Dienstgüte in Mobilen Ad Hoc Netzen

Dennis Gräff

Abstract

In mobilen Ad Hoc Netzen Dienstgüte zu unterstützen ist durch begrenzte Bandbreiten und Rechenleistung, sowie durch die dynamische Natur dieser Netze sehr komplex. Die Forschung, die in Dienstgüte im Internet und andere drahtgebundene Netzarchitekturen investiert wurde, lässt sich aus diesem Grund nicht einfach auf MANETs übertragen.

Diese Arbeit befasst sich zum einen mit allgemeinen Ansätzen zur Dienstgüteunterstützung in einem MANET, mit konkreten Modellen und Protokollen, die bereits entwickelt wurden und zum anderen mit den besonderen Herausforderungen und Problemen, die dieses Gebiet mit sich bringt und wie man lösen kann.

1. Einleitung

Drahtlose Netze erfreuen sich immer größerer Beliebtheit. Sie bieten Benutzern Dienste an, die sie unabhängig von ihrem Aufenthaltsort benutzen können. Sie lassen sich mit konventionellen, drahtgebundenen Netzen verbinden, so dass die Benutzer der drahtlosen Netze beispielsweise Internet-Angebote in Anspruch nehmen können, ohne an die begrenzte Reichweite eines Kabels gebunden zu sein. Momentane Lösungen das Internet auf drahtlose Netze zu erweitern verlangen noch, dass sich jeder mobile Knoten in direkter Kommunikationsreichweite einer Basisstation befindet. Es wäre aber durchaus denkbar, völlig ohne feste Infrastruktur auszukommen. Die mobilen Knoten stehen nur untereinander in Verbindung. Jeder Knoten fungiert dabei als Router. Dadurch ist es einem Benutzer möglich, Dienste in Anspruch zu nehmen, die außerhalb der Reichweite seines mobilen Gerätes liegen, aber über andere Knoten als Zwischenstationen erreicht werden können. Diese Art von Netz nennt man Mobiles Ad Hoc Netz (MANET). Mit ihnen kann Kommunikation in Umgebungen, in denen keine drahtgebundenen Netze verfügbar sind, ermöglicht werden; wie beispielsweise in Katastrophengebieten. Auch als Campusnetz, bei Konferenzen oder in elektronischen Klassenzimmern wären ihr Einsatz denkbar.

Ein MANET kann völlig autonom betrieben werden – mit eigenen Routing-Protokollen und Mechanismen zum Netzwerkmanagement. Oder es kann als Erweiterung des Internets dienen und den Benutzern des MANETs Zugang zu Diensten im Internet verschaffen.

Der zunehmenden Beliebtheit und Verbreitung von Multimedia-Applikationen und der möglichen kommerziellen Nutzung von MANETs würde die Einführung von Dienstgüte zu Gute kommen.

Dienstgüte kann in einem Netzwerk einer Applikation bzw. einem Benutzer bestimmte Garantien zu verfügbarer Bandbreite, zu Latenz, Jitter oder maximaler Paketverlustrate geben.

Für das Internet oder infrastruktur-basierte Netze im Allgemeinen, existieren bereits diverse Möglichkeiten zur Dienstgüteunterstützung. Leider gibt es wesentliche Unterschiede zwischen diesen, „klassischen“ Netzen und MANETs. Eine direkte Übertragung der Lösungen ist nicht möglich. Die verfügbare Bandbreite ist in diesen Netzen viel geringer, also muss Overhead so weit wie möglich vermieden werden. Die dynamische Topologie der Netze durch die Mobilität der einzelnen Knoten erschwert es, Informationen über alle Links in einem größeren Gebiet zu halten. Dazu kommen die möglicherweise stark begrenzten Ressourcen der einzelnen Knoten. Daher müssen für MANETs völlig neue Dienstgütemodelle entwickelt werden.

In Kapitel 2 werden zunächst die Dienstgütemodelle des Internets und FQMM, das erste Modell für MANETs, vorgestellt. Kapitel 3 befasst sich mit den Signalisierungsprotokollen RSVP, INSIGNIA und SWAN. In Kapitel 4 wird das Dienstgüte-Routing in MANETs behandelt, insbesondere CEDAR und Ticket-Based Probing. Schließlich wird in Kapitel 5 ein kurzer Blick auf die besonderen Probleme beim Medienzugriff geworfen.

2. Dienstgütemodelle

Ein Dienstgütemodell spezifiziert die grundlegende Architektur des Systems, in der bestimmte Dienstklassen angeboten werden. Für das Internet hat die Internet Engineering Task Force (IETF) zwei Dienstgütemodelle entwickelt: IntServ und DiffServ. Auf Grund der besonderen Eigenschaften von MANETs, z.B. der dynamischen Topologie, lassen sich diese jedoch nicht direkt auf MANETs anwenden. Dennoch sind einzelne Ansätze aus beiden Modellen durchaus brauchbar und werden deshalb im Folgenden kurz vorgestellt, bevor gezeigt wird, wie in FQMM, einem Dienstgütemodell speziell für MANETs, diese Ansätze umgesetzt wurden.

2.1 IntServ

Die Idee hinter dem Integrated-Services-Modell (IntServ) [XN99] ist, in jedem Router flussspezifische Zustandsinformationen, wie erforderte Bandbreite, Verzögerung, etc. zu speichern. Diese Vorgehensweise wird per-flow-Management genannt. IntServ geht von zwei Service-Klassen zusätzlich zum üblichen Best-Effort Service aus: Guaranteed-Service und Controlled-Load Service.

Guaranteed-Service bietet, wie der Name schon sagt, einer Anwendung Garantien für verfügbare Bandbreite und Verzögerung. Controlled-Load Service stellt der Anwendung einen erweiterten und zuverlässigen Best-Effort Service bereit, in dem es ein unbelastetes Netz vortäuscht.

IntServ wird mit vier Komponenten implementiert:

- einem Signalisierungsprotokoll
- der Zugangskontrolle
- dem Classifier und

- dem Paket-Scheduler.

Als Signalisierungsprotokoll wird das Resource-Reservation-Protocol, kurz RSVP, benutzt. Anwendungen, die eine der erweiterten Serviceklassen in Anspruch nehmen wollen, benutzen RSVP um vor der eigentlich Übertragung Ressourcen zu reservieren. Die Zugangskontrolle entscheidet, ob die Reservierung erfüllt werden kann. Dabei überprüft jeder Router entlang des Weges unabhängig, ob er ausreichend verfügbare Ressourcen hat, um die Anforderung zu erfüllen. Beginnt nach bestätigter Reservierung die Datenübertragung, so sorgt der Classifier für die Identifizierung der gesondert zu behandelnden Pakete und der Paket-Scheduler schließlich für die Einhaltung der geforderten Parameter.

IntServ hat zwei wesentliche Nachteile: Es ist kompliziert und es skaliert schlecht. Daher ist es für MANETs nicht geeignet. Jeder mobile Netzteilnehmer müsste in diesem Modell die Zugangskontrolle, einen Classifier und einen Paket-Scheduler betreiben. Dazu kommt der Speicherbedarf für die Zustandsinformationen der einzelnen Datenströme. Diese Aufgaben würden die mobilen Hosts auf Grund ihrer limitierten Ressourcen übermäßig stark belasten. Außerdem würden durch die RSVP-Pakete ein zu großer Teil der verfügbaren Bandbreite in einem MANET verloren gehen, da die Mobilität der Knoten häufige Neureservierungen nötig macht.

2.2 DiffServ

Wegen der Nachteile von IntServ wurde Differentiated Service (DiffServ) [XN99] als Alternative entwickelt. Das DiffServ-Konzept sieht einige wenige Dienstgüteklassen vor. Jede dieser Klassen wird kombiniert mit einem Satz von Regeln, genannt Per-Hop-Behavior.

Zentrale Elemente eines Netzes sind „DiffServ Clouds“ oder „DS-Domänen“. Zwischen den Betreibern dieser Domänen gibt es Service-Level-Agreements, mit denen festgelegt wird, wie einzelne Datenströme innerhalb der Domänen zu behandeln sind. Sie definieren also letztendlich das Per-Hop-Behavior.

Kommt ein Datenpaket an einem Router an der Grenze einer solchen Domäne an, erhält es im DS-Feld des Headers eine Markierung entsprechend der vorgesehenen Dienstklasse des Paketes. Als DS-Feld wird das Type-of-Service-Byte bei IPv4 bzw. das Traffic-Class-Octett von IPv6 verwendet. Alle folgenden „inneren“ Router schauen sich nur das DS-Feld an und behandeln das Paket entsprechend des der Serviceklasse entsprechenden Per-Hop-Behaviors. Sie brauchen also keine Informationen über einzelne Datenströme zu speichern.

Das DiffServ-Modell unterstützt mehrere Serviceklassen, wie z.B. Premium-Service mit garantierter Bandbreite und geringem Verlust, geringem Jitter und geringer Verzögerung, Assured-Service, inklusive der drei Varianten des Olympic-Service (Gold, Silber, Bronze), für Anwendungen, die höhere Zuverlässigkeit als Best-Effort benötigen.

DiffServ wäre durchaus ein mögliches Modell für Dienstgüte in MANETs, da es die inneren Router einer Domäne nicht übermäßig belastet. Allerdings gibt es einige Probleme bei der Anwendung von DiffServ auf MANETs.

Zum einen gibt es keine eindeutigen Grenzen einer DS-Domäne. Jeder Knoten könnte „innerer“ oder „Rand-Router“ sein. Also müsste auch jeder Knoten alle Funktion beherrschen, was wiederum zu hoher Belastung der Knoten führt.

Das zweite Problem ist das Konzept der Service-Level-Agreements (SLA). Diese Vereinbarung legt fest, welche Dienstklassen zu welchen Bedingungen zur Verfügung

stehen. Im Internet schliesst ein Kunde ein solches SLA mit seinem Service Provider ab. Solche Vereinbarungen für MANETs zu treffen ist schwierig.

2.3 FQMM

Als erstes Dienstgütemodell für MANETs wurde FQMM (A Flexible QoS-Model for MANET) [WH] entwickelt. Die Lösungen, die in drahtgebunden Netzen existieren, wurden kombiniert und ein Modell kreiert, dass die besonderen Eigenschaften von MANETs beachtet.

Das Modell bietet die Möglichkeit von per-flow-Management wie IntServ, als auch Serviceklassen vergleichbar zu denen von DiffServ. Dabei werden Daten mit der höchsten Priorität per-flow gehandhabt, alles andere durch Serviceklassen abgedeckt. Es wird von der Annahme ausgegangen, dass nur ein kleiner Teil der Datenströme die höchste Priorität braucht. Ansonsten würden die Probleme von IntServ auftreten, also der Aufwand um jeden Strom einzeln zu verwalten zu groß werden.

FQMM sieht drei Arten von Knoten vor: Ausgangsknoten, die Daten aussenden, innere Knoten, die die Daten weiterleiten und Endknoten, die Empfänger der Daten (ingress, interior und egress nodes). Ein einzelner Knoten kann dabei verschiedene Rollen übernehmen, abhängig davon, an welcher Position entlang eines bestimmten Datenstromes er sitzt, aber unabhängig von seiner physikalischen Position.

Um die Einhaltung der Dienstgüteparameter kümmert sich ein Traffic-Conditioner am Ausgangsknoten. Er ist dafür verantwortlich, Pakete entsprechend dieser Parameter und dem Verkehrsprofil, also den momentanen Gegebenheiten im Netz, zu markieren, anzupassen oder zu verwerfen.

FQMM ist der erste Versuch, Dienstgüte in MANETs zu bringen. Allerdings gibt es noch einige Probleme, die gelöst werden müssen. Es können zum Beispiel nicht beliebig viele Anwendungen die per-flow Dienstgüte benutzen. Andernfalls würde das Skalierungsproblem von IntServ wieder auftreten.

3. Signalisierung

Signalisierung wird benötigt, um Ressourcen zu reservieren, wieder freizugeben und wenn nötig zwischendurch die Parameter neu zu verhandeln. Ein Signalisierungsmechanismus muss zwei Vorraussetzungen erfüllen: die Signalisierungsinformationen müssen zuverlässig zwischen den Routern transportiert werden und sie müssen korrekt interpretiert und nötige Maßnahmen ausgeführt werden.

3.1 In-Band vs. Out-of-Band

Signalisierung kann In-Band oder Out-Of-Band erfolgen. Bei einer In-Band-Signalisierung werden alle für das Netzwerk-Management nötigen Daten in die eigentlichen Datenpakete eingebettet. Out-of-Band-Signalisierung benutzt zusätzliche, dedizierte Datenpakete um die Netzwerk-Managementinformationen zu transportieren.

Für MANETs sollte ein Signalisierungsverfahren möglichst geringen Overhead bedeuten. Es sollte in der Lage sein, Datenströme aufrecht zu erhalten, selbst wenn die Netztopologie sich

ändert und der Verkehr umgeleitet werden muss. Ausserdem sollte das Verfahren leicht erweiterbar sein, um neue Dienste in existierende Netzwerke einzubauen.

Out-of-Band-Signalisierung macht diese unabhängig von der eigentlichen Datenübertragung. Außerdem können mit ihr vielfältigere Dienste angeboten werden. In einigen Fällen, zum Beispiel bei vollkommen unidirektionalem Datenfluss, ist es nur schwer möglich ausschließlich mit In-Band-Signalisierung zu arbeiten, da die Daten überwiegend in eine Richtung fließen und damit kaum Signalisierungsdaten in die andere Richtung geschickt werden können. Allerdings nimmt Out-of-Band-Signalisierung mehr Bandbreite in Anspruch, die in drahtlosen Netzen begrenzt ist. Da Signalisierungspakete eine höhere Priorität haben sollten, würde diese Signalisierungsmethode die Performance eines MANETs negativ beeinflussen. In-Band-Signalisierung dagegen geht deutlich schonender mit den Ressourcen um. Auch wenn sie natürlich nicht völlig ohne zusätzliche Bandbreite auskommt, sind die Auswirkungen auf das Netz deutlich geringer.

Da für MANETs die verfügbare Bandbreite und Leistungsbegrenzung Ausschlag gebend sind, ist es wichtiger, dass die Signalisierung möglichst einfach gehalten wird und so wenig Ressourcen wie möglich in Anspruch nimmt. Auf ein umfangreiches und damit komplexes System sollte vorerst verzichtet werden.

3.2 RSVP

Das Resource-Reservation-Protocol (RSVP) [XN99] ist das Signalisierungsprotokoll im Internet und wird unter anderem in Verbindung mit IntServ benutzt. Es ist ein Out-of-Band-Verfahren. Will ein Sender Daten an einen Empfänger senden, so schickt er als erstes eine Path-Nachricht an den Empfänger. Diese Nachricht enthält Informationen über die Charakteristik des Datenstroms. Jeder Router auf dem Weg leitet die Path-Nachricht entsprechend des Routing-Protokolls weiter. Erhält der Empfänger die Path-Nachricht, sendet er eine RESV-Nachricht zurück an den Sender. Diese RESV-Nachricht enthält Informationen über die Ressourcen, die für den Datenstrom benötigt werden. Die Router, die diese Nachricht erhalten, überprüfen, ob die geforderten Ressourcen bereit gestellt werden können. Ist dies möglich reservieren werden sie reserviert und die RESV-Nachricht weiter geleitet. Sind nicht genügend Ressourcen verfügbar wird eine Fehlermeldung an den Empfänger geschickt.

RSVP hat zwei wichtige Merkmale: Zum einen stellt der Empfänger des Datenstroms, nicht der Sender, die Reservierungsanfrage. Im Multicast-Fall können unterschiedliche Empfänger so verschiedene Anforderungen stellen. Außerdem werden die Informationen über den Datenfluss und die Reservierung regelmäßig aktualisiert. Damit kann der Ausfall einer Verbindung entdeckt werden.

Zum Einsatz in MANETs ist RSVP ungeeignet. Der Signalisierungs-Overhead von RSVP würde mobile Knoten zu stark belasten und viel Bandbreite in Anspruch nehmen. Außerdem kann RSVP nicht mit der dynamischen Topologie der mobilen Netze umgehen.

3.2 INSIGNIA

INSIGNIA [LC99] ist ein In-Band-Signalisierungsprotokoll, dass die Dienstgüte in MANETs unterstützt. Es benutzt das IP-Options-Feld im Header jedes IP-Paketes, hier genannt INSIGNIA-Option, für die Signalisierungsinformation. Wie RSVP benutzt INSIGNIA per-flow-Management der einzelnen Datenströme, d.h. jeder Datenstrom muss von Ende zu Ende aufgebaut, möglicherweise angepasst und wieder abgebaut werden.

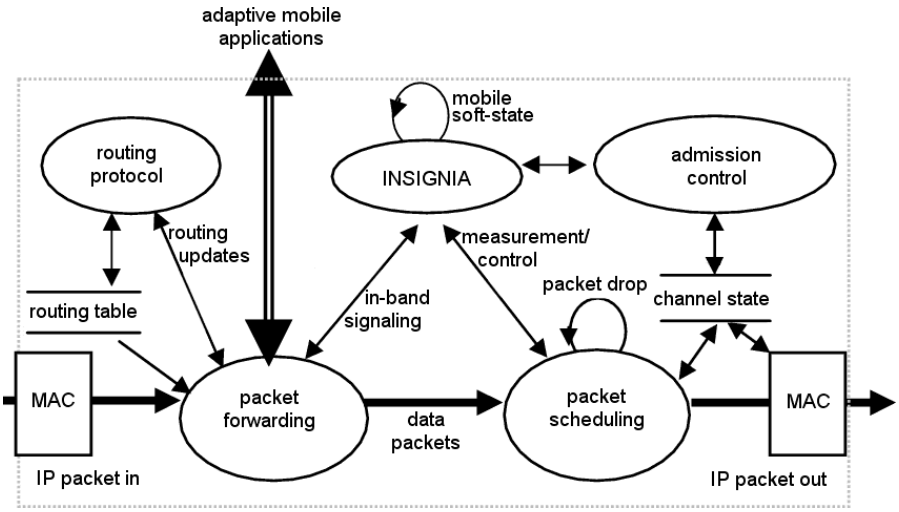

Abb. 3.1 – INSIGNIA Modell

Abbildung 3.1 zeigt, wie INSIGNIA in die mobilen Hosts eingebettet wird. Der Packet-Forwarder klassifiziert ankommende Pakete und leitet sie an die entsprechenden Module (Routing, INSIGNIA, lokale Anwendungen oder Packet-Scheduler) weiter. Wird ein IP-Paket mit einer INSIGNIA-Option empfangen, wird die Kontrollinformation an das INSIGNIA-Modul weitergegeben und dort verarbeitet, während das Pakets je nach Ziel entweder an eine lokale Anwendung oder an den Paket-Scheduler übergeben wird. Der Paket-Scheduler kümmert sich darum, ausgehenden Daten ihrer Dienstgüteanforderung entsprechende Priorität zu geben.

Das INSIGNIA-Modul ist verantwortlich für Reservierungen, den Auf- und Abbau der Echtzeitdatenströme, sowie für deren Anpassung an aktuelle Gegebenheiten im Netz. Die Informationen über die einzelnen Datenströme, die lokal gespeichert werden, müssen periodisch erneuert werden, sonst verfallen sie. Veränderungen im Netzwerk werden dadurch automatisch berücksichtigt. Zusammen mit der Zugangskontrolle kümmert sich INSIGNIA um die Zuweisungen von Bandbreite. Um die Verarbeitung einfach zu halten, werden im Fall, dass die Reservierung nicht erfüllt werden kann, keine Ablehnungs- oder Fehlermeldungen verschickt, sondern der Datenstrom auf Best-Effort Service herunter gestuft.

Um auf Änderungen in der Netzwerktopologie schnell reagieren zu können und die Einhaltung der Dienstgüteparameter zu garantieren, beobachtet der Empfänger der Daten ständig den Datenfluss und ermittelt statistische Werte zu Verlustrate, Verzögerung, Durchsatz usw. Die Daten werden regelmäßig an den Sender geschickt, der gegebenenfalls den Datenfluss oder die Dienstgüteparameter an veränderte Netzwerkzustände anpassen kann.

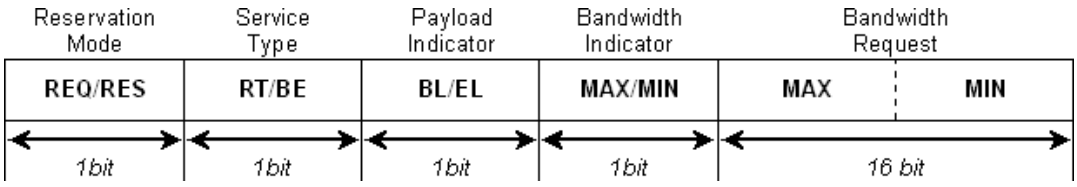

Abb 3.2 INSIGNIA-Options

Abbildung 3.2 zeigt den Aufbau der INSIGNIA Options im IP Header. Die einzelnen Felder haben folgende Funktion:

1. *Reservation-Mode:* Ein REQ in diesem Feld steht für eine Reservierungsanfrage. Ein Knoten, der ein REQ erhält, gibt die Reservierungsparameter an das Admission-Control-Modul weiter, der die Reservierung annimmt oder ablehnt. Wenn Pakete an dieser Stelle ein RES stehen haben, wurde der Datenstrom bereits vom Admission-Control-Modul zugelassen und die Ressourcen bereits reserviert.

2. *Service-Type:* Der Servicetyp kann entweder RT (Real-Time) oder BE (Best-Effort) sein. Um eine Echtzeitreservierung durchzuführen, sendet ein Knoten die Kombination REQ/RT mit den Bandbreitenanforderungen im Bandwidth-Request Feld. RES/RT zeigt an, dass die Reservierung bereits durchgeführt wurde. Ein BE-Packet erfordert keine Reservierung. Der Empfang eines RES/BE-Paketes bedeutet, dass ein Echtzeitdatenstrom auf Best-Effort heruntergestuft wurde.

3. *Bandwidth-Request:* Der Bandwidth-Request gibt der Datenquelle die Möglichkeit, Werte für die minimal und maximal benötigte Bandbreite für Echtzeitverkehr anzugeben. Der Sender kann auch nur eine minimale oder maximale Bandbreite angeben.

4. *Payload-Indicator:* INSIGNIA unterstützt zwei verschiedene Typen von Nutzdaten (payload): Base- (BL) und Enhancement-Layer (EL). Der Base-Layer ist dabei der Verkehr, der mit der minimalen angeforderten Bandbreite übermittelt werden kann. Die Summe von Verkehr des Base- und Enhancement-Layer bildet die maximal angeforderte Bandbreite.

5. *Bandwidth-Indicator:* Der Bandwidth-Indicator spielt bei der Einrichtung der Datenströme und der Anpassung an aktuelle Netzgegebenheiten eine große Rolle. Während der Einrichtung zeigt er die Verfügbarkeit von Ressourcen auf den Zwischenknoten an. Hat ein empfangenes Paket dabei das Bandwidth-Indicator auf MAX gesetzt, so ist entlang der Route die maximal angeforderte Bandbreite verfügbar. Steht es dagegen auf MIN kann mindestens ein Knoten nicht die maximale Bandbreite zur Verfügung stellen, sondern nur die minimale. Der Bandwidth-Indicator wird auch bei der Anpassung der Datenströme benutzt. Dabei überwacht der Empfänger den Bandwidth-Indicator um festzustellen, ob die zusätzliche Bandbreite für den Enhancement-Layer verfügbar ist.

Der Ablauf einer Echtzeitdatenübertragung durch INSIGNIA könnte so aussehen:

Der Sender schickt eine ein Datenpaket mit einer Reservierungsanfrage ab. Für die Reservierungsanfrage wird in den INSIGNIA Options im Header der Reservation-Mode auf REQ, Service-Type auf RT, Payload-Indicator entsprechend den Daten auf BL oder EL und die minimal und maximal benötigte Bandbreite gesetzt. Dieses Paket passiert auf allen Zwischenknoten seiner Reise das Admission-Control-Modul und es wird dort Bandbreite zugewiesen. Der Sender schickt Datenpakete mit Reservierungsanfragen solange, bis ihm der Empfänger mit QoS-Reports (s.u.) das Ergebnis der Anfrage mitteilt. Diese zusätzlichen Reservierungsanfragen lösen keine zusätzliche Reservierung von Ressourcen aus, sondern aktualisieren nur die Soft-State-Informationen auf den Zwischenknoten.

Kann ein Knoten nur die minimale Bandbreite einer Reservierung bereitstellen, so setzt er den Bandwidth-Indicator auf MIN und alle Pakete die als EL im Payload-Indicator gekennzeichnet sind, werden auf Best-Effort herunter gestuft.

Um den Sender über den Status der empfangenen Datenströme auf dem laufenden zu halten, werden ihm QoS-Reports geschickt. Der Empfänger der Daten überwacht die ankommenden Datenströme und deren INSIGNIA Options und berechnet daraus Statistiken über die Dienstgüteparameter. Diese werden regelmäßig an den Sender geschickt, der dann

gegebenenfalls Anpassungen vornehmen kann. Diese Reports werden als einfache Best-Effort-Daten verschickt und müssen nicht die selbe Route nehmen, über die der eigentliche Datenstrom gekommen ist. Die QoS-Reports sind abhängig von den Anwendungen; sie bestimmen die Häufigkeit dieser Meldungen.

Erhält der Sender per QoS-Report die Mitteilung, dass seiner Reservierungsanfrage nachgekommen wurde, ändert er bei ausgehenden Paketen den Reservation Mode von REQ auf RES. Für den Fall, dass die Reservierung nur für den Base-Layer (also die minimale Bandbreite) erfüllt werden kann, werden alle Enhancement-Layer Pakete fallen gelassen. Da die Zwischenknoten nun keine EL-Daten mit RT-Bit mehr bekommen, können die dafür reservierten Ressourcen für andere Datenströme frei gegeben werden.

INSIGNIA arbeitet mit Soft-State-Reservierungen. Solange ein Zwischenknoten Daten eines bestimmten Datenstromes empfängt, frischt es die Reservierung immer wieder auf. Bleiben Daten aus, verfällt die Reservierung nach Ablauf einer bestimmten Zeit. Werden danach wieder Daten empfangen, müssen die Ressourcen neu reserviert werden. Diese Arbeitsweise erlaubt sehr flexiblen Umgang mit den Ressourcen. Wird ein Datenstrom auf Grund von Topologieänderungen umgeleitet, ist es nicht nötig, die Reservierung auf den nun nicht mehr beteiligten Knoten explizit aufzulösen.

INSIGNIA ist ein effektives Signalisierungsprotokoll für MANETs. Es kann in Verbindung mit anderen geeigneten Netzwerkkomponenten (Routing-Protokoll, Scheduling und Zugangskontrolle) effizient für adaptiven Echtzeitdatenverkehr in MANETs benutzt werden. Allerdings benutzt es per-flow-Management, erfordert also das Halten von Daten über jeden Datenstrom auf den einzelnen Knoten über die der Strom geführt wird. Daher könnte ein Skalierungsproblem dem tatsächlichen Einsatz des Verfahrens im Wege stehen.

3.3 SWAN

SWAN (Service Differentiation in Stateless Wireless Ad hoc Networks) [ACV02] bietet, ähnlich wie DiffServ, Serviceklassen für IP-Datenverkehr in MANETs. Neben dem traditionellen Best-Effort Service unterstützt es UDP Echtzeitverkehr. Dabei kommt es ohne eine Dienstgüteunterstützung auf der MAC-Ebene aus. Es gibt kein per-flow-Management, also werden keine explizite Signalisierung oder komplexe Mechanismen benötigt, um Informationen über einzelne Datenströme auf allen beteiligten Knoten aktuell zu halten. Um Echtzeitverkehr bei Änderungen im Netzwerk durch Mobilität oder temporäre Überlastungen dynamisch regulieren zu können, benutzt man Explicit Congestion Notifications (ECN), ähnlich wie in TCP.

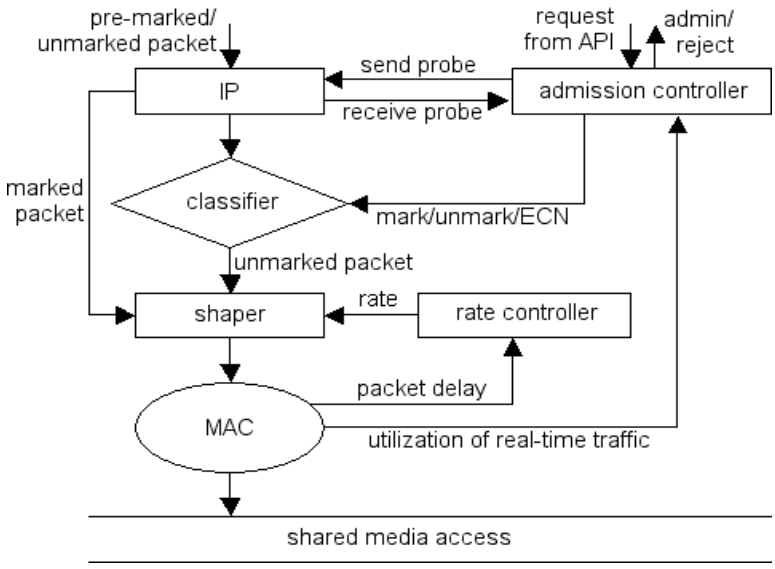

Abb. 3.3 – SWAN Modell

Abbildung 3.3 zeigt die Komponenten von SWAN und wie sie untereinander und mit der IP bzw. MAC-Schicht kommunizieren.

SWAN benutzt das DS-Feld im IP-Header um zwischen Echtzeit- und Best-Effort-Verkehr zu unterscheiden. Wenn eine Applikation einen Echtzeitdatentransfer benötigt, wendet sie sich zuerst an den Admission-Controller. Dieser schickt einen Bandwidth-Probe-Request aus, um das Netzwerk zwischen sich und dem Empfänger auf verfügbare Bandbreite zu untersuchen und entscheidet dann, ob den Anforderungen der Applikation nachgekommen werden kann. Wird der Anwendung ein Echtzeittransfer bewilligt, kümmert sich nur der Admission-Controller an der Quelle der Daten um die Einhaltung der Bandbreitengrenze. Dadurch entfällt die Speicherung der Flussdaten bei allen anderen Knoten entlang des Weges der Daten.

Best-Effort-Daten werden an jedem Knoten des Netzes, den sie passieren, der Ratenkontrolle unterzogen. Damit wird sichergestellt, dass Bandbreiten- und Verzögerungsanforderungen des Echtzeitverkehrs eingehalten werden können. Die Ratenkontrolle stellt außerdem sicher, dass die Bandbreite, die nicht von Echtzeitdaten in Anspruch genommen wird, effektiv mit Best-Effort-Verkehr gefüllt wird. Um dies umzusetzen, benutzt SWAN einen AIMD-Ansatz (Additive Increase, Multiplicative Decrease).

Jeder Knoten überwacht ständig die Auslastung des geteilten Mediums mit Echtzeitverkehr, um die verfügbare Bandbreite abschätzen zu können. Fällt die verfügbare Bandbreite unter eine bestimmt Grenze, werden in Echtzeitpaketen die ECN-Bits im DS-Feld gesetzt. Erhält ein Empfänger Pakete mit gesetzten ECN-Bits, schickt er Regulate-Messages an den Sender, um ihn von der Überlastung in Kenntnis zu setzen. Der Sender kann nun seinen Echtzeitdatenstrom neu initialisieren.

Auch SWAN würde den Anforderungen eines Signalisierungssystems für MANETs gerecht werden. Da SWAN keine flussspezifischen Daten speichert, skaliert es besser als INSIGNIA.

4. Routing

Damit der eigentliche Datenverkehr und die Signalisierung funktionieren können, muss ein Routing-Verfahren zu Verfügung stehen, dass in der Lage ist, Wege zwischen zwei mobilen

Geräten zu finden, die geforderte Dienstgüteanforderungen erfüllen. Erst wenn solche Wege gefunden wurden, kann mit der Signalisierung ein Pfad etabliert werden und entlang diesen Pfades können die Ressourcen reserviert werden.

Dienstgüte-Routing für MANETs ist schwierig. Die mobilen Knoten müssen Informationen über andere Knoten in ihrer Nähe speichern und sie regelmäßig aktualisieren. Die hohe Beweglichkeit der Knoten erschwert diese Aufgabe noch. Der Overhead, der dadurch entsteht, belastet die MANETs sehr. Und selbst wenn eine Verbindung mit Dienstgüte aufgebaut ist, kann sie jederzeit wieder zusammenbrechen, weil sich ein Knoten auf der Route zu weit von einem anderen entfernt, oder er einfach abgeschaltet wird. Ein Routing-Protokoll sollte in der Lage sein, möglichst schnell einen alternativen Weg finden.

Verglichen mit der vielen Arbeit, die in Dienstgüte-Routing für drahtgebundene Netze investiert wurde, sind brauchbare Ergebnisse für MANETs in diesem Bereich wegen der genannten Probleme eher rar. Selbst die Frage, ob man überhaupt Dienstgüte in MANETs unterstützen sollte, wird noch diskutiert. Es gibt allerdings einige wenige Ansätze, wie CEDAR und ticket-based probing, die durchaus vielversprechend sind. Auch für das AODV-Protokoll gibt eine Erweiterung (QoS over AODV [PH02]), die Dienstgüte ermöglicht.

4.1 CEDAR

Ein Dienstgüte-Routing-Algorithmus, der den besonderen Eigenschaften von MANETs gewachsen ist, ist der Core-Extraction Distributed Ad Hoc Routing Algorithm (CEDAR) [SSB99]. Es ist ein link-state-Protokoll und besteht im Wesentlichen aus drei Teilen: Core Extraction, Link-State Propagation, und Route Computation. Im Folgenden werden die Hauptfunktionen dieser drei Algorithmen kurz erläutert.

4.1.1 Core Extraction

Die Core Extraction ermittelt den Kern eines Netzes, d.h. eine Approximation der minimalen dominierenden Menge (dominating set, DS) von Knoten des Netzes. Die dominierende Menge DS eines Netzes ist eine Menge von Knoten, bei der jeder Knoten des Netzes entweder selbst Element von DS oder Nachbar eines Elements von DS ist.

Der Core-Extraction-Algorithmus ermittelt eine Menge von Knoten DS, die den Kern des Netzes bilden. Jeder Knoten, der zu DS gehört wird Core Host genannt. Die Knoten, die nicht in DS sind, suchen sich einen ihrer Nachbarn, der zu dieser Menge gehört als sogenannten Dominator aus. Der Dominator eines Core Host ist er selbst. Ist der Abstand zwischen zwei Core Hosts nicht mehr als 3 Hops, so nennt man sie benachbart. Ein Pfad zwischen zwei benachbarten Core Hosts wird als Virtual Link bezeichnet. Der Graph, der aus den Core Hosts und ihren Virtual Links besteht, ist der Core Graph.

CEDAR bietet einen verteilten Algorithmus, um die Core Hosts zu bestimmen. Diese Bestimmung erfolgt auf den einzelnen Knoten nur mit lokalen Berechnungen. Er funktioniert wie folgt:

Jeder Knoten sendet regelmäßig einen Broadcast mit seiner Identität, der Anzahl seiner Nachbarn, wie viele seiner Nachbarn ihn als Dominator gewählt haben und seinen eigenen Dominator. Sollte dieser Knoten keinen Dominator haben, sucht er sich einen seiner Nachbarn oder sich selbst als Dominator aus. Außerdem sendet der Knoten seinem Dominator seine Identität, die aller seiner Nachbarn, und die Dominatoren der einzelnen Nachbarn. Ist ein Knoten Dominator mindestens eines Knotens, so tritt er dem Kern des

Netzes bei. Durch diesen Mechanismus erfährt ein Core Host zum einen, wer sich alles in seiner Nachbarschaft befindet und zum anderen, welche anderen Core Hosts sich in seiner Nähe befinden und wie er sie erreichen kann.

Abbildung 4.1 zeigt ein MANET mit einigen Knoten. Vier davon wurden als Core Hosts ausgewählt. Um die Übersichtlichkeit möglichst hoch zu halten, wurde auf das Einzeichnen der Reichweite der einzelnen Knoten verzichtet. Dafür wurde zwischen zwei Knoten eine Kante eingezeichnet, wenn diese sich in Reichweite von einander befinden. Den Core Graph dieses Netzes erhält man, wenn man nur die schwarz eingezeichneten Core Hosts und die grauen Virtual Links betrachtet.

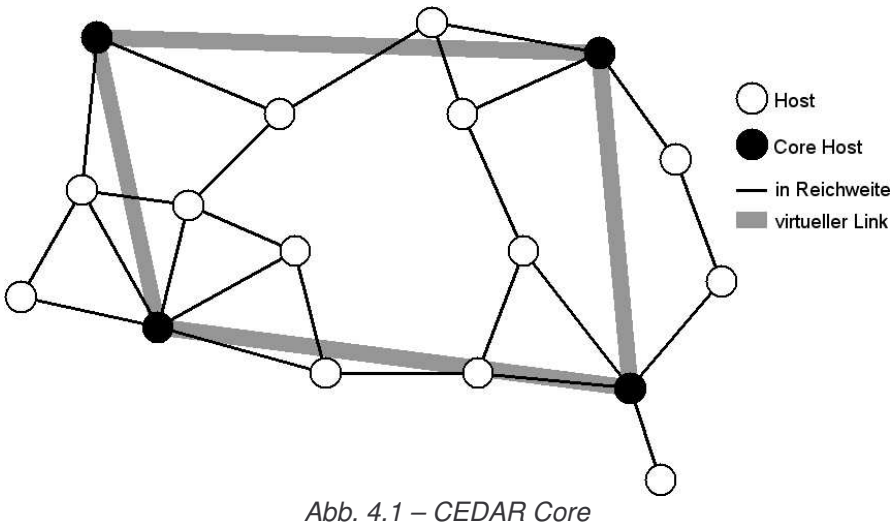

Abb. 4.1 – CEDAR Core

4.1.2 Link State Propagation

Um mit CEDAR Netzwerkpfade zu finden, auf denen Datenströme Dienstgütemechanismen benutzen können, führt jeder Core Host Daten über die Netzwerktopologie in seiner Umgebung, sowie stabile Routen mit hoher verfügbarer Bandbreite zu weiter entfernten Knoten. Routen, die unzuverlässig sind oder nur über geringe Bandbreite verfügen, werden nicht beachtet, da sie für das Auffinden von Routen auf, denen Dienstgüte angeboten werden kann, nicht relevant sind. Um dies zu erreichen, benutzt CEDAR sogenannte Increase bzw. Decrease Waves.

Wird zwischen zwei Hosts ein Link aufgebaut, oder die Bandbreite eines bestehendes Links steigt über eine bestimmte Schwelle, so benachrichtigen diese Hosts ihre jeweiligen Dominatoren, die ihrerseits eine Increase Wave aussenden. Das heißt, sie informieren die Core Hosts des Netzwerks über den neuen Link.

Vergleichbar dazu wird eine Decrease Wave ausgesendet, wenn zwei Hosts ihren Dominatoren mitteilen, dass ein Link zusammen gebrochen oder unter die Bandbreitenschwelle gefallen ist.

Die Increase Waves breiten sich langsamer aus als ihre Gegenstücke. Für einen bestimmten Link, der erst entsteht und später wieder verschwindet, überholt die Decrease-Nachricht das vorherige Increase und löscht es aus. Stabile Links werden durch Increase Waves an alle Core Hosts weitergeleitet.

4.1.3 Route Computation

Die eigentliche Routen-Berechnung in CEDAR erfolgt in drei Schritten:

1. Aufspüren des Zieles und Finden eines Core Paths dorthin;
2. Suche nach einer stabilen Route, die den Dienstgüteanforderungen entspricht, wobei aus dem Core Path die grobe Richtung abgeleitet wird;
3. dynamische Neuberechnung der ermittelten Route wenn ein Link ausfällt oder sich die Netztopologie ändert.

Wenn ein Knoten **s** Nachrichten an ein Ziel **d** schicken will, sendet er zuerst Quell- und Zieladresse, sowie die benötigte Bandbreite an seinen Dominator. Kann dieser mit seinen lokalen Link-State-Informationen aus der Core-Computation eine mögliche Route zu **d** bestimmen, antwortet er sofort. Kann er es nicht, kennt aber den Dominator von **d** und einen Core Path zu diesem, beginnt er mit der Suche nach einer geeigneten Route. Kennt er den Dominator von **d** oder einen Core Path dorthin nicht, so versucht er, ihn mittels Core Broadcast ausfindig zu machen. Core Hosts, die einen solchen Broadcast erhalten, fügen ihre ID ein, und senden sie an alle bekannten Core Hosts weiter. Erreicht diese Nachricht den Dominator von **d**, schickt dieser ein Acknowledgement an den Dominator von **s** mit dem sich aus den IDs der Zwischenknoten ergebenden Pfad. Damit steht dem Dominator von **s** ein Core Path zur Verfügung, mit dessen Hilfe er die eigentliche Route zu ermitteln beginnen kann.

Der Dominator von **s** kennt nur aktuelle Details der Topologie seiner unmittelbaren Nachbarschaft. Die Informationen, die er über entfernte Teile Netzes hat, beschränken sich auf einzelne stabile Links mit hoher Bandbreite. Daher kann er möglicherweise keinen geeigneten Pfad zu **d** für die Anforderung von **s** finden. In diesem Fall versucht er, aus den ihm zur Verfügung stehenden Daten, einen Pfad mit ausreichender Bandbreite zu einem möglichst weit entfernten Knoten **t** zu finden, dessen Dominator auf dem vorher ermittelten Core Path liegt. Dem Dominator von **t** sendet er eine Nachricht, dass dieser die Suche nach einem Pfad von **t** bis zum Empfänger fortführen soll. Kann dieser Core Host einen Pfad zu **d** mit ausreichender Bandbreite ermitteln, ist das Problem gelöst. Ansonsten wiederholt er die Prozedur, die der Dominator von **s** benutzt hat und ermittelt wiederum einen neuen Core Host, der wieder etwas dichter an **d** liegt. Diese Verfahren setzt sich fort, bis ein Core Host einen Pfad bis zu **d** findet oder kein möglicher Weg gefunden werden kann. Im ersten Fall setzt sich der komplette Pfad von **s** zu **d** aus den berechneten Teilstücken zusammen. Wird kein Weg gefunden, so wird der Verbindungswunsch von **s** abgelehnt.

Link-Ausfälle handhabt CEDAR auf zwei Arten: entweder versucht der Knoten, an dem der Ausfall stattfand, eine neue geeignete Route von sich bis zum Ziel zu finden; oder die Quelle der Datenübermittlung wird benachrichtigt, damit sie selbst eine neue Route suchen kann. Beide Methoden arbeiten zusammen, um auf Topologieänderungen zu reagieren.

In Simulationen hat CEDAR gezeigt, dass es in der Lage ist, gute und zuverlässige Routen zu finden und es sich trotz geringem Overhead sehr effektiv an Änderungen in der Netzwerktopologie anpassen kann. Für Dienstgüte in MANETs würde es also durchaus in Frage kommen.

4.2 Ticket-Based Probing

Der Ticket-Based-Probing Algorithmus [CN99] ist ein weiteres Routing-Verfahren, mit dem man Dienstgüte in MANETs unterstützen kann. Es benutzt sogenannte Tickets um die

Anzahl der Suchen nach möglichen Pfade einzuschränken. Ein Knoten, der eine Route zu einem Ziel sucht, schickt eine Probe-Message mit einem oder mehr Tickets ab. Jedes Ticket entspricht genau einer Pfadsuche. Knoten, die eine solche Probe-Message mit n Tickets erhalten, machen bis zu n Kopien davon, verteilen die empfangenen Tickets auf die neuen Probes und schicken sie anschließend über geeignete Links weiter. Jede Probe-Message akkumuliert die Zeit, die seit dem Aussenden vom Sender vergangen ist. Wird die von den geforderten Dienstgüteparametern festgelegte Verzögerung überschritten, so wird die Anforderung verworfen. Jede Probe-Message, die beim Empfänger ankommt, hat also einen möglichen Pfad vom Sender zum Empfänger gefunden – genau den, den sie genommen hat.

Die Anzahl der vom Sender ausgesendeten Tickets wird anhand der Anforderungen an die Verbindung bestimmt. Je höher oder strenger die Voraussetzungen, desto mehr Tickets werden benutzt. Dies erhöht die Wahrscheinlichkeit, dass ein geeigneter Pfad gefunden wird. Für weniger strenge oder hohe Anforderungen können auch mit weniger Tickets geeignete Pfade gefunden werden.

Wenn ein Knoten die erhaltenen Tickets an seine Nachbarn weiterleitet, werden sie normalerweise ungleichmäßig aufgeteilt. Die Anzahl der über einen bestimmten Link weitergeleiteten Tickets hängt ab von der Wahrscheinlichkeit, dass dieser Link zu einem geeigneten Pfad mit möglichst geringen Kosten führt. Über Links mit hoher Bandbreite und niedriger Verzögerung werden also mehr Tickets ausgesendet als über andere. Ständige Links werden kurzlebigen Links vorgezogen. An einige Nachbarn werden gar keine Tickets weitergeleitet werden, wenn nicht genügend in der erhaltenen Probe-Message enthalten waren.

In regelmäßigen Abständen oder bei Bedarf können für Datenströme durch das Aussenden neuer Tickets andere Routen gesucht werden. Durch dieses Re-Routing werden Veränderungen in der Netzwerktopologie erkannt und der Datenverkehr auf möglicherweise existierende, günstigere Routen umgeleitet. Ebenso wird bei unterbrochenen Links der Datenstrom entweder über eine existierende alternative Route geschickt, oder wenn nötig wird ein komplett neuer Pfad gesucht.

5. Medienzugriff

Dienstgütekomponenten auf höheren Schichten, wie Signalisierung oder Routing, gehen im Allgemeinen davon aus, dass es eine funktionierende Medienzugriffskontrolle (Medium-Access-Control, MAC) gibt, die zuverlässige Unicast Kommunikation ermöglicht und Ressourcenreservierungen für Echtzeitdatenverkehr unterstützt. Bei MANETs gibt es zusätzlich die „hidden/exposed terminal“-Probleme, mit denen ein MAC-Protokoll umgehen können muss.

Abbildung 5.1 zeigt das hidden-terminal-Problem: A sendet Daten an B. C will ebenfalls Daten an B senden, kann aber nicht sehen, dass A bereits sendet. Es kommt zur Kollision bei B.

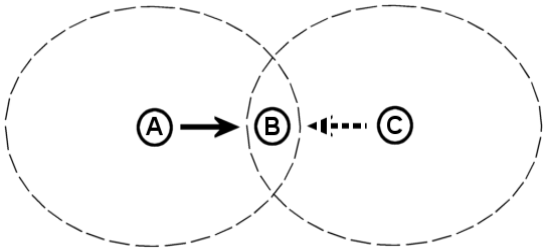

Abb. 5.1 – hidden terminal

In Abbildung 5.2 wird das exposed-terminal-Problem gezeigt: B sendet Daten an A. C will Daten an D senden, stellt aber fest, dass B das Medium belegt hat und sendet daher nicht. Eine Übertragung von C nach D würde allerdings die Übertragung von B nach A nicht beeinträchtigen, da A die „Störung“ von C gar nicht empfängt. Das Medium wird also ineffizient genutzt.

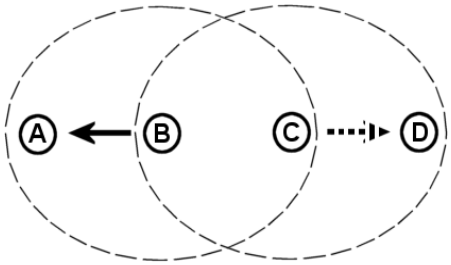

Abb. 5.2 – exposed terminal

Beide Probleme lassen sich mit einem RTS-CTS-Dialog lösen: In Abbildung 5.1 sendet A einen Request-to-send (RTS), worauf B mit Clear-to-send (CTS) antwortet. C empfängt dieses CTS ebenfalls, weiß also, dass B gerade Daten empfängt. Ebenso kann in Abbildung 5.2 C seine Datenübertragung an D einleiten, da es das CTS von A an B nicht erhalten hat.

Ein mögliches MAC-Protokol für MANETs ist MACA/PR (Multiple Access Collision Avoidance with Piggyback Reservation). Es benutzt RTS-CTS um die oben erwähnten Probleme zu lösen und bietet schnelle und zuverlässige Übertragung von Nicht-Echtzeitdaten sowie Bandbreitengarantien für Echtzeitübertragung. Reservierungsdaten für Echtzeitverkehr werden dabei im Header der Datenpaketen mitgeschickt.

6. Zusammenfassung

Diese Arbeit befasst sich mit dem Problem, Dienstgüte in mobilen Ad hoc Netzen zu unterstützen. Es wurden die Ergebnisse einiger Forschungsarbeiten vorgestellt. Auch wenn diese sich unterschiedlichen Aspekten der Dienstgüte widmen, haben sie doch mit den gleichen Problemen zu kämpfen:

Die Bandbreite in MANETs ist limitiert und die Rechenleistung der Netzteilnehmer gering. Komplexe Dienstgütemodelle mit hohem Overhead für Signalisierung und Routing kommen also nicht in Frage.

Die Knoten sind mobil, was eine sich ständig ändernde Netzwerktopologie zur Folge hat. Der Ansatz aus klassischen Netzen, dass eine einmal aufgebaute Verbindung auf absehbare Zeit verfügbar bleibt, trifft hier nicht zu. Es kann sich jederzeit ein Knoten aus der Reichweite eines Anderen bewegen und ein Re-Routing nötig werden.

Außerdem müssen die besonderen Eigenschaften eines drahtlose Mediums berücksichtigt werden, z.B. hidden/exposed Terminals.

Diese Schwierigkeiten machen die Lösungen, die beispielsweise für das Internet entwickelt wurden, in der neuen Umgebung unbrauchbar. Wie die vorgestellten Arbeiten jedoch zeigen, sind diese Probleme sind nicht unlösbar.

Literatur und Quellen

[ACVS02] SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks
Gahng-Seop Ahn, Andrew T. Campbell, Andras Veres, Li-Hsiang Sun
IEEE INFOCOM'2002, June 2002

[CN99] Distributed Quality-of-Serice Routing in Ad-hoc Networks
Shigang Chen, Klara Nahrstedt
IEEE Journal on Selected Areas in Communications
Vol 17, No 8, August 1999

[LC98] INSIGNIA: In-band signaling support for QOS in mobile ad hoc networks
Seoung-Bum Lee and Andrew T. Campbell
Proc of 5th International Workshop on Mobile Multimedia Communications
Berlin, Germany, October 1998

[PH02] A survey on Quality-of-Service support for mobile ad hoc networks
Dimitri D. Perkins, Herman D. Hughes
Wireless Communications and Mobile Computing, 2002

[SSB99] CEDAR: a Core-Extraction Distributed Ad Hoc Routing algorithm
Raghupathy Sivakumar, Prasun Sinha, Vaduvur Bharghavan
IEEE Journal on Selected Areas in Communications
Vol 17, No 8, August 1999

[WH] QoS Support in Mobile Ad Hoc Networks
Kui Wu, Janelle Harms
CS-Department, University of Alberta

[XN99] Internet QoS: A Big Picture
Xipeng Xiao, Lionel M. Ni
IEEE Network, März/April 1999