Optimierung und Evaluation eines Sicherheitskonzepts für mobile Ad-hoc-Netze

Diplomarbeit von

cand. inform. Hans-Joachim Hof

Betreuer:

Prof. em. Dr. Dr. h.c. mult. Gerhard Krüger[+]
Prof. Dr.-Ing. Lars Wolf[*]
Dipl.-Inform. Marc Bechler[*]
Dipl.-Inform. Daniel Müller[+]

+ Institut für Telematik, Universität Karlsruhe (TH)
* Institut für Betriebssysteme und Rechnerverbund, Technische Universität Braunschweig

Ich erkläre hiermit, die vorliegende Arbeit selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

_____

Karlsruhe, den 20.12.2002

Inhaltsverzeichnis

1. Motivation

Immer mehr elektronische Geräte des täglichen Bedarfs werden mit funktechnischen Schnittstellen ausgestattet. Mobiltelefone, PDAs und Laptops können drahtlos miteinander kommunizieren. Sogar Kleidungsstücke mit integrierten elektronischen Geräten und Funkanbindung wurden bereits vorgestellt. Spätestens seit der Entwicklung von Bluetooth und den damit einhergehenden billigen Chipsätzen mit geringen Energieanforderungen verfügen mehr und mehr Geräte um uns herum über drahtlose Kommunikationsmöglichkeiten.

Lag in früheren Jahren das Hauptaugenmerk bei Mobilkommunikation noch auf infrastrukturbasierter Kommunikation, so tragen die oben beschriebenen Entwicklungen dazu bei, dass Ad-hoc-Netzen immer mehr Aufmerksamkeit zuteil wird. Die heute verfügbaren Kommunikationsstandards für Ad-hoc-Netze legen Wert darauf, Kommunikation zu ermöglichen, stellen aber die Sicherheit der entstehenden Kommunikationsnetze hinten an oder lassen sie ganz außen vor. Wird ein hohes Maß an Sicherheit gewünscht, so muss eine aufgesetzte Architektur zum Einsatz kommen. Anders als in drahtgebundenen Netzen ist es bei drahtloser Kommunikation einfach, die Kommunikation abzuhören, zu stören oder zu verfälschen, da der Zugang zum Medium Luft, im Gegensatz z.B. zu einem Kabel, nicht kontrolliert werden kann. Jeder kann in das Medium senden oder laufende Kommunikation abhören. Deshalb gewinnt der Sicherheitsaspekt in Ad-hoc-Netzen eine noch höhere Bedeutung als in drahtgebundenen Netzen. Die Verwendung einer Sicherheitsarchitektur ist also dringend zu empfehlen. Bisherige Sicherheitslösungen für mobile Ad-hoc-Netze beschränken sich meist auf spezielle Sicherheitsaspekte. So stehen z.B. viele sichere Routingverfahren für mobilen Ad-hoc-Netzen zur Verfügung. Eine Komplettlösung für einen Großteil der Sicherheitsaspekte gibt es dagegen noch nicht.

1.1 Aufgabenstellung

Die vorliegende Arbeit greift das Konzept einer viel versprechenden Sicherheitsarchitektur, die am Institut für Telematik der Universität Karlsruhe (TH) im Rahmen einer Diplomarbeit entwickelt wurde, heraus und betrachtet sie unter folgenden Aspekten:

- Der Entwurf der Sicherheitsarchitektur wurde auf Stärken und Schwächen untersucht.

- Die Architektur wurde als Simulation implementiert.
- Anhand der Simulation wurde die Leistungsfähigkeit der Architektur getestet.
- Auf Basis der Messergebnisse und Implementierungserfahrung wurden Optimierungsvorschläge herausgearbeitet. Die Zentrale Frage war hier, wie die Parameter zu wählen sind.

1.2 Gliederung

Die vorliegende Arbeit gliedert sich in sieben Kapitel:

Kapitel 2 beschreibt existierende Sicherheitsarchitekturen und gibt einen Überblick über die Architektur, die dieser Arbeit zugrunde liegt. Das Kapitel gibt außerdem einen Überblick über gebräuchliche Bewegungsmodelle. Aus diesen werden später die Modelle der Simulation abgeleitet.

Kapitel 3 stellt Konzepte für diese Arbeit vor. Zwei Anwendungsszenarien für die betrachtete Sicherheitsarchitektur werden präsentiert. Anhand der Anwendungsszenarien und mit Hilfe der Modelle aus Kapitel 2 werden Bewegungsmodelle für die Simulation entwickelt.

Kapitel 4 erläutert die Implementierung der Simulation im Detail. Die verwendeten Werkzeuge werden vorgestellt.

Kapitel 5 enthält die Messergebnisse und Analysen aus verschiedenen Simulationsläufen. Das Kapitel schließt mit einer kurzen Bewertung ab.

Kapitel 6 schlägt auf Basis der in Kapitel 4 aufgezeigten Erkenntnisse Optimierungen und Verbesserungen der Sicherheitsarchitektur vor.

Kapitel 7 fasst schließlich die Arbeit zusammen und gibt einen Ausblick.

An die sieben Kapitel schließen sich drei Anhänge an, die Begriffserklärungen, ein Abbildungs- und ein Literaturverzeichnis enthalten.

2. Grundlagen

An Sicherheitsarchitekturen für mobile Ad-hoc-Netze werden die gleichen Anforderungen gestellt wie an Sicherheitsarchitekturen in drahtgebundenen, statischen Netzen. Dies sind unter anderem:

- *Verfügbarkeit (availability)*: Dienste und Ressourcen des Netzes sollen den Nutzern kontinuierlich zur Verfügung stehen.

- *Authentizität* (authenticity): Die Authentizität von Nachrichten soll gewährleistet werden. Dies setzt eine erfolgreich durchgeführte Authentisierung voraus.

- *Integrität (integrity)*: Nachrichte soll gegen Änderungen gesichert sein.

- *Vertraulichkeit (privacy)*: Informationen aller Art sollen vor nicht autorisierten Entitäten verborgen bleiben.

- *Autorisierung (authorization)*: Autorisierung bezeichnet einen Vorgang, mit dem Benutzer für gewisse Dienste oder Ressourcen bevollmächtigt werden können.

Mobile Ad-hoc-Netze verfügen über besondere Eigenschaften, die sie von drahtgebundenen, statischen Netzen unterscheiden. Diese Eigenschaften stellen eine besondere Herausforderung beim Entwurf von Sicherheitsarchitekturen dar. Im Einzelnen sind dies:

- *Asymmetrische Verbindungen*: Aufgrund verschiedener Sendeleistungen, Antennenbeschaffenheiten, etc. ist es möglich, dass ein Knoten A einen anderen Knoten B zwar empfangen kann, Knoten B den Knoten A aber nicht hört. Zwischen beiden Knoten besteht dann eine asymmetrische Verbindung, die Kommunikation nur in eine Richtung erlaubt. Eine asymmetrische Verbindung liegt aber auch vor, wenn Hin- und Rückrichtung eines Kommunikationskanals verschiedene Bandbreiten haben.

- *Dynamische Topologie*: In Ad-hoc-Netzen können sich Knoten typischerweise frei bewegen. Dabei sind die Zeitpunkte der Lokationsänderungen nicht vorhersehbar. Insbesondere können Ortswechsel sehr schnell eintreten. Dadurch kann es zu Verbindungsabbrüchen und zu neuen Verbindungen kommen. Auch durch das Ein- und Ausschalten von Geräten entsteht Dynamik in der Topologie.

- *Bandbreitenbegrenzung*: Typischerweise ist die Bandbreite in Ad-hoc-Netzen stark eingeschränkt. Der tatsächliche Durchsatz wird durch Störeinflüsse, Rauschen und Kollisionen beim Vielfachzugriff oft deutlich unter die maximal mögliche Übertragungsrate gesenkt.

Die besonderen Eigenschaft von Ad-hoc-Netzen machen es schwierig, Sicherheitslösungen aus drahtgebundenen, statischen Netzen zu übernehmen. Der folgende Abschnitt gibt einen Überblick über die bereits verfügbaren Lösungen. Ein allumfassendes Komplettsystem besteht allerdings noch nicht.

2.1 Verfügbare Architekturen

Im Bereich Sicherheit für Ad-hoc-Netze findet sehr viel Forschung statt. Sicheres Routing stellt dabei einen Forschungsschwerpunkt dar. Ariadne [HU01] und SEAD (Secure Efficient Distance Vector Routing [HU02]) sind Beispiele aus diesem Bereich. Eine Liste mit weiteren Arbeiten enthält [ZHU02]. Ariadne ist ein reaktives (on-demand) Routingprotokoll, d.h. ein Knoten versucht erst dann eine Route zu einem Ziel zu finden, wenn er etwas senden möchte. Ariadne bietet Schutz gegen einen kompromittierten Knoten und aktive Angriffe. Es kommt nur symmetrische Verschlüsselung zum Einsatz. SEAD ist ein proaktives (tabellenbasiertes) Routingprotokoll. Es basiert auf dem Destination-Sequenced-Distance-Vector-Routingprotokoll (DSDV, [PER94]). SEAD ist robust gegenüber mehreren unkoordinierten Angreifern, die versuchen, inkorrekte Routinginformationen zu verbreiten. Auf aufwendige kryptographische Verfahren wird verzichtet, stattdessen kommen Einweg-Hashfunktionen zum Einsatz.

Andere Arbeiten beschäftigen sich mit sicherem Schlüsselaustausch und Schlüsselverteilung in mobilen Ad-hoc-Netzen. In [CAP02] wird z.B. eine selbstorganisierende Public-Key-Infrastruktur vorgestellt, die ohne eine vertrauenswürdige Wurzelinstanz oder einen Schlüssel-Server auskommt. Jeder Knoten ist selbst Wurzelinstanz und vergibt Zertifikate an andere Knoten. Er speichert eine begrenzte Anzahl an Zertifikaten anderer Knoten. Schlüsselauthentisierung wird über eine Zertifikatkette realisiert.

Sichere Gruppenkommunikation in mobilen Ad-hoc-Netzen wird in [YAS02] untersucht. Die Arbeit verwendet ein Konzept von Diffie-Hellman zur Verbreitung von Schlüsseln in einer Gruppe.

Die vorgestellten Arbeiten beschäftigen sich alle mit einem Teilbereich der Sicherheit in Ad-hoc-Netzen. Sie stellen „Insellösungen“ dar. Bislang fehlt es aber noch an Gesamtkonzepten, die verschiedene Sicherheitsaspekte, wie z.B. Routing, Gruppenkommunikation und Public-Key-Infrastruktur, in einer einzigen Architektur vereinen. DAHNI (Driver Ad Hoc Networking Infrastructure, [ZAR02]) stellte einen ersten Versuch dar, eine Gesamtsicherheitsstruktur zu umreißen. Die Arbeit beschäftigt sich mit der Kommunikation zwischen verschiedenen Fahrzeugen auf einer Autobahn und zeitweiser Infrastrukturanbindung.

