DFA weiter.
- Daraufhin sendet der DFA eine Subscription Reply an die BS zurück, die eine lokale Multicast-Adresse Y enthält, die vom DFA der vom MH gewünschten Multicast-Gruppe zugeordnet wurde. Ist der DFA bereits Mitglied in X, so kann er die zugehörige Adresse von Y aus seiner Zuordnungstabelle entnehmen. Ansonsten erzeugt der DFA die Adresse von Y neu, fügt sie in die Zuordnungstabelle ein und registriert sich in der Multicast Gruppe X.
- Nach Erhalt der Subscription Reply Nachricht vom DFA tritt die BS der lokalen Multicast Gruppe Y bei und informiert die anderen WLAN-Basisstationen in ihrer DVM, dass sie ebenfalls Y beitreten sollen.

Nach der Registrierung in der Multicast-Gruppe durch den DFA empfängt dieser alle an die Gruppe gerichteten Multicast-Pakete und leitet sie an die lokal zugeordnete Multicast-Gruppe Y weiter. Die BS, die den MH momentan bedient, erhält das Paket, ändert die Empfänger-Adresse wieder auf die Adresse von X und leitet das Paket an den MH weiter. Die übrigen BSs speichern die Pakete nach dem FIFO-Prinzip in ihrem Puffer. Für den MH bleibt das MobiCast Protokoll also weitgehend transparent: Er registriert sich via „normalem“ IGMP Report und erhält die Pakete scheinbar direkt, ohne Zwischenstationen, geliefert.

7.5 Bewertung

MobiCast bietet folgende Vorteile...:

- Abschirmung der Mobilität des MHs innerhalb der Domain vor dem Rest der Multicast-Gruppe und somit Verhinderung der Neuberechnung des Multicast-Delivery Trees bei Intra-Domain Mobilität. Der Aufwand zur Neuberechnung des Multicast Delivery Trees innerhalb der Domain, der zur Auslieferung der vom DFA empfangenen Pakete an die BSs genutzt wird, ist vernachlässigbar.

- Effizientes Routing der Multicast-Pakete aufgrund der notwendigen Neuregistrierung des MHs in den gewünschten Multicast-Gruppen bei Eintritt in die Domain.
- Durch die Verwendung von Multicast zur Verteilung ' 'der vom DFA empfangenen Multicast-Pakete innerhalb der Domain wird einerseits die Verteilung effizient gelöst und andererseits der DFA von der Aufgabe befreit, stets den aktuellen Aufenthaltsort des MHs zu kennen, um die korrekte Weiterleitung der Pakete sicherzustellen.
- Schnelle Übergabe beim Wechsel des MH zwischen BSs bei minimaler Unterbrechung des Multicast-Paketstroms.
- Der MobiCast-Prozess zur Registrierung in einer Multicast-Gruppe sowie zum Empfang von Multicast-Paketen durch den MH bleiben für diesen unsichtbar und erfordern somit keine Änderungen am MH.

...hat aber auch einige Schwächen:

- Die Netzwerkbelastung durch die Weiterleitung von Multicast-Paketen an die benachbarten BSs ist nicht vernachlässigbar.
- Inter-Domain-Mobilität wird nicht behandelt und führt weiterhin zur Neuberechnung des Multicast Delivery Trees.
- Am Foreign Agent sind Änderungen nötig.

8 Zusammenfassung

Die von der IETF vorgeschlagenen Protokolle Remote Subscription und Bi-directional Tunneling weisen zu viele Schwächen auf, um für den praktischen Einsatz geeignet zu sein. Das MoM Protokoll und das MobiCast Protokoll verwenden je eines dieser Protokolle als Grundlage und erweitern dieses - erfolgreich - um Mechanismen zur Umgehung der bekannten Schwächen. Während das MoM Protokoll auf dem Bi-directional Tunneling Ansatz basiert und diesen um Mechanismen wie zum Beispiel die Ernennung von HAs zu DMSPs erweitert, setzt MobiCast auf den Remote Subscription Ansatz auf und erweitert ihn um das hierarchische Mobilitätsmanagement zur Abschirmung der Intra-Domain Mobilität vor dem Multicast-Delivery Tree.

Es wird vom jeweiligen Einsatzgebiet abhängen, welches Protokoll die bessere Wahl darstellt. MobiCast ist schon aufgrund seiner Ausrichtung auf Umgebungen mit drahtlosen, aus vielen kleinen Zellen bestehenden Netzen gut für Firmen- und Campus-Netze geeignet, wo die Mobilität des MH größtenteils auf die Mobilität innerhalb der Domain begrenzt ist. Benötigt man hingegen ein flexibleres Protokoll, das die Mobilität des Geräte - wenn auch zu Lasten der Routing Effizienz - global unterstützt, wird die Wahl wohl eher zugunsten des MoM Protokolls ausfallen.

Literatur

[Ball97] A. Ballardie. Core Based Trees (CBT) Multicast Routing Architecture, September 1997.

[Cain02] B. Cain. Internet Group Management Protocol, Version 3. RFC 2236, October 2002.

[D. J03] J. Arkko D. Johnson, C. Perkins. Mobility Support in IPv6. Internet-Draft, June 2003.

[D. W88] S. Deering D. Waitzman, C. Partridge. Distance Vector Multicast Routing Protocol. RFC 1075, November 1988.

[Drom93] R. Droms. Dynamic Host Configuration Protocol. RFC 1514, November 1993.

[eal94] S. Deering et al. An Architecture for Wide-Area Multicast Routing. In *Proceedings of the 1994 ACM SIGCOMM Conference*, 1994, S. 126–135.

[Mont98] G. Montenegro. Reverse Tunneling for Mobile IP. RFC 2344, May 1998.

[Moy94] J. Moy. Multicast Extensions to OSPF. RFC 1584, März 1994.

[Perk96] C. Perkins. IP Encapsulation within IP. RFC 2003, October 1996.

[TaPi00] Cheng Lin Tan und Stephen Pink. MobiCast: A Multicast Scheme for Wireless Networks. *Mobile Networks and Applications*, 2000.

[TGHBu97] Wayne L. Mackrell Tim G. Harrison, Carey L. Williamson und Richard B. Bunt. Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts. *Mobile Computing and Networking*, 1997.

[VCMa98] Richard B. Bunt Vineet Chikarman, Carey L. Williamson und Wayne L. Mackrell. Multicast Support for Mobile Hosts Using Mobile IP: Design Issues and Proposed Architecture. *Mobile Networks and Applications*, 1998.

Abbildungsverzeichnis

Mobile IP - Erweiterungen zur Unterstützung schneller Handover

Frederic Majer

Kurzfassung

Mit der wachsenden Mobilität der Internetnutzer werden neue Protokolle, die eine sinnvolle und zuverlässige Kommunikation zwischen den Kommunikationsteilnehmern ermöglichen, nötig. Die vorliegende Ausarbeitung baut auf dem Protokoll Mobile IPv6 auf, welches die Erreichbarkeit eines mobilen Knoten und seiner Dienste über die Grenzen seiner Heimatnetzes hinaus gewährleistet. Problematisch bleibt dabei nur der Vorgang des Handover. Hierfür werden drei Erweiterungen vorgestellt, die die Probleme bei einem Netzwerkwechsel minimieren. Ungefähr 10–12 Zeilen schreiben und bitte als ein Absatz.

1 Einleitung

Mit der stetigen Weiterentwicklung und der verstärkten Einbindung neuer Technologien in unseren Alltag wachsen die Ansprüche an die zugrundeliegenden Protokolle. Einen der größten Trends kann man kurz als ubiquitous Internet - die allgegenwärtige Erreichbarkeit und Kommunikation beschreiben. Um diese uneingeschränkte Kommunikation zu ermöglichen, entfernt man sich zunehmend von den altbekannten verdrahteten Infrastrukturnetzen, welche sich durch feste Adresszuweisungen sowie seltene Paketverluste und Verbindungsabbrüche auszeichnen, hin zu drahtlosen Erweiterungen dieser Festnetze, sogenannten „drahtlosen Zugangsnetzen“. Über drahtlose Verbindungen wird hier eine dynamische Infrastruktur von mobilen Knoten, die meist über schwache Rechen- und Energieleistung verfügen, geschaffen. Zur Unterstützung einer sinnvollen Nutzung dieser neuartigen Netze sind neue Mechanismen nötig.

Die vorliegende Ausarbeitung konzentriert sich dabei nicht auf die Schwierigkeiten, die durch den Wechsel vom Medium Kabel zur Luft entstehen, sondern auf die wachsende Mobilität der Knoten. In Zukunft wird die Gesamtzahl der Internetnutzer unter Berücksichtigung der mobilen Teilnehmer, welche von einem Ort zum anderen reisen und dort immer unter ihrer Heimatadresse erreichbar sein wollen, stark wachsen. Mit der Spezifikation des „Internet Protokolls Version 6“ (IPv6) hat man den nun schon über dreißig Jahre alten Standard des „Internet Protokolls Version 4“ (IPv4) deutlich verbessert und an die neuen Anforderungen angepasst. Z. B. wurde der Adressraum stark ausgeweitet und somit eines der gravierendesten bereits absehbaren Probleme - die Adressknappheit durch die steigende Anzahl der in Zukunft verwendeten Geräte - gelöst. Unter dem Namen „Mobility Support in IPv6“ (Mobile IPv6) wurde eine spezielle Unterstützung für die Mobilität der Geräte definiert.

Mobile IPv6 bildet die Grundlage dieser Ausarbeitung und wird in Kapitel 3 neben Adresskonfigurationsverfahren für mobile Knoten vorgestellt. Da das Protokoll noch nicht alle Probleme, die mit der Mobilität der Clients einhergehen, behebt, werden in Kapitel 4 Erweiterungen vorgestellt, welche diese Lücken schließen und es zum Protokoll der Zukunft machen.

2 Grundlagen

Neben der Vermittlung des Wissens über die Funktionsweise der allgemeinen Internetprotokolle bildet dieser Abschnitt die zentrale Grundlage für das Verständnis der Ausarbeitung. Es werden Verfahren zur Vergabe von IP Adressen an mobile Clients sowie das Protokoll Mobile IPv6 vorgestellt.

2.1 Host Configuration (DHCP und Stateless Address Auto-Configuration)

DHCP (Dynamic Host Configuration Protocol) und Stateless Address Auto-Configuration sind für die Mobilitätsunterstützung von mobilen Geräten (z.B. Notebook) in einer fremden Umgebung unerlässlich geworden. Neben der Übermittlung von für die Kommunikation wichtigen Informationen, wie die Subnetzmaske und der Adresse des Domain Name Servers und des Standard-Gateways, bieten die Protokolle einem Client Mechanismen eine freie und topologisch korrekten IP-Adresse zu erhalten. Optional können noch weitere Informationen wie z.B. die Adresse des Webservers des Subnetzes an den Knoten übermittelt werden.

DHCP ermöglich die Vergabe der IP-Adresse „manuell“ durch den Administrator, „automatisch“ auf unbegrenzte und „dynamisch“ auf begrenzte Zeitdauer. Für mobile Geräte bietet sich gerade der dritte Modus - die dynamische Vergabe der IP-Adresse - an. Wie bei den anderen beiden Modi erhält der Client seine Konfigurationsinformationen indem er per Link-Level-Broadcast ein DHCPDISCOVER sendet. Der DHCP-Server antwortet mit einem DHCPOFFER, einer möglichen Adresskonfiguration. Nimmt der Host diese an, sendet er ein DHCPREQUEST mit der gewählten Adresse direkt an den Server, welcher dann wiederum mit DHCPACK antwortet.

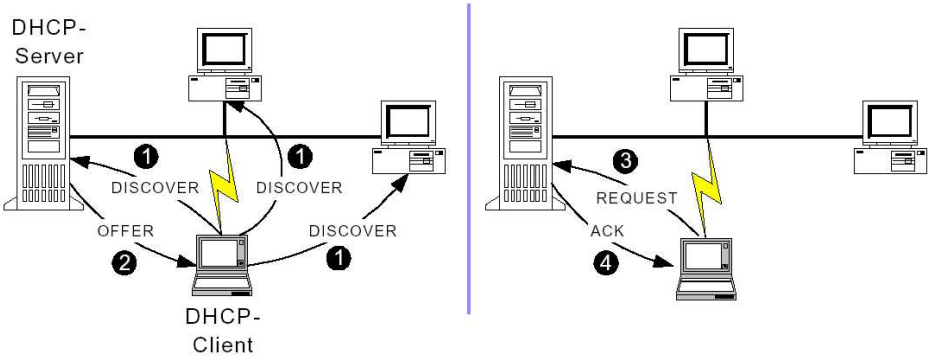

Abbildung 1: Protokollablauf von DHCP

Bei der dynamischen Adressvergabe ist der Client im Besitz eines Leases, einer Zeitdauer für die Benutzung. Nach Ablauf der Hälfte der Lease-Zeit versucht der Host mittels DHCPREQUEST an den DHCP-Server die IP-Zuweisung zu verlängern. Erhält er nach 7/8 der Lease-Zeit keine Antwort wendet er sich mittels Broadcast an alle DHCP-Server, um eine neue IP-Adresse zu erhalten.

Als weiteres Protokoll - gerade im Zusammenhang mit IPv6 - muss die „Stateless Address Auto-Configuration“ genannt werden. Auch über diesen Mechanismus erlangt der Client eine korrekte IP-Adresse. Im Gegensatz zu DHCP bekommt ein IPv6-fähiges Gerät die Adresse nicht zugewiesen, sondern generiert sie sich aus seiner MAC-Adresse und sendet eine Neighbor Soliticiation an seine Nachbarn. Ein Router antwortet mit einer Router-Advertisement-Nachricht und übermittelt dem Client weitere für die Kommunikation wichtige Details. Das

Endgerät überprüft mittels Duplicate Address Detection, ob die Adresse noch nicht belegt ist.

Zusammenfassend kann man sagen, dass mittels DHCP oder Stateless Address Auto-Configuration auf einfache Art und Weise ein mobiler Host in ein Netz eingebunden werden und dieser dann die Dienste des lokalen Netzes nutzen (z.B. Internet) kann. Allerdings entstehen - neben sicherheitsrelevanten Fragestellungen wie das ungewollte Öffnen des Netzes für jeden mobilen Knoten und dem ungenügenden Schutz von mobilen Rechnern vor Falschinformationen - Probleme, falls der Host selbst Dienste anbietet. In diesem Fall sind die Dienste durch den ständigen Adresswechsel von außen nicht auffindbar und können nicht unterbrechungsfrei angeboten werden, sobald sich der Host in ein neues Subnetz bewegt. Diese Schwierigkeiten werden durch Mobile IP gelöst, welches einem mobilen Host eine feste IP-Adresse zuweist, die er in allen Subnetzen sowie im Heimatnetz nutzen kann.

2.2 Mobile IPv6

Die neueste Version von Mobile IPv6 wurde im Sommer 2003 verabschiedet und ist die Weiterentwicklung und Verbesserung von Mobile IPv4. Das Protokoll ermöglicht das Routen von IP-Paketen zu mobilen Knoten im Internet, da diese unabhängig von ihrer Position über ihre Heimatadresse angesprochen werden. Außerhalb des Heimatnetzwerks wird dem Knoten eine zusätzliche Adresse zugewiesen unter welcher er direkt oder indirekt erreichbar ist. Dadurch kann sich der Client frei bewegen und über die Grenzen seines Heimatnetzwerks hinaus für andere Kommunikationspartner erreichbar sein und Dienste anbieten.

Zur näheren Erläuterung von Mobile IPv6 sind einige Begriffsdefinitionen nötig. Man bezeichnet den Knoten bzw. das Gerät, welches sich über verschiedene Netze hinwegbewegt und überall unter seiner Heimatadresse ansprechbar sein soll als Mobile Node (MN). In jedem Netz publizieren die vorhandenen Router sogenannte Router-Advertisement-Nachrichten und teilen allen im Netz befindlichen Knoten Informationen über ihre Adresse und Lebensdauer mit und geben Auskunft, ob sie als Home Agent fungieren können (Router Discovery). Router können auch explizit durch eine Router-Solicitation-Nachricht zur Abgabe der Informationen aufgefordert werden. Ein weiterer wichtiger Mechanismus ist die Überprüfung der Erreichbarkeit bestimmter Knoten. Hierbei kann ein Host die Adresse aller benachbarten Knoten über eine Neighbor-Solicitation-Nachrichten, welche mit einem Neighbor-Advertisement beantwortet wird, erhalten (Neighbor Discovery). Das Senden dieser Pakete an einen bestimmten Knoten zur Überprüfung seiner Erreichbarkeit nennt man Neighbor Unreachability Detection. Sobald der mobile Knoten bemerkt, dass er das Heimatnetzwerk verlassen hat, verschafft er sich mittels z.B. Stateless Address Auto-Configuration eine für das fremde Netzwerk topologisch korrekte IP-Adresse, einer Care-of-Address (CoA). In seinem Heimatnetzwerk hat der Client einen Home Agent (HA), welcher in vertritt und an ihn adressierte Pakete an seinen aktuellen Aufenthaltsort weiterleitet. Ihm teilt der mobilen Knoten seine aktuelle Care-of-Address nach jedem Netzwerkwechsel mittels einem Binding Update (BU) mit. Der Kommunikationspartner in diesem System wird als Correspondent Node (CN) bezeichnet.

Jeder Client erhält in seinem Heimatnetzwerk eine IP-Adresse über die er auf das Medium zugreifen kann und unter der er selber sowie seine angebotenen Dienste erreichbar sind. Bewegt sich der mobile Knoten aus seinem Heimatnetzwerk heraus ergibt sich das Problem, dass ihm im Fremdnetz eine neue IP-Adresse zugewiesen wird, wodurch aktive Verbindungen zu Kommunikationspartnern unterbrochen werden. Des Weiteren ist der mobile Knoten dem Netz unter der neuen IP-Adresse unbekannt und niemand könnte seinen angebotenen Dienst in Anspruch nehmen. Würde sich der Client nun kontinuierlich bewegen, wäre eine sinnvolle Kommunikation unter diesen Bedingungen kaum möglich.

Mobile IPv6 umgeht diese Problematik, indem jeder mobile Knoten in seinem Heimatnetzwerk über einen Home Agent verfügt, welcher für den mobilen Knoten bestimmte Pakete an den aktuellen Aufenthaltsort des mobilen Knoten weiterleitet. Der mobile Knoten bleibt allen Kommunikationspartnern unter seiner alten Heimatadresse bekannt und teilt vorerst nur seinem Home Agent Bewegungen und damit verbundene, neu erhaltene IP-Adressen mit.

Eine Bewegung kann der Client einfach mittels Neighbor Unreachability Detection überprüfen. Dabei kontrolliert er, ob sein Default Router noch erreichbar ist. Bleibt eine Antwort aus, sucht er sich mittels Router Discovery einen neuen Default Router. Hat dieser eine IP-Adresse mit einem anderen Netz-Präfix, muss er sich bewegt haben - entspricht sie der alten Adresse, ist der alte Default Router temporär nicht erreichbar oder hat sich komplett aus dem Netz bewegt.

Falls der mobile Knoten eine Bewegung seinerseits erkannt hat, muss er sich eine Care-of-Address beschaffen. Diese Adresse entspricht einer normalen IP-Adresse, welche er über DHCP bzw. Stateless Address Auto-Configuration bekommen hat. Durch eine Binding-Update-Nachricht, welche die Heimatadresse und die neue Care-of-Address beinhaltet, teilt er die neue Adresse seinem Home Agent mit, welcher die Nachricht mit einem Binding Acknowledgement bestätigt. Falls die Bestätigung ausbleibt und davon ausgegangen werden muss, dass der Home Agent nicht mehr verfügbar ist, kann der mobile Knoten dynamisch einen neuen Home Agent in seinem Heimatnetzwerk suchen und ihm seinen aktuellen Standort übermitteln.

Mit der Bestätigung des Binding Updates übernimmt der Router, der somit als Home Agent agiert, mehrere Aufgaben.

Die wohl wichtigste Aufgabe ist das Weiterleiten aller Pakete, welche an die Heimatadresse des mobilen Knotens gerichtet sind, an die aktuell verwendete Care-of-Address. Dabei kapselt der Home Agent alle Pakete in ein neues IP-Paket. Da er das ursprüngliche Paket nicht verändert, kann der mobile Knoten beim Entpacken den Absender des ursprünglichen Pakets identifizieren. Dieser Vorgang der unveränderten Weiterleitung - IP-Pakete werden in das Nutzdatenfeld eines neuen IP-Paketes gepackt - nennt man Tunneln.

Des Weiteren hat der Home Agent die Aufgabe, den mobilen Knoten im Heimatnetz zu vertreten. Er schickt deshalb periodisch Neighbor-Advertisement-Nachrichten im Namen des mobilen Knotens und gibt seine eigene MAC-Adresse an. Somit ist der Knoten noch immer für die anderen Knoten im Heimatnetzwerk erreichbar und Pakete, welche für den Knoten bestimmt sind, werden immer über den Home Agent geschickt, falls sich der mobile Knoten außerhalb des Heimatnetzwerkes befindet. Auch das Antworten auf Neighbor -Solicitations übernimmt der Home Agent.

Erhält der mobile Knoten ein durch seinen Home Agent weitergeleitetes Paket, kann er dem Paket die Adresse des Correspondent Node entnehmen und diesem ein Binding Update schicken. Ist dies geschehen, kann die weitere Kommunikation, anstatt bidirektional über den HA getunnelt, direkt zwischen Care-of-Address und der Adresse des Correspondent Node ablaufen.

Grundsätzlich gewährleistet das Protokoll Mobile IPv6 die Erreichbarkeit eines mobilen Knoten und seiner Dienste über die Grenzen seiner Heimatnetzes hinweg. Problematisch bleibt dabei nur der Vorgang des Handover. Wenn sich der Knoten in ein neues Netz bewegt, muss er sich einen neuen Default Router suchen und ein Binding Update an seinen Home Agent und an alle Correspondent Nodes, welche teilweise weit entfernt sein können, senden. Währenddessen ist seine Erreichbarkeit eingeschränkt und Pakete von bestehenden Verbindungen können verloren gehen, da sie noch an die alte Care-of-Address gesendet werden. Dies kann, je nach Art der Verbindung und Dauer des Irrtums, zu Engpässen führen. Auf der Transportschicht werden beim Einsatz von TCP länger ausbleibende Paketbestätigungen möglicherweise als

Stausituation interpretiert und es wird fälschlicherweise mit einer Reduzierung der Datenrate reagiert.

Für diese Probleme bietet Mobile IPv6 keine ausreichende Lösung, aber es gibt mehrere Alternativen und Erweiterungen - in Kapitel 3 beschrieben - die das bestehende Protokoll verbessern und, neben der Erreichbarkeit, die Performanz aktiver Verbindungen über einen Handover hinweg verbessern.

3 Erweiterungen von Mobile IPv6

In diesem Kapitel werden drei Erweiterungen zu Mobile IPv6 vorgestellt. Dabei wird jeweils das Verfahren in groben Zügen erklärt und vor allem der Nutzen für die Unterstützung schneller Handover heraus gearbeitet.

3.1 Hierarchical Mobile IPv6 mobility management (HMIPv6)

HMIPv6 erweitert das System von Mobile IPv6 um einen weiteren Knoten, den Mobility Anchor Point (MAP). Dieser Knoten befindet sich auf einer beliebigen Ebene der Fremdnetzhierarchie und fungiert dort für einen bei ihm registrierten mobilen Knoten als lokaler Home Agent. Alle an den Client adressierten Pakete werden an dessen aktuellen Mobility Anchor Point gesendet - dieser tunnelt sie zu dem bei sich gespeicherten Aufenthaltsort des mobilen Knotens. Wenn sich der mobile Knoten im Fremdnetzwerk zwischen den verschiedenen Subnetzen bewegt, sendet er nur ein einziges Binding Update an den MAP, anstatt sich an seinen Home Agent und alle anderen mit ihm in Verbindung stehenden Correspondent Nodes zu wenden.

Zur Erläuterung der Erweiterung sind nur wenige neue Begriffsdefinitionen nötig. Mit der Einführung des bereits kurz erwähnten Mobility Anchor Points (MAP) ist eine Differenzierung des bereits bekannten Care-of-Address nötig. Zum einen verfügt der mobile Knoten in jedem Subnetz über eine On-Link Care-of-Address (LCoA), welche nur lokal - innerhalb der Domäne eines MAPs - eindeutige ist. Beim Eintritt in die Domäne eines MAPs konfiguriert er sich aber auch eine Regional Care-of-Address (RCoA). Diese ist global eindeutig und kann dem Fremdnetzwerk zugewiesen werden. Das Binding Update zwischen LCoA und RCoA wird beim MAP durchgeführt und als Local Bindung Update bezeichnet.

Sobald ein mobiler Knoten das fremde Netz betritt, bekommt er Router Advertisements mit Informationen über den oder die MAPs im Netz (MAP discovery). Nun kann er seine lokale (LCoA) und globale Adresse (RCoA) des Fremdnetzes konfigurieren und diese Information an einen ausgewählten MAP senden. Nachdem dieser eine Duplicate Address Detection durchgeführt hat, bestätigt er das Binding und teilt dem Client die Dauer der Zusammenarbeit mit. Optional kann durch den MAP noch überprüft werden, ob sich der mobile Knoten wirklich an der Stelle befindet und das System keiner fflooding attackëines anderen Knotens unterliegt. Sind diese Schritte erfolgreich durchgeführt, tritt der MAP von nun an als lokaler Home Agent auf, empfängt alle Pakete, welche für den Client bestimmt sind und leitet diese an seine aktuelle LCoA weiter. Neben der Übermittlung der mit dem MAP vereinbarten voraussichtlichen Dauer der Zusammenarbeit, teilt der mobile Knoten seinem Home Agent im Heimatnetz und allen Correspondent Nodes vor allem seine neue Adresse durch die Übermittlung der RCoA mit. Dabei bleibt im weiteren Verlauf sein genauer Standort verborgen, da der für ihn zuständige MAP ihm in der jeweiligen Domäne die Pakete weiterleitet.

Die Domäne eines MAP bestimmt sich über den Bereich, in dem die Access Router seine Informationen publizieren. Wechselt der Client sein Netz, kann er anhand der Router Advertisements überprüfen, ob sein MAP auch für dieses Netz zuständig ist und er somit nur

das Subnetz gewechselt hat. Ist dies der Fall muss er nur ein Binding Update mit seiner aktuellen LCoA an den MAP schicken. Dies funktioniert durch die Übermittlung des RCoA in einem speziellen Bereich des IP-Headers zur Identifikation - die LCoA kann der MAP der Absenderadresse des Paketes entnehmen und somit das Binding aktualisieren. Die Bindings zwischen Heimatadresse und RCoA bei dem Home Agent und Correspondent Nodes bleiben unverändert. Wechselt der mobile Knoten komplett die Domäne und erhält somit Router Advertisements, denen er entnehmen kann, dass sein vorheriger MAP nicht in diesem Bereich verfügbar ist, bringt die Erweiterung keine Verbesserung. Wie beim Wechsel vom Heimatnetz ins Fremdnetz muss nun wieder eine LCoA und RCoA neu bestimmt werden und die verschiedenen Bindings bei dem neuen MAP bzw. seinem Home Agent und Correspondent Nodes durchgeführt werden.

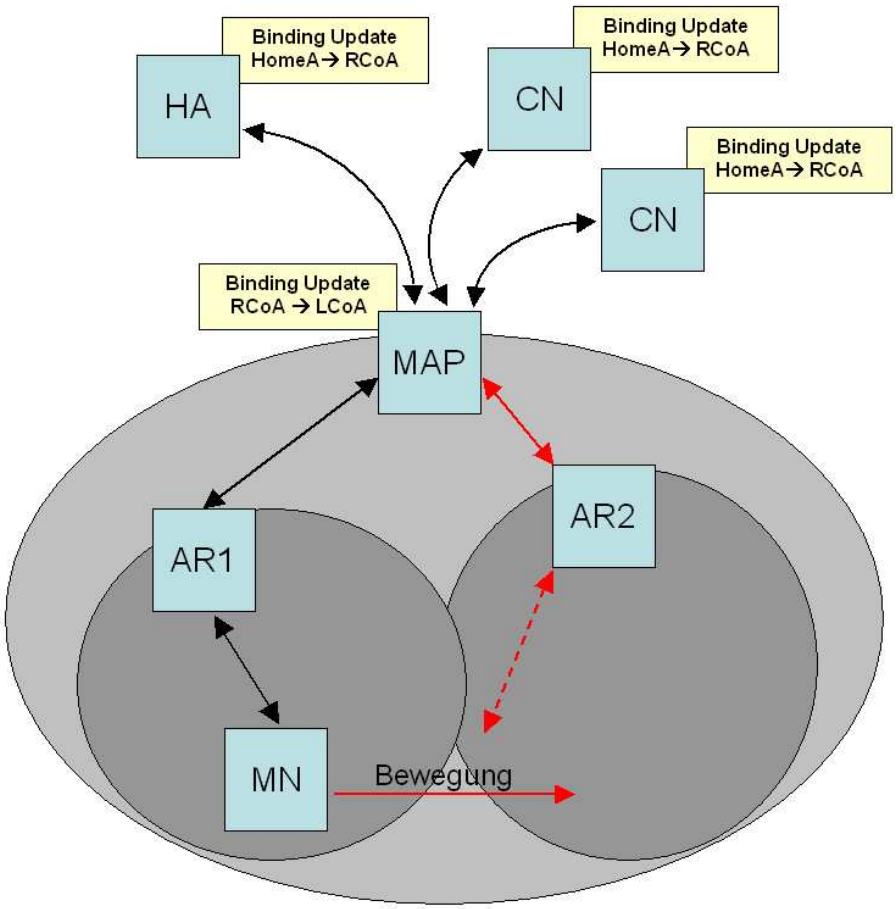

Abbildung 2: HMIPv6 im Überblick

Da man die Anzahl der Handover und somit die Anzahl der Domänen so gering wie möglich halten möchte, decken einzelne MAPs teilweise große Bereiche ab. Dadurch können sie sich nicht immer in unmittelbarer Nähe der mobilen Knoten befinden und die Distanz zwischen mobilem Knoten und MAP wachsen. Um dennoch die Vorteile eines llokalen Home Agentünd die Bandbreite des Mediums effizient zu nutzen, hat jeder Client die Möglichkeit, sich parallel bei mehreren MAPs zu registrieren und einzelne MAPs für eine gewisse Gruppe an Correspondent Nodes zu benutzen. Für die Auswahl eines MAPs bezüglich eines Kommunikationspartners können Faktoren wie die Kapazitäten der jeweiligen MAPs, Geschwindigkeit und Richtung der Bewegung des mobilen Knotens und die Entfernung und Art der Verbindung zwischen den beiden kommunizierenden Partnern ausschlaggebend sein. Je unwahrscheinli-

cher es erscheint, dass sich ein mobiler Knoten über mehrere Netze hinweg bewegen wird, desto näher kann der MAP gewählt werden. Ein Client sollte immer die Strategie verfolgen, aktiv neue Bindings zu erstellen und sich somit dem Geschehen anzupassen. Bestehende Bindings sollte er nur zögernd auflösen.

Des Weiteren sei noch erwähnt, dass der Client auf unterster Ebene - im eigenen Subnetz - seine aktuelle LCoA ganz normal als CoA, wie schon aus Mobile IPv6 bekannt, nutzen kann. Infolgedessen entsteht kein overhead durch das Versenden der Pakete über einen MAP, aber man verliert auch den Vorteil der Mikromobilität, den HMIPv6 bietet.

Sollte es zu einem Wechsel des MAPs kommen, so hat der mobile Knoten die Möglichkeit - nachdem er sich bei seinem neuen MAP registriert hat und über eine neue RCoA und LCoA verfügt - den alten MAP zu bitten, weiterhin alle an ihn adressierten Pakete an seine neue Adresse weiterzuleiten. Sozusagen sendet er ein Binding Update mit seiner neuen CoA, welche außerhalb der Domäne des alten MAPs liegt. Ob der alte MAP dieser Bitte nachkommt, hängt von seiner Konfiguration ab. Empfohlen wird das Weiterleiten von Paketen an Adressen, welche Access Routern zugewiesen werden können, die sich in angrenzenden MAPs und innerhalb der selben administrativen Domain befinden. Somit können weiche Handover zwischen MAPs durchgeführt werden, falls das Konfigurieren und Binden des Clients ausreichend schnell durchgeführt wird. Um den Vorgang des Binding Updates zu beschleunigen, gibt es die Möglichkeit in dem BU-Paket an den neuen MAP das BU für den Home Agent zu kapseln. Dadurch wird Zeit gespart, aber die jeweiligen Informationen für den Home Agent und den MAP sind nicht verifiziert und die Option sollte überlegt verwendet werden. Ein erhöhter Aufwand kann entstehen, falls der MAP die Registrierung aufgrund einer bereits vergebenen CoA (lokal überprüfbar mit Duplicate Address Detection) ablehnt oder die Anfrage mit einer geringeren Lebensdauer bestätigt.

Im nächsten Unterkapitel „Fast Handovers for Mobile IPv6“ werden weitere Mechanismen vorgestellt, die den Erhalt einer aktiven Verbindung zwischen einem mobilen Knoten und einem Kommunikationspartner über die Grenzen einer Domäne hinweg verbessern, indem der Vorgang des Handovers beschleunigt und teilweise überbrückt wird. Diese Mechanismen lassen sich dann mit dem Hierarchical Mobile IPv6 mobility management kombinieren.

HMIPv6 bietet aber bereits entscheidende Verbesserungen gegenüber Mobile IPv6 an, ohne große Veränderungen vorzunehmen. Durch das Hinzufügen des Mobility Anchor Point wird das Netz bei einem Wechsel des mobilen Knotens innerhalb der Domäne des ihm zugewiesenen MAPs entscheidend entlastet. Im Gegensatz zu Mobile IPv6 muss hier nur ein Binding Update (an den MAP) gesendet werden. Des weiteren befindet sich der MAP meist näher am mobilen Knoten als der Home Agent, wodurch die Information einer Bewegung (Binding Update) viel schneller wahrgenommen und darauf reagiert werden kann. Ein Großteil der Pakete, welche bereits auf dem Weg zum mobilen Knoten sind, gehen dadurch nicht verloren und werden bereits an die richtige Adresse gesendet. Die mit einem kompletten Domänenwechsel verbundenen Probleme der falsch gerouteten IP-Pakete kann man eindämmen, indem man dem alten MAP mitteilt, wohin sich der mobile Knoten bewegt hat. Unterstützt dieser die Weiterleitung an Access Router anderer Domänen, so werden die Pakete dem mobilen Knoten hinterhergesendet. Kapitel 3.3 „Simultaneous Bindings for Mobile IPv6 Fast Handovers“ bietet weitere interessante Ansätze zu diesem Bereich der Optimierung.

3.2 Fast Handovers for Mobile IPv6

Obwohl Mobile IPv6 das freie Bewegen eines mobilen Knotens auch in Fremdnetze hinein unterstützt, bleibt dennoch das Problem der Handover und die damit verbundenen Kommunikationsverzögerungen. Hauptsächlich kann man die Verzögerung darauf zurückführen, dass ein mobiler Knoten erkennen muss, dass sich seine Position geändert hat und nun ein

anderer Access Router für ihn zuständig ist; er die nötigen Informationen für eine CoA-Konfiguration in Erfahrung bringen und die neu erstellte Adresse als Binding Update an seine Kommunikationspartner senden muss. Dieses Kapitel stellt ein Verfahren vor, welches den Zeitabschnitt der Kommunikationsunterbrechung minimiert und somit Echtzeitanwendungen wie „Voice over IP“ oder durchsatzgebundene Verbindungen im Zusammenhang mit Mobile IPv6 ermöglicht.

Die beiden beteiligten Access Router werden als Previous bzw. New Access Router (PAR / NAR) und die jeweiligen in diesen Netzen gültigen IP-Adressen des mobilen Knoten dementsprechend als Previous bzw. New Care-of-Address (PCoA / NCoA) bezeichnet. Des Weiteren ergänzt die Erweiterung das Protokoll um sieben neue Nachrichtenformate, welche im Protokollverlauf erläutert werden.

Über Mechanismen auf der Sicherungsschicht erfährt der mobile Knoten, dass neu erreichbare Access Points und Netze vorhanden sind und ein Handover bevorsteht (abnehmende Signalstärke des PAR und zunehmende Präsenz anderer Knoten). Falls der Client das Protokoll unterstützt, initiiert er einen Fast Handover und versucht noch während er bei seinem momentanen Access Router angemeldet ist, Informationen über den neuen Zugangspunkt - zu dem er sich hinbewegt - in Erfahrung zu bringen. Hierfür sendet er ein Router Solicitation for Proxy (RtSolPr) mit der Link-Layer-Adresse oder einer ID des neuen Zugangspunkts (AP-ID) an seinen PAR. Als Antwort erhält der mobile Knoten das Proxy Router Advertisement (PrRtAdv) mit einem Tupel der Art [AP-ID, AR-Info] vom PAR mit den nötigen Informationen zum NAR und zur Konfiguration einer Care-of-Address (NCoA) mittels Stateless Address Auto-Configuration. Das PrRtAdv kann den mobilen Knoten auch ohne Aufforderung erreichen, sobald der Handover durch das Netz initiiert wird. Wechselt der Client nun das Netz, so hat er bereits eine CoA für das neue Netz, wobei diese nur voraussichtlich ist, da sie noch nicht mittels Duplicate Address Detection überprüft wurde.

Die Überprüfung erfolgt am NAR und wird durch das Senden des Fast Binding Updates (FBU) des mobilen Knotens an seinen PAR, welche unter anderem die voraussichtliche NCoA des mobilen Knotens enthält und die Weiterleitung aller für den mobilen Knoten bestimmten und am PAR ankommenden Pakete aktiviert, eingeleitet. Der PAR sendet ein Handover Initiate (HI) an den NAR und leitet die NCoA weiter. Ist die Adresse zulässig übernimmt der NAR die Zuständigkeit - optional puffert er alle ankommenden Pakete für diese Adresse. Sollte eine Adresskollision vorliegen, sendet er im Handover Acknowledge (HAck) eine alternative NCoA, die der mobile Knoten anstelle der selbstkonfigurierten Adresse verwenden muss. Diese Adresse kann dem Client durch das Fast Binding Acknowledgement (FBAck), welches ihm den erfolgreichen Abschluss aller Vorkehrungen und den Beginn des Tunnelns aller am PAR ankommenden und für den Client bestimmten Pakete an die NCoA mitteilen soll, übermittelt werden. Damit das FBAck den mobilen Knoten mit Sicherheit erreicht - er könnte das ursprüngliche Netz bereits verlassen haben - wird es an seine PCoA und NCoA gesendet. Wechselt der mobile Knoten den Access Router nach Erhalt des FBAck, hat er nun eine verifizierte NCoA und teilt dem NAR bei Ankunft im neuen Netz mittels Fast Neighbor Advertisement (FNA) seine Anwesenheit mit, woraufhin ihm alle gepufferten Pakete übermittelt werden. Alle weiteren am PAR eintreffenden Pakete werden ohne Pufferung an den mobilen Knoten getunnelt. Zum Senden von Paketen kann der Client auch auf den bestehenden Tunnel zugreifen und Pakete mit seiner alten Care-of-Address (PCoA) über den PAR schicken. Dieser Mechanismus ist besonders wichtig, da der mobile Knoten, obwohl er eine verifizierte NCoA besitzt und somit im neuen Netz IP-fähig ist, bei den Kommunikationspartnern noch kein Binding Update durchgeführt hat und infolgedessen unter der NCoA nicht bekannt ist.

Falls der mobile Knoten das FBU nicht mehr aus seinem ursprünglichen Netz senden kann - z.B. wegen dem Verlust der Verbindung zu seinem PAR - oder kein FBAck erhalten hat und

davon ausgeht, dass das FBU nicht erfolgt war, muss er dies schnellstmöglich im neuen Netz durchführen. Hier bietet es sich an, das FBU in einem Fast-Neighbor -Advertisement-Paket gekapselt zu versenden, damit der NAR zuerst lokal die NCoA aus dem FNA überprüfen kann, bevor er den inneren Teil - das FBU - an den PAR weiterleitet. Falls eine Kollision vorliegt, wird dem Client eine neue Adresse zugewiesen, die er in einem neuen FBU an den PAR sendet. Dieser akzeptiert und leitet die Pakete in der oben beschriebenen Art und Weise an die angegebene Adresse weiter.

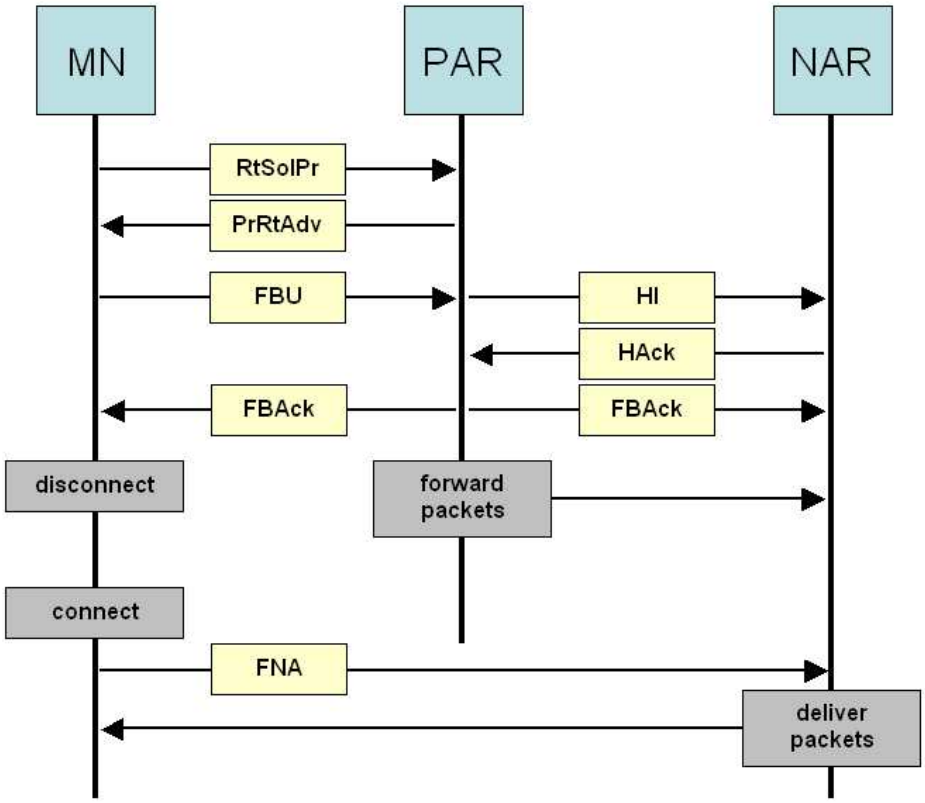

Abbildung 3: Protokollablauf eines Fast Handovers

Ob der mobile Knoten nun das FBU in seinem alten oder neuen Netz gesendet hat und daraufhin in die Netztopologie integriert wurde, macht für die weitere Kommunikation keinen Unterschied. Der Client wird im nächsten Schritt unter Verwendung seiner neuen Care-of-Address die Binding Updates bei seinem Home Agent und seinen Correspondent Nodes durchführen, damit zukünftige Pakete direkt an seine NCoA gesendet werden, das Tunneln der Pakete nicht mehr nötig ist und der Netzwechsel endgültig abgeschlossen ist.

Die vorgestellte Methode ermöglicht durch die Erweiterung von Mobile IPv6 um die sieben neuen Nachrichtentypen (s. Abb. 4) erhebliche Verbesserungen. Die Verzögerung der Adresskonfiguration, die normalerweise bei einem Netzwechsel entstand, kann mittels einem Fast Handovers umgangen werden, indem der mobile Knoten sich schon vor dem Handover die nötigen Informationen besorgt und somit mit einer meist topologisch korrekten IP-Adresse das neue Netz betritt. Auch die beiden involvierten Access Router sind in der Regel über den Wechsel unterrichtet und arbeiten bereits zusammen, damit an den alten Aufenthaltsort des Clients adressierte Pakete an dessen neue Adresse gesendet werden und keine zu großen Verzögerungen entstehen, die aktive Verbindungen stören würden.

3.3 Simultaneous Bindings for Mobile IPv6 Fast Handovers

Dieses Verfahren baut auf der Erweiterung „Fast Handovers for Mobile IPv6“ auf und erweitert es um eine Funktionalität der gleichzeitigen Auslieferung aller für den Client bestimmten Pakete am PAR und NAR.

Wie bereits vorgestellt, verbessert der Einsatz von Fast Handover die Qualität aktiver Verbindungen während eines Handovers. Doch es bliebt die Schwierigkeit den richtigen Zeitpunkt zu wählen, ab dem Pakete nicht mehr vom PAR an die PCoA gesendet, sondern an die NCoA weitergeleitet werden sollen. Das eigentliche Problem besteht in der Ungewissheit, ab wann der mobile Knoten nach erfolgreicher Einleitung eines Handovers wirklich das ursprüngliche Netz verlässt und sich in das neue Netz einbindet. Durch das Versenden des FBU des Clients an den PAR ist die Weiterleitung aktiv, aber dies sagt noch nichts über den tatsächlichen Wechsel auf der Ebene der Access Points aus. Wird die Weiterleitung zu früh eingesetzt, befindet sich der mobile Knoten noch im Ursprungsnetz und bekommt die Datenpakete aktiver Verbindungen nicht mehr zugestellt und kann somit auch keine Antworten oder Bestätigungen senden. Auch wenn die Pakete bei dem NAR gepuffert und nach längerem Warten auf den Client nicht verworfen werden, kann der Empfang, nach erfolgreicher Anmeldung des mobilen Knotens im neuen Netz, zu spät sein.

Da in drahtlosen Netzen eine präzise Vorhersage über den Zeitpunkt des Ab- und Zugangs des mobilen Knotens im Bezug auf die jeweiligen Netze nicht möglich ist und das Gerät den Wechsel des Access Points nicht aktiv steuern kann, setzt man die parallele Auslieferung der Pakete ein. Dadurch wird der mobile Knoten noch für eine maximale Dauer im ursprünglichen Netz bedient und erhält „bis zum letzten Moment“ Pakete aktiver Verbindungen. Pakete die nach seinem Austritt aus dem Netz „ins Leere laufen“ sind aber auch am NAR vorhanden und können ihm dort nach seiner Anmeldung sofort zugestellt werden. Infolgedessen besteht die Zeit, die der Client nicht ansprechbar ist nur aus der Dauer des Handovers auf Schicht 2, welcher sich relativ schnell zwischen 10-100 ms vollzieht.

Wünscht der mobile Knoten den Einsatz des Bicastings - das parallele Versenden aller für den Client bestimmten Paketen an die PCoA und NCoA durch den PAR - kann er dies in seinem FBU dem PAR mitteilen. Hat er keine Zeitdauer spezifiziert wird die voreingestellte Dauer von zwei Sekunden verwendet. Ist zu dem Zeitpunkt der Einleitung des Handoves noch nicht klar in welches Netz sich der Client bewegen wird und es kommen mehrer NCoA in Frage, so kann er auch das n-casting nutzen. Der Ablauf entspricht dem des Bicastings nur werden die eintreffenden Pakete nun an mehrere potentielle NCoAs weitergeleitet.

Ausschlaggebend für den Erhalt der Verbindung ist die Behandlung der ankommenden Pakete durch den NAR für einen ihm bis dahin vielleicht noch nicht bekannten Client. Je nach Art der Verbindung und lokaler Bewertung wird er alle, einen Teil oder keine der ankommenden Pakete puffern. Um direkt das Netz und den NAR zu entlasten und keine unnötige Netzlast zu schaffen, kann der mobile Knoten das Bi-/n-casting nur für bestimmte Datenströme anfordern.

„Simultaneous Bindings for Mobile IPv6 Fast Handovers“ ist eine sinnvolle Erweiterung für Mobile IPv6 und vervollständigt das Verfahren der „Fast Handovers for Mobile IPv6“. Der mobile Knoten kann sich durch die beiden Verbesserungen des Ausgangsprotokolls frei zwischen verschiedenen Netzen bewegen und bestehende Verbindungen zu Kommunikationspartnern bleiben bestehen, sobald der Handover auf der Sicherungsschicht ausreichend schnell vollzogen wird.

4 Zusammenfassung

Nachdem die Ausarbeitung auf die Dringlichkeit der Verbesserung bestehender Kommunikationsprotokolle im Hinblick auf die steigende Mobilität der Nutzer hingewiesen hat und eine Einführung in die Grundlagen von Adresskonfigurationsprotokolle und Mobile IPv6 erfolgte, wurde der Handover - der Wechsel eines Knotens zwischen zwei Netzwerken - als Hauptproblem für den Erhalt aktiver Verbindungen zwischen Kommunikationspartnern identifiziert.

Die drei präsentierten Lösungen „Hierarchical Mobile IPv6 mobility management“, „Fast Handovers for Mobile IPv6“ und „Simultaneous Bindings for Mobile IPv6 Fast Handovers“ lassen sich in zwei Kategorien einteilen. Der erste Ansatz verringert die Verzögerung eines Handovers, indem die Dauer der Binding Updates minimiert wird. Mit dem hierarchischen Aufbau eines Netzes und der Einführung der Mobility Anchor Points (MAP) sind die Binding Updates nur noch lokal und nicht mehr im weit entfernten Heimatnetzwerk und bei den Correspondent Nodes nötig. Die beiden anderen Lösungswege ermöglichen eine schnellere Einbindung und längere Versorgung des mobilen Knotens mit Datenpaketen bei einem Netzwerkwechsel, indem der Client schon vor dem Handover bestimmte Informationen über das Zielnetzwerk in Erfahrung bringt und den beteiligen Access Routern den Wechsel ankündigt.

Schlussendlich lässt sich mit der Kombination von Mobile IPv6 mit allen drei Erweiterungen eine weitere Verbesserung der Ergebnisse erzielen. Voraussetzung hierfür ist die Unterstützung der Protokolle durch alle Geräte. Die entgültige Struktur und der Ablauf eines Handovers kann man Abbildung 5 entnehmen. Dabei ermittelt der mobile Knoten schon frühzeitig über den PAR die Informationen, welche er zur Konfiguration einer neuen CoA benötigt. Das FBU sendet der Client an den MAP, welcher die Adressprüfung in die Wege leitet und das parallele Versenden ankommender Pakete an die PCoA und NCoA übernimmt. Eine Weiterleitung aller Pakete durch den PAR an die NCoA ist somit nicht nötig, da die Aufteilung des Datenstromes schon auf höherer Ebene geschieht. Informiert der mobile Knoten über seine erfolgreiche Ankunft im neuen Netz, werden die Pakete nur noch an die NCoA geschickt.

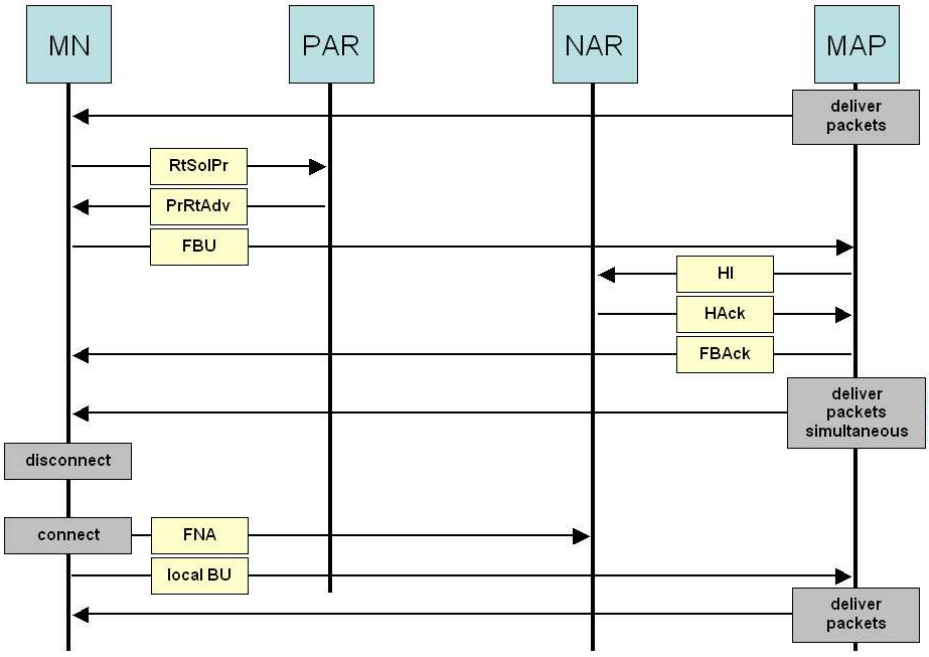

Abbildung 4: Mobile IPv6 mit Erweiterungen

Diese Kombination stellt einen solides Protokoll dar und bietet eine gute Unterstützung von mobilen Knoten. Es profitiert von den jeweiligen Vorteilen der beinhalteten Erweiterungen und gestattet vor allem ausreichenden Schutz vor Verbindungsabbrüchen im Falle eines Handovers. Dadurch wird sich Mobile IPv6 durchsetzen können und „das“ Protokoll in Sachen Mobilität werden.

Literatur

[eal] R. Koodli et al. Fast Handover for Mobile IPv6. draft-ietf-mipshop-mipv6-00.txt (Work in progress).

[H. S] K. El Malki und L. Bellier H. Soliman, C. Castelluccia. Hierarchical MIPv6 mobility management. draft-ietf-mobileip-hmipv6-08.txt (Work in progress).

[Jed] Karim El Jed. Mobile IPv6: Mobilität im zukünftigen Internet.

[K. E] h.Soliman et al K. El Malki. Simultaneous Bindings for Mobile IPv6 Fast Handovers. draft-elmalki-mobileip-bicasting-v6-05.txt (Work in progress).

Abbildungsverzeichnis

Kapazitätsbetrachtungen zu mobilen Ad-hoc-Netzwerken

Moritz Engel

Kurzfassung

Wenn man n identische Knoten, die jeweils W $\frac{bits}{sec}$ übertragen können, optimal auf einer Kreisscheibe mit der Fläche 1 platziert und die Übertragungsreichweite und die Wegewahl ebenfalls optimal wählt, ist das Bit-Distanz-Produkt, welches das drahtlose Netzwerk unter einem Protokoll ohne Interferenzen übertragen kann, $\Theta(W\sqrt{n})$ $\frac{bit-meters}{sec}$. Pro Knoten ergibt sich demnach nur eine Kapazität von $\Theta(\frac{W}{\sqrt{n}})$ $\frac{bit-meters}{sec}$.

Werden die Knoten zufällig platziert und senden mit fester Reichweite, ist der erreichbare Durchsatz jedes Knoten für einen zufällig gewählten Empfänger $\Theta(\frac{W}{\sqrt{n \log n}})$.

Betrachtet man die Netzwerke unter einem Physikalischen Modell erhält man unter Berücksichtigung eines Mindest-Signal-Interferenz-Verhältnisses ähnliche Beschränkungen des Durchsatzes.

1 Einleitung

Die Anzahl mobiler Endgeräte wie PDAs, Laptops, MP3-Player und auch Mobiltelefone, die ein Mensch heute benutzt, wird täglich größer. Dabei kann der Anwender diese allerdings meistens nur separat nutzen. Ein dezentrales Zusammenspiel all dieser beruflich und privat genutzten Komponenten wäre ein gutes Beispiel für ein drahtloses *ad hoc*-Netzwerk. Weitere Beispiele wären das „intelligente Heim“ oder die Vernetzung von Kraftfahrzeugen auf Straßen.

Ein drahtloses *ad hoc*-Netzwerk besteht aus einer beliebigen Anzahl von Knoten, die untereinander über eine drahtlose Verbindung kommunizieren. Im Gegensatz zum Mobilfunk (GSM o.ä.), wo nur der letzte Hop nicht drahtgebunden ist, sind hier alle Verbindungen drahtlos. Außerdem existiert keine zentralisierte Kontrolle der Kommunikation. Damit auch eine Übertragung zwischen zwei nicht direkt erreichbaren Knoten stattfinden kann, müssen alle Knoten Daten anderer Knoten richtig weiterleiten können. Der Umstand, dass keine zentrale Kontrollinstanz existiert und die Knoten sich bewegen können, erschwert die Implementierung eines solchen Netzwerkes im Bezug auf Belegung des Mediums (medium access) und der physikalischen Schicht, die hier kein Gegenstück zum drahtgebundenen Netzwerk wie dem Ethernet oder Mobilfunk hat.

Vor allem die sich in der Zeit verändernde Netzwerktopologie erschwert die Mediumsbelegung und das Routing, welches auch durch Beschränkung der Sendeleistung und die allgemeinen Charakteristika eines drahtlosen Kanals beeinträchtigt wird. Eine Regelung des Zugriffs durch Verfahren wie TDMA oder eine dynamische Frequenzvergabe scheitern an der nicht vorhandenen zentralen Kontrolle. FDMA ist in dichten Netzwerken ineffizient und CDMA aufgrund der Mobilität der Knoten kaum zu implementieren, so dass ein zufälliger Zugriff auf das Medium noch die erfolgversprechenste Methode sein dürfte. Auf der physikalischen Schicht stellt sich vor allem die Frage der Leistungssteuerung der Übertragung. Die Leistung muss einerseits groß genug sein, um den gewünschten Empfänger zu erreichen. Andererseits soll sie so klein wie möglich sein, damit sie nur minimale Interferenzen bei anderen Knoten auslöst.

In dieser Seminararbeit wird anhand der Arbeit von Gupta und Kumar [GuKu99] die theoretisch erreichbare Kapazität in drahtlosen Netzwerken analysiert. Ausführlich wird die beliebige Anordnung der Knoten auf einer definierten Fläche in einer Ebene behandelt.

Auf einer Fläche von $1m^2$ befinden sich n Knoten. Jeder von ihnen kann W $[\frac{bits}{sec}]$ über einen gemeinsamen Kanal übertragen. Solange die Gesamtkapazität ($\sum_{m=1}^{M} W_m = W$) gleich bleibt, hat es keine Auswirkungen, ob der Kanal in M Subkanäle unterteilt wird. Einzelne Pakete werden in einem *multi-hop*-Verfahren von Knoten zu Knoten bis zum Ziel übertragen. Die Knoten speichern, wenn nötig, die Daten zwischen. Solange keine Interferenzen auftreten, können verschiedene Knoten gleichzeitig über den gleichen Subkanal senden.

Im folgenden Kapitel 2 werden die notwendigenigen Definitionen bei beliebiger Verteilung spezifiziert. Es wird beschrieben unter welchen Bedingungen eine drahtlose Übertragung von einem Knoten zum gewünschten Empfänger erfolgreich ist und es werden die Ergebnisse der Kapazitäten vorgestellt. Kapitel 3 gibt die Ergebnisse wieder, wenn die Knoten zufällig platziert wurden. In Kapitel 4 folgt die Herleitung für eine Obergrenze und eine Untergrenze der Kapazität bei beliebig verteilten Knoten. Kapitel 5 zeigt die Bedeutung der Ergebnisse für mobile *ad hoc*-Netzwerke auf. Abschließend fasst Kapitel 6 diese Arbeit zusammen.

2 Beliebige Verteilung der Knoten

Bei beliebiger Verteilung der Knoten sind n Knoten beliebig (arbiträr) auf einer Kreisscheibe mit einer Fläche von 1 in einer Ebene angeordnet. Beliebig bedeutet hier, dass alle Knoten willkürlich nach eigenem Ermessen platziert werden können. Jeder Knoten sendet mit beliebiger Frequenz, beliebiger Reichweite oder Sendeleistung auf beliebigem Weg zu einem beliebigen Empfänger.

Um zu klären, wann eine Übertragung erfolgreich vom nächsten gewünschten Empfänger erhalten wurde, werden zwei Modelle, das Protokoll Modell und das Physikalische Modell, eingeführt. Dafür sei X_i die Lage des Knoten i und geichzeitig der Knoten i selbst.

2.1 Das Protokoll Modell

Angenommen Knoten X_i sendet über den Subkanal m zum Knoten X_j. Dann ist die Übertragung erfolgreich, wenn für alle anderen Knoten X_k, die gleichzeitig über Subkanal m senden, gilt:

$$|X_k - X_j| \geq (1 + \Delta)|X_i - X_j|. \tag{3}$$

Die Größe $\Delta \geq 0$ erlaubt Ungenauigkeiten in der erreichten Reichweite der Übertragungen und hilft Situationen abzubilden, in denen ein Nachbarknoten daran gehindert werden soll auf dem gleichen Subkanal m zu senden. Gleichung (3) besagt, dass eine Übertragung erfolgreich ist, wenn die Entfernung zwischen Sender X_i und Empfänger X_j kleiner als die Entfernung zwischen den übrigen Sendern X_k und dem ursprünglichen Empfänger X_j ist. Der Faktor $(1+\Delta)$ erhöht die Entfernung der Knoten der gewünschten Übertragung um die eben beschriebene Schutzzone.

2.2 Das Physikalische Modell

Sei nun $\{X_k; k \in \tau\}$ die Teilmenge an Knoten, die gleichzeitig über den gleichen Subkanal senden und P_k die Sendeleistung von Knoten X_k, $k \in \tau$. Dann ist eine Übertragung vom Sender X_i zum Empfänger X_j erfolgreich wenn gilt:

$$\frac{\frac{P_i}{|X_i - X_j|^\alpha}}{N + \sum\limits_{\substack{k \in \tau \\ k \neq i}} \frac{P_k}{|X_k - X_j|^\alpha}} \geq \beta. \tag{4}$$

Das bedeutet, dass ein Signal-Störinterferenz-Verhältnis von mindestens β für eine erfolgreiche Übermittlung nötig ist. Wobei N das Umgebungsrauschen ist und bei einer Entfernung von r eine Signalverschlechterung von $\frac{1}{r^\alpha}$ angenommen wird. Dass $\alpha > 2$ gilt, sollte in nicht direkter Nachbarschaft zum Sender üblich sein.

2.3 Transportkapazität

Um die Transportkapazität des Netzwerkes zu messen, wird der Begriff *bit-meter* wie folgt definiert: das Netzwerk hat ein *bit-meter* transportiert, wenn ein Bit einen Meter in Richtung seines Ziels transportiert wurde. Für den Fall, dass ein Bit von einem Sender zu mehreren Empfängern gesendet werden soll (multicast oder broadcast), werden diese Wege nicht mehrfach gezählt.

Die folgenden Resultate müssen mit $\sqrt{A}$ skaliert werden, wenn die betrachtete Fläche nicht $1m^2$ entspricht.

Ergebnis unter Protokoll Modell:

Die Transportkapazität bei beliebiger Verteilung von n Knoten ist $\Theta(W\sqrt{n})$ $\frac{bit-meters}{sec}$ bei optimaler Platzierung der Knoten, optimaler Reichweiteneinstellung jeder Übertragung und optimaler Wegewahl (traffic pattern).

Eine Obergrenze für alle möglichen räumlichen und zeitlichen Planungen (Schedules) ist $\sqrt{\frac{8}{\pi}}\frac{W}{\Delta}\sqrt{n}$ $\frac{bit-meters}{sec}$. Bei angemessener Wahl der Parameter (Verteilung der Knoten, Wegewahl, Reichweite, Scheduling) erhält man so noch $\frac{W}{1+2\Delta}\frac{n}{\sqrt{n}+\sqrt{8\pi}}$ $\frac{bit-meters}{sec}$. (Für n ist ein Vielfaches von vier.)

Teilt man diese Transportkapazität nun auf die vorhandenen n Knoten auf, enthält jeder Knoten $\Theta(\frac{W}{\sqrt{n}})$ $\frac{bit-meters}{sec}$. Oder auch $\Theta(\frac{W}{\sqrt{n}})$ $\frac{bits}{sec}$ Durchsatzkapazität, wenn der Abstand zwischen jedem Sender-Empfänger-Paar ungefähr 1 Meter entspricht.

Ergebnis unter dem Physikalischen Modell:

Maximal möglich sind $cW\sqrt{n}$ $\frac{bit-meters}{sec}$. Bei angemessener Wahl der Parameter im Netzwerk sind $\frac{1}{(16\beta(2^{\frac{\alpha}{2}}+\frac{6\alpha-2}{\alpha-2})^{\frac{1}{\alpha}}}\frac{Wn}{\sqrt{n}+\sqrt{8\pi}}$ $\frac{bit-meters}{sec}$ realisierbar; wobei $\frac{1}{\sqrt{\pi}}(\frac{2\beta+2}{\beta})^{\frac{1}{\alpha}}Wn\frac{\alpha-1}{\alpha}$ $\frac{bit-meters}{sec}$ eine Obergrenze der Transportkapazität darstellt.

Wichtig anzumerken ist, dass im Spezialfall, $\frac{P_{max}}{P_{min}} > \beta$, $\sqrt{\frac{8}{\pi}}\frac{W}{(\frac{\beta P_{min}}{P_{max}})^{\frac{1}{\alpha}}-1}\sqrt{n}$ $\frac{bit-meters}{sec}$ eine Obergrenze ist. Deshalb sollte man annehmen können, dass $\Theta(W\sqrt{n})$ wirklich eine Obergrenze der Transportkapazität ist. Außerdem ist auffällig, dass die Transportkapazität mit zunehmendem α ebenfalls steigt. Dass also mit einer mit der Distanz schnell fallenden Signalstärke, die Tranportkapazität ansteigt.

3 Zufällige Verteilung der Knoten

In diesem Szenario sind die n Knoten zufällig auf einer $1m^2$ großen Kreisscheibe oder Oberfläche einer Kugel angeordnet. Es wird auch die Verteilung auf einer Kugeloberfläche betrachtet, da so die Effekte an Kanten in einer Ebene ausgeschlossen werden können. Jeder Knoten sendet $\lambda(n)$ $[\frac{bits}{sec}]$ zu einem zufällig gewählten Empfänger. Dieser ist der zu einem zufällig gewählten Punkt der nächste Knoten. Damit beträgt der Abstand zwischen Sender und Empfänger im Durchschnitt 1 Meter. Zufällig bedeutet hier, dass die Wahl unabhängig und gleichverteilt ist. Alle Übertragungen aller Knoten werden mit der gleichen Reichweite oder Leistung gesendet.

Wie bei beliebiger Verteilung werden zur Erfolgsmessung wieder das Protokoll Modell und das Physikalische Modell betrachtet.

3.1 Das Protokoll Modell

Wenn Knoten X_i über Subkanal m zu Knoten X_j sendet, ist die Übertragung erfolgreich, wenn gilt:

$$|X_i - X_j| \leq r \tag{5}$$
$$|X_k - X_j| \geq (1+\Delta)r \tag{6}$$

Die Distanz zwischen Sender und Empfänger darf nicht größer als die Reichweite r sein. Die Distanz zwischen allen anderen gleichzeitig auf dem gleichen Subkanal sendenden Knoten und dem ursprünglichen Empfänger muss kleiner sein als die um den Schutzfaktor $(1+\Delta)$ erhöhte Reichweite r.

3.2 Das Physikalische Modell

Sei wieder $\{X_k; k \in \tau\}$ die Teilmenge an Knoten, die gleichzeitig über den gleichen Subkanal senden und die Sendeleistung P für alle Übertragungen und Knoten gleich. Dann ist eine Übertragung von Knoten X_i, $i \in \tau$ an Knoten X_j erfolgreich, wenn gilt:

$$\frac{\frac{P}{|X_i - X_j|^\alpha}}{N + \sum\limits_{\substack{k \in \tau \\ k \neq i}} \frac{P}{|X_k - X_j|^\alpha}} \geq \beta. \tag{7}$$

3.3 Durchsatzkapazität

Der Begriff Durchsatz gibt wie üblich die Anzahl der Bits pro Sekunde an, die von jedem Sender zum zugehörigen Empfänger übertragen werden.

Ergebnis:

Die Größenordnung der Durchsatzkapazität unter Protokoll Modell bei einer zufälligen Verteilung von n Knoten auf einer Kugeloberfläche von $1m^2$ ist $\Theta(\frac{W}{\sqrt{n \log n}})$ $\frac{bits}{sec}$ pro Knoten unter Protokoll Modell. Es lässt sich zeigen, dass für Konstanten c'' und c''' eine Durchsatzkapazität von $\frac{c''W}{(1+\Delta)^2\sqrt{n \log n}}$ $\frac{bits}{sec}$ möglich, eine Kapazität von $\frac{c'''W}{\Delta^2\sqrt{n \log n}}$ $\frac{bits}{sec}$ aber nicht möglich ist.

Unter dem Physikalischen Modell ist für angemessene Konstanten c und c' ein Durchsatz von $\frac{cW}{\sqrt{n \log n}}$ $\frac{bits}{sec}$ möglich, während $\frac{cW}{\sqrt{n}}$ $\frac{bits}{sec}$ unmöglich sind. Es lässt sich für die Konstanten c'' und c''' ableiten, dass eine Kapazität von $\frac{c''}{(2(c'''\beta(3+\frac{1}{\alpha-1}+\frac{2}{\alpha-2}))^{\frac{1}{\alpha}}-1)^2}\frac{W}{\sqrt{n \log n}}$ $\frac{bits}{sec}$ machbar ist.

4 Herleitung: Beliebige Verteilung der Knoten

Es gelten zusätzlich zu den Bedingungen für beliebig verteilte Knoten (Seite 106) weitere Annahmen:

(A.i) n Knoten sind beliebig auf einer Kreisscheibe mit Fläche 1 platziert.

(A.ii) Das Netzwerk transportiert λnT Bits in T Sekunden. Wobei λ der Durchsatz pro Knoten, n die Anzahl der Knoten und T die Zeit ist.

(A.iii) Die durchschnittliche Distanz vom Sender zum Empfänger für ein Bit ist $\bar{L}$. Daraus und aus Annahme (A.ii) folgt, dass eine Transportkapazität von $\lambda n\bar{L}$ erreicht werden kann.

(A.iv) Jeder Knoten kann über einen von M Subkanälen mit jeweils einer Kapazitäten W_m $\frac{bits}{sec}$ übertragen. Es gelten: $1 \leq m \leq M$ und $\sum_{m=1}^{M} W_m = W$.

(A.v) Die Übertragungen sind in synchronisierte Zeitschlitze mit Länge τ Sekunden unterteilt. (Diese Annahme könnte auch ausgelassen werden, vereinfacht aber den Beweis.)

(A.vi) Zusätzlich zu Restriktion (4) unter dem Physikalischen Modell, kann entweder die Restriktion (3) oder die folgende Alternative gewählt werden: Wenn Knoten X_i über einen bestimmten Subkanal in einem bestimmten Zeitschlitz an Knoten X_j sendet und die Entfernung zwischen beiden r beträgt, dann kann kein anderer Knoten im Umkreis Δr von X_j auf dem gleichen Subkanal und im gleichen Zeitschlitz senden.

4.1 Obergrenze der Transportkapazität

Satz:

- Die Transportkapazität unter Protokoll Modell ist wie folgt beschränkt:

$$\lambda n\bar{L} \leq \sqrt{\frac{8}{\pi}}\frac{1}{\Delta}W\sqrt{n} \qquad \frac{bit-meters}{sec} \tag{8}$$

- Unter dem Physikalischen Modell folgt:

$$\lambda n\bar{L} \leq (\frac{2\beta+2}{\beta})^{\frac{1}{\alpha}}\frac{1}{\sqrt{\pi}}Wn^{\frac{\alpha-1}{\alpha}} \qquad \frac{bit-meters}{sec} \tag{9}$$

- Wenn das Verhältniss zwischen maximaler und minimaler Sendeleistung $\frac{P_{max}}{P_{min}}$ immer größer als β ist, gilt sogar:

$$\lambda n\bar{L} \leq \sqrt{\frac{8}{\pi}}\frac{1}{(\frac{\beta P_{min}}{P_{max}})^{\frac{1}{\alpha}}-1}W\sqrt{n} \qquad \frac{bit-meters}{sec} \tag{10}$$

- Wenn die Grundfläche nicht $1m^2$ sondern A m^2 beträgt, müssen alle Grenzen durch $\sqrt{A}$ angepasst werden.

Beweis:

Jedes Bit b, wobei $1 \leq b \leq \lambda nT$, gelangt in einer Folge von $h(b)$ Hops vom Sender zum Empfänger. r_b^hist dann die überwundene Distanz im h-ten Hop von Bit b. Hieraus und aus Annahme (A.iii) ergibt sich:

$$\sum_{b=1}^{\lambda nT}\sum_{h=1}^{h(b)} r_b^h \geq \lambda nT\bar{L} \tag{11}$$

Der zurückgelegte Weg eines Bits aufsummiert über alle Hops und alle Bits muss mindestens so groß wie die Transportkapazität in $[\frac{bit-meters}{sec}]$ in der Zeit T sein.

Wenn man davon ausgeht, dass in jedem Zeitschlitz höchstens die Hälfte aller Knoten $(\frac{n}{2})$ gleichzeitig senden können, ergibt sich für jeden Subkanal m und jeden Zeitschlitz s:

$$\sum_{b=1}^{\lambda nT} \sum_{h=1}^{h(b)} 1(\text{Der } h\text{-te Hop von Bit } b \text{ über Subkanal } m \text{ in Schlitz } s) \leq \frac{W_m \tau n}{2} \tag{12}$$

Die Anzahl der Bits, die über einen Subkanal im Zeitschlitz τ übertragen werden, aufsummiert über alle Hops und alle Bits kann höchstens so groß wie die Übertragungsrate mal $\frac{n}{2}$ Knoten mal Zeit τ sein.

Über alle Subkanäle und alle Zeitschlitze aufsummiert und der logischen Annahme, dass in T Sekunden nicht mehr Zeitschlitze als $\frac{T}{\tau}$ möglich sind, erhält man:

$$H := \sum_{b=1}^{\lambda nT} h(b) \leq \frac{WTn}{2} \tag{13}$$

Betrachte man nun die Annahmen für eine erfolgreiche Übertragung unter dem Protokoll Modell (3). Dann ergeben sich, wenn erstens Knoten X_i zu X_j (14) und zweitens Knoten X_k zu X_l (15) über den gleichen Subkanal und zur gleichen Zeit senden, zwei Gleichungen.

$$|X_k - X_j| \geq (1+\Delta)|X_i - X_j| \tag{14}$$
$$|X_i - X_l| \geq (1+\Delta)|X_k - X_l| \tag{15}$$

Außerdem erhält man durch umstellen der Dreiecksungleichung $|X_k - X_j| \leq |X_j - X_l| + |X_l - X_k|$:

$$|X_j - X_l| \geq |X_k - X_j| - |X_l - X_k| \tag{16}$$

Und durch einsetzen von (14) in (16):

$$|X_j - X_l| \geq (1+\Delta)|X_i - X_j| - |X_l - X_k| \tag{17}$$

Durch umstellen der Dreicksungleichung $|X_i - X_l| \leq |X_l - X_j| + |X_j - X_i|$ und einsetzen von (15) erhält man auf gleiche Weise:

$$|X_l - X_j| \geq (1+\Delta)|X_k - X_l| - |X_j - X_i| \tag{18}$$

Addiert man nun (17) und (18) resultiert daraus:

$$|X_l - X_j| \geq \frac{\Delta}{2}(|X_k - X_l| + |X_i - X_j|) \tag{19}$$

Das bedeutet, dass zwei Kreise mit Mittelpunkt im jedem der beiden Empfänger und Radius ($\frac{\Delta}{2}$*Länge des jeweiligen Hops) disjunkt sein müssen. Selbiges folgt aber auch schon aus Annahme (A.vi). Erlaubt man auch Randeffekte, wo Knoten am Rand der Fläche platziert sind, und berücksichtigt, dass eine Reichweite größer als der Durchmesser der Scheibe unnütz ist, sieht man, dass mindestens ein viertel dieser Kreise die Gesamtfläche überdecken. Außerdem können in Zeitschlitz s und über Subkanal m höchstens $W_m\tau$ Bits übertragen werden. Daraus folgt:

$$\sum_{b=1}^{\lambda nT} \sum_{h=1}^{h(b)} 1(\text{Der } h\text{-te Hop von Bit } b \text{ ist über Subkanal } m \text{ in Schlitz } s) \frac{1}{4}\pi(\frac{\Delta}{2} r_h^b)^2 \leq W_m \tau \tag{20}$$

Über alle Subkanäle und alle Zeitschlitze aufsummiert ergibt sich:

$$\sum_{b=1}^{\lambda nT}\sum_{h=1}^{h(b)}\frac{\pi\Delta^2}{16}(r_h^b)^2 \leq WT \tag{21}$$

Unter Berücksichtigung von (13) gilt dann:

$$\sum_{b=1}^{\lambda nT}\sum_{h=1}^{h(b)}\frac{1}{H}(r_h^b)^2 \leq \frac{16WT}{\pi\Delta^2 H} \tag{22}$$

Weil eine quadratische Funktion konvex ist, muss gelten:

$$(\sum_{b=1}^{\lambda nT}\sum_{h=1}^{h(b)}\frac{1}{H}r_h^b)^2 \leq \sum_{b=1}^{\lambda nT}\sum_{h=1}^{h(b)}\frac{1}{H}(r_h^b)^2 \tag{23}$$

Kombiniert man (22) und (23) erhält man:

$$\sum_{b=1}^{\lambda nT}\sum_{h=1}^{h(b)}(r_h^b) \leq \sqrt{\frac{16WTH}{\pi\Delta^2}} \tag{24}$$

Setzt man nun (11) in (24):

$$\lambda nT\bar{L} \leq \sqrt{\frac{16WTH}{\pi\Delta^2}} \tag{25}$$

Und schließlich (13) in (25) so erhält man das Endergebnis:

$$\lambda n\bar{L} \leq \sqrt{\frac{8}{\pi}}\frac{1}{\Delta}W\sqrt{n} \qquad \frac{bit-meters}{sec} \tag{26}$$

Betrachte man nun die Annahmen für eine erfolgreiche Übertragung unter dem Physikalischen Modell (4). Knoten X_i sendet über Subkanal m mit Sendeleistung P_i an Knoten $X_{j(i)}$. τ sei wieder die Menge aller Knoten, die zur gleichen Zeit und über den gleichen Subkanal senden. Berücksichtigt man in (4) das Signal von X_i rechts im Nenner des Signal-Störinterferenz-Verhältnisses und summiert links dann über alle sendenden Knoten, ergibt sich:

$$\frac{\frac{P_i}{|X_i-X_{j(i)}|^\alpha}}{N+\sum\limits_{k\in\tau}\frac{P_k}{|X_k-X_{j(i)}|^\alpha}} \geq \frac{\beta}{\beta+1} \tag{27}$$

Stellt man diese um zu:

$$|X_i-X_{j(i)}|^\alpha \leq \frac{\beta+1}{\beta}\frac{P_i}{N+\sum\limits_{k\in\tau}\frac{P_k}{|X_k-X_{j(i)}|^\alpha}} \tag{28}$$

und geht davon aus, dass der Abstand vom Empfänger $X_{j(i)}$ zu allen anderen Sendern X_k immer kleiner als $\frac{2}{\sqrt{\pi}}$ sein muss (Fläche $= \pi r^2 = \pi(\frac{d}{2})^2 = \pi(\frac{\frac{2}{\sqrt{\pi}}}{2})^2 = 1$) ergibt sich:

$$|X_i-X_{j(i)}|^\alpha \leq \frac{\beta+1}{\beta}\frac{P_i}{N+(\frac{\pi}{4})^{\frac{\alpha}{2}}\sum\limits_{k\in\tau}P_k} \tag{29}$$

Summiert über alle Sender-Empfänger-Paare erhält man:

$$\sum_{i\in\tau}|X_i-X_{j(i)}|^\alpha \leq \frac{\beta+1}{\beta}\frac{\sum\limits_{i\in\tau}P_i}{N+(\frac{\pi}{4})^{\frac{\alpha}{2}}\sum\limits_{k\in\tau}P_k} \tag{30}$$

Zusammengefasst gilt:

$$\sum_{i\in\tau} |X_i - X_{j(i)}|^\alpha \leq (\frac{4}{\pi})^{\frac{\alpha}{2}} \frac{\beta+1}{\beta} \tag{31}$$

Summiert über alle Subkanäle und alle Zeitschlitze ergibt endlich:

$$\sum_{b=1}^{\lambda nT} \sum_{h=1}^{h(b)} (r_h^b)^\alpha \leq 2^\alpha \pi^{\frac{\alpha}{2}} \frac{\beta+1}{\beta} WT \tag{32}$$

Wie die quadratische Funktion r^2 unter dem Protokoll Modell (23) ist auch r^α konvex. Somit verläuft der Rest des Beweises ab hier parallel zum vorherigen. Als Obergrenze der Transportkapazität unter dem Physikalischen Modell erhält man deshalb:

$$\lambda n \bar{L} \leq (\frac{2\beta+2}{\beta})^{\frac{1}{\alpha}} \frac{1}{\sqrt{\pi}} W n^{\frac{\alpha-1}{\alpha}} \qquad \frac{bit-meters}{sec} \tag{33}$$

Für den Spezialfall, dass $\frac{P_{max}}{P_{min}} \leq \beta$ ist, betrachten wir die Bedingung für erfolgreiche Übertragung unter dem Physikalischen Modell (4). Sendet Knoten X_i zu X_j und Knoten X_k gleichzeitig über den gleichen Subkanal zu X_l kann folgendes Verhältnis abgeleitet werden (N weggelassen - nach oben abgeschätzt):

$$\frac{\frac{P_i}{|X_i - X_j|^\alpha}}{\frac{P_k}{|X_k - X_j|^\alpha}} \geq \beta \tag{34}$$

Umgestellt:

$$|X_k - X_j| \geq (\frac{\beta P_{min}}{P_{max}})^{\frac{1}{\alpha}} |X_i - X_j| \tag{35}$$

Gleichzeitig gilt aber auch (3):

$$|X_k - X_j| \geq (1+\Delta)|X_i - X_j|$$

Daraus folgt direkt, dass das Protokoll Modell das selbe Ergebnis wie das Physikalische Modell liefert, wenn $\Delta := (\frac{\beta P_{min}}{P_{max}})^{\frac{1}{\alpha}} - 1$ definiert ist.

4.2 Konstruktive Untergrenze der Tranportkapazität

Es soll jetzt gezeigt werden, dass die Größenordnung der Obergrenzen, wie sie im letzten Kapitel hergeleitet wurden, schon sehr genau sind. Dafür wird ein Szenario entworfen, indem eben diese Größenordnungen der Transportkapazität erreicht werden.

Satz:

Es existiert eine Verteilung von n Knoten und eine Festlegung der Wegewahl, so dass im Netzwerk unter Protokoll Modell $\frac{1}{1+2\Delta} \frac{Wn}{\sqrt{n}+\sqrt{8\pi}} \frac{bit-meters}{sec}$ und unter dem Physikalischen Modell $\frac{1}{(16\beta(2^{\frac{\alpha}{2}} + \frac{6^\alpha - 2}{\alpha-2}))^{\frac{1}{\alpha}}} \frac{Wn}{\sqrt{n}+\sqrt{8\pi}} \frac{bit-meters}{sec}$ erreichbar sind. (Für n ist ein Vielfaches von 4.)

Beweis:

Unter Protokoll Modell: Die Reichweite r jedes Knoten sei definiert durch $r := \frac{1}{1+2\Delta} \frac{1}{\sqrt{\frac{n}{4}}+\sqrt{2\pi}}$. Die Fläche beträgt wie bisher $1m^2$; der Radius einer Kreisscheibe betrage demnach zum Beispiel $\frac{1}{\sqrt{\pi}}$. Ausgehend vom Mittelpunkt der Scheibe als Ursprung platziert man Sender an den Stellen $(j(1+2\Delta)r \pm \Delta r, k(1+2\Delta)r)$ und $(j(1+2\Delta)r, k(1+2\Delta)r \pm \Delta r)$, wenn $|j+k|$ gerade ist. Empfänger platziert man an den Stellen $(j(1+2\Delta)r \pm \Delta r, k(1+2\Delta)r)$ und $(j(1+2\Delta)r, k(1+$

$2\Delta)r \pm \Delta r)$, wenn $|j + k|$ ungerade ist. Jeder Sender kann dann zum mit einer Distanz von r nächstgelegenen Empfänger ohne Interferenz durch andere Sender-Empfänger-Paare übertragen. Dementsprechend gibt es $\frac{n}{2}$ Sender-Empfänger-Paare und auch $\frac{n}{2}$ gleichzeitige Übertragungen mit Reichweite r und W $\frac{bits}{sec}$. Damit ergibt sich eine Transportkapazität von $\frac{n}{2} * W * r = \frac{1}{1+2\Delta} \frac{Wn}{\sqrt{n}+\sqrt{8\pi}}$ $\frac{bit-meters}{sec}$.

Unter dem Physikalischen Modell: Eine Berechnung des Signal-Interferenz-Verhältnisses zeigt, dass es für alle Empfänger nach unten mit $\frac{(1+2\Delta)^\alpha}{16(2^{\frac{\alpha}{2}}+\frac{6^{\alpha-2}}{\alpha-2})}$ begrenzt ist. Der Ergebnis erhält man, wenn man Δ so wählt, dass diese Untergrenze gleich β ist:

$$\frac{1}{(16\beta(2^{\frac{\alpha}{2}} + \frac{6^{\alpha-2}}{\alpha-2}))^{\frac{1}{\alpha}}} \frac{Wn}{\sqrt{n} + \sqrt{8\pi}} \qquad \frac{bit - meters}{sec}$$

Satz:

Für ein Netzwerk mit wenigen Knoten kann man eine Verteilung und eine Wegewahl der Übertragung so wählen, dass folgende Kapazitäten mindestens erreichbar sind:

$\frac{2W}{\sqrt{\pi}}$	$\frac{bit-meters}{sec}$	für n ≥ 2
$\frac{4W}{\sqrt{\pi}(1+\Delta)}$	$\frac{bit-meters}{sec}$	für n ≥ 8
$\frac{W}{1+2\Delta} \frac{n}{\sqrt{n}+\sqrt{8\pi}}$	$\frac{bit-meters}{sec}$	für n $= 2, 3, ..., 21$
$\frac{W}{1+2\Delta} \frac{4\lfloor \frac{n}{4} \rfloor}{\sqrt{4\lfloor \frac{n}{4} \rfloor} + \sqrt{8\pi}}$	$\frac{bit-meters}{sec}$	für alle n

Beweis: Mit mindestens 2 Knoten sind leicht verständlich $\frac{2W}{\sqrt{\pi}}$ $\frac{bit-meters}{sec}$ im Netzwerk erreichbar, wenn man zwei Knoten an zwei diametrisch gegenüberliegende Stellen positioniert. Bei mindestens 8 Knoten können 4 Sender an vom Mittelpunkt der Fläche gegenüberliegende Punkte platziert werden. Jeder dieser Sender kann dann über eine Entfernung von $\frac{1}{\sqrt{\pi}(2+2\Delta)}$ in Richtung des Mittelpunktes zu seinem Empfänger übertragen. Die weiteren Grenzen folgen dann jeweils aus den vorangegangenen.

5 Bedeutung der Ergebnisse

Die hergeleiteten Ergebnisse beruhen alle auf der Annahme eines perfekten Scheduling und Routing, da der Übertragungsbedarf und auch die Standorte aller Knoten bekannt sind. So können jegliche zeitlich oder räumlich bedingte Kollisionen ausgeschlossen werden. Außerdem sind die Knoten nicht mobil. Sollten sich die Knoten bewegen oder ein Teil der für das Scheduling und Routing notwendigen Information nicht zur Verfügung stehen, sollte man annehmen, dass die Kapazität eines solchen Netzwerkes geringer ausfallen muss.

Es ergeben sich verschiedene Hinweise für das Design von Netzwerken: Anstatt riesige Netze für sehr viele Nutzer aufzubauen, in denen der Durchsatz mit der Anzahl der Nutzer dramatisch abfällt, sollte man eher versuchen, viele unabhängige Netzwerke für kleine Anzahlen von Nutzern zu entwerfen. Ein weitere mögliche Chance wäre, dass jeder Knoten jeweils nur mit seinem direkt benachbarten Knoten kommuniziert, was zum Beispiel in „smart homes“ leicht umgesetzt werden könnte. So würde die Kapazität nicht mit steigendem n abfallen.

Beachtet man die Kapazitätsgrenzen unter dem Physikalischen Modell, sollte man außerdem berücksichtigen, dass sich mit schneller Abnahme der Signalstärke gegenüber der Distanz die

Transport- und auch Durchsatzkapazität steigern lässt. Zum Beispiel könnte man ein Netzwerk konstruieren, in dem zu n Knoten zusätzlich m Knoten mit ausschließlich Relaisfunktion zufällig platziert werden. Man könnte dadurch zwar die Signalstärken absenken, um einen nennenswerten Kapazitätsgewinn zu erreichen müsste aber eine sehr große Anzahl solcher Relayknoten verwendet werden. Um zum Beispiel bei n = 100 Knoten fünf mal so viel Durchsatz zu erreichen, müssten man mindestens 4476 Relayknoten platzieren. Dies folgt aus der Größenordnung des Durchsatzes in diesem Szenario von $\Theta(\frac{(n+m)W}{n\sqrt{(n+m)\log(n+m)}})$. kn zusätzliche Relayknoten würden nur einen Anstieg um weniger als $\sqrt{k+1}$ hervorbringen.

Eine anderweitige Möglichkeit um das Kapazitätsproblem von drahtlosen Netzen zu lösen, wäre die gleiche Vorgehensweise wie in Mobilfunknetzen: man verbindet die wenigen Basisstationen drahtgebunden. Wenn man diese Verbindung dann aber doch auch drahtlos gestalten will, sollte man beachten, dass dieses drahtlose Netzwerk den gleichen hergeleiteten Kapazitätsbeschränkungen unterliegt; dass bei b Basisstationen auch nur ein Durchsatz von $\Theta(\frac{1}{\sqrt{b\log b}})$ möglich ist.

5.1 Kompromiss zwischen Relaying-Last und Interferenzen

Warum fällt der Durchsatz überhaupt, wenn die Anzahl der Knoten steigt? Dazu betrachte man folgenden Fall: Die durchschnittliche Distanz, die ein Paket zurücklegen muss sei $\bar{L}$ und $r(n)$ die gemeinsame Reichweite aller Übertragungen. Dann ist im Durchschnitt die Anzahl der Hops eines Pakets nicht kleiner als $\frac{\bar{L}}{r(n)}$. Daraus folgt, dass jeder Knoten mindestens $\frac{L\bar{\lambda}(n)}{r(n)}$ $\frac{bits}{sec}$ an Paketverkehr für andere Knoten erzeugt. Damit ist für insgesamt n Knoten der gesamte Verkehr nicht kleiner als $\frac{Ln\bar{\lambda}(n)}{r(n)}$ $\frac{bits}{sec}$. Dieser muss aber von n Knoten mit jeweils W $\frac{bits}{sec}$ geleistet werden. Es gilt also: $\frac{Ln\bar{\lambda}(n)}{r(n)} \leq nW$.

Eine Obergrenze für den Durchsatz wäre deshalb $\lambda(n) \leq \frac{Wr(n)}{\bar{L}}$. Da die Grenze hier linear in $r(n)$ wächst, sollte man meinen, dass durch Erhöhung der Reichweite der Durchsatz ansteigt und jeder einzelne Knoten durch die reduzierte Anzahl an Hops weniger Pakete der anderen Knoten weiterleiten müsste. Durch Steigerung der Reichweite erhöht sich allerdings gleichzeitig die Zeit, die ein Sender warten muss, um Kollisionen mit anderen Sendern zu verhindern und somit verringert sich die Kapazität. Tatsächlich steigen diese Einbußen quadratisch mit der Reichweite $r(n)$. Dementsprechend muss ein Kompromiss zwischen Reduzierung der Relaying-Last der Knoten und dem Anstieg der Interferenzen gefunden werden.

Da der Kapazitätsverlust durch erhöhte Interferenzen, also durch Erhöhung der Reichweite, (quadratisch in $r(n)$) den Gewinn (linear in $r(n)$) überwiegt, muss man die Reichweite $r(n)$ so klein wie möglich halten. Nach unten ist $r(n)$ dadurch beschränkt, dass bei zu niedriger Reichweite die Netzwerkverbindung verloren geht. Grundsätzlich lässt sich festhalten, dass der Grund für die Begrenzung der Kapazität im wesentlichen ist, dass ein Knoten sich ein und desselben Kanal mit anderen benachbarten Knoten teilen muss.

6 Zusammenfassung und Ausblick

Es wurde unter einem Modell ohne Interferenzen hergeleitet, das wenn n Knoten optimal auf einer Kreisscheibe der Fläche 1 platziert sind und ihre Reichweiten ebenfalls optimal eingestellt sind, ein drahtloses Netzwerk insgesamt nicht mehr als $\Theta(W\sqrt{n})$ $\frac{bit-meters}{sec}$ transportieren kann. Oder anders ausgedrückt lässt sich über eine Entfernung von einem Meter kein Durchsatz erreichen, der größer als $\Theta(\frac{W}{\sqrt{n}})$ $\frac{bit}{sec}$ zu jedem Knoten ist. Sind die Knoten zufällig

platziert, egal ob auf einer Kreisscheibe oder einer Kugeloberfläche, ist die Durchsatzkapazität des gesamten Netzwerkes $\Theta(\frac{W}{\sqrt{n \log n}})$ $\frac{bit}{sec}$. Den Kanal in einzelne Subkanäle aufzuteilen, ändert diese Ergebnisse nicht.

Als Konsequenz sollte man beim Design neuer drahtloser Netzwerke eher auf viele Netze mit geringerer Anzahl an Knoten zurückgreifen. Da die Begrenzung der Kapazitäten der Netzwerke im wesentlichen durch die Interferenzen bei großen Reichweiten der Knoten bedingt sind, sollten Netzwerke mit kurzen Sender-Empfänger-Entfernungen auch mit einer großen Anzahl an Knoten möglich sein, solange die Knoten nur mit benachbarten Knoten und nicht weit entfernten Knoten kommunizieren müssen.

Nicht beachtet wurden eventuelle Kapazitätsverschlechterungen durch Mobilität der Knoten, durch Koordination des Mediumzugriffs oder verschiedene Routing-Algorithmen. Unter diesen zusätzlichen Bedingungen müssten die Kapazitäten aber geringer ausfallen. Auch wurden Verzögerungen (Delay) nicht weiter betrachtet. Grossglauser und Tse [GrTs01] können aber zeigen, dass die Durchsatzkapazität bei mobilen drahtlosen Netzwerken gegenüber den Ergebnissen hier gesteigert werden kann, wenn gerade Verzögerungen für die zu nutzenden Anwendungen keine wesentliche Rolle spielen. Sie zeigen, dass langfristig der durchschnittliche Durchsatz vom Sender zum Empfänger konstant gehalten werden kann, obwohl die Anzahl der Knoten ansteigt.

In einer späteren Arbeit untersuchen Gupta, Gray und Kumar [GuGK01] anhand eines Experimentes die gefundenen Resultate. Sie messen in einem üblichen IEEE 802.11 WLan - Netzwerk den Durchsatz bei unterschiedlicher Anzahl von Knoten (1 - 12 Laptops). Der Durchsatz pro Knoten beträgt nur $\frac{c}{n^{1,68}}$ $\frac{bits}{sec}$. Der Durchsatz, wie auch die Anzahl der Knoten logarithmisch aufgetragen zeigt eine fast lineare Beziehung mit einer Steigung von $-1,68$.

In der vorhandenen Hardware und den heutigen Protokollen ist also noch genug Potential vorhanden, den Durchsatz an das theoretische Maximum anzunähern, was man im Falle von IEEE 802.11 an den verschiedensten proprietären Erweiterungen, die zur Zeit veröffentlicht werden, sehen kann.

Literatur

[GrTs01] Matthias Grossglauser und David N. C. Tse. Mobility Increases the Capacity of Ad-hoc Wireless Networks. In *INFOCOM*, 2001, S. 1360–1369.

[GuGK01] P. Gupta, R. Gray und P. Kumar. An experimental scaling law for ad hoc networks, 2001.

[GuKu99] P. Gupta und P. Kumar. Capacity of wireless networks, 1999.

Unterstützung der Dienstsuche in mobilen Ad-hoc-Netzen

Marco Schäfer

Kurzfassung

In mobilen Ad-hoc Netzen wird ein effizientes System zur Dienstsuche benötigt, um die von den Dienstgebern bereitgestellten Dienste zu finden. Obwohl es schon eine Vielzahl von Ansätzen für das Internet gibt, kommen bei dem Einsatz in mobilen Ad-hoc-Netzen Schwierigkeiten hinzu, da Ad-hoc-Netze sich von festverdrahteten Netzen stark unterscheiden. Deshalb werden in dieser Ausarbeitung die zusätzlichen Anforderungen, welche sich für mobile Ad-hocNetze ergeben untersucht. Es werden die vorhandenen Implementierungen kurz vorgestellt, und danach ausführlicher auf die viel versprechenden Ansätze eingegangen, welche eine effiziente Dienstsuche in mobilen Ad-hoc-Netzen ermöglichen.

1 Einleitung

Mobile Ad-hoc-Netzwerke entstehen durch den spontanen Zusammenschluss mobiler drahtloser Computer, die im folgenden als mobile Knoten bezeichnet werden. Sie setzen keine zugrunde liegende Infrastruktur voraus. Diese Netze kommen oft zum Einsatz um zum Beispiel bei Rettungseinsätzen, beim Militär oder auf Konferenzen, Informationen schnell und ohne Nutzung einer vorhandenen Infrastruktur austauschen zu können. Die Verbreitung mobiler Geräte wie PDA, Laptop, Handy oder Tablet PC hat in den letzten Jahren stark zugenommen. Ständig trägt man die Geräte mit sich und benutzen sie, was die Art und Weise wie Computer im täglichen Leben eingesetzt werden stark beeinflusst hat. Trotz der Mobilität der Geräte, erwartet der Benutzer, dass vergleichbare Dienstmerkmale und Dienstgüte wie in festverdrahteten Netzwerken zur Verfügung gestellt wird. Unglücklicherweise folgen die Netzwerk Ressourcen (z.B. Drucker, Fax) und Anwendungen (DNS, GPS, pdf viewer) dem Benutzer nicht an einen anderen Ort, wenn er das Büro oder Haus verlässt. Deshalb müssen diese Dienste in der neuen Umgebung zuerst gesucht werden.
Für festverdrahtete Netze gibt es bereits viele Ansätze um nach Diensten zu suchen und sie dann auch zu nutzen. Allerdings treten einige Probleme aufgrund der Eigenschaften eines Ad-hoc-Netzes auf, weshalb diese Ansätze nicht ohne weiteres auf mobile Ad-hoc-Netze übertragbar sind.

- Da ein mobiles Ad-hoc-Netz keine feste Infrastruktur voraussetzt und die einzelnen Knoten frei beweglich sind, ist das Netz hoch dynamisch und verändert sich somit ständig. Dies führt zu dem Problem, dass das Finden und die Konfiguration eines zentralen Registers, wie es in fest verdrahteten Netzen benutzt wird um Informationen über die vorhandenen Dienste zu speichern, nicht im voraus geschehen kann. Die Tatsache, dass kein Ansatz verwendet werden kann bei dem alle Informationen über die Dienste auf einem Knoten verwaltet werden, liegt unter anderem daran, dass die Knoten jederzeit den Empfangsbereich des Netzes verlassen können, was im Fall des zentralen Knotens zu einem Verlust der Funktionalität Dienstsuche führen würde.

- In mobilen Ad-hoc-Netzen müssen Dienstanfragen effizient weitergeleitet werden können. Ein Broadcast Ansatz, wie er in festverdrahteten Netzen benutzt wird, ist hier nicht angebracht, da die mobilen Geräte oft nur über begrenzte Ressourcen (Batterielaufzeit, Bandbreite, Reichweite) verfügen. Es gilt somit den Kommunikations- und Verwaltungsaufwand möglichst gering zu halten, um die knappen Ressourcen nicht unnötig zu vergeuden.

Bei kleinen Netzen mit ca. 10 Geräten mögen die oben angesprochenen Probleme auf den ersten Blick nicht so gravierend aussehen, allerdings sollte das Netz auch volle Funktionalität bieten, wenn bis zu hunderten Knoten oder mehr ein mobiles Ad-hoc-Netz bilden. Die Knoten unterscheiden sich dann sehr stark voneinander und kein Knoten weiss über die Eigenschaften oder Ressourcen des anderen Bescheid. Aus diesen Problemen kann man nun direkt Voraussetzungen ableiten, welche ein System zur Dienstsuche in mobilen Ad-hoc-Netzen, erfüllen muss:

- Skalierbar: Anwendungen die eine Vielzahl von Sensoren im Netzwerk benötigen sollten im Bezug auf Netzwerkgröße und Verwaltungsaufwand skalieren
- Effizient: die begrenzten Ressourcen der Geräte sollten bestmöglich eingesetzt werden
- Robust: bei Ausfall von Knoten oder Verbindungen sollte die Funktionalität bewahrt werden
- Dezentral: keine Abhängigkeit von einem zentralen Knoten
- Selbstorganisiert: Netz sollte sich eigenständig aufbauen und nicht von Ortsangaben abhängig sein

In Abschnitt 2 wird auf die bereits vorhandenen Implementierungen zur Dienstsuche eingegangen und erörtert, weshalb sie sich für dieses Szenario nicht eignen. In Abschnitt 3 werden dann verschiedene neue Ansätze vorgestellt und in Abschnitt 4 ihre Grundstrukturen verglichen. Eine kurze Zusammenfassung wird in Abschnitt 5 gegeben.

2 Bestehende Ansätze

Um mobile Ad-hoc-Netze zu realisieren, wurde bereits viel Arbeit in Architekturen investiert, die es ermöglichen, dass sich die einzelnen Geräte finden, miteinander kommunizieren und zusammenarbeiten können. Techniken, die zur Dienstsuche eingesetzt wurden, lassen sich dabei in 3 Kategorien unterteilen: *Rundruf basierte Anfragen, Server basierte Verzeichnisse, verteilte Verzeichnisse.*

- Bei den auf *Rundruf basierten Anfragen* (Gnutella, Bluetooth), wird eine Nachricht an alle sich im Netz befindlichen Geräte gesendet. Das Empfangende Knoten kann nun darüber entscheiden wie er mit der Anfrage weiter verfährt, je nachdem ob er im Stande ist den erwünschten Dienst zu erbringen oder nicht. Die Zeitintervalle innerhalb derer ein solcher Rundruf abgesetzt werden kann sollten aber mit Vorsicht gewählt werden, da eine zu hohe Abfragerate zu viel Bandbreite und Rechenleistung verbraucht, was in Ad-hoc-Netzen nicht wünschenswert ist

- Diejenigen Systeme, welche einen *zentralen Server* zum unterhalten des Diensteverzeichnisses benutzen, haben sich für Anwendungen im Internet als äußerst bewährt erwiesen. So benutzt der IETF-Standard Service Location Protocol (SLP) [GPVD99] einen ausgewählten Server, um innerhalb einer Domäne Dienste aufzufinden oder sie anderen über eine Service-URL bekannt zu machen. Ähnlich verfährt Sun Microsystem's Jini [Micr99], indem es in einem zentralen Verzeichnis (lookup service) Stellvertreter der einzelnen Dienste vorhält, die sich dann der jeweilige Client herunterlädt und über Stellvertreter den tatsächlichen Dienst in Anspruch nehmen kann. Der Tatsache, dass es in einem Netz sehr viele Knoten geben kann, begegnen diese Ansätze mit einer hierarchischen Strukturierung mehrerer Verzeichnis Server. Jedoch ist eine solche Struktur für mobile Netze nicht einfach zu realisieren bzw. durch die Mobilität der Knoten nicht immer die beste Lösung, da sie die dynamischen Beziehungen des Netzes nicht erfasst. Es ist somit fast nicht möglich immer eine aktuelle Version der Netzstruktur vorliegen zu haben, da sie sich bei großen Netzen sehr schnell ändern kann und dadurch zu sehr viel Verwaltungsaufwand führen würde.
- In Systemen mit *verteilten Verzeichnissen* wird kein zentraler Server benötigt, was sie sehr attraktiv für peer-to-peer Netzwerke macht (Freenet, OceanStore). Jeder Knoten enthält einen Teil des Verzeichnisdienstes. Einem Dienst wird ein eindeutiger Schlüssel zugewiesen und eine Hash-Funktion bestimmt die Abbildung des Schlüssels auf den Knoten, welcher die Informationen über diesen Dienst in seinem Verzeichnis hat. Dies ermöglicht zwar ein sehr schnelles Auffinden, allerdings ist der Kommunikationsaufwand bei der Wegewahl zu diesem Knoten nicht zu unterschätzen.

Ein weiteres Problem dieser Ansätzen, ist die Tatsache, dass sie alle von einem sehr großen festverdrahtetem Netzwerk (z.B. Firmen-Netz, Internet) ausgehen. Sie wählen die Dienste nur nach statischen Merkmalen aus, nicht nach dynamischen, wie es für mobile Ad-hoc-Netze der Fall sein sollte. Weiterhin fehlen ausdrucksstarke Sprachen und Werkzeuge, welche eine große Palette von Dienstbeschreibungen vorhalten um auf Merkmale und Funktionalitäten der Dienste schließen zu können.

3 Neue Ansätze

In diesem Abschnitt werden nun neue Ansätze zur Unterstützung der Dienstsuche in mobilen Ad-hoc-Netzen betrachtet. In Abschnitt 3.1 wird ein Ansatz vorgestellt, in dem ein mobiles Ad-hoc-Netz in „Umgebungen“ aufgeteilt und diese über „Kontakte“ miteinander verbunden sind. In Abschnitt 3.2 wird eine Architektur vorgestellt, die mit Hilfe einer Ontologie das Netz in mehrschichtige Kluster unterteilt. In Abschnitt 3.3 wird ein ähnlicher Ansatz wie im vorherigen Abschnitt verwendet, da auch hier semantische und physikaliche Nähe beim gruppieren der Dienste zu „Dienst-Ringen“ eine Rolle spielt. Eine auf einem „virtuellen Backbone“ basierende Architektur wird in Abschnitt 3.4 vorgestellt.

3.1 Kontakt basierte Dienstsuche

Mit CARD (Contact-Based Architecture for Resource Discovery) [HSPN03] soll eine Dienstsuche ermöglicht, und gleichzeitig die Wegewahl für kleine Datenpakete realisiert werden, bei denen die Optimierung der Wegewahl keine Rolle spielt. Dies wird dadurch erreicht, dass das vorliegende Netz in eine so genannte „kleine Welt“ [WaSt98] transformiert wird. Das Netz wird in „Umgebungen“ aufgeteilt, deren Größe durch eine zuvor festgelegte Anzahl von Hops bestimmt wird. Jede dieser Umgebungen hat eine bestimmte Anzahl an „Kontakten“, welche sie

mit den anderen Umgebungen verbindet und dadurch eine bessere Sicht auf die Umgebungen jenseits der eigenen ermöglicht. Diese Kontakte fungieren somit als Abkürzungen innerhalb des Netzes und indem sie den Grad der Unterteilung reduzieren, transformieren sie das Netz in eine „kleine Welt“. Es handelt sich hier um einen hybriden Ansatz, bei dem periodisch die Knoten in der eigenen Umgebung abgefragt und aktuell gehalten werden müssen und reactive Anfragen über die Kontakte hinaus eine Sicht auf das ganze Netz ermöglichen.

Versucht nun ein Knoten S sich eine Umgebung aufzubauen, so beginnt er damit Knoten zu seine Umgebung hinzuzufügen, die maximal R Hops von S entfernt sind. Liegt ein Knoten ausserhalb dieses Radius R, so wird er nicht in die Umgebung von S aufgenommen. Hat S seine Umgebung festgelegt, liegt also kein Knoten, der Teil dieser Umgebung ist mehr als R Hops von S entfernt. Knoten, die exakt R Hops von S entfernt liegen nennt man Kanten-Knoten. Damit nun Verbindungen zu anderen Umgebungen entstehen können, müssen vorher noch die Kontakte ausgewählt werden. Dies geschieht, indem S eine Kontaktwahl Nachricht (contact selection message, CS) an einen Kanten-Knoten schickt, welche die Suchtiefe d, die Kontakt Liste und die Knoten ID von S enthält. Der Kanten Knoten leitet die CS dann an einen beliebigen Nachbarknoten weiter. Der empfangende Knoten X entscheidet nun entweder anhand des Wahrscheinlichkeits-Verfahrens (probabilistic method, PM) oder der Kanten Methode (edge method, EM) ob er als Kontakt für S in Frage kommt.

Bei der Auswahl der Kontakt Knoten sollte berücksichtigt werden, dass sich die Umgebung des neuen Kontaktes und die von S nicht überschneiden und auch die Kontakt-Listen der beiden Umgebungen disjunkt sind. Bei dem Wahrscheinlichkeit-Verfahren wird zuerst überprüft, ob sich S in der Umgebung von X befindet und ob die Kontakt-Liste von S bereits einen Knoten aus dieser Umgebung enthält. Ist dies nicht der Fall, so wird die Wahrscheinlichkeit P mit der X als Kontakt ausgewählt wird bestimmt durch:

$$P = (d - 2R)/(r - 2R) \tag{36}$$

Der Kontakt wird also ausgewählt, wenn er zwischen $2R$ und r liegt, wobei r die maximale Distanz (Anzahl der Hops) eines Knotens ist damit er noch als Kontakt ausgewählt wird. Gleichung 36 garantiert jedoch nicht in jedem Fall, dass sich die Umgebungen der Knoten nicht überschneiden. Denn die Kontaktwahl-Nachricht kann zwar $2R$ Hops gewandert sein, der Knoten kann aber in Wirklichkeit viel näher an S liegen. Ebenso erzeugt der Auswahlprozess sehr viel Overhead, was ein weiterer Grund ist, weshalb die Kanten Methode der Wahrscheinlichkeits-Methode vorgezogen wird.

Bei der Kanten-Methode wird der Kontaktwahl-Nachricht noch eine Liste aller Kanten-Knoten beigefügt und der Empfänger Knoten X prüft nun zusätzlich, ob sich die Umgebungen der Kanten- Knoten überschneiden. Da jeder Knoten der weniger als R Hops von der Kante der Umgebung weg liegt sich mit der Umgebung von S überschneidet, können somit disjunkte Umgebungen und eine Auswahl des Kontaktes zwischen $2R$ und r Hops garantiert werden. Die Knoten des Netzes werden mit Tiefensuche abgearbeitet und bei Auswahl eines Knotens als Kontakt, wird der Pfad der zu ihm führt an S zurückgeliefert.

Abbildung 1 erklärt die Kontaktwahl anhand eines Beispiels. Hier wurde $R = 3$ und $r = 6$ gewählt. Knoten *a, b, c,* und *d* sind die Kanten Knoten von *S*. Sendet S nun eine CS an Knoten *a*, der die Nachricht an einen beliebigen Nachbarknoten (hier e) weiterleitet. Knoten *e* berechnet nun die Wahrscheinlichkeit als Kontakt ausgewählt zu werden, an Hand von Gleichung 36 aus und schickt die CS dann weiter. Dies geschieht bei allen Knoten die auf dem Pfad liegen. Da g exakt r Hops von S entfernt ist, ergibt sich für Knoten g die Wahrscheinlichkeit $P = 1$. Er wird allerdings nicht zu einem neuen Kontakt von S, da Knoten h schon länger ein Kontakt von S ist und in der Umgebung von g liegt. Die CS wird an f zurückgesendet (backtracking) und von dort an einen anderen Nachbarknoten weitergeleitet.

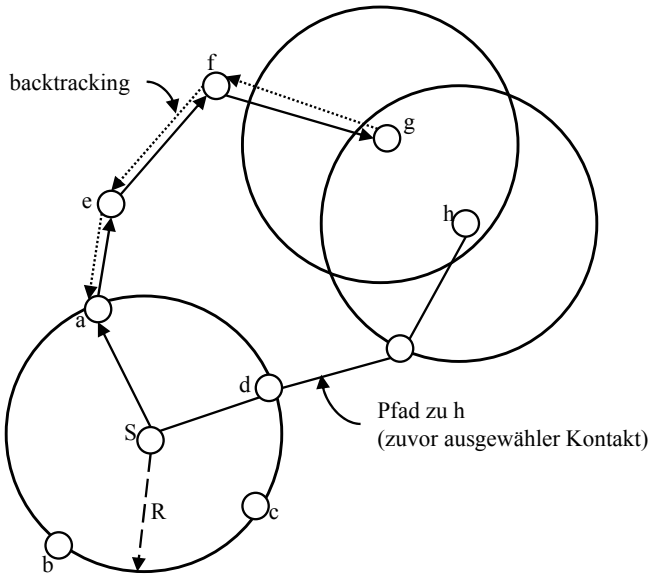

Abbildung 1: Kontakt Wahl nach Wahrscheinlichkeits Methode

Wird nun eine Dienstsuche von Knoten S aus initiiert, wird zuerst in der Umgebung von S gesucht. Danach wird eine Suchanfrage, welche die Suchtiefe D und die ID des Zieles enthält, nacheinander an alle Kontakte von S geschickt. Zuerst ist $D = 1$ und die Kontakte durchsuchen ihre eigene Umgebung nach der ID des Zieles. Ist die Suche erfolglos, wird eine neue Anfrage mit $D = 2$ verschickt. Die Kontakte von S bemerken, dass die Anfrage nicht für sie bestimmt ist, erniedrigen D um eins und schicken die Anfrage weiter an ihre eigenen Kontakte. Somit wird nach und nach das komplette Netz abgesucht.

Im folgenden soll nun kurz auf den Effekt, den die verschiedenen Variablen auf die Erreichbarkeit eines Netzes haben eingegangen werden.

- Vergrößert man zum Beispiel die Größe R einer Umgebung, nimmt die Erreichbarkeit zuerst deutlich zu. Je näher R jedoch an die maximale Kontakt Distanz r herankommt um so schlechter wird die Erreichbarkeit des Netzes, da nur noch sehr wenige Knoten zur Kontaktauswahl stehen.

- Variiert man r, so lässt sich zunächst eine deutliche bessere Erreichbarkeit messen, die aber sobald r einen bestimmten Wert überschreitet, nicht mehr merkbar zunimmt.

- Erhöht man die Suchtiefe D, wird über Breitensuche das Netz abgesucht, was in einer Baumstruktur der Kontakte resultiert und so eine starke Auswirkung auf die Forderung der Skalierbarkeit des Netzes hat.

Der Overhead der bei der Aufrechterhaltung der Netzstruktur (Kontakte und Umgebungen) hängt natürlich direkt von den zuvor genannten Variablen ab. So nimmt der Overhead logischer Weise mit der Anzahl der Kontakte zu, wohin gegen er bei Vergrößern von r abnimmt. Dies liegt daran, dass die Knoten die bei der Knotenwahl in Frage kommen weiter von S entfernt liegen und sich ihre Umgebung nur selten überschneiden, wodurch der Aufwand der durch Backtracking entsteht merklich reduziert wird. Bei Vergleich mit Ansätzen wie „Flooding“ oder „Broadcasting“ lässt sich feststellen, das CARD in statischen als auch in hoch dynamischen Netzen signifikant weniger Overhead erzeugt.

3.2 Multi-Layer Cluster

Mit diesem Ansatz [KRKl02] wird versucht, die verschiedenen Dienste innerhalb des Netzes zu mehrschichtigen Klustern zusammenzufassen. Die Kluster werden auf Grund semantischer und physikalischer Nähe der Dienste zueinander gebildet. Die Entscheidung, ob sich zwei Dienste semantisch ähnlich sind, wird mit Hilfe einer Ontologie getroffen. Sie beschreibt die Dienste mit Hilfe eines Sets von Termen $\{t_1, \ldots, t_r\}$ und der Relation *isSubTopicOf*. Der oberste Term dieser Hierarchie wird Wurzelterm genannt und lautet "database" in dem Beispiel in Abbildung 2. Er umfasst die ganze Domäne der Ontologie. Alle anderen Terme sind dem Wurzelterm untergeordnet und die Beziehung zwischen zwei Termen t_1 und t_2 wird durch $t_1.isSubTopicOf = t_2$ dargestellt. Sei $S = \{s_1, \ldots, s_n\}$ die Menge aller Dienste und $T = \{t_1, \ldots, t_r\}$ die Menge aller Terme der Ontologie, wobei t_r der Wurzelterm ist, dann lautet die Eltern-Funktion $p : S \cup T \to T$:

$$p(x) = \begin{cases} t, & mit \quad t = x.isDescribedBy & : & x \in S \\ t, & mit \quad t = x.isSubTopicOf & : & x \in T \backslash \{t_r\} \\ x & & : & x = t_r \end{cases}$$

Da die Ontologie eine Baumstruktur besitzen soll, sind Zyklen verboten. Die Zahl der Schichten der Ontologie ist durch die Anzahl der Knoten definiert, welche auf dem längsten Weg von einem Blatt zur Wurzel liegen. Die dynamische Relation $\longrightarrow \subseteq S \times S$ bezieht sich darauf, ob die Geräte erreichbar sind. Sind zum Beispiel die Geräte i und j direkt miteinander verbunden, so wird die Bezeichnung $i \to j$ benutzt. Sind mehrere Schritte nötig, um ein anderes Gerät zu erreichen, wird auf Grund der Transitivität der Relation aus $i \to \ldots \to j$ zu $i \xrightarrow{*} j$.

Bisher wurden die semantische und die physikalische Struktur nur getrennt betrachtet. Um in einem realen Ad-hoc-Netz mehrschichtige Kluster zu bilden, müssen sie gemeinsam betrachtet werden. Die entstandene Architektur besteht aus *n+1* Schichten, wobei n für die Anzahl der Schichten der Ontologie steht, zu der dann noch die Schicht mit den konkreten Diensten hinzukommt. Auf der untersten Schicht, werden diejenigen Geräte zusammengefasst, deren Dienst durch den gleiche Blatt-Term der Ontologie beschrieben werden und sich gleichzeitig im selben Empfangsbereich befinden. Auf den Schichten darüber, werden diejenigen Kluster zusammengefasst, die den gleichen Eltern-Term haben, d.h. deren semantische Beschreibung allgemeiner wird, wenn man sich in der Baumstruktur der Ontologie nach oben bewegt. Zusätzlich, müssen sich die Kluster „erreichen" können, was bedeutet, dass mindestens zwei Geräte unterschiedlicher Kluster miteinander direkt kommunizieren können. Die oberste Schicht besteht dann aus einem großen Kluster, welcher alle Geräte mit ihren angebotenen Diensten enthält.

Durch die Art und Weise, wie die Cluster gebildet wurden erhält man nun eine Schichtenarchitektur, welche es erlaubt komplexe Funktionen (z.B Netzweite Dienstsuche) in Teilschritte aufzuspalten. Eine Schicht *l* benutzt dann immer die von der darunter liegenden Schicht *l-1* bereitgestellten Funktionen um ihrerseits wieder der Schicht *l+1* höhere Funktionalität zur Verfügung zu stellen. Jede Schicht hat eine unterschiedliche Sicht auf das Netz und stellt zwei grundlegende Funktionen bereit: *Suchen* und *Senden*. Mit der Suchen-Funktion wird der aktuelle Kluster nach einem Dienst durchforstet und mit der Senden-Funktion eine Nachricht an alle Kluster geschickt, die erreichbar sind. Beide Funktionen sind aber nur innerhalb ihrer Schicht aktiv und nutzen die respektiven Funktionen der darunter liegenden Schicht.

Wird nun eine Dienstsuche initiiert, wird auf der Geräte-Schicht (Schicht 0) zuerst überprüft ob das Gerät selbst nicht bereits den Dienst zur Verfügung stellen kann. Hier wird die Ontologie noch nicht benötigt, da jedes Gerät einfach seine eigene Dienstbeschreibug mit der Anfrage vergleichen kann. Ist die Suche erfolglos, wird eine Nachricht an alle direkt erreichbaren Gerät (i → j) mit ähnlicher Semantik innerhalb des Klusters (Schicht 1) geschickt,

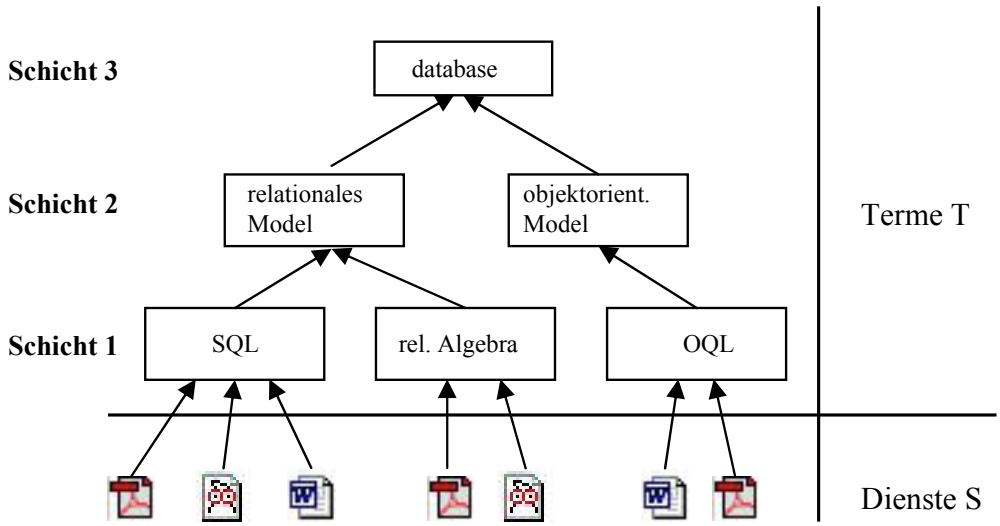

Abbildung 2: Ontologie einer Datenbank und die beschriebenen Dokumente

welche dann wieder ihre Dienstbeschreibung mit der Anfrage abgleichen. Bleibt auch diese Suche erfolglos, so wird die Anfrage an alle erreichbaren Kluster weitergereicht. Somit wird die Dienstbeschreibung innerhalb der man nach dem Dienst sucht immer allgemeiner und es wird auch nach und nach ein immer größeres Gebiet (physikalisch) abgesucht. Ist man bei dem Wurzelterm angelangt, hat man eine Sicht über das komplette Netz und hat dann den gesuchten Dienst gefunden oder kann mit Sicherheit sagen, dass er in diesem Ad-hoc-Netz nicht angeboten wird.

Diese Schichten-Architektur ist sehr natürlich und intuitiv, da sie nur zwei externe Parameter betrachtet: die Semantik einer Ontologie und einen Graphen, welcher auf Grund von Empfangsbereichen der Geräte zustande kommt. Somit ist ein sehr einfaches aber gleichzeitig auch stabiles Rahmenwerk zur Dienstsuche gegeben. Der Ansatz beruht auf keiner zentralen Architektur, da jedes Gerät gleich behandelt wird. In mobile Ad-hoc-Netzen ist es notwendig sich darüber im klaren zu sein, dass Ressourcen knapp sind und deshalb ein gesuchter Dienst nicht nur der Anfrage entsprechen, sondern auch dem Dienstnehmer physikalisch nahe zu sein sollte, um eine „teure“ Paket-Weiterleitung zu vermeiden. Dies geschieht bei diesem Ansatz automatisch, indem er zuerst die nahe gelegenen Kluster durchsucht, bevor er seine Suche ausweitet. Durch die Einteilung in Kluster kann auch schnell auf Veränderungen innerhalb des Netzes reagiert werden, indem in den verschiedenen Klustern unterschiedliche Strategien zur Organisation angewendet werden können.

3.3 Dienst-Ringe

In diesem Abschnitt soll eine Architektur, welche aus so genannten Dienst-Ringen besteht, vorgestellt werden [KlKRO03]. Ein sematisches Overlay, das als hierarchisch angeordneter Ringe auf der Transportschicht aufbaut, soll dabei die Dienstsuche in mobilen Ad-hoc-Netzen ermöglichen.
Wie bereits in Abschnitt 3.2 wird auch hier ein hybrider Ansatz verwendet, der die verschiedenen Dienste auf Grund von physikalischer und semantischer Nähe zu Ringen zusammenfasst. Jeder dieser Ringe hat genau einen Dienstzugangspunkt, welcher Informationen über alle

Dienste vorhält, die innerhalb des Ringes angeboten werden. Mit $v_i \rightarrow v_j$ wird die Möglichkeit beschrieben, auch über andere Geräte hinweg Daten von Gerät i nach Gerät j zu schicken. Da aber die Minimierung der Kommunikation eine wichtige Zielvorgabe beim Entwurf der Architektur ist, werden nur diejenigen Verbindungen aufrecht erhalten, die zur Dienstsuche notwendig sind. Diese Overlay-Links werden durch $v_i \stackrel{*}{\rightarrow} v_j$ dargestellt und werden zu einem Ring zusammengefügt. Der einzelne Knoten v_i speichert die Adresse seines Vorgängers in $v_i.succ_0$, die Adresse seines Nachfolgers in $v_i.pred_0$ und den Namen seines Ringes in $v_i.ring_0$. Die Dienstzugangspunkten solcher Level-0-Ringe haben die kompletten Dienstbeschreibungen der jeweiligen Ringe in $v_0.sum_0$ gespeichert und, werden auf der darüber liegenden Schicht zu einem neuen Ring von Dienstzuganspunkten zusammengefasst. Eine gültiges Overlay ist somit gegeben, falls: (a) alle Geräte Teil von genau einem Level-0-Ring sind, (b) alle Dienstzugangspunkte der Level *l-1* Ringe Teil eines Level *l* Ringes sind und (c) es auf der obersten Schicht L nur einen Ring, den „Welt Ring“ gibt.
In Abbildung 3 wird ein gültiges Overlay für Dienst-Ringe dargestellt, wobei V_3, V_6 und V_{10} Dienstzugangspunkte der Ringe R_1, R_2 und R_3 sind, die zu dem neuen Ring R_4 mit Dienstzugangspunkt V_6 zusammen geschlossen werden.

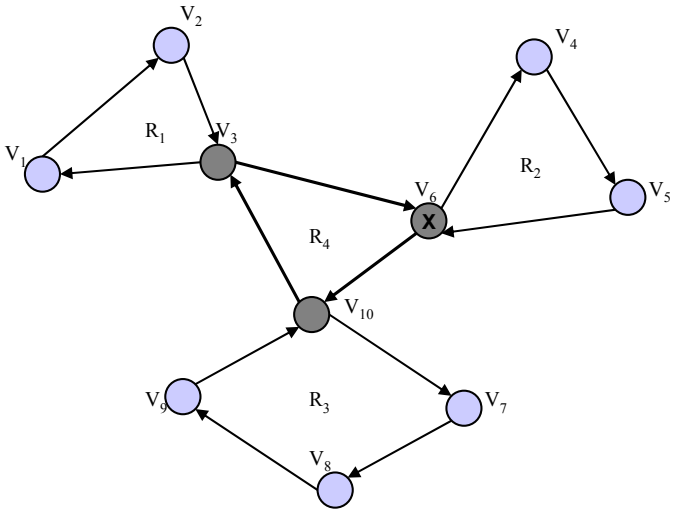

Abbildung 3: Gültiges Overlay für Dienst-Ringe.

Da die strukturellen Voraussetzungen für ein gültiges Overlay geklärt wären, gilt es nun die Bedingungen für ein semantisch gutes Overlay zu betrachten. Es wird eine Möglichkeit benötigt, die vorhandenen Dienste zu beschreiben und miteinander zu vergleichen. Als erstes wird eine Distanz Funktion *dist* definiert die es ermöglicht vorhandene Dienstbeschreibungen miteinander zu vergleichen und zu entscheiden ob die Dienste eine ähnliche Funktionalität besitzen. Desweiteren wird eine Funktion *sum* benötigt, welche in der Lage ist Dienstbeschreibungen zusammenzufassen und eine neu allgemeinere Beschreibung zu erzeugen, welche die vorherigen Beschreibungen überdeckt. Diese neue Beschreibung wird nun im Dienstzugangspunkt des Ringes gespeichert, der die semantisch ähnlichen Dienste zusammenfasst. Jedoch sollte darauf geachtet werden, dass die Ringe weder zu groß noch zu klein sind und „gute“ Verbindungen (wenige Hops) zu anderen Ringen haben, damit Nachrichten schnell und effizient verschickt werden können. Sind diese Voraussetzungen erfüllt, so spricht man von einem semantisch korrektem Overlay.

Kommt nun eine Gerät mit neuen Diensten hinzu, so wird es nach einem Algorithmus in die Struktur des Ad-hoc-Netzes eingebunden. Die bereitgestellten Dienste des Knotens werden innerhalb des Level-0-Rings, dem das Gerät nun angehört, bekanntgemacht. Dies geschieht über das Versenden einer Nachricht an den Nachfolgerknoten. Handelt es sich bei diesem nicht um den Dienstzugangspunkt des Ringes, wird die Nachricht so lange weitergeleitet, bis sie diesen erreicht. Der jeweilige Dienstzugangspunkt sorgt nun dafür, dass die Änderung in die höheren Ringe weitergeleitet wird.
Bei der Suche nach einem bestimmten Dienst ruft der Knoten, von dem die Anfrage ausgeht einen Dienstsuche-Algorithmus auf, welcher zuerst den eigenen Ring nach dem Dienst untersucht und dann nach und nach die Anfrage nur in solche Ringe weiterleitet für die, aufgrund der Dienstbeschreibung, es möglich wäre einen solchen Dienst zu erbringen. Dadurch wird erreicht, dass nicht das komplette Netz durchsucht werden muss, sondern nur einzelne Teilbereiche, was der Forderung nach Effizienz nachkommt. Das Konzept der Dienst-Ringe ist dezentral organisiert, und es enthält Mechanismen, welche die Grundstruktur aufrechterhalten, aufgebrochene Ringe wieder zusammenfügen, die Ringe in der richtigen Größe halten und somit den Ausfall oder das Hinzukommen von einzelnen Knoten kompensieren, was es nicht nur Robust gegen Fehler macht, sondern auch in einer guten Skalierbarkeit des Netzes resultiert.

3.4 Dienstsuche durch Unterstützung der Vermittlungsschicht

Bei diesem Ansatz wird das Dienstverzeichniss auf mehrere Knoten verteilt. Diese Dienstvermittlungsknoten (Service Broker Nodes) schließen sich dann zu einem „virtuellen Backbone“ zusammen [HSPN03]. Damit an die einzelnen Dienstvermittlungsknoten keine zu hohen Anforderungen gestellt werden müssen, werden nur einfache Datensätze mit Dienstbeschreibungen und Wegewahl Informationen auf ihnen gespeichert, wodurch jeder beliebige Knoten im Netz diese Aufgabe erfüllen kann. Somit erreicht man, dass das Netz gut skaliert und die Suchzeiten verkürzt werden. Ist der virtuelle Backbone erstellt, muss man sich nun Gedanken machen, wie man Nachrichten effektiv innerhalb des Netzes verbreitet. Somit ergeben sich zwei unterschiedliche Phasen bei diesem Ansatz: die Verwaltung des Backbones (Backbone Managment Phase) und die Phase der verteilten Dienstsuche (Distributed Service Discovery Phase).

Zu Beginn der Backbone Managment Phase (BBM) sammelt jeder Knoten so genannte „Hello-Beacons“, welche von den Knoten verschickt werden. Sie enthalten Informationen wie Anzahl der Nachbaren, Frequenz des Verbindungsverlustes, Netz ID eines Knotens und Wegewahl Informationen. Somit kann jeder Knoten eine „Neighborhood Information Table“ (NIT) und eine Routingtabelle aufbauen. Hello-Beacons werden nur über einen Hop gesendet und helfen nicht nur den Backbone zu formieren, sondern auch virtuelle Verbindungen zwischen den Dienstvermittlungsknoten aufzubauen und die Backbone Struktur aufrecht zu erhalten.

Um den Algorithmus leichter beschreiben zu können, werden im Folgenden den unterschiedlichen Gruppen, denen Knoten angehören können, verschiedene Farben zugeordnet. Da die Knoten zu Beginn noch keine Funktion innerhalb der Architektur übernehmen sind sie zuerst „weiß“. Danach wird eine Reihe von Knoten ausgewählt, deren Verbindungsverlust-Frequenz unterhalb eines vorgegebenen Wertes liegt und gleichzeitig viele Verbindungen zu anderen Knoten haben. Diese „weißen“ Knoten werden nun zu Backbone Knoten, die „schwarz“ sind. Empfängt ein weißer Knoten ein Hello-Beacon eines Bachkbone Knotens, so wird der Backbone Knoten zu einem virtuellen Zugangspunkt (Virtual Access Point, VAP) für den weißen Knotens. Alle Knoten, die einen VAP zum Backbone als Nachbar haben werden dann „grün“. Die Zeit in der die Hello-Beacons gesammelt werden, kann verlängert werden. Falls gegen Ende dieser Zeitspanne nur noch ein weißer Knoten übrig ist, so wählt er denjenigen grünen Knoten als virtuellen Zugangspunkt, der die geringste Verbindungsverlust-Frequenz hat.

Durch diesen Algorithmus wird garantiert, dass die Entscheidung über die Zugehörigkeit eines Knotens nach einer bestimmten Zeit getroffen wird, ein Satz von Knoten ein stabiles Gerüst bildet (Backbone) und jeder virtuelle Zugangspunkt maximal drei Hops vom nächsten virtuellen Zugangspunkt entfernt ist. Ein Beispiel für einen virtuellen Backbone sieht man in Abbildung 4, wobei die großen schwarzen Knoten den Backbone darstellen und bei den kleinen grünen Knoten die höherwertige Ziffer die Zugehörigkeit zu einem virtuellen Zugangspunkt anzeigen.

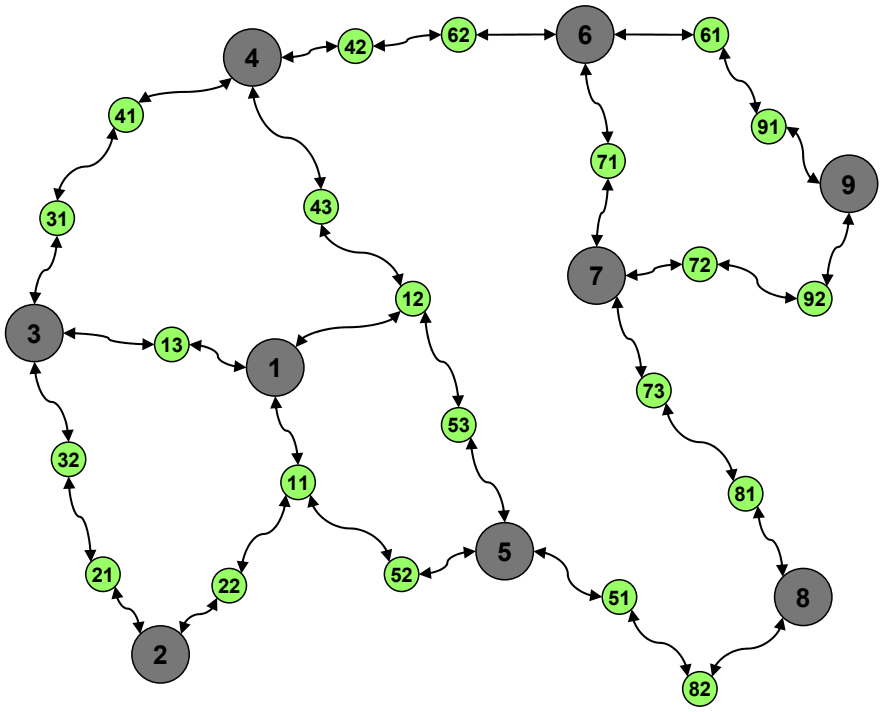

Abbildung 4: Virtueller Backbone

Die Verzeichnisse mit den Dienstbeschreibungen befinden sich auf den Knoten des virtuellen Zugangspunkts. Um einen von Knoten i erbrachten Dienst zu registrieren, muss er dies seinem virtuellen Zugangspunkt (VAP_i), dem Quell-VAP, mitteilen. Möchte man den Dienst auch bei anderen virtuellen Zugangspunkten registrieren, benötigt man einen Multi- oder Broadcast Mechanismus um die Nachrichten an die andern virtuellen Zugangspunkte weiterzuleiten. Der virtuelle Backbone hilft zwar dabei den Kommunikationsoverhead niedrig zu halten, jedoch kann dieser durch das einfache „Fluten“ von Nachrichten und die auftretenden Schleifen innerhalb des Netzes erheblich erhöht werden. Um diesem Phänomen entgegen zu wirken, wird ein Algorithmus zur Erstellung eines quellen-basierten Multicast Baumes benutzt. Der Algorithmus wird durch das Verschicken von Such- oder Registrierungs-Nachrichten an das Backbone Managment ausgelöst. Jeder Backbone Knoten hält eine Liste mit Weiterleitungsmöglichkeiten seiner benachbarten virtuellen Zugangspunkte vor, welche für jeden Multicast Baum eindeutig durch den Quell-VAP gekennzeichnet ist. Multicast Nachrichten enthalten Angaben wie: *Quell Knoten, Quell-VAP Knoten, Sequenznummer, last-hop Knoten, next-hop Knoten.* Durch das Versenden dieser Informationen können Multicast-Nachrichten eindeutig identifiziert werden. Nachrichten, die über andere Pfade empfangen werden kann man ausgefiltern und Informationen für die Erstellung eines Multicast-Baumes sammeln.

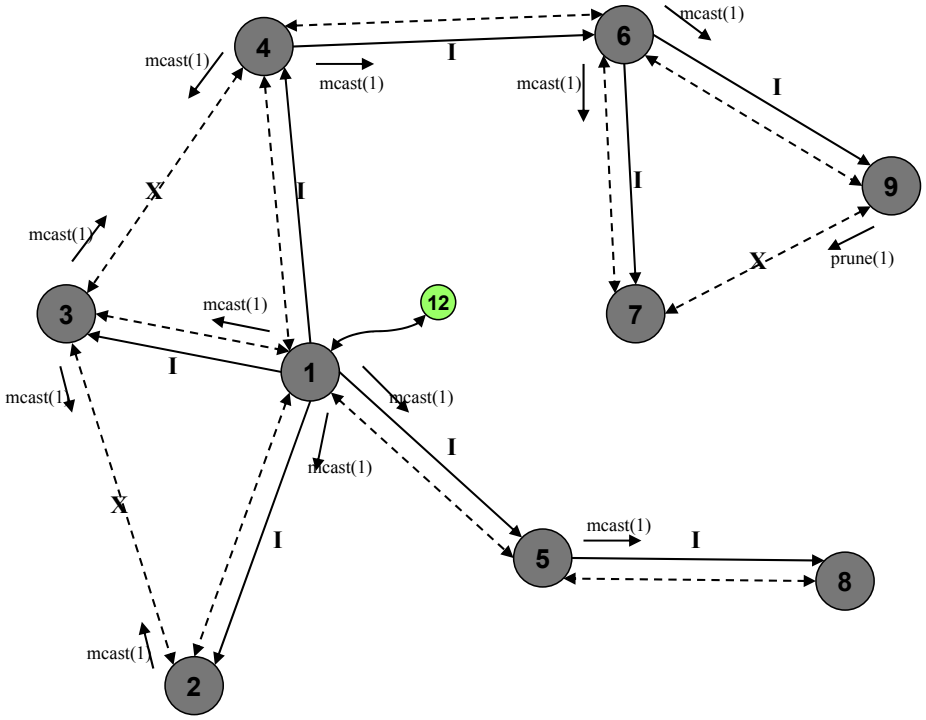

Abbildung 5: Quellen-basierter Multicast Baum

In Abbildung 5 wird an einem Beispiel erläutert, wie ein Multicast-Baum entsteht. Es wird dieselbe Topologie wie in Abbildung 4 verwendet mit dem Unterschied, dass die grünen Knoten zwischen den VAP-Knoten durch Pfeile ersetzt wurden, welche die virtuellen Verbindungen darstellen. Möchte nun Knoten 12 seinen angebotenen Dienst bei allen VAP's registrieren, so sendet er zuerst eine Nachricht an seinen VAP_1 (Quell-VAP), um sich bei ihm anzumelden. Dann sendet VAP_1 Nachrichten an all seine Nachbar-VAP's. Knoten 2, 3, 4 und 5 erhalten zum ersten mal eine Multicast-Nachricht von Knoten 1 und deshalb leiten sie eine Kopie dieser Nachricht an all ihre Nachbar-VAP, mit Ausnahme dessen Knoten der die Nachricht geschickt hat, weiter. Im Beispiel wird nun Knoten 2 die selbe Nachricht von Knoten 3 empfangen, wie er sie schon zuvor von Knoten 1 erhalten hat. Somit weiß er, dass Knoten 3 bereits die Nachricht von 1 empfangen hat und stoppt somit die Weiterleitung von Nachrichten zu Knoten 3, falls die Nachricht von Knoten 1 ausging. Das selbe Prinzip gilt auch für Knoten 3, der Duplikate von Knoten 2 und 4 erhält und als Konsequenz die Nachrichten von Knoten 1 nicht mehr an 2 und 4 weiterleitet. Ein spezieller Fall tritt in Knoten 9 auf, der zuerst eine Multicast-Nachricht von Knoten 6 und kurz darauf von Knoten 7 empfängt. Da dieser aus der Weiterleitungsliste von Knoten 9 gestrichen wurde, muss Knoten 9 explizit eine PRUNE-Nachricht an Knoten 7 schicken, damit dieser weiß, dass er keine Nachrichten mehr von Knoten 1 an Knoten 9 weiterleiten muss. Somit ist ein Multicast-Baum über dem virtuellen Backbone entstanden, mit dem Wurzel Knoten 1 und den mit I markierten Pfaden. Alle Dienstsuchen und Registrierungen, welche von Knoten ausgehen die zu VAP 1 gehören, benutzen diesen Multicast Baum um Nachrichten zu verschicken.

In [HSPN03] wird dieser Ansatz mit anderen Multi- oder Anycast Implementierungen verglichen. Es lässt sich sagen, dass sich bei fester Knoten Anzahl und gleich bleibender Dienste die Performanz nicht stark verändert, wenn man das Client/Server-Verhältnis oder die Geschwindigkeit mit der sich Knoten bewegen verändert. Dies kommt unter anderem daher, dass der Großteil des erzeugten Overheads (75% bis 94%) durch die Hello-Beacons entsteht.

Die Laufzeit der Anfragen ist ebenfalls besser als bei anderen Implementierungen, da weniger Knoten in die Suche mit einbezogen werden müssen, wodurch Netze mit diesem Aufbau für Echtzeitanwendungen sehr attraktiv werden.

4 Vergleich

Der in Abschnitt 3.1 vorgestellte Ansatz CARD teilt das vorhandene Netz in „Umgebungen“, deren Größe durch eine zuvor bestimmte Anzahl von Hops vorgegeben ist. Durch so genannte „Kontakte“, welche als Schnittstellen zwischen den einzelnen Umgebungen wirken, wird erreicht, dass Nachrichten über sie sehr schnell verbreitet werden können und damit alle Umgebungen des Netzes erreicht werden.
Bei der Entscheidung ob man eine Architektur mit oder ohne ausgewähltem Verzeichnis zum Speichern der Dienstbeschreibungen bevorzugt, scheint es unlogisch in mobilen Ad-hoc-Netzen einen bestimmten Knoten auszuwählen, der diese Aufgabe übernehmen soll. Durch die Tatsachen, dass keine Infrastruktur vorausgesetzt wird, ein solch zentraler Ansatz sehr fehleranfällig ist und eine Menge Kommunikation mit diesem Knoten stattfinden muss (Flaschenhals), wird dieser Ansatz oft nicht weiter verfolgt. Allerdings zeigt Abschnitt 3.4, dass es möglich ist Nachteile, die man sich mit einem solchen Ansatz einhandelt durch geringfügige Änderungen zu kompensieren. Anstatt einem Knoten diese Aufgabe zu über lassen, schließen sich mehrere solche Dienstvermittlungsknoten (Service Broker Nodes) zu einem „virtuellen Backbone“, unterhalb der Vermittlungsschicht, zusammen. Damit an die einzelnen Dienstvermittlungsknoten keine zu hohen Anforderungen gestellt werden müssen, werden nur einfache Datensätze mit Dienstbeschreibungen und Wegewahl Informationen auf ihnen gespeichert, damit jeder beliebige Knoten im Netz diese Aufgabe erfüllen kann. Dadurch erreicht man, dass das Netz gut skaliert, die Suchzeiten verkürzt und die Dienstanbieter nicht mit Suchanfragen überschüttet werden. Ist der virtuelle Backbone erstellt, muss man sich Gedanken machen, wie man Nachrichten effektiv innerhalb des Netzes verbreitet, wobei es wünschenswert ist, dass die einzelnen Kommunikationsschichten eng zusammenarbeiten um mögliche Redundanzen zu vermeiden, die durch das ausführen ähnlicher Aufgaben auf den verschiedenen Schichten entstehen. Hat man also einen Mechanismus um einen virtuellen Backbone unterhalb der Vermittlungsschicht aufzubauen und aufrechtzuerhalten, so ist der dadurch entstehende Overhead gerechtfertigt, da die Wegewahl- und Dienstsuche-Algorithmen diese Struktur gemeinsam nutzen können. Durch den Vergleich wurde ebenfalls deutlich, dass Architekturen die ihre Dienstverzeichnisse über eine virtuellen Backbone bereitstellen sehr gut mit den Problemen eines mobilen Ad-hoc-Netzes klar kommen, was den Charakteristiken des Ansatzes (Lastverteilung, Ressourcen Zuteilung, gute Skalierbarkeit) zuzschreiben ist.
Die Ansätze in Abschnitt 3.2 und 3.3 sind sich in der Hinsicht ähnlich, dass sie beide semantische und physikalische Nähe von Diensten betrachten um die Dienste sinnvoll zu gruppieren. Bei dem Ansatz Multi-Layer Cluster, wird eine Ontologie benutzt um die vorhandenen Dienste zu beschreiben. Auf der untersten Schicht werden Dienste, die den gleichen Blattterm haben und direkt miteinander kommunizieren können, zu Klustern zusammengefasst. Während auf den darüber liegenden Schichten die semantische Nähe mit Hilfe der Ontologie bestimmt wird und sich die Kluster erreichen können müssen, damit sie zu neuen Klustern zusammengefasst werden können.
Dienst-Ringe entscheiden auf eine andere Art und Weise, ob sich Dienste semantisch nahe sind. Die Architektur stellt eine Funktion bereit, um zu entscheiden wie stark sich zwei Dienste unterscheiden und eine weitere Funktion um ähnliche Dienstbeschreibungen zusammenzufassen und eine neue allgemeinere Beschreibung zu generieren, welche dann in den Dienstzugangspunkten der Ringe gespeichert wird. Die Dienstzugangspunkte der Ringe auf der untersten Schicht, werden dann zu neuen Ringen zusammengeschlossen, bis es auf der obersten Schicht

nur noch einen Ring gibt, dessen allgemeine Dienstbeschreibung alle angebotenen Dienste überdeckt.

5 Zusammenfassung

Auf der technischen Seite von mobilen Ad-hoc-Netzen wurde bereits viel Forschungsarbeit geleistet, um die einzelnen Geräte miteinander zu verbinden und effiziente Wegewahl Verfahren zu entwickeln. Jedoch wurde bisher sehr wenig Arbeit darin investiert, den Nutzern eines solchen Netzes gezielt die Möglichkeit zu bieten, nach den angebotenen Dienste zu suchen und sie dann auch zu nutzen. Alle vorgestellten Ansätze erfüllen die Forderungen nach einem dezentralen Aufbau, der Skalierbarkeit und der Effizienz. Ebenso verfügen sie über Mechanismen, die beim Ausfall von Knoten greifen und die grundlegende Struktur aufrecht erhalten, was sie unabhängig und selbstorganisiert macht. Eine besondere Stellung nehmen die Ansätze aus Abschnitt 3.2 und 3.3 ein, bei denen auch semantische Aspekt bei der Gruppierung der Dienste eine Rolle spielen was in einer wesentlich besseren Performanz bei Suchanfrage resultiert. Welcher dieser viel versprechenden Ansätze sich durchsetzen wird, ist jetzt noch nicht abzuschätzen, da noch nicht für alle Ansätze Simulationsergebnisse vorliegen. Dies wird sich wohl erst nach konkreter Implementierung und ausgiebigem Testen zeigen.

Literatur

[GPVD99] E. Guttman, C. Perkins, J. Veizades und M. Day. Service Location Protocol, Version 2. *IETF RFC 2608*, Juni 1999.

[HSPN03] A. Helmy, S.Garg, P. Pamu und N. Nahata. Contact-Based Architecture for Resource Discovery (CARD) in Large Scale MANets. *International Parallel and Distributed Processing Symposium (IPDPS'03)*, April 2003.

[KlKRO03] M. Klein, B. König-Ries und P. Obreiter. Service Rings - A Semantic Overlay for Service Discovery in Ad hoc Networks. *14th International Conference on Database and Expert Systems Applications DEXA'2003*, September 2003.

[KRKl02] B. König-Ries und M. Klein. Multi-Layer Clusters in Ad-hoc Networks - An Approach to Service Discovery. *International Workshop on Peer-to-Peer Computing, im Rahmen der Networking 2002 Konferenz*, Mai 2002.

[Micr99] Sun Microsystems (Hrsg.). Jini Technology Architectural Overview. White Paper, Sun Microsystems, Januar 1999.

[WaSt98] D. Watts und S. Strogatz. Collective dynamics of „small world" networks, Juni 1998.

Abbildungsverzeichnis

On-demand Link-state Routing in Ad-hoc-Netzen

Boudigue Alioum

Kurzfassung

Die Merkmale von Ad-hoc-Netze wie die Mobilität der Netzteilnehmer und die Abwesenheit einer zentralen Struktur stellen eine echte Herausforderung für Routingprotokolle in Ad-hoc-Netzen auf. Bis jetzt wurden zwei Klassen von Routingprotokolle in Ad-hoc-Netzen eingesetzt: die Table-driven Routingprotokolle wie Link-state Routingprotokolle(OLSR) und die On-demand Routingprotokolle wie AODV und DSR. Hier wird eine neue Familie von Routingprotokolle, bei dem die Eigenschaften von Link-state Routingprotokolle und On-demand Routingprotokolle in einem einzigen Protokoll zusammengeführt werden, vorgestellt. Zur Beschreibung dieses Ansatzes werden zwei solche Protokolle untersucht. Es handelt sich um die Protokollen SOAR (Source Tree On Demand Adaptive Routing) und OLIVE (On-Demand Link Vector).

1 Einführung ins Routing in Ad-hoc-Netzen

1.1 Einführung in Ad-hoc-Netzen

Ein Ad-hoc-Netz ist ein autonomes System von mobilen Rechnern, die drahtlos verbunden sind. Einer der wichtigsten Merkmale von Ad-hoc-Netzen ist die Abwesenheit einer zentralen Struktur wie einer Basisstation. Die Kommunikation zwischen zwei Knoten geht über anderen Netzwerknoten, die sich als Router verhalten. Die Netzwerkknoten haben eine Bewegungsfreiheit und ändern damit ständig die Netztopologie. Dies stellt eine grosse Herausforderung für Routingprotokolle dar, die in der Lage sein sollen, Wege zu garantieren trotz der Mobilität und der ständigen Änderung der Netztopologie.

1.2 Funktionsweise von Link-state und On-demand Routingprotokolle

Die Table-driven Routingprotokolle wie Link-state Routingprotokolle und On-demand Routingprotokolle sind zwei Familien von Routingprotokolle, die in Ad-hoc-Netze eingesetzt werden.

Link-state Routingprotokolle
Als Modell zur Beschreibung der Routingprotokolle wird einen Graph eingesetzt mit Routern als Knoten und die Verbindungen zwischen Routern als Kante. Im Link-state Routing beschreibt jeder Knoten seine eigene Umgebung, dass heisst den Zustand der Verbindungen mit seinen Nachbarn. Die Nachbarn werden erkundet, indem der Router einfache Hello-Paketen mit denen austauscht. Diese Beschreibung wird bei jedem Router im Netz gespeichert. Damit ist eine verteilte Datenbank aufgebaut, die die Beschreibung der Umgebung jeder Knoten enthält und die bei jedem Knoten vorhanden ist. Diese Datenbank wird bei jeder Änderung in der Umgebung eines Knotens aktualisiert. Mit dieser Technik vefügt jeder Router jederzeit über genügende Informationen um sich einen Überblick des ganzen Netz zu schaffen. Damit

kann er den kürzesten Weg zu jedem anderen Knoten im Netz finden. Ein Beispiel von einem Link-state Routingprotokoll ist OLSR (Optimized Link State Routing) [Jacq03].

On-demand Routingprotokolle
Die On-demand Routingprotokolle funktionieren auf Basis der sogenannten Flooding Technik. Die Quelle oder jeder Knoten, der nach einem Weg zu einem anderen Knoten im Netz sucht sendet einen Abfragepaket an das ganze Netz. Jeder Knoten, der den Abfragepaket bekommt, strahlt ihn an seinen Nachbarn aus, wenn er nicht der Zielknoten ist. Jede Abfrage trägt dabei einen eindeutigen Idenfikator. Mit Hilfe diesem Identifikator wird eine mehrmalige Propagation einer Abfrage durch denselben Knoten verhindert. Mit dieser Technik werden alle Knoten erreicht. So wird den Zielknoten von der Abfrage erreicht, wenn es möglich ist ihn von der Quelle aus überhaupt zu erreichen. Nachdem er die erste Abfrage bekommen hat antwortet der Zielknoten, indem er eine Antwort (reply) zur Quelle sendet. Die Antwort geht zurück zur Quelle auf dem gefundenen Pfad, aber in der umgekehrten Richtung (es wird angenommen, dass Links symmetrisch sind). Dies wird aber von der Protokollen unterschiedlich implementiert. DSR (Dynamic Source Routing) und AODV (Ad-hoc On-Demand Distance Vector) sind zwei bekannte on-demand Routingprotokolle.

2 Source Tree On-demand Adaptive Routing(SOAR)

2.1 Einführung

SOAR steht für Source Tree On-demand Adaptive Routing. SOAR ist ein On-demand Routingprotokoll, das Link-state Routinginformationen verwendet. Die Hauptidee von SOAR ist, dass drahtlose Routers sich minimale Source Trees austauschen. Diese minimale Source Trees bestehen aus Zuständen der Links in den von Routers verwendeten Pfade um Ziele von Datenpaketen zu erreichen. Um sein Source Tree zu bilden, verwendet ein Router seine ausgehende Links und die von seinen Nachbarn berichteten Source Trees. Dazu verwendet der Router einen Algorithmus zur Pfadsauswahl in seiner partiellen Topologie. Wie jeder On-demand Routingprotkoll, findet SOAR einen Pfad zum Ziel auf Basis einer Nachfrage in dem er Anfragen schickt und Antworten bekommt(Queries und Replies). Dieses Verfahren funktioniert nach dem folgenden Schema. Wenn ein Router einen Datenpaket zum Weiterleiten bekommt können sich folgende Scenarii spielen:

- Der Router hat einen Eintrag für das entsprechende Ziel in seiner Routingtabelle: der Paket wird bei dem nächsten Sprung(hop) weitergeleitet.
- Der Router hat keinen Weg für das Ziel des Datenpakets: er sendet seinen Nachbarn eine Anfrage (Query) nach den nötigen Informationen zum Bilden eines kompletten Pfads zum Ziel. Wenn ein Router die Anfrage empfängt und über keinen Pfad zum Ziel verfügt, leitet er die Anfrage an den eingenen Nachbarn weiter. Sonst antwortet er (Reply).

Um Schleifen und inkorrekte Paketweiterleitungen zu vermeiden, tauschen sich die Router Informationen über Aktualisierungen (Updates).

Modell:

Um SOAR zu beschreiben wird einen gerichteten Graph als Modell eingesetzt, mit folgenden Annahmen:

- Der Graph ist G=(V,E) wobei V die Menge der Knoten und E die Menge der Kanten sind.

- Jeder Knoten(Router) hat einen eindeutigen Identifikator.
- Router funktionieren fehlerfrei und die Information ist fehlerfrei gespeichert.
- Wenn ein Router einen neuen Knoten findet dann entsteht einen Link mit diesem. Mit jedem Link können Kosten verbunden sein. Diese Kosten sind unendlich wenn der Nachbarn nicht direkt erreichbar ist.

2.2 Gespeicherte Information

Einige Begriffe und Konzepte von SOAR müssen erstmals erläutert werden um das Protokoll richtig zu verstehen.

Schema:

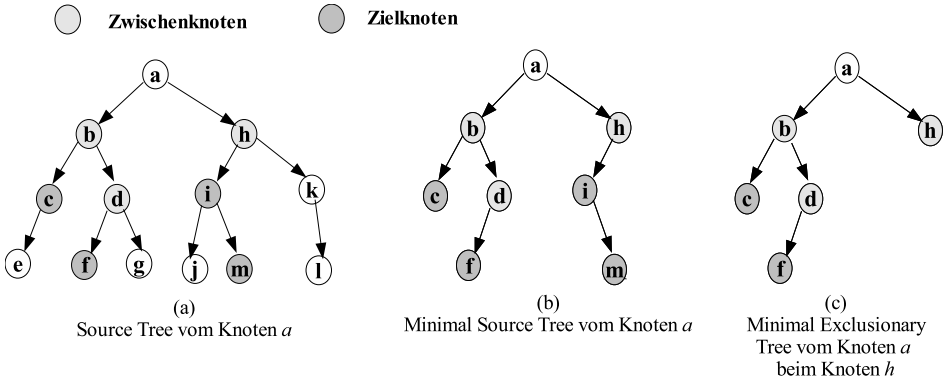

Abbildung 1: Gespeicherte Routinginformationen

- Wichtige Ziele eines Routers
  Es sind Knoten mit denen der Router einen aktiven Fluss hat.
- Source Tree
  Um bekannte Ziele zu erreichen benutzt ein Router eine Folge von Links. Ein Source Tree von einem Router ist ein Baum, der aus diesen Links besteht. Die Abbildung 1 (a) zeigt den Source Tree vom Router a.
- Minimaler Source Tree
  Der minimale Source Tree ist ein Unterbaum vom Source Tree. Er besteht aus Links zur wichtigen Zielen für den Router. Alle unwichtige Ziele werden im minimalen Source Tree nicht mitgennommen. Die Abbildung 1 (b) zeigt den minimalen Source Tree vom Router a. Auf dem Baum erkennt man nur Zielknoten und Zwischenknoten.
- Minimal Exclusionary Tree
  Wenn ein Router den minimalen Source Tree eines Nachbarns empfängt, bestimmt er die Wurzel dieses minimalen Source Tree. Der Unterbaum, der nur aus dieser Wurzelknoten besteht wird in der Topologie vom Router aufgenommen und heisst Minimal Exclusionary Tree. Es wird davon ausgegangen, dass aus dem Wurzel können die Knoten im Unterbaum gefunden werden. Die Abbildung 1 (c) zeigt der vom Knoten h benutzte Minimal Exclusionary Tree für den Knoten a.
- Routingtabelle
  Jeder Eintrag in der Routingtabelle enthält eine Zielidentität und der nächste Sprung (next hop) für dieses Ziel.

- Zieltabelle
  Enthält und pflegt Informationen, die andeuten ob ein Knoten k für das Ziel z wichtig ist oder nicht.

- partielle Topologie am Knoten i
  Die partielle Topologie am Knoten i wird durch die Aggregation von Minimal Exclusionary Trees von allen Nachbarn erzeugt.

Um seine Funktion zu erfüllen speichert und pflegt der Router fogende Informationen:

- eine partielle Topologietabelle
- seine Source Tree
- eine Routingtabelle
- Der minimale Source Tree von jedem Nachbarn
- eine Zieltabelle

Ein Router hat zusätzlich noch eine Abfragetabelle und eine Datenpuffer.

2.3 Auffinden einer Route

Einer der wichtigsten Aspekte der Routing ist die Wege- Nachbarnwahl. Dieses Verfahren wird realisiert indem die Router sich Nachrichten austauschen. Es handelt sich um Abfragen(queries), Antworte(replies), Aktualisierungsnachrichten (updates) und gedrängte Antworten und Aktualisierungsnachrichten(forced queries, forced replies).

- Abfragen(queries)
  Wenn ein Router einen Datenpaket, für den er keinen gültigen Pfad zum Ziel hat empfängt, sendet er eine Abfrage um einen Weg zu finden. Diese Abfragen werden in zwei Gruppen klassifiziert.
  - propagierenden Abfragen: die Abfrage wird an das ganze Netz gesendet.
  - nicht-propagierenden Abfragen: die Abfrage wird nur an Nachbarn gesendet.

  Nicht-propagierenden Abfrage werden gegenüber propagierenden Abfrage bevorzugt. So wird eine unnötige Überflutung des Netzes mit Abfragen vermieden(flooding), in dem Fall wo ein Nachbarn über den Weg zum Ziel verfügt.

- Antworten(replies)
  Ein Knoten sendet eine Antwort zu einer Abfrage wenn er einen Pfad zum angefordeten Ziel hat.

- Aktualisierungsnachrichten(updates)
  In dem Fall eines Wegabbruchs verwendet SOAR Aktualisierungsnachrichten(updates) um alternative Pfade zu finden und benachbarte Routers von der Wegänderung zu informieren. Wenn der Router feststellt, dass eine Weiterleitung eines Datenpakets zu eine Schleife führen wird, setzt er den Datenpaket ab und sendet eine Aktualisierungsnachricht. Auch wenn der Router keinen Weg für eine Paketweiterleitung hat, sendet er Aktualisierungsnachricht.

- gedrängte Antworte(replies)/Aktualisierungsnachrichten(updates)
  Es sind Kontrollpaketen, die nach einem Wegabbruch zum Einsatz kommen. Sie werden eingesetzt um einige relevante Knoten zu drängen, Pfade nach bestimmten Zielen aufzubauen.

Verfahren:
Wenn ein Router einen Datenpaket von einem Nicht-Nachbarn-Router empfät, nimmt er an, dass ein Link mit einem neuen Nachbarn aufgebaut ist. Bei einem Kantenausfall wird es angenommen, dass entweder eine Link-Layer-Protokoll SOAR informiert oder SOAR kann es selber feststellen nach einigen Sendungen zum Nachbarn.

Wenn ein Router einen Datenpaket empfängt und wenn er einen gültigen Pfad zum Paketziel hat, leitet er den Paket sofort weiter. Wenn er das Paketziel ist, antwortet er einfach. Wenn keinen Pfad für das Ziel bei dem Router vorhanden ist, leitet er ein Verfahren zum Auffinden einer Route ein. Während dieses Verfahrens behält er den Datenpaket in seinem Datenpuffer.

Das Auffinden einer Route wird gestartet indem der Router eine Abfrage sendet. Eine Abfrage nach einem bestimmten Ziel wird von einem Knoten nur unter folgenden Bedingungen weitergeleitet:

- Der Router hat keinen Pfad zum Ziel
- Die Abfrage hat seine festgelegte Anzahl von Sprünge(hops) nicht überschritten.
- Der Unterschied zwischen die aktuelle Zeit und die Zeit der letzten Weiterleitung für das Ziel ist grösser als ein vordefiniertes Zeitintervall zwischen zwei Weiterleitungen:$query\_fwd\_time$

Die zweite Bedingung schränkt die Abfragen in einem bestimmten Bereich. Die dritte Bedingung reduziert die Propagation von Abfragen im Netz wenn sie verschiedene Quellen und dasselbe Ziel haben.

Ein Knoten sendet eine Antwort zu einer Abfrage wenn er einen Pfad zum angefordeten Ziel hat. Er leitet eine Antwort unter folgende Bedingungen weiter:

- Der Router hat einen Pfad zum Antwortziel
- Der Router hat den neuen Weg zur Paketquelle erst nach dem Empfang der Antwort entdeckt (vermeidet mehere Antworte)
- Der Router ist ein Knoten im Pfad von der Quelle zum Ziel(vermeidet Überflutung von Antwortswege).

Wenn ein Router eine Abfrage empfängt und weiterleitet, markiert er die Abfragequelle als wichtig. So beinhaltet der Abfragepaket Links, die den Pfad zur Quelle bilden. Damit werden die Antworten an die richtige Quelle zurückgesendet. Wenn ein Router sendet oder leitet eine Antwort weiter, markiert er das Ziel, für das das Auffinden einer Route initiiert wurde als wichtig. So trägt der minimale Source Tree den Pfad zum Ziel. Nach der Zusammensetzung des Pfads werden Datenpakete Knoten pro Knoten weitergeleitet. Wenn der Weg abbricht, verwendet SOAR Aktualisierungsnachrichten(updates) und gedrängte Aktualisierungsnachrichten und Antworten(forced updates, forced replies) um alternative Pfade aufzubauen und

den Nachbarn zu benachrichtigen.

Falls der Auslöser der Abfrage überhaupt keine Antwort bekommt, versucht der Router immer wieder bis eine bestimmte Anzahl von Versuche erreicht ist. Danach benachrichtigt der Router seine obere Schicht, dass das Ziel nicht erreichbar ist. Die obere Schicht musst danach bestimmen ob der Fluss beendet wird oder nicht.

2.4 Behandlung von Kantenausfällen

Wenn ein Router entdeckt, dass die Kosten von seinem Pfad zum Ziel sich erhöht haben, benachrichtigt er seinen Nachbarn indem er Aktualisierungen sendet. Wenn dagegen Kosten zu wichtigen Zielen senken oder unverändert bleiben, kann ein Knoten Nachfolger wechseln ohne die Nachbarn zu benachrichtigen, weil diese Operation keine Schleifen verursachen kann.

Schema:

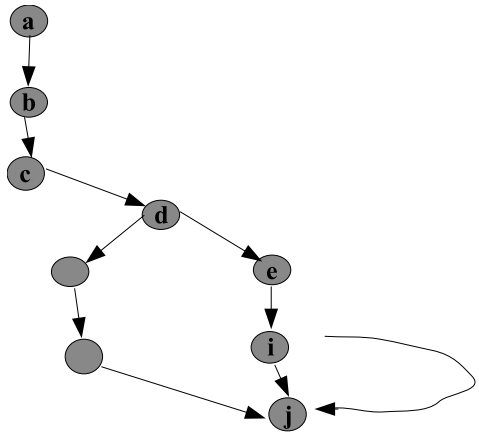

Abbildung 2: Behandlung von einem Wegfehler

Betrachten wir das Scenario auf der Abbildung 2. Nehmen wir an es existiert einen aktiven Fluss zwischen a und j und der Pfad ist *abcdeij*. Und nun fällt die Kante(i,j) aus! Wie soll diesen Ausfall behandelt werden? Wenn der Knoten i den Ausfall entdeckt, versucht er einen alternativen Pfad zu finden. Wenn der alternative Pfad höhere Kosten als die Kosten des bisherigen Pfad hat, benachrichtigt i seinen Nachbarn. Der Source Tree von i enthält implizit die Information über den Ausfall von (i,j).Diese Information propagiert zu den obenliegenden Knoten bis ein Knoten, der einen alternativen Pfad zu j mit den gleichen oder niedrigeren Kosten gefunden wird. Wenn so ein Knoten nicht gefunden wird, propagiert die Information bis zur Quelle zurück. Unter diesen Umständen initiiert die Quelle einen neuen Prozess zum Auffinden einer Route. Hier ist zu beachten, dass die Knoten zwischen Quelle und Senke lokal den Weg zu reparieren versuchen. Das zeigt ein Vorteil von SOAR gegenüber DSR und AODV Protokolle wo einen Ausfall immer zur Quelle zurück geführt wird. Damit ist in SOAR die Anzahl der beteiligten Knoten im Netzverkehr niedriger als in AODV und DSR. Wenn ein Router keinen Weg zu einem Ziel für einen empfangenen Datenpaket hat, sendet er eine Aktualisierungsnachricht(update). Im Falle einer Folge von Datenpaketen, für die den Router keinen Weg hat kann es zu einer potentiellen Schleife führen. Um dies zu vermeiden, ist eine Aktualisierungszeit(update time) in der Sendung von konsekutiven Aktualisierungen gesetzt.

2.5 Source Tree Darstellung

Schema:

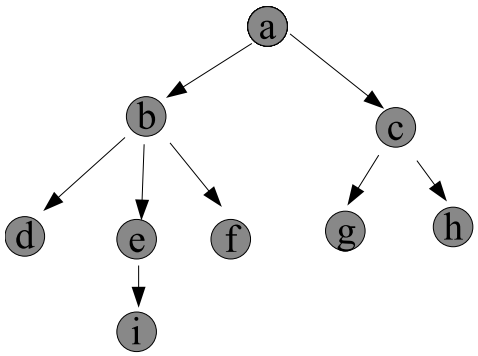

Abbildung 3: Bitmap-Darstellung von minimal source tree

Bitmapdarstellung - Breadth First Search Walk
a,b,c,d,e,f,g,h,i
Bitmap 1,0,0,1,0,0,0,1,0,0,1,1,0,1,1,1,1
Die kommenden Nullen nach einem Eins repräsentieren die Kinderknoten. Eins repräsentiert der Vaterknoten. Bei einem Anzahl von n Knoten sind 2n-1 Bits in der Bitmapdarstellung vorkommen.

2.6 Beispiel einer SOAR-Operation

Mit Hilfe der Abbildung 4 wird hier eine SOAR-Operation vorgestellt. Aus Vereinfachungsgründen, werden folgende Annahmen getroffen:

- Jeder Knoten hat Datenpakete von jedem anderen Knoten im Netz, so dass jeder ist für jeden wichtig.
- Das Netz hat diesselbe Sequenznummer für jeden Knoten

Das Beispiel auf dem Abbildung 4 zeigt die Änderung der partiellen Topologietabellen beim Knoten a nachdem der Link (b,c) ausgefallen ist. Die Regeln zur Behandlung von Linkausfällen werden hier verwendet.

Bild a: Netztopologie. Die Sequenznummern für jeden Knoten sind in Klammern eingetragen. Die aktuelle Sequenznummer eines Kontens ist von allen anderen bekannt.

Bild b: Hier ist die Darstellung von dem Miminal Exclusionary Tree vom Knoten b beim Knoten a

Bild c: Hier ist die Darstellung von dem Miminal Exclusionary Tree vom Knoten f beim Knoten a

Bild d: Das Bild zeigt die partielle Topologie vom Knoten a. Der Miminal Exclusionary Tree von b und der von f sind dort erkennbar.

Bild e: Hier ist die Netztopologie nachdem der Link (b,c) ausgefallen ist.

Bild f: Wenn (b,c) ausfällt erhöht b seine Sequenznummer um 1; sie wird 35. Der Pfad nach dem Knoten c bricht bei b ab und sendet eine Aktualisierung um die Änderungen in seinem minimalen source tree zu melden. Der Knoten a empfängt die Aktualisierung und aktualisiert die Einträgen des minimalen source tree von b.

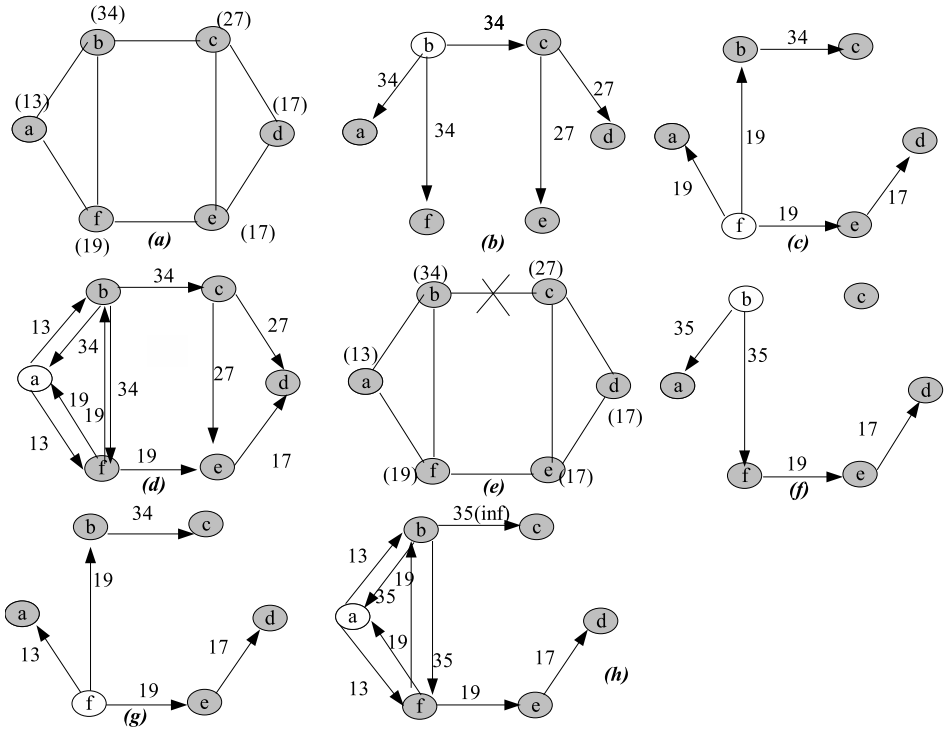

Abbildung 4: Beispiel einer SOAR-Operation

Bild g: Da keine Aktualisierung vom Knoten f gekommen ist, bleibt sein minimal exclusionary tree unverändert. Die Links (c,d) und (c,e) gehört nicht zum minimalen Source Tree von f. Sie werden von der partiellen Topologie von a gelöscht.

Bild h: Der von f verwaltete Link (b,c) hat die Sequenznummer 34(kleiner als 35). Der Knoten a markiert daher (b,c) als einen Link mit Squenznummer 35 und setzt die Kosten auf Unendlich. Die Kosten von (b,c) wurden auf undenlich gesetzt weil der benachbarte Knoten d, der aufgehört hat den Link zu benutzen, hat die höchste Sequenznummer für diesen Link. Der Vorteil dieser Technik ist, dass Routers über die Unbenutzbarkeit eines Links informiert werden ohne eine explizite Benachrichtigung von Linkausfällen. Die modifizierte Netztopologie wird zur Wegberechnungen verwendet [Roy01].

2.7 Vorteile von SOAR

Aufgrund der Benutzung vom minimalen Source Tree verwendet SOAR eine kleine Anzahl von Kontrollpaketen. In der Behebung von Wehgfehlern bietet SOAR im Vergleich zu AODV und DSR eine bessere Lösung. Wenn ein Link ausfällt hat SOAR öfters als andere Protokolle die Möglichkeit einen alternativen Pfad einzusetzen ohne eine Routeauffindung neu initiieren zu müssen. Durch den Source Tree Technik in SOAR wird eine umfangreiche Routingdatenbank aufgebaut.

3 On-Demand Link Vector Protocol(OLIVE)

3.1 Einführung

Wie SOAR ist OLIVE ein weiteres On-demand Link-state Routingprotokoll. OLIVE bringt mit sich zwei Vorteile gegenüber SOAR:

- OLIVE garantiert ein schleifenfreies Routing(loop free routing) jederzeit
- OLIVE braucht braucht keine Informationen über traversierte Pfade um Zyklen zu entdecken.

Die Anforderungen an OLIVE waren:

- Ein Routingalgorithmus, dass Pfadinformationen nach Abfrage verwendet(on-demand)
- Schleifenfreies Routing
- Der Algorithmus soll eine lokale Behebung von Wegfehelern ermöglichen und soll keine Informationen im Kopf der Datenpaketen verwenden trotz der Einführung der ersten Anforderung.

So legen die Anforderungen fest, dass OLIVE ein schleifenfreies Protokoll, das Source Tree Informationen verwendet sein soll.

In OLIVE sendet der Router Wegabfragen(Route Requests - RREQS) um einen Weg zu den Zielen, mit den er kommunizieren will zu wählen. Die entsprechende Ziele bzw. die Knoten, die aktive Pfade zu diesen Zielen haben antworten mit den sogennanten Route Replies (RREP). Die RREP enthalten Pfaden zu den Zielen und werden an den Sender zurückgeschickt. Die Aggregation von Pfadinformationen in RREP bilden eine partielle Topologie. Ein Router wählt nur Pfade, die von seinen Nachbarn ausgesucht wurden.

Algorithmus zum Auffinden einer Route:
Sei $P_j^{n_i}$ der Pfad von i nach j, vom Nachbarn n_i berechnet wobei $n_i \epsilon N_i$. N_i ist die Menge von Nachbarn von i.
Sei $P_j^{n_i}.cost$ die zum Pfad $P_j^{n_i}$ assoziierten Kosten.
Der Pfad $P_j^{n_x}$ ist gewählt wenn $P_j^{n_x} = min(P_j^{n_i}.cost)$
Also, von den Nachbarn von i berechneten Pfade wird den Pfad mit den geringsten Kosten ausgewählt.
Wenn eine Kante ausfällt, der benachbarte Knoten im Pfad versucht der Weg zu reparieren in dem er nach einen alternativen Pfad sucht ohne seinen im Pfad obenliegenden Knoten zu informieren. Dazu benutzt er seine partielle Topologie um herauszufinden ob einen alternativen Weg überhaupt existiert. Wenn so einen Pfad existiert, die Knoten auf diesem Pfad tauschen sich gedrängte Anfragen und gedrängte Antworten(FRREQS - FRREPS) aus um die Anlegbarkeit des Pfads zu überprüfen. Im negativen Fall, ein Fehler(RERR - Request Error) wird an den obenliegenden Knoten gemeldet.

3.2 Beispiel einer OLIVE-Operation

Um einer OLIVE-Operation zu beschreiben werden zwei Fälle betrachtet. Der Fall wo alle Kanten vorhanden sind und der Fall eines Kantenausfalls. Das Scenario ist auf dem Abbildung 4 dargestellt.

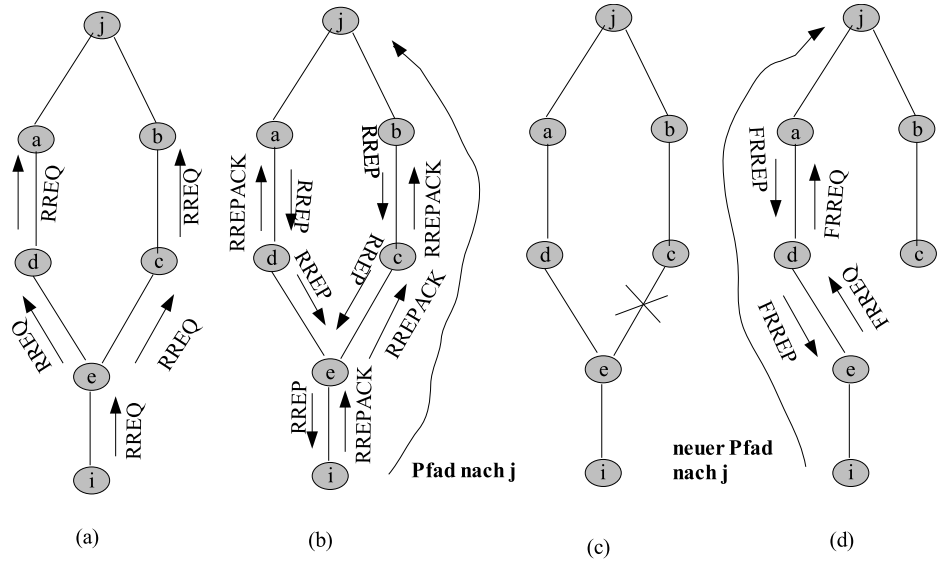

Abbildung 5: Beispiel einer OLIVE-Operation

Der Knoten i muss einen Paket zum Knoten j schicken. Der Knoten i steht also vor der Aufgabe einen Weg nach j zu finden. Es wird auch angenommen, dass a und b einen aktiven Fluss mit j haben. Erstmal sendet i eine Abfrage an seinen eigenen Nachbarn. Wenn er keine Antwort bekommt, sendet er erneut eine Abfrage, aber dieses Mal über das ganze Netz(Bild a). Bild b zeigt wie RREPs, die Wege zum Ziel tragen, und RREPACKs zwischen den Routern ausgetauscht werden. RREQS beinhalten Informationen über den Auslöser des Prozesse(Quelle) und das Ziel. Das Ziel oder jeder Zwischenknoten mit einem gültigen Pfad zum Ziel können RREQs mit RREPs beantworten. So transportieren RREQs Informationen über traversierte Pfade und diese Informationen stehen RREPS zu Verfügung um den Rückweg zu finden. Im Beispiel haben a und b aktive Wege nach j. Der RREP vom Knoten a ist für d bestimmt. Wenn a diesen RREP an d sendet, meint er damit einen aktiven Pfad zu j zu haben, und hat auch damit d in der Liste seiner Vorgänger aufgenommen. So kann d benachrichtigt werden, wenn der Weg nach j sich ändert. Wenn a nach einem bestimmten Zeit keine Bestätigung bekommt, entfernt er d von seiner Vorgängerliste. Daher muss d einen RREPACK nach a schicken um seinen Weg nach j auszuwählen. Derselbe Scenario findet mit den Knoten b und c statt. Der Knoten e bekommt beide RREPs aus den Knoten c und d. Da die Pfade in Kosten gleich sind, nimmt e den ersten bei ihm eingetroffenen Weg an. Nehmen wir an der Knoten e empfängt ein RREP vom c zuerst, dann sendet e ein RREPACK an c(nicht an d!). Es ist wichtig anzumerken, dass den von d gefundenen Weg in der Netztopologie von e eingetragen wird. Diese Information wird verwendet um alternative Wege zu finden im Falle eines Ausfalls. Zum Schluss sendet e den RREP an i und i sendet ein RREPACK zurück an e und wählt den endgültigen Pfad *iecbj* aus. Jetzt wird den zweiten Fall, wo ein Link ausfällt betrachtet. Der Link (e,c) fällt aus(Bild c). Damit ist der Nachfolger von e auf dem Weg nach j nicht mehr erreichbar. Aus diesem Grund entfernt e den Knoten c von seiner Routingtabelle. Da seinen Weg zu j abgebrochen ist, versucht e eine lokale Reparation des Weges anhand seine Topologie ohne i zu informieren. In seiner eigenen Netztopologie findet i den Pfad *edaj* als Alternative um j zu erreichen. Aber e ist nicht der aktuelle Vorgänger von d für das Ziel j. Das heisst der Pfad ist für e nicht aktuell(up-to-date). Der Knoten e initiiert jetzt eines Prozess namens *forced routing*, bei dem die Knoten sich Informationen austauschen um die Gültigkeit des Pfades zu überprüfen. FRREQs(Forced Routing Requests) werden dem Pfad entlang weitergeleitet(*edaj*) und jeder Knoten, der einen aktiven Pfad zu j hat sendet einen FRREP(Forced Routing Response). Ein FRREP wird von a nach d und von d nach e gesendet und e stellt den alternativen Weg *edaj* auf. Wenn es keine alternativen Wege gäbe,

hätte der Knoten e an i eine RERR(Request Error) weitergeleitet und i hätte einen neuen Prozess zum Auffinden einer Route gestartet, weil ihm keinen alternativen Weg bekannt ist [Roy02].

3.3 Ausfürliche Beschreibung von OLIVE

OLIVE Operationen können in drei Phasen klassifiziert werden.

- Auffinden einer Route um neue Pfade zu setzen
- lokale Behandlung von Ausfällen um alternative Pfade zu finden wenn der Weg abbricht
- Benachrichtigung eines Wegfehlers um benachbarte Knoten zu aktualisieren.

Auffinden einer Route:
In diesem Prozess gibt es drei verschiedene Typen von Kontrollpaketen: RREQ, RREP, RREPACK

- Route Request (RREQ)
  Diese Pakete werden verwendet um Wege für unbekannte Ziele zu finden. Solche Abfragen werden an Nachbarnknoten gesendet. Wenn diese keinen Weg für das Ziel haben, wird die Abfrage über das ganze Netz geschickt. Ein Knoten leitet einen Paket unter folgenden Bedingungen weiter:
  - Er hat keinen gültigen Weg zum Ziel
  - Kein RREQ wurde aus derselben Quelle innerhalb einer bestimmten Zeit weitergeleitet.
  - RREQ ist nicht ausserhalb der Suchbereich

  Jeder RREQ beinhaltet Informationen über den Sender, das Ziel und eine Liste der Identitäten der weiterleitenden Knoten. Alle diese Informationen werden verwendet um Antworten zur Quelle zurückzuschicken.
- Route Response (RREP)
  RREP werden von Knoten, die aktive Wege zum Ziel haben gesendet. Wenn ein RREP als unmittelbare Antwort zu einem RREQ gesendet wird, wird er an allen Nachbarn ausgestrahlt(broadcast). Bevor ein Knoten eine RREP austrahlt wartet er eine Back-off Periode und bricht die Operation ab, wenn er ein RREP von einem anderen Knoten bekommt, der für dieselbe Quelle bestimmt ist. Wenn es dagegen um eine Weiterleitung geht, wird der RREP genau an einem Knoten(der Nachbarn auf dem Pfad) gesendet(unicast). Jeder Knoten stellt erstmals fest, dass sein Nachbarn in seiner Vorgängerliste steht bevor er ihm ein RREP sendet. So kann der Knoten benachrichtigt werden wenn der ursprüngliche Weg abbricht.
- Route Response Acknowledgement(RREPACK)
  Ein Knoten bekommt Pfadinformationen von seinen Nachbarn, über die er einen Weg aufbaut. Dann sendet der Knoten eine Bestätigung zum Nachbarn(RREPACK). Wenn der Nachbarn die Bestätigung empfängt , löscht er den *ReplyAckTimer*. Der *ReplyAckTimer* ist ein Timer, der nach der Sendung eines RREPs gestartet wird. Ist eine Bestätigung innerhalb einer bestimmten Zeit(timeout) nicht eingetroffen, wird der Vorgängerknoten von der Vorgängerliste gelöscht.

Lokale Behandlung von Ausfällen:
Wenn der ursprüngliche Weg abbricht, versucht der benachbarte Knoten zum ausgefallenen Link lokal eine Alternative zu finden, ohne die obenliegenden Knoten zu informieren. Diese Methode ist vorteilhaft, da nur ein Teil des Netzes sich mit der Fehlerbehandlung beschäftigt. Im Laufe dieses Verfahren werden zwei Typen von Datenpaketen ausgetauscht.

- Forced Route Request (FRREQ)
  Die gefundene alternative Wege sind warscheinlich nicht aktuell, weil sie kürzlich nicht für Datenübermittlung nicht verwendet wurden. Daher werden FRREQ diese Pfade entlang gesendet um ihre Anlegbarkeit zu überprüfen. Jeder FRREQ trägt Informationen über den möglichen Weg zum Ziel, so dass jeder Knoten auf dem Pfad seinen aktuellen Weg mit dem Pfad vergleichen kann. Der Router leitet den FRREQ weiter, wenn er nicht über die Informationen über die Anlegbarkeit des Pfades verfügt. Sonst antwortet er mit einem FRREP. Wenn FRREQs innerhalb einer gewissen Zeit keine Antwort bekommen, gelten alternative Pfade als nicht vorhanden. Dies wird dem Vorgänger gemeldet.

- Forced Route Reply (FRREP)
  FRREP sind die Antworten an FRREQ wie der Name schon verrät. FRREPs beinhalten entweder einen gültigen Pfad zum Ziel oder keinen Weg in dem Fall wo der alternative Pfad nicht existiert. In dem zweiten Fall wird im FRREP eingetragen, dass der erste Link in dem vermuteten alternativen Pfad unendlich ist. Im Anschluss ist ein RERR an den Auslöser des Verfahrens gesendet, wenn er ein Zwischenknoten ist und eine RREQ wenn er die Quelle ist.

Benachrichtigung eines Wegfehlers
Bei einem Wegabbruch, wenn es keine billigere Alternative gefunden wird, startet ein Router einen Prozess zur Benachrichtigung eines Wegfehlers. Während dieses Verfahrens werden Vorgänger von dem Fehler benachrichtigt, so dass jeder nach dem anderen an seiner Stelle den Weg zu reparieren versucht. Dabei werden RERRs und RERRACKs ausgetauscht.

- Route Error (RERR)
  RERRs werden an den Vorgänger gesendet um einen Wegfehler zu melden, und der Sender wird nicht als Nachfolger benutzt. RERRs tragen Informationen über den ersten Link im vermuteten Alternativpfad. Die Kosten des Links werden auf Unendlich gesetzt.

- Route Error Acknowledgement(RERRACK)
  RERRACK wird als Bestätigung zum Empfang eines RERRs gesendet. Damit ist der RERR-Sender informiert, dass seine Nachricht angekommen ist und er wird nicht mehr als Nachfolger benutzt. Genauso wird der RERRACK-Sender nicht mehr als Vorgänger behandelt.

3.4 Nachbarnbeziehungen

Um richtige und korrekte Informationen zu bekommen, hängt einen Router von Benachrichtigungen aus dem Netzwerkschicht oder aus dem Link-Schicht(link layer)ab. Bei jedem Knoten ist ein Link zum Nachbarn vorhanden, wenn einer von den drei folgenden Ereignissen eintritt:

- Der Knoten bekommt zum ersten Mal einen Kontrollpaket vom Nachbarn.
- Der Knoten bekommt die erste Meldung(hello Message) vom Nachbarn.
- Ein Nachbarn-Protokoll bei dem Linkschicht meldet einen neun Nachbarn.

Ein Knoten erklärt seinen Link zu einem Nachbarn als ausgefallen wenn einer der folgenden Ereignissen eintritt:

- Der Router empfängt eine Benachrichtigung vom Linkschicht wenn er einen Datenpaket über den Link nicht senden kann.
- Die Hello-Meldungen der Netzwerkschicht werden mehrmals vermisst.
- Ein Nachbarn-Protokoll in der Linkschicht meldet einen Link-Ausfall.
- Keine Bestätigung wurden bekommen nach wiederholten Sendungen aus dem Netzwerschicht.

3.5 Sequenznummern

Jeder Knoten verwaltet seine eigene Sequenznummern und wenn einer von seinen benachbarten Links ausfällt, erhöht er seine Sequenznummer. Diese Sequenznummer wird dem ausgefallenen Link zugewiesen. Konflikten können entstehen, da Links von verschiedenen Nachbarn verwaltet werden. In solche Situationen werden Links mit höheren Sequenznummer bevorzugt. Wenn es keinen Eintag für einen Link vorhanden ist, nimmt der Router der erste Link-Zustand(link-state), der er bekommt. Links mit unendlichen Kosten werden nie gelöscht. Links mit endlichen Kosten werden gelöscht nachdem jeder benachbarte Knoten diesen Link von seinem eingetragenen Pfad entfernt hat. Wenn die Information über ausgefallenen Links in jedem Router im Netz gespeichert ist, kann die Bandbreite gespart werden.

3.6 Vorteile von OLIVE

Routers in OLIVE tauschen sich Pfadinformationen aus um eine partielle Netztopologie zu bilden. Der Algorithmus zur Routeauffindung ist auf der partiellen Netztopologie ausgeführt um den *source graph* zu berechnen. Mit Hilfe dieser Topologieinformationen wird eine netzweite Suche verhindert und es wird genauso die Möglichkeit einer lokalen Behebung von Wegfehlers gegeben. OLIVE bietet jederzeit ein schleifenfreies Routing und findet einen korrekten Pfad zum Ziel in endlicher Zeit.

4 Fazit

Die Untersuchung von OLIVE und SOAR hat gezeigt, dass diese Protokolle effizienter als aktuelle state-of-the-art On-demand Routinprotokolle und Link-state Routingprotokolle wie AODV und DSR. OLIVE und SOAR verwenden sowohl die Link-state Technik, bei der eine verteilte Datenbank aufgebaut und gepflegt wird als die On-demand Technik, bei der eine Abfrage über das ganze Netz gesendet ist. Der gezielte und effiziente Einsatz beider Techniken durch On-demand Link-state Routingprotokolle in Ad-hoc-Netze ist vorteilhaft und macht den Nutzen dieser neuen Protokolle deutlich. Ein qualitiativ besseres Routing wird erreicht, weil Schleifen werden vermeidet und Wege nach einem Ziel werden immer innerhalb akzeptabler Zeit gefunden. Damit ist eine neue Klasse von Routingprotokolle in Ad-hoc-Netzen entwickelt, die sich vielleicht durchsetzen werden.

Literatur

[Jacq03] C. Adjih E. Bacelli P. Jacquet. Link State Routing in Ad-Hoc Networks. Forschungsbericht - http://www.inria.fr/rrrt/rr-4874.html, Juli 2003.

[Roy01] J.J. Garcia-Luna-Aceves Soumya Roy. Using Minimal Source Trees for On-Demand Routing in Ad-hoc Networks. Paper, April 2001.

[Roy02] J.J. Garcia-Luna-Aceves Soumya Roy. An Efficient Path Selection Algorithm for On-Demand Link-State Hop-by-Hop Routing. Paper, Oktober 2002.

[Roy03] Soumya Roy. *On-Demand Link-State Routing in Ad-hoc Networks*. University of California Santa Cruz. 2003.

Abbildungsverzeichnis