2.2 Die betrachtete Architektur

Arbeiten am Institut für Telematik der Universität Karlsruhe (TH) befassen sich ebenfalls mit der Sicherheit in mobilen Ad-hoc-Netzen. [HAU01] fasst viele Aspekte früherer Forschung in einem Gesamtkonzept zusammen. Dabei wurde besonderen Wert auf einen völlig dezentralen Entwurf gelegt, um Angreifern keine Angriffsflächen zu bieten. Die Lösung ist clusterbasiert, das Ad-hoc-Netz ist also in einzelne Cluster partitioniert, denen die Knoten (vom Aufenthaltsort abhängig) zugeteilt sind. In jedem Cluster gibt es einen ausgezeichneten Knoten, den Clusterhead (CH), der den Cluster aufbaut und organisiert. Jeder Knoten kann zum Clusterhead werden. Gateways (GW) können zwischen benachbarten Clustern vermitteln und nehmen eine besondere Stellung ein. Gateways sind alle jene Rechner, die mehr als einen Cluster empfangen können, d.h. sie liegen im Schnittbereich der Cluster. Knoten erkennen einen Cluster an dem so genannten CH-Beacon, das vom Clusterhead in regelmäßigen Abständen als Broadcast in den Cluster verschickt wird und Informationen über den Cluster enthält, wie z.B. eine Liste der Knoten und Gateways im Cluster. Knoten, die ein CH-Beacon empfangen,

senden dieses als Broadcast weiter. Innerhalb eines Clusters empfangen alle Knoten das CH-Beacon. Ein Parameter legt fest, wie viele Hops ein CH-Beacon weitergeleitet wird. Abbildung 1 zeigt ein clusterbasiertes CH-Netz. In der hier betrachteten Sicherheitsarchitektur bilden die Clusterheads der einzelnen Cluster ein logisches Netzwerk, das Clusterhead-Netzwerk. Dieses ist in Abbildung 1 mit einer gepunkteten Linie angedeutet. Durch die Organisation des Ad-hoc-Netzes in Cluster zerfällt das Routing-Protokoll in zwei Teile: Intra-Cluster-Routing und Inter-Cluster-Routing. Es gibt spezielle Routing-Verfahren für clusterbasierte Ad-hoc-Netze, z.B. das Zone Routing Protokoll [HAA00]. Dieses verwendet für die Wegewahl innerhalb eines Clusters ein proaktives Verfahren (auf Tabellen basierend). Für die Wegewahl bei cluster-übergreifender Kommunikation kommt ein reaktives Verfahren (on-demand) zum Einsatz

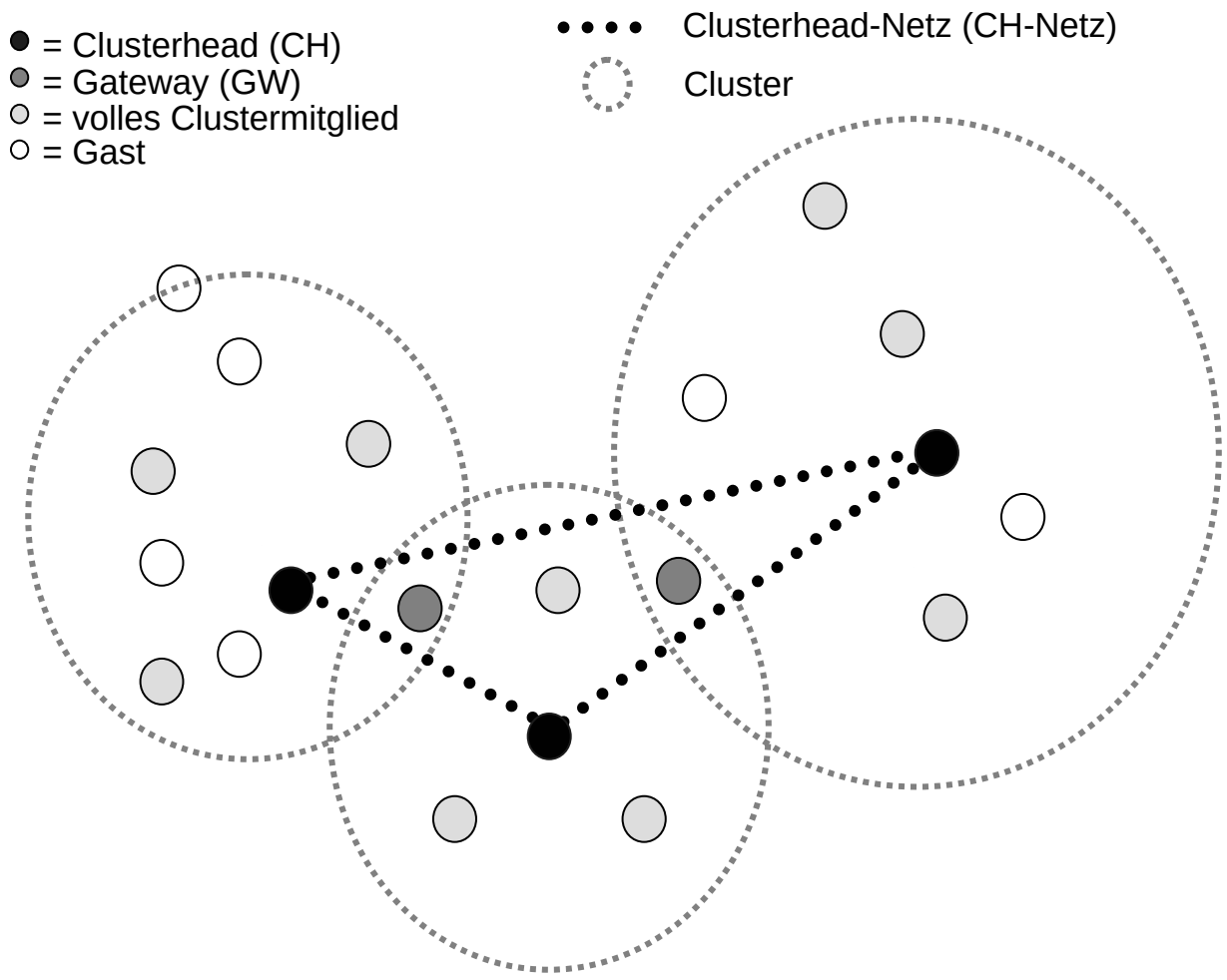

Abbildung 1: Clusterbasiertes Ad-hoc-Netz

2.2.1 Übersicht

Die hier betrachtete Sicherheitsarchitektur realisiert eine verteilte Public-Key-Infrastruktur. Es gibt also keinen zentralen Server, auf dem alle Schlüssel hinterlegt sind etc. Alle Clusterheads gehören einem logischen Netz, dem Clusterhead-Netzwerk (gepunktete Linien in Abbildung 1), an. In diesem wird mittels proaktiver Geheimnisteilung (siehe unten) ein privater Schlüssel von den Clusterheads verwaltet, der im Rahmen einer proaktiven digitalen Signatur eingesetzt wird, um netzweit gültige Zertifikate für öffentliche Schlüssel von Knoten innerhalb der Sicherheitsarchitektur zu erzeugen. Ein Knoten fragt mehrere Clusterheads an und bekommt von diesen jeweils einem Teil des identitätsbezogenen Schlüsselzertifikats. Hat der Knoten genügend Teilzertifikate gesammelt, dann kann er diese zum Schlüsselzertifikat zusammensetzen. Das Zertifikat ist netzwerkweit gültig, weil der öffentliche Schlüssel des CH-Netzwerks in den CH-Beacons aller Cluster bekannt gegeben wird.

Geheimnisteilung bezeichnet das Prinzip, ein Geheimnis über mehrere Instanzen zu verteilen. Dadurch wird der Grad der erreichbaren Geheimhaltung und Integrität erhöht, da ein potentieller Angreifer mehrere Instanzen kompromittieren muss, um an das Geheimnis zu kommen. Ebenso werden Denial-of-Service-Attacken erschwert, da es nicht mehr nur einen zentralen Angriffspunkt gibt. Geheimnisteilung wird über Schwellwert-Kryptographie erreicht. Dazu wird ein Geheimnis in n Teile zerlegt. Das Wissen über k oder mehr der Teile macht es möglich, das Geheimnis leicht zu rekonstruieren, während das Wissen über k-1 oder weniger Teile es unmöglich macht, das Geheimnis wiederherzustellen. Dieses Verfahren bezeichnet man auch als (k,n)-Schwellwert-Schema. Erreicht werden kann dies z.B. mit Hilfe von Lagrange-Interpolation [SHA79]. Proaktive Geheimnisteilung bezeichnet ein Verfahren, bei dem Teile des Geheimnisses periodisch geändert werden, ohne dass sich das Geheimnis an sich verändert. Dies hat den Vorteil, dass ein Angreifer nur eine gewisse Zeit hat, um k der n Geheimnisträger zu kompromittieren, ohne dass es nötig ist, das Geheimnis regelmäßig zu ändern. [HER96] beschreibt ein allgemeines Verfahren, wie Schwellwert-Public-Ley-Signaturschemata über diskrete Logarithmen in proaktive Signaturschemata transformiert werden können. Die Proaktive Signatur ist eine Anwendung der Proaktiven Geheimnisteilung. Es wird ein geheimer Schlüssel geteilt, mit Hilfe dessen verteilte Signaturen erstellt werden können.

Knoten, die über einen zertifizierten öffentlichen Schlüssel verfügen, erhalten von ihrem Clusterhead einen symmetrischen Schlüssel, der zur Kommunikation innerhalb des Clusters verwendet werden kann. Der Besitz des symmetrischen Clusterschlüssels ist gleichbedeutend mit voller Mitgliedschaft im Cluster. Kommunikation mit symmetrischen Schlüsseln ist effizienter als Public-Key-Kommunikation.

2.2.2 Der Anmeldevorgang

Um sicher kommunizieren zu können, muss sich ein neu ins Netz kommender Knoten erst einmal einem Cluster zuordnen. Dazu wartet er eine gewisse Zeit auf ein CH-Beacon (diesen Vorgang nennt man „CH-Discovery“). CH-Beacons werden in regelmäßigen Abständen vom Clusterhead (CH) des Clusters verschickt. Abbildung 2 zeigt ein CH-Beacon. Enthalten sind wichtige Informationen über den Cluster: der öffentliche Schlüssel des Clusterheads (PubCH), der öffentliche Schlüssel des Clusterhead-Netzwerks (PubCN) dem der Clusterhead angehört, die Clients, die zum Cluster gehören (K_1 bis K_i) und die im Cluster verfügbaren Gateways (G_1 bis G_k).

Abbildung 2: CH-Beacon

Empfängt ein Knoten kein CH-Beacon, so ernennt er sich selbst zum Clusterhead, gründet ein eigenes Clusterhead-Netzwerk und sendet fortan selbst CH-Beacons. Dadurch entsteht ein neuer Cluster.

Hat der Knoten jedoch ein CH-Beacon empfangen, so zeigt er dem Clusterhead an, dass er dem Cluster beitreten möchte. Der Clusterhead übermittelt dem Knoten die im Cluster gültige Sicherheitsrichtlinie. Der Knoten entscheidet dann, ob er immer noch dem Cluster beitreten möchte. Die Sicherheitsrichtlinien innerhalb des Clusters werden vom Clusterhead autonom bestimmt. Sie können zum Beispiel festlegen, ob der symmetrische Clusterschlüssel, den nur volle Clustermitglieder erhalten, zur Kommunikation im Cluster obligatorisch oder optional

eingesetzt wird. Ist der anfragende Knoten mit den Sicherheitsrichtlinien des Clusters einverstanden, so wird er Gast im Cluster. Er muss sich jetzt authentisieren. In der hier betrachteten Sicherheitsarchitektur wird die Authentisierung über Bürgen vorgenommen. Dabei sammelt ein Knoten eine Anzahl von Bürgen-Zertifikaten von Bürgen. Bürgschaften werden erteilt, aufgrund einer Identifizierung des anfordernden Knotens auf höherer Ebene. Abbildung 3 zeigt ein Bürgen-Zertifikat. Es enthält die Adresse des anfordernden Knotens (Knoten K), den zu signierenden öffentlichen Schlüssel von K (PubK), die Dauer, die der Bürge bereit ist für den Knoten K zu bürgen (Dauer t), und die Zustimmung („ich bürge“) zur Bürgschaft. Das Bürgen-Zertifikat wird mit der Signatur des Bürgen abgesichert (SigB).

Knoten K	PubK	Dauer t	„ich bürge“	SigB

Abbildung 3: BürgZert

Darüber hinaus schickt der Bürge ein Bürgen-Autorisierungszertifikat (BürgAutoZert) mit, welches beweist, dass der Bürge vom Clusterhead-Netzwerk autorisiert wurde, als Bürge aufzutreten. Dieses wurde dem Bürgen nach seiner Anmeldung am Clusterhead übermittelt.

Bürge B	PubB	„darf bürgen“	SigNetz

Abbildung 4: BürgAutoZert

Abbildung 4 zeigt das Zertifikat. Enthalten ist die Adresse des Bürgen (Bürge B), der öffentliche Schlüssel des Bürgen (PubB) und seine Berechtigung („darf bürgen“). Abgesichert wird das Zertifikat durch die Signatur des Clusterhead-Netzwerks. Bürgen-Autorisierungszertifikate werden wie andere Autorisierungszertifikate vergeben. Als autorisierende Instanz tritt hier das gesamte Clusterhead-Netzwerk auf. Sowohl Bürgen-Zertifikate als auch Bürgen-Autorisierungszertifikate können eine Gültigkeitsdauer beinhalten.

Ein Knoten muss mindestens eine vom Clusterhead-Netzwerk festgelegte Anzahl an Bürgen-Zertifikate sammeln. Es steht den Clusterheads frei, eine höhere Anzahl an Bürgen in den Sicherheitsrichtlinien für den Cluster festzulegen. Mit den Bürgen-Zertifikaten kann der Client nun versuchen, sich beim Clusterhead seines Clusters anzumelden. Der Clusterhead prüft die Bürgen-Zertifikate sowie die Bürgen-Autorisierungszertifikate und schickt bei positivem Ergebnis einen Teil des Identitätszertifikat (IdZert) sowie eine Liste weiterer Clusterheads an den anfragenden Knoten. Der Knoten muss eine gewisse Anzahl von weiteren Cluster-heads um die Signatur seines öffentlichen Schlüssels bitten (durch Übergabe von BürgZert und BürgAutoZert) und erhält weitere Teile des Identitätszertifikats, die er schließlich zu einem vollständigen Zertifikat seines öffentlichen Schlüssels zusammensetzen kann. Abbildung 5 zeigt ein Identitätszertifikat. Es enthält die Adresse des Knotens (Knoten K), seinen öffentlichen Schlüssel (PubK, der signiert werden soll) und die Gültigkeitsdauer (Dauer t) des Zertifikats. Geschützt wird das Zertifikat durch eine Signatur des Clusterhead-Netzewerks (SigNetz).

Knoten K	PubK	Dauer t	SigNetz

Abbildung 5: IdZert

Der Knoten verfügt nun über ein identitätsbezogenes Zertifikat seines öffentlichen Schlüssels. Damit kann er nun bei seinem Clusterhead den symmetrischen Clusterschlüssel anfordern und wird volles Clustermitglied. Das Zertifikat wurde vom Clusterhead-Netzwerk signiert und ist im gesamten Einflussbereich des CH-Netzwerks von jedem Knoten verifizierbar, da die Clusterheads in regelmäßigen Abständen den öffentlichen Schlüssel des Clusterhead-Netzwerks propagieren.

Neben den Clusterheads nehmen auch die Gateways eine besondere Stellung in der Sicherheitsarchitektur ein. Sie sind verantwortlich für die Weiterleitung von Daten aus dem eigenen Cluster in andere Cluster. Dazu kann ein Gateway (ebenso wie jeder andere Knoten) Mitglied in mehreren Clustern werden. Der oben beschriebene Anmeldevorgang läuft dann in jedem

Cluster getrennt ab, allerdings müssen (je nach Sicherheitsrichtlinie im entsprechenden Cluster und Zugehörigkeit des Clusters zum gleichen CH-Netzwerk) die Bürgen-Zertifikate eventuell nur einmal gesammelt werden und können zur Anmeldung an mehreren Clustern verwendet werden.

Gateways senden ebenso wie Clusterheads in regelmäßigen Abständen ein Datenpaket, das so genannte GW-Beacon. Abbildung 6 zeigt den Aufbau eines GW Beacons. In ihm enthalten sind der öffentliche Schlüssel des Gateways (PubGW), die Cluster, denen das Gateway angehört ($Cluster_1$ bis $Cluster_n$) und der Status, den das Gateway im entsprechenden Cluster hat (Status in $Cluster_i$). Der Status kann „Gast“, „volles Mitglied“ oder „Clusterhead“ sein. Ein Gateway kann auch in einem Cluster volles Mitglied sein, während es im benachbarten Cluster nur Gastmitglied ist. Bei der Weiterleitung von Datenverkehr ist dies in Bezug auf die Sicherheit unter Umständen zu beachten. Somit können z.B. Routen spezifiziert werden, die ausschließlich über volle Mitglieder gehen.

PubGW	$Cluster_1,\ldots,Cluster_n$	Status in $Cluster_i$

Abbildung 6: GW Beacon

Abbildung 7 zeigt zusammenfassend den Protokollverlauf im Überblick. An den Pfeilen wird jeweils angegeben, welche Aktion (action) ein Knoten ausführt und durch welches Ereignis (event) der Übergang verursacht wird. Die Kreise stellen interne Zustände des Knotens dar. Manche Zustände sind gesondert markiert („Guest“ und „Full cluster member“). Sie repräsentieren das Erlangen eines besonderen Status innerhalb der Sicherheitsarchitektur.

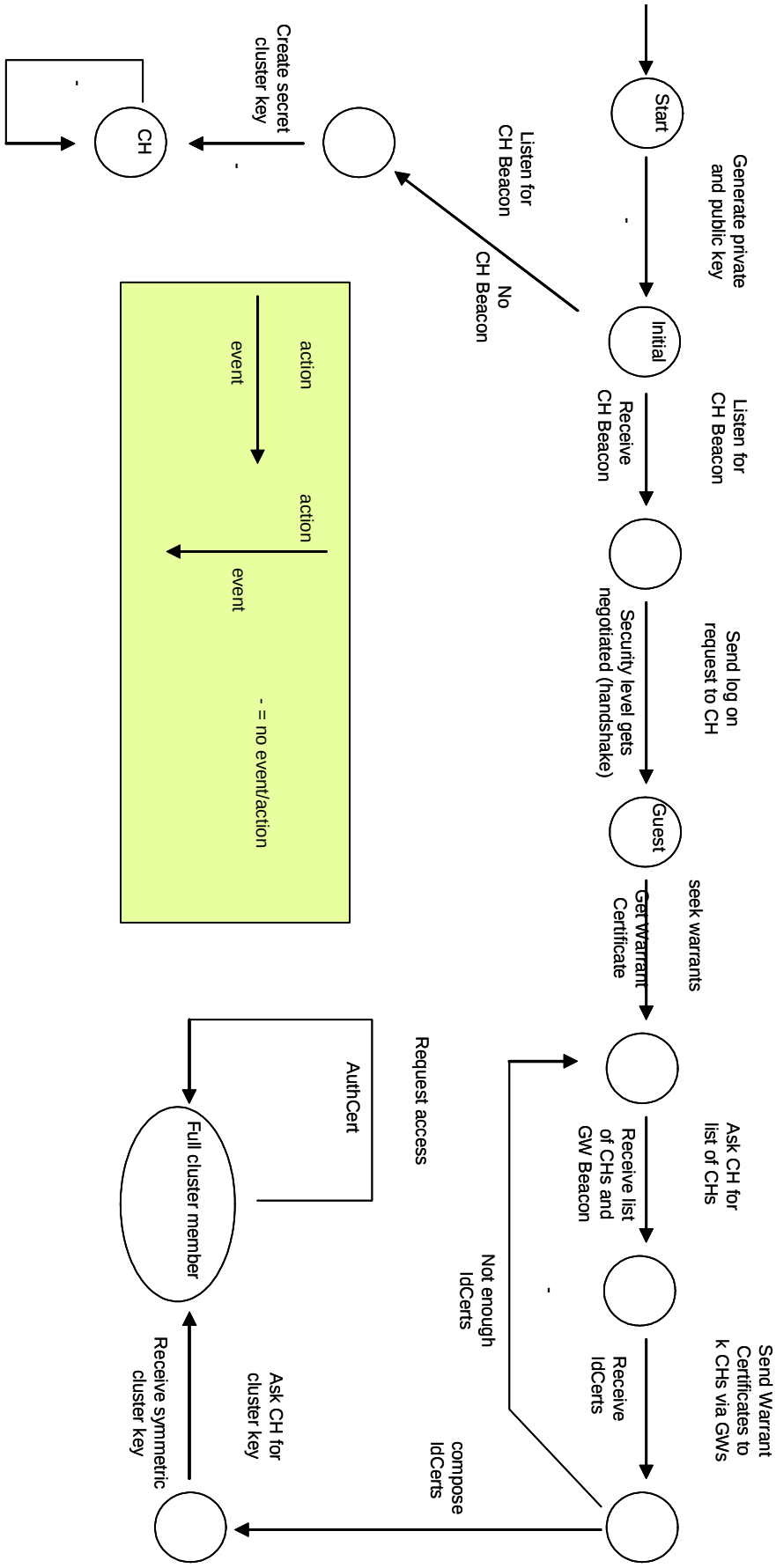

Abbildung 7: Der Protokollablauf im Überblick

2.2.3 Offene Fragen im Entwurf

Die Sicherheit einer mobilen Sicherheitsarchitektur basiert maßgeblich auf der Qualität der Authentisierung. Kann ein Angreifer sich gegenüber einem Bürgen als ein anderer Knoten ausgeben, so ist die Sicherheitsarchitektur obsolet. In Kapitel 3 Abschnitt 2 wird eine Methode zur sicheren Authentisierung vorgestellt.

Treffen mehrere Clusterhead-Netzwerke aufeinander, so sieht der Entwurf der Architektur eine Vereinigung der Netze vor. Dabei wird ein neues CH-Netzwerk erstellt. Dieses Netzwerk besitzt einen neuen privaten Schlüssel. Damit werden alle zuvor ausgegebenen Zertifikate ungültig, da sie mit dem alten Schlüssel erstellt wurden. Die Knoten der beiden ehemaligen CH-Netzwerke müssen sich neu anmelden. Es ist zu erwarten, dass dabei sehr viel Overhead entsteht und der Schlüsseltausch durch eine Phase begleitet wird, in der keine Kommunikation möglich ist. Kapitel 3 Abschnitt 3.5 stellt eine Ergänzung des ursprünglichen Protokolls vor, um die Netzvereinigung möglichst reibungslos zu realisieren. In Kapitel 6 wird eine Methode vorgeschlagen, die Vereinigungen, welche zu einem CH-Netzwerk führen, das wahrscheinlich schon bald wieder partitioniert wird, dynamisch erkennt und verzögert.

Die Clusterheads teilen sich den privaten Schlüssel des CH-Netzwerks. Eine proaktive digitale Signatur kommt zur Zertifizierung der öffentlichen Schlüssel der Knoten zum Einsatz. Jeder Knoten muss mehrere Clusterheads anfragen und erhält jeweils nur einen Teil des identitätsbezogenen Schlüsselzertifikats. Ein Schwellwert gibt an, wie viele Clusterheads ein Knoten mindestens ansprechen muss. Es stellt sich die Frage, wie dieser Wert ideal gewählt werden kann. Ebenso wird in [HAU01] keine Aussage darüber gemacht, wie viele Bürgen nötig sind, um bei einem Clusterhead anzufragen. Kapitel 6 widmet sich dieser Fragestellung.

Generell stellt sich die Frage, wie die Parameter der Architektur, z.B. das Intervall in dem CH-Beacons gesendet werden, möglichst optimal gewählt werden können. Kapitel 6 gibt Hinweise für eine optimale Wahl.

Die Sicherheitsarchitektur setzt nur sehr wenige Rahmenbedingungen voraus. In vielen Szenarien liegen unter Umständen günstigere Bedingungen vor. In Kapitel 3 Abschnitt 3.3 wird gezeigt, wie z.B. ein Zertifikat einer bekannten Wurzelinstanz optional verwendet werden

kann, um den Overhead der Sicherheitsarchitektur zu verringern, ohne damit die Sicherheit zu gefährden.

2.3 Bewegungsmodelle

Zur Simulation von Ad-hoc-Netzen gehören nicht nur die korrekte Nachbildung der Luftschnittstelle und deren Charakteristika. In hohem Maße hängt die Aussagekraft der erzielten Ergebnisse von dem in der Simulation eingesetzten Bewegungsmodell ab. Kommen in einer Simulation nur sehr allgemeine Modelle zum Einsatz, so sind die Ergebnisse zwar mit anderen Arbeiten leichter zu vergleichen. Die Ergebnisse sind aber nicht unbedingt aussagekräftig für konkret gegebene Szenarien. Dieses Kapitel stellt einige der wichtigsten Bewegungsmodelle vor. Die in der Simulation verwendeten Bewegungsmodelle wurden von diesen Modellen abgeleitet.

Es gibt zwei Arten von Bewegungsmodellen:

- *Spurenmodelle* verwenden Bewegungsmuster, die in der realen Welt aufgezeichnet wurden. Sie sind sehr genau, besonders wenn sehr viele Bewegungen über eine lange Zeit beobachtet wurden.

- *Synthetische Modelle* dagegen sind mathematische Modelle, die ohne die Hilfe von Spuren versuchen, ein realistisches Verhalten nachzubilden.

Spurenmodelle sind synthetischen Modellen gegenüber vorzuziehen, da sie die Realität besser nachbilden. Allerdings ist es in vielen Fällen schwierig beziehungsweise zu aufwändig, Spuren in ausreichender Qualität aufzuzeichnen. Hier bieten sich möglichst realitätsnahe synthetische Modelle an.

Im Folgenden werden einige der gebräuchlichsten synthetischen Modelle vorgestellt.

2.3.1 Random-Walk-Modell (Brownian Motion)

Einstein formulierte das Random-Walk-Bewegungsmodell bereits 1926 mathematisch. Historisch gesehen ist Random Walk eines der ersten Modelle, die benutzt wurden, um die Bewegung von Fußgängern zu beschreiben[1].

Ein Teilnehmer bewegt sich mit zufälliger Geschwindigkeit (gleichverteilt zwischen 0 und einem Maximum max) in eine zufällige Richtung (gleichverteilt zwischen 0 und 360°). Jede Bewegung geschieht entweder innerhalb eines konstanten Zeitintervalls oder einer konstanten zurückzulegenden Entfernung. Am Ende der Bewegung werden Geschwindigkeit und Richtung erneut wie oben beschrieben festgelegt. Random-Walk ist also gedächtnislos; die jeweils nächste Bewegung hängt nicht von der vorherigen ab. Erreicht ein Teilnehmer den Rand des simulierten Gebiets, so „prallt“ er dort ab und bewegt sich weiter. Dabei ist der Einfallswinkel gleich dem Ausfallswinkel. Abbildung 8 zeigt die Spur eines mit Random Walk bewegten Knotens auf einem 1000 mal 1000 großen Feld. [SAN02] stellt ein Java-Applet bereit, mit dem Random Walk simuliert werden kann.

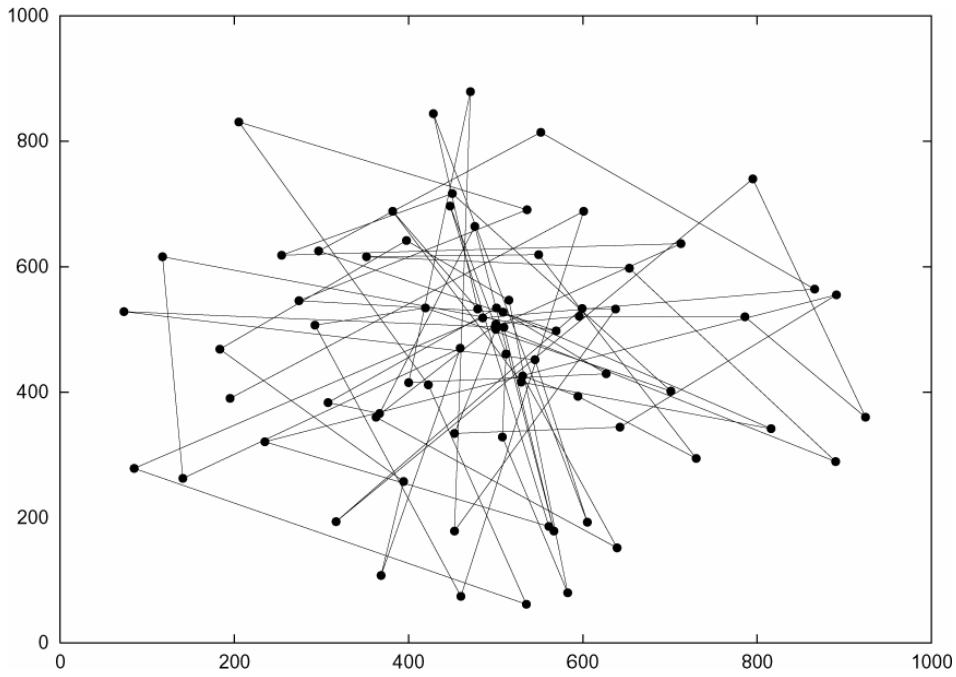

Abbildung 8:Random-Walk

[1] Heute gibt es zur Simulation der Bewegung von Fußgängern wesentlich besser geeignete Modelle. Random-Walk stellt kaum eine realistische Simulation für dieses Szenario dar.

Aus dieser Abbildung wird deutlich, dass häufig abrupte Richtungswechsel vorkommen. Daneben gibt es wegen der zufälligen Geschwindigkeitswahl ohne Rücksicht auf die vorige Geschwindigkeit auch zu ruckartigen Änderungen in der Bewegung. Deswegen finden sich in der Realität wenige Bewegungen, die durch dieses Modell simuliert werden können.

2.3.2 Probabilistische Version von Random-Walk

Bei diesem Modell handelt es sich um eine Abwandlung vom Random-Walk-Modell, die in [CHI98] veröffentlicht wurde. Es verwendet eine Wahrscheinlichkeitsmatrix und ein Zustandübergangsdiagramm, um die Position eines Teilnehmers im nächsten Schritt zu bestimmen. Für die nächste einzunehmende Position gibt es jeweils drei Zustände im Zustandsübergangsdiagramm zur Berechnung der neuen x- und y-Werte (siehe Abbildung 9) : Ist das Zustandsübergangsdiagramm in Zustand 0, so verbleibt der Teilnehmer an der aktuellen Position („stehen bleiben“). Geht das Zustandsübergangsdiagramm in den Zustand 1 über, so bewegt sich der Teilnehmer auf seine vorige Position („ein Schritt zurück“). Zustand 3 im Zustandsübergangsdiagramm weist den Teilnehmer an, sich in der aktuellen Richtung weiterzubewegen („ein Schritt vorwärts“). Die x- und y-Werte des Teilnehmers werden mit Hilfe des Zustandübergangsdiagramms jeweils getrennt berechnet. Die verwendete Wahrscheinlichkeitsmatrix hat folgendes Aussehen:

$$P = \begin{pmatrix} P(0,0) & P(0,1) & P(0,2) \\ P(1,0) & P(1,1) & P(1,2) \\ P(2,0) & P(2,1) & P(2,2) \end{pmatrix}$$

wobei $P(a,b)$ die Wahrscheinlichkeit angibt, dass der Automat vom Zustand a in den Zustand b übergeht. In [CHI98] wird folgende Matrix verwendet:

$$P = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.3 & 0.7 & 0 \\ 0.3 & 0 & 0.7 \end{pmatrix}$$

Abbildung 9 zeigt das Zustandübergangsdiagramm, der sich aus der obigen Matrix ergibt. In der probabilistischen Version von Random-Walk bewegen sich alle Teilnehmer zufällig mit einer festgelegten durchschnittlichen Geschwindigkeit.

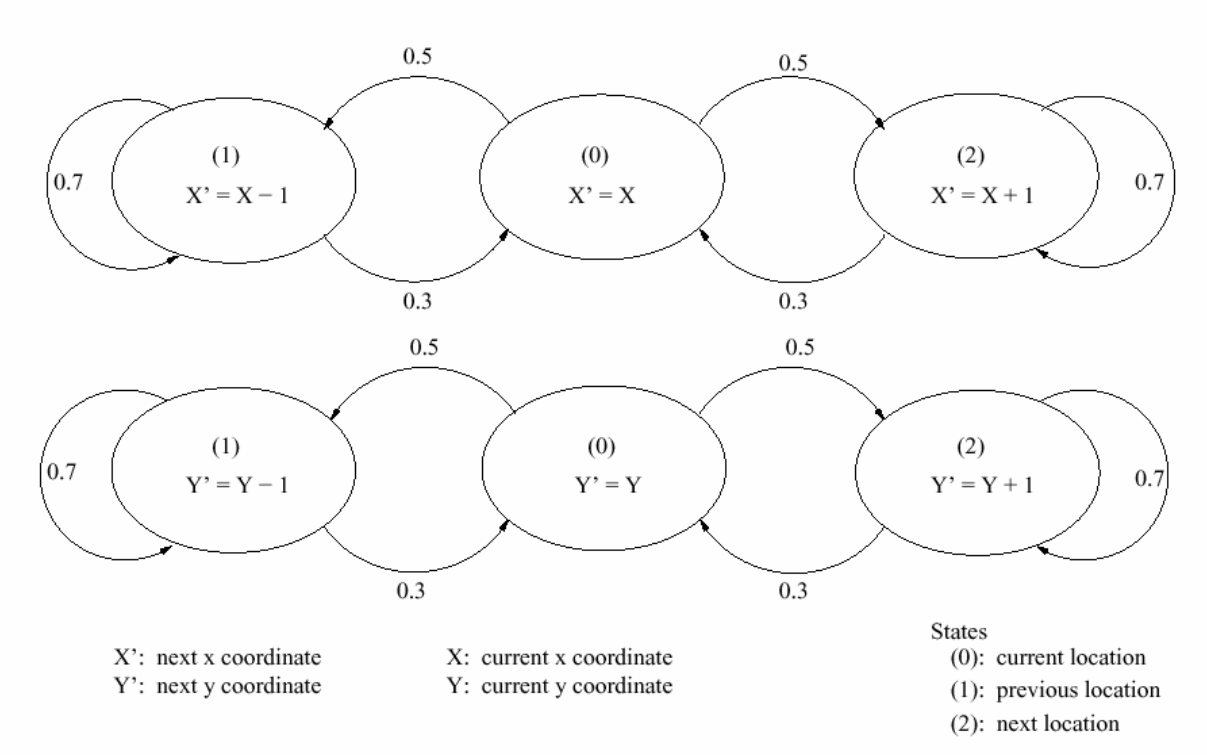

Abbildung 9: Automat zur Berechnung der neuen Position im probabilistischen Random-Walk

Das Probabilistische Random-Walk-Modell macht die Zufälligkeit im Random-Walk-Modell durch Vergabe fester Wahrscheinlichkeiten berechenbarer. Dabei ist es allerdings sehr schwer, geeignete Wahrscheinlichkeiten zu wählen, wenn diese nicht anhand von vorhandenen Spuren ermittelt werden können.

2.3.3 Incremental-Modell

In diesem Modell nach [HAA97] gibt es keine plötzlichen Stop-and-Go-Bewegungen wie z.B. in Random-Walk. Geschwindigkeit und Bewegungsrichtung werden nur graduell geändert. In jedem Zeitintervall werden x- und y-Position neu berechnet basierend auf der letzten Geschwindigkeit und Bewegungsrichtung. Darüber hinaus werden die Geschwindigkeit und Bewegungsrichtung neu bestimmt. Alles in allem kommt dadurch eine weiche Bewegung ohne abrupte Richtungs- oder Geschwindigkeitswechsel zustande.

Die Werte des nächsten Schritts zum Zeitpunkt $t + \Delta t$ werden aus den Werten des aktuellen Schritts zum Zeitpunkt t durch folgende Formeln berechnet:

$$v(t+\Delta t) = \min[\max(v(t)+\Delta v, 0), v_{MAX}]$$
$$\Theta(t+\Delta t) = \Theta(t) + \Delta\Theta$$
$$x(t+\Delta t) = x(t) + v(t) * \cos\Theta(t) * \Delta t$$
$$y(t+\Delta t) = y(t) + v(t) * \sin\Theta(t) * \Delta t$$

Dabei ist die Änderung der Geschwindigkeit gleichverteilt zwischen –A und A (maximale Verzögerung/Beschleunigung). Die Bewegungsrichtungsänderung ist ebenfalls gleichverteilt zwischen $-\alpha$ und α (α ist der maximale Winkel, um den die Richtung geändert wird). v_{MAX} ist die maximale Geschwindigkeit, $\Theta(t)$ die Bewegungsrichtung zum Zeitpunkt t und $\Delta\Theta$ die Änderung der Bewegungsrichtung.

2.3.4 Boundless-Simulation-Area-Modell

Das Boundless-Simulation-Area-Modell erweitert das Incremental-Modell. Es unterscheidet sich von anderen Modellen in der Behandlung der Simulationsgrenzen. In den bisher betrachteten Modellen wird ein Teilnehmer von den Grenzen des simulierten Gebiets reflektiert oder stoppt seine Bewegung ganz. Im Boundless-Simulation-Area-Modell erscheint ein Teilnehmer jedoch auf der entgegengesetzten Seite wieder im simulierten Gebiet, wenn er eine Gebietsgrenze überschreitet. Die Formeln zur Berechnung der Position eines Teilnehmers sind die gleichen wie beim Incremental-Modell. Abbildung 10 zeigt die Spur eines Teilnehmers, dessen Bewegung mit dem Boundless-Simulation-Area-Modell in einem Gebiet der Größe 1000 mal 1000 simuliert wird.

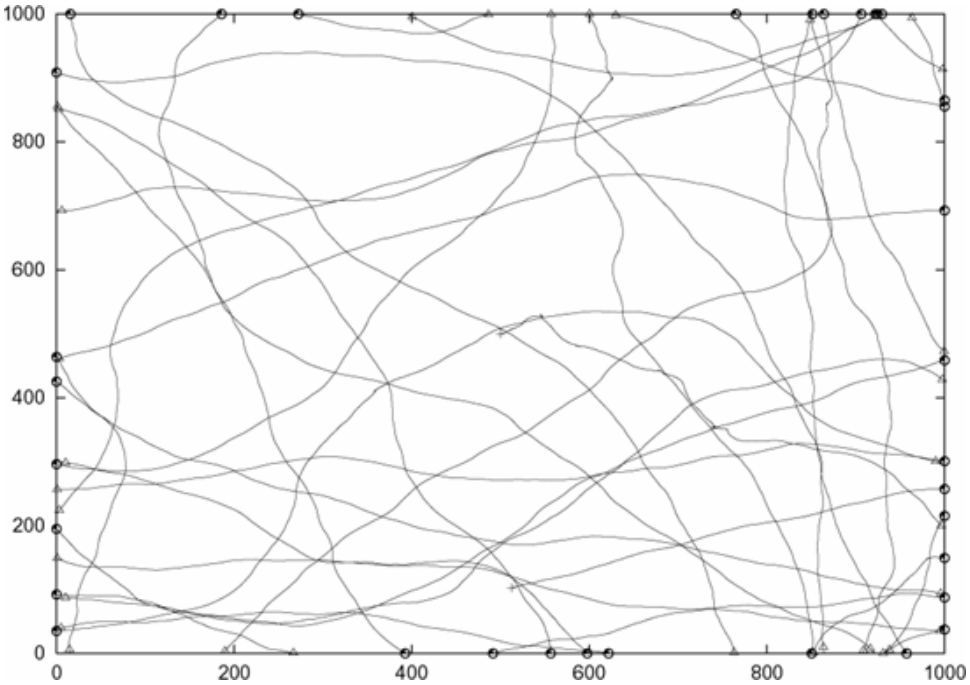

Abbildung 10: Boundless-Simulation-Area-Mobility-Model

Vergleicht man Abbildung 8 mit Abbildung 10, so erkennt man, dass beim Boundless-Simulation-Area-Modell die Aufenthaltswahrscheinlichkeit für einen Knoten im gesamten Simulationsgebiet gleich ist. Beim Random-Walk-Modell ist die Aufenthaltswahrscheinlichkeit jedoch in der Mitte des Simulationsgebiets größer, d.h. bei einer Simulation mit mehreren Teilnehmern ist zu erwarten, dass die Teilnehmerdichte dort ebenfalls höher ist. Die Teilnehmerdichte bestimmt maßgeblich die Anzahl der Nachbarn.

2.3.5 Fluid-Flow-Modell

Dieses Modell wurde entwickelt, um die typische Bewegung von Flugzeugen zu simulieren. Jeder Teilnehmer wählt in diesem Modell zufällig Richtung und Geschwindigkeit und beginnt, sich zu bewegen. Nach einer festgelegten Zeit endet die Bewegung, der Teilnehmer wählt abermals Richtung und Geschwindigkeit beginnt von vorne. Diese Bewegung ist besser vorhersehbar als das Random-Walk-Modell. Beschrieben wird dieses Modell in [THO88], [LEU94] und [FRO94]

2.3.6 Random-Gauß-Markov-Modell

In [HAA99] wird das Random-Gauß-Markov-Modell vorgestellt. Es ist eine Mischung aus Random-Walk und Fluid-Flow. [TOL99] entwickelt dieses Modell weiter und adaptiert es für mobile Ad-hoc-Netze. Die beiden oben erwähnten Modelle werden dabei als Extreme angesehen. Die meisten mobilen Knoten bewegen sich mit einer Mischung aus Random-Walk und Fluid-Flow. Das Modell hat zum Ziel, verschiedene Stufen von Zufälligkeit zu ermöglichen. Dabei soll der Grad, in welchem der Zufall Einfluss auf die Bewegung hat, mit nur einem Parameter einstellbar sein.

Zu Beginn bekommt jeder Teilnehmer eine Geschwindigkeit und Bewegungsrichtung zugewiesen. In festen Zeitintervallen werden die Werte für Geschwindigkeit und Richtung jedes Knotens geändert. Geschwindigkeit und Richtung im n-ten Zeitintervall werden aus den Werten im n-1 ten Intervall nach folgender Formel berechnet:

$$s_n = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}} * \Delta t$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}} * \Delta t$$

mit s_n und d_n neue Geschwindigkeit und Richtung im Intervall n, $0 \le \alpha \le 1$ Parameter, der den Grad des Zufalls wählt, $\bar{s}$ und $\bar{d}$ vorgegebene durchschnittliche Geschwindigkeit und Richtung für $n \to \infty$, $s_{x_{n-1}}$ und $d_{x_{n-1}}$ gaußverteilte Zufallsvariablen. Random-Walk[2] kann durch $\alpha = 0$ erreicht werden, total lineares Verhalten (Fluid-Flow[3]) durch $\alpha = 1$. In jedem Zeitintervall wird die nächste Position auf Basis der aktuellen Position, Geschwindigkeit und Richtung berechnet. D.h. im Intervall n:

$$x_n = x_{n-1} + s_{n-1} * \cos d_{n-1}$$

$$y_n = y_{n-1} + s_{n-1} * \sin d_{n-1}$$

Teilnehmer, die sich in einer bestimmten Distanz zu einem Rand der Simulation befinden, werden gezwungen, sich von dort wegzubewegen. Erreicht wird das, indem $\bar{d}$ (durchschnittliche Richtung) aus obiger Gleichung entsprechend angepasst wird. Abbildung 11 zeigt die

[2] entspricht einem hohen Zufallsanteil
[3] entspricht einem niedrigen Zufallsanteil

Spur eines mit Random Gauß-Markov bewegten Teilnehmers mit $\alpha = 0{,}25$ in einem Simulationsfeld von 1000 mal 1000.

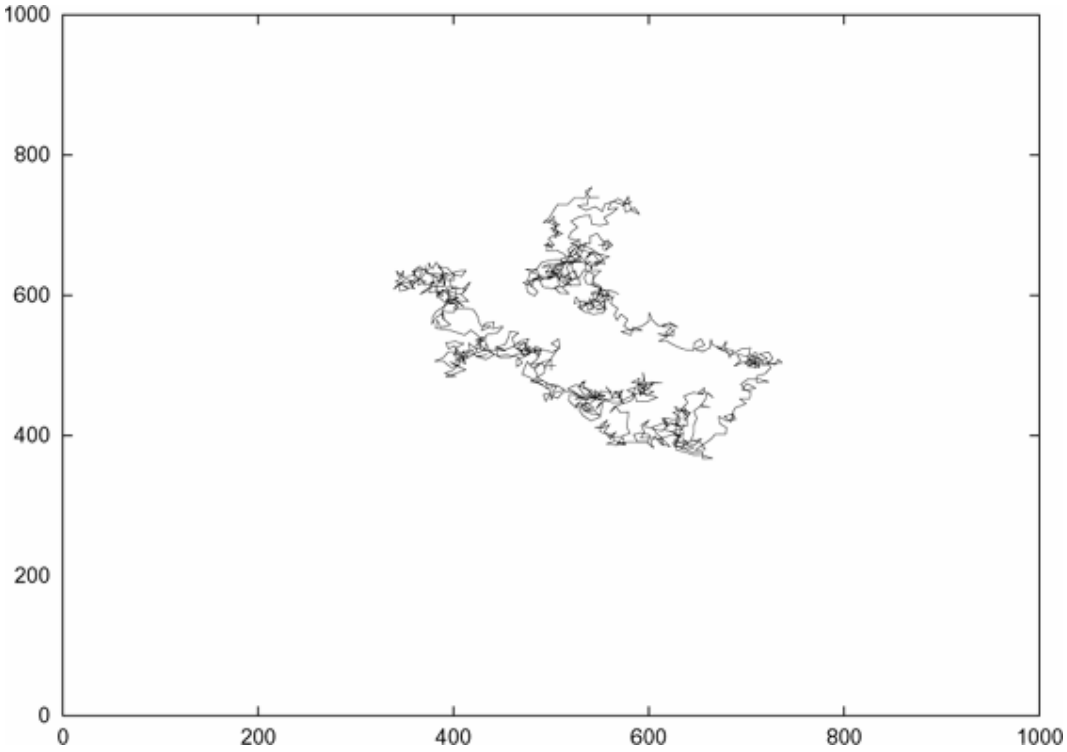

Abbildung 11: Random-Gauß-Markov-Modell

2.3.7 Random-Mobility-Modell

Dieses Modell aus [HON99] erweitert das Random-Gauß–Markov-Modell um Gruppenmobilität. Dabei bewegen sich sowohl die Gruppe als auch die Teilnehmer der Gruppe zufällig. Die tatsächliche Bewegung eines mobilen Teilnehmers ist die Summe aus seinem individuellen Bewegungsmoment und der Gruppenbewegung. Hiermit kann zum Beispiel das Verhalten von Infanterieeinheiten in einem Gefecht simuliert werden.

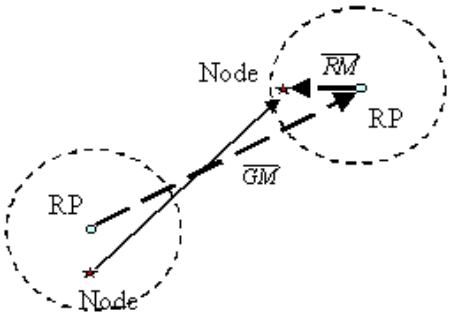

Abbildung 12: Gruppenbewegung im Random-Mobility-Modell

Die Gruppenbewegung basiert auf dem Weg, den das logische Zentrum (Ortsvektor GP) jeder Gruppe zurücklegt. Aus dem logischen Zentrum wird die Gruppenbewegung in Form des Gruppenbewegungsvektors $\overline{GM}$ berechnet. Jede Gruppe hat einen Radius, innerhalb dessen sich die Teilnehmer der Gruppe bewegen. Verschiedene Gruppen können sich überlappen. Die Bewegung des Gruppenzentrums legt bereits die Bewegung der Teilnehmer der Gruppe in groben Zügen fest. Jeder Gruppenteilnehmer berechnet seinen individuellen Referenzpunkt. Dazu verwendet er einen beim Start festgelegten Verschiebungsvektor $\overline{RP}$. $\overline{RP}$ gibt in jedem Schritt die relative Position des Referenzpunkts zum logischen Gruppenzentrum GP an. Um den Referenzpunkt herum kann sich der Knoten nun zufällig bewegen. Dies wird durch einen Vektor $\overline{RM}$ bewerkstelligt, der die individuelle Mobilität eines jeden Knoten definiert und in jedem Schritt zufällig neu gewählt wird. Die Position zur Zeit t+1 berechnet sich also nach folgender Formel:

$$GP(t+1)=GP(t)+GM$$

$$Pos(t+1)=GP(t+1)+RP+RM$$

2.3.8 Reference-Point-Group-Mobility-Modell

Reference-Point-Group-Mobility (RPGM) ist eine Erweiterung von Random-Mobility. In diesem Modell muss die Gruppe festgelegte Wegpunkte erreichen, bevor sie am Ziel ankommt. Damit lassen sich Hindernisse für die Bewegung modellieren.

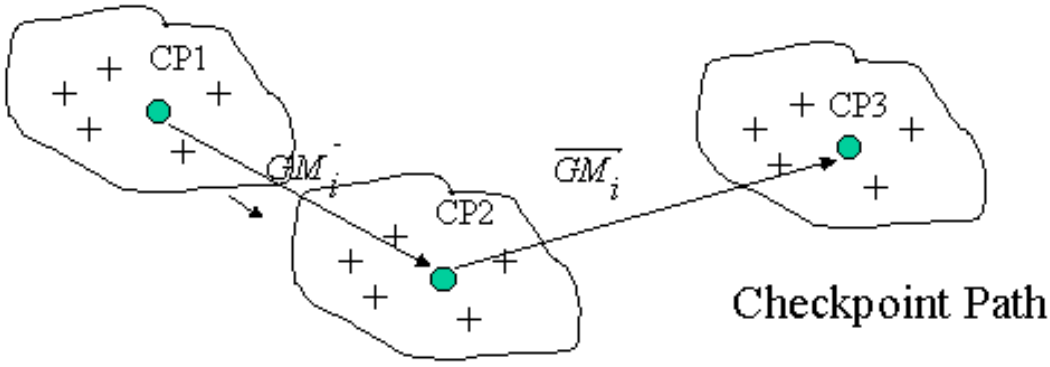

Abbildung 13: Gruppenbewegung im RPGM

2.3.9 Random-Waypoint-Modell

Random-Waypoint wurde erstmals in [JOH96] verwendet. Das Modell teilt die Bewegung eines Teilnehmers in Phasen der Ruhe und Phasen der Bewegung auf. Jedem Teilnehmer wird eine gewisse Ruhezeit zugeordnet. Er wartet diese Zeitspanne, sucht sich dann im Simulationsgebiet ein beliebiges Ziel, wählt eine zufällige Geschwindigkeit (gleichverteilt zwischen min_speed und max_speed) und bewegt sich mit den gewählten Werten auf sein Ziel zu. Wurde das Ziel erreicht, dann wartet der Teilnehmer der ihm zugewiesenen Wartezeit entsprechend, bevor er sich wieder wie oben in Bewegung setzt. Die Wartezeit ist zwischen 0 und max_Wartezeit normalverteilt. Bei einer Wartezeit von 0 ähnelt die erzeugte Bewegung der von Random-Walk. Das Random-Waypoint-Modell ist im Simulationsbereich sehr beliebt und wird oft eingesetzt. Allerdings beobachtet man in Random-Waypoint ebenso wie bei Random-Walk, dass die Aufenthaltswahrscheinlichkeit der Teilnehmer im Zentrum des Simulationsgebiets höher ist als an den Rändern. Eine höhere Aufenthaltswahrscheinlichkeit bedeutet in der Simulation, dass dort die durchschnittliche Teilnehmeranzahl und damit auch die durchschnittliche Anzahl von Nachbarn größer sind. [ROY01] beschäftigt sich mit diesem Problem und schlägt als Lösung unter anderem das Random-Direction Modell (siehe Kapitel 2, Absatz 3.10) vor.

Random-Waypoint unterscheidet sich von Fluid-Flow vor allem dadurch, dass hier das Ziel ausgewählt wird und die Bewegung erst an diesem Ziel endet. Bei Fluid-Flow dagegen bewegen sich alle Knoten im Takt, d.h. es wird immer nach derselben Zeitspanne die Richtung und Geschwindigkeit gewechselt.

Abbildung 14 zeigt die Spur eines mit Random-Waypoint bewegten Knotens auf einem Simulationsfeld der Größe 1000 mal 1000.

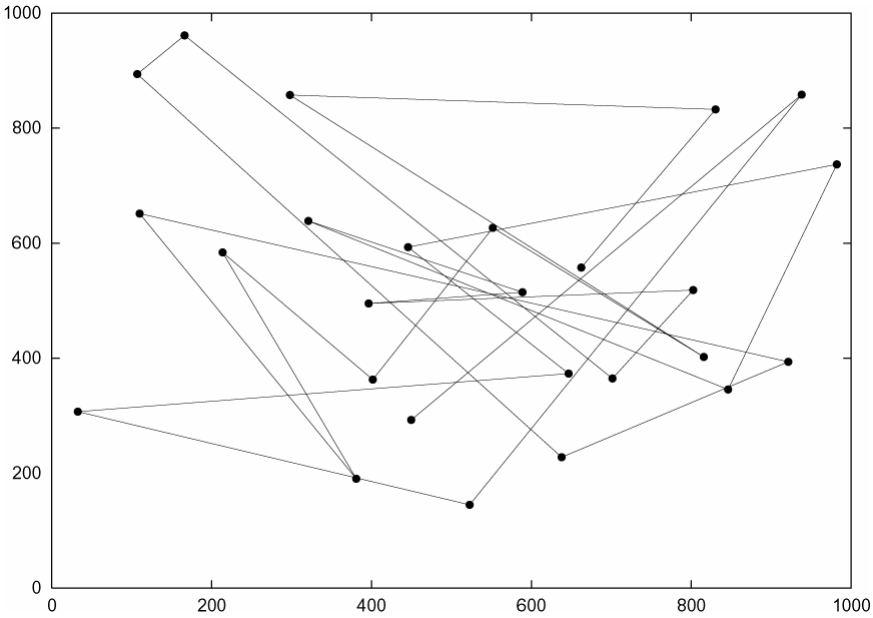

Abbildung 14: Random-Waypoint-Modell

2.3.10 Random-Direction-Modell

[ROY01] beschäftigt sich mit dem oben angesprochenen Problem der unterschiedlichen Teilnehmerdichteverteilung im Simulationsgebiet. Als Simulationsmodell wird das Random-Direction-Modell vorgeschlagen, welches eine gleichmäßige Dichteverteilung erzeugt. In diesem Modell wählen die Teilnehmer eine zufällige Bewegungsrichtung ähnlich Random-Waypoint. Dann bewegen sie sich zum Rand des Simulationsgebiets in der gewählten Bewegungsrichtung. Ist der Rand erreicht wartet jeder Teilnehmer eine vorgegebene Zeit, wählt eine andere Richtung (Winkel zwischen 0 und 180°) und wiederholt obiges Vorgehen. Dieses Modell verhindert „Dichtewellen“, wie sie beim Random-Waypoint-Modell vorkommen, indem die Teilnehmer besser über das Simulationsgebiet verteilt werden. Dadurch wird eine über den gesamten Simulationslauf annähernd konstante Anzahl von Nachbarn erreicht. In Abbildung 15 wird dieses Verhalten deutlich. Die Abbildung zeigt die Spur eines Knotens, der mit dem Random-Direction-Modell bewegt wurde.

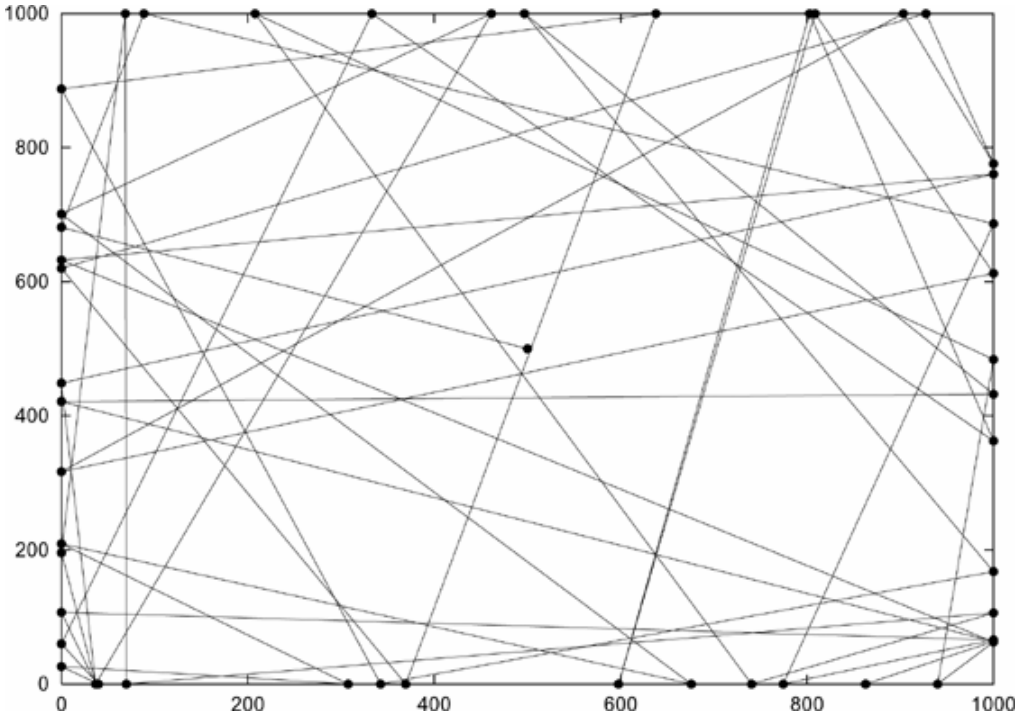

Abbildung 15: Random-Direction-Modell

2.3.11 City-Section-Modell

In [MAR97] werden die Anforderungen an ein Modell formuliert, das einen Teil des Straßennetzes einer Stadt nachbildet. [DAV00] konkretisiert diese Anforderungen. Danach besteht das Simulationsgebiet aus einem Straßennetz. Der Verlauf der Straßen und deren Charakteristik (Geschwindigkeitslimits etc.) werden durch die zu simulierende Stadt bestimmt. Jeder Teilnehmer beginnt an einem vordefinierten Punkt auf einer Straße. Er sucht sich dann zufällig ein Ziel (dies ist ebenfalls ein Punkt auf einer Straße). Der Bewegungsalgorithmus ermittelt den kürzesten Pfad zwischen dem Anfangspunkt und dem Ziel und bewegt den Teilnehmer dort hin. Dabei werden lokale Geschwindigkeitsbegrenzungen und minimaler Abstand zwischen den Fahrzeugen berücksichtigt. Am Ziel wartet der Teilnehmer eine vorgegebene Zeit, sucht dann ein neues Ziel und wiederholt die Prozedur. In diesem Modell müssen alle Teilnehmer vordefinierten Pfaden folgen.

Abbildung 16 zeigt die Spur eines Knotens, der im Rahmen des City-Section-Modells bewegt wurde. Simuliert wurden hier 6*6 Häuserblocks und Straßen dazwischen.

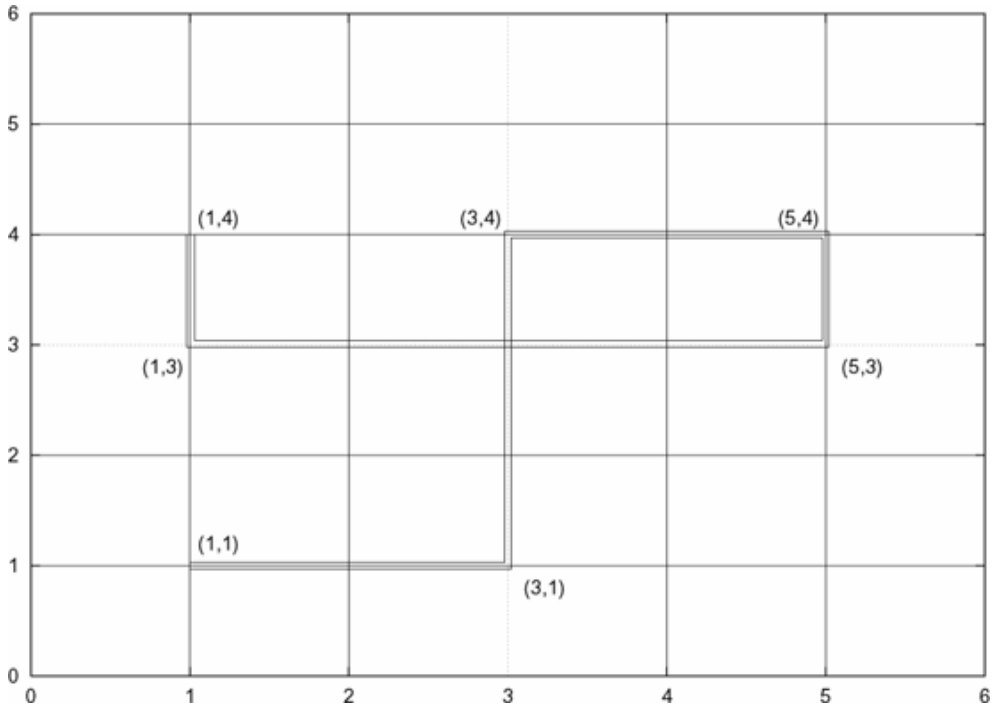

Abbildung 16: City Section Modell

Das City-Section-Modell beschränkt die Bewegung der Knoten im Simulationsgebiet stark und legt der erzeugten Bewegung bestimmte Charakteristika auf. Dieses Modell ist beliebig erweiterbar. So können z.B. an Kreuzungen und Zielen Wartezeiten eingeführt, typische Beschleunigungs- und Bremsvorgänge nachgebildet und für Tageszeiten typische Knotenanzahlen verwendet werden.

2.3.12 Expontential-Correlated-Random-Modell

In diesem Modell wird eine Bewegungsfunktion festgelegt, die auf jeden Teilnehmer angewendet wird. Die Funktion lautet:

$$b(t+1) = b(t)e^{-\frac{1}{\tau}} + \left(\sigma\sqrt{1-\left(e^{-\frac{1}{\tau}}\right)^2}\right)r$$

Dabei bezeichnet $b(t)$ die Position zum Zeitpunkt t, τ bestimmt den Grad der Änderung von der vorigen Position zur nächsten (kleines τ bedeutet große Änderungen) und r ist eine gaußsche Zufallsvariable mit Varianz σ. Dieses Modell war das erste Gruppenmobilitätsmodell.

Nach [HON99] ist es ist schwer, zu einem gegebenen Bewegungsmuster die passenden Parameter für die Bewegungsfunktion zu finden.

2.3.13 Markovian-Modell

Dieses Bewegungsmodell wird gewählt, um ruhige Flugbahnen zu simulieren. Zur Berechnung der Bewegung wird ein Zustandsübergangsdiagramm verwendet (siehe Abbildung 17). Es gibt im Zustandsübergangsdiagramm drei Zustände: Linksbewegung, Stillstand und Rechtsbewegung. In jedem Zeitintervall kann das Zustandsübergangsdiagramm entweder in einem Zustand bleiben oder in einen anderen wechseln. Die Position eines Teilnehmers berechnet sich dann aus der aktuellen Position und einer Linksbewegung, Rechtsbewegung oder Stillstand.

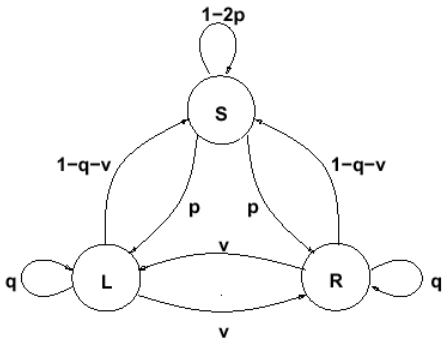

Abbildung 17: Zustandsautomat für das Markovian-Modell

Es handelt sich um ein verfeinertes Zufallsmodell. Der nächste Standort hängt vom gegenwärtigen Standort ab und berücksichtigt kleine zufällige Änderungen. Dadurch können weichere Bewegungen als bei Random-Walk erzeugt werden. In Abbildung 17 bedeutet der Zustand „L“ Bewegung nach links, „R“ steht für Rechtsbewegung und „S“ für Stillstand. Die Beschriftung der Kanten gibt die Übergangswahrscheinlichkeit von einem Zustand in einen anderen an. Beschrieben wird dieses Modell unter anderem in [CHI98] und [BAR95].

2.3.14 Pursue-Modell

Wie der Name schon sagt, gibt es in diesem Gruppenbewegungsmodell einen Knoten, dem alle anderen folgen. Die Verfolger bewegen sich in Richtung des Zielknotens. Die Geschwindigkeit der Verfolger ist normalerweise begrenzt. Das Bewegungsziel der Verfolger wird durch einen Zufallsvektor (random vector) um einen gewissen Grad zufällig verfälscht, um Umgebungsbedingungen zu simulieren. Der Zufallsvektor wird aus einem Bewegungsmodell für einzelne Knoten gewonnen (z.B. Random-Walk). Dabei werden allerdings nur kleine Abweichungen erlaubt, um nach wie vor die Verfolgung des Zielknotens zu erreichen. Die Werte für den Zufallsvektor können berechnet werden, indem ein Bewegungsmodell mit sehr kleinem Simulationsgebiet parallel zur Ausführung des Pursue-Modells ausgeführt wird.

Das Modell besteht aus einer einzigen Gleichung:

New_position=old_position+acceleration(target-old_position)+random_vector.

Der Ortsvektor New_position berechnet sich also aus dem Ortsvektor old_position, einem Zufallsvektor (random_vektor) und einem Bewegungsvektor, der Ergebnis der Funktion accleration() ist. Diese Funktion berechnet eine beschleunigte Bewegung zwischen der alten Position (old_position) und dem Ziel (target). Die Geschwindigkeit ist dabei nach oben begrenzt. [SAN02] stellt ein Java-Applet zur Verfügung, das dieses Modell veranschaulicht.

2.3.15 Column-Modell

In diesem Modell nach [SAN02] sind alle Teilnehmer anfangs in einer Zeile oder Spalte angeordnet. Dann bewegen sie sich alle in die gleiche Richtung, wobei es kleine zufällige Unterschiede in der Geschwindigkeit gibt, um das Feld auseinander zu ziehen. Mit diesem Modell kann zum Beispiel ein Fischschwarm simuliert werden, der flussabwärts schwimmt.

Das Modell kann auch so implementiert werden, dass sich die Teilnehmer in einer festen Entfernung um eine Linie herum bewegen, die sich vorwärts bewegt. Realisiert werden kann dies, indem für jeden Teilnehmer anhand der sich bewegenden Linie in jedem Zeitintervall ein Referenzpunkt auf der Linie festgelegt wird, um den herum sich der Teilnehmer nach ei-

nem individuellen Bewegungsmodell (z.B. Random-Walk) bewegen kann. Der neue Referenzpunkt wird festgelegt als:

new_reference_point= old_reference_point + advance_vector.

Durch die Referenzpunkte aller Teilnehmer entsteht ein Referenz-Gitter. Diese Implementierung eignet sich z.B. um Soldaten zu simulieren, die in Schützenreihen vorrücken.

[SAN02] stellt ein Java-Applet zur Verfügung, das dieses Modell veranschaulicht.

2.3.16 Nomadic-Community-Mobility-Modell

Jeder Teilnehmer dieses Gruppenbewegungmodells benutzt sein eigenes Bewegungsmodell (z.B. Random-Walk), um sich um einen gegebenen Referenzpunkt herum zu bewegen. Wird ein neuer Referenzpunkt für eine Gruppe gesetzt, dann bewegen sich alle Gruppenmitglieder dort hin und setzten dort ihre Bewegung nach dem eigenen Bewegungsmodell fort. Die Parameter der individuellen Bewegungsmodelle legen fest, wie weit sich die Teilnehmer von ihrem Referenzpunkt entfernen können. Im Nomadic-Community-Mobility-Modell teilen sich alle Mitglieder einer Gruppe denselben Referenzpunkt, während im Column-Modell jeder Knoten seinen eigenen Referenzpunkt hat. Mit diesem Modell kann zum Beispiel eine Besuchergruppe während einer Führung durch ein Museum beschrieben werden. [SAN02] stellt ein Java-Applet zur Verfügung, das dieses Modell veranschaulicht.

2.3.17 Mobility-Vector-Modell

In diesem Modell nach [KWO99] wird die Mobilität eines Knotens durch einen Vektor (x_v, y_v) beschrieben, der die (zweidimensionale) Geschwindigkeit eines Knotens angibt. Die Norm des Vektors ist die Geschwindigkeit des Knotens, die durch die Entfernung zwischen dem augenblicklichen Standort und dem Standort in der nächsten Zeiteinheit beschrieben wird. Der Mobilitätsvektor $\vec{M} = (x_m, y_m)$ oder (r_m, θ_m) ist die Summe von zwei Untervektoren: dem Basisvektor $\vec{B} = (bx_v, by_v)$ oder (r_b, θ_b) und dem Abweichungsvektor $\vec{V} = (vx_v, vy_v)$

oder (r_v, θ_v). Der Basisvektor bestimmt dabei die grobe Richtung und Geschwindigkeit des Knotens. Er wird zum Beginn der Bewegung gewählt. Der Abweichungsvektor speichert die Mobilitätsabweichung vom Basisvektor. Er wird während der Bewegung immer wieder neu berechnet. $\overrightarrow{M}$ berechnet sich als: $\overrightarrow{M} = \vec{B} + \alpha \times \vec{V}$ mit dem Beschleunigungsfaktor α. Werden α und die Geschwindigkeit (aus dem Intervall $[\min, \max]$) geeignet gewählt, so wird eine weiche Bewegung erreicht, die frei von abrupten Richtungsänderungen ist, wie sie z.B. bei Random-Walk vorkommen.

2.3.18 Gravity-Modell

In einigen Funkszenarien neigen Empfänger dazu, sich auf die Quelle des Signals hinzuzubewegen, um einen besseren Empfang zu bekommen. Diese Art der Bewegung wird durch das Gravity-Modell beschrieben. Jeder Teilnehmer besitzt eine Polung, die entweder positiv oder negativ ist. Es können auch Teilnehmer ohne Polung vorkommen. Diese sind frei von Gravitation (Gravity). Teilnehmer gleicher Polung stoßen sich ab, Teilnehmer unterschiedlicher Polung ziehen sich an. Die Anziehungs- oder Abstoßungskraft zwischen zwei Teilnehmern kann durch obiges Mobility-Vector-Modell beschrieben werden (Anpassung der Basisvektoren).

Eine sinnvolle Erweiterung dieses Modells ergibt sich, wenn man jedem Knoten noch zusätzlich eine Masse zuordnet. Aus der Anziehungs- oder Abstoßungskraft und der Masse lässt sich die Beschleunigung des Knotens nach dem Newtonschen Gesetz nach folgender Formel berechnen:

$$a = \frac{F}{m}$$

Stationäre Teilnehmer wie z.B. Infrastruktureinrichtungen können dabei mit einer unendlichen Masse an der Bewegung gehindert werden. In manchen Fällen ist auch eine Abstoßung zwischen Teilnehmern gleicher Polung uninteressant und kann ausgeklammert werden. Das Gravity-Modell wird in [HON01] beschrieben.

2.3.19 Location-Dependent-Modell

Ebenso wie das Gravity-Modell ist auch dieses Modell eine Abwandlung des Mobility-Vector-Modells. Hier werden gemeinsame Bewegungen in einem bestimmten Bereich simuliert, wie sie zum Beispiel bei Fahrzeugen auf einer Straße vorkommen. Jeder Punkt im Simulationsgebiet erhält einen Vektor, der die Bewegungsrichtung in diesem Punkt angibt. Teilnehmer in diesem Modell bewegen sich wie im Mobility-Vector-Bewegungsmodell angegeben, allerdings ist der Basisvektor eines jeden Teilnehmers nicht konstant. Der Teilnehmer nimmt jeweils den Vektor, der für die aktuelle Position im Simulationsgebiet definiert ist, auf. Das Modell wird in [HON01] beschrieben.

3. Konzepte

Dieses Kapitel stellt Konzepte vor, die als theoretische Grundlage für die Implementierung dienen. Außerdem wird auf konzeptionelle Änderungen und Erweiterungen der beschriebenen Sicherheitsarchitektur eingegangen.

3.1 Szenarien für den Einsatz von Ad-hoc-Netzen

Die vorhergehenden Abschnitte stellten einige gebräuchliche Bewegungsmodelle vor. Dabei handelt es sich um formale Modelle für konkrete Szenarien aus der Realität. So kann das Column-Modell z.B. zur Simulation von Soldaten im Gefecht und das Fluid-Flow-Modell zur Simulation der Bewegung von Flugzeugen verwendet werden. Im Folgenden werden zwei alltägliche Szenarien für den Einsatz von Ad-hoc-Netzen vorgestellt. Dabei wird Wert darauf gelegt, Szenarien mit ganz unterschiedlichen Eigenschaften zu finden. Anhand der Szenarien werden später Bewegungsmodelle für die Simulation abgeleitet.

3.1.1 Konferenz

Eine Konferenz ist das erste von zwei betrachteten Anwendungsszenarien für die Sicherheitsarchitektur. Auf einer Konferenz gibt es verschiedene Veranstaltungen, an denen die Besucher teilnehmen. Jede Veranstaltung hat eine gewisse Dauer. Nach Ende verteilen sich die Teilnehmer üblicherweise relativ schnell auf andere Veranstaltungen. Veranstaltungen können sehr unterschiedliche Größen haben. Workshops, Besprechungen, Vorträge und Führungen sind Beispiele für Veranstaltungen verschiedener Größe. In den Arbeitsräumen, in denen einige Veranstaltungen stattfinden, stehen verschiedene Peripheriegeräte zur Verfügung, die von den Teilnehmern der Arbeitsgruppen genutzt werden können. Unter anderem sind Drucker und Internetanbindung vorhanden. Typischerweise findet eine Konferenz auf einem begrenzten Gebiet statt, in dem sich eine große Zahl von Besuchern aufhält. Die durchschnittliche Dichte der Teilnehmer ist also hoch. Dadurch können die entstehenden Ad-hoc-Netze sehr groß werden, selbst wenn die Kommunikationseinrichtungen der Besucher nur über eine geringe Reichweite verfügen.

Die Kommunikationsgeräte der Besucher reichen von PDAs mit Bluetooth-Chips (siehe Anhang A) bis zu Laptops mit IEEE 802.11b Funk-LAN-Karten (Siehe Anhang A). Typischerweise verfügen diese Geräte nur über eine geringe oder mittlere Sendeleistung. Da die Geräte batteriebetrieben sind, ist energiesparender Umgang mit den vorhandenen Ressourcen nötig. Die Sendereichweiten sind beschränkt und reichen im Ad-hoc-Modus von 10 bis 180 m.

3.1.2 Autobahn

Ein weiteres Einsatzszenario ist eine Autobahn. Dieses Szenario bietet ein breites Spektrum an Situationen, die sich hinsichtlich der Übertragungscharakteristika unterscheiden. Eine Sicherheitsarchitektur, die diesem Szenario gewachsen ist, wird auch einer Vielzahl von anderen Anwendungsfällen genügen.

Auf einer Autobahn fahren verschiedene Arten von sehr unterschiedlichen Fahrzeugen. Man kann die Fahrzeuge anhand ihrer Geschwindigkeitsprofile einteilen in die Klassen „schnelle PKWs“, „langsame PKWs“ und „LKWs“. In allen drei Fällen spricht nichts dagegen, diese Fahrzeuge mit leistungsfähiger Elektronik zur Kommunikation auszustatten, d.h. die Prozessorleistung der mobilen Geräte kann als hoch angenommen werden. Auch gibt es kaum Einschränkungen bezüglich der Sendevorrichtungen. Anders als bei Mobiltelefonen, bei denen nur eine begrenzte Länge für die Antenne zur Verfügung steht, sind bei Fahrzeugen in gewissem Rahmen keine solchen Einschränkungen zu erwarten, da genügend Raum zur Verfügung steht. Auch eine gesundheitliche Belastung durch Funkwellen ist nicht zu erwarten, da die Fahrzeugkabine wie ein Faradayscher Käfig wirkt, d.h. eine Begrenzung der Funkleistung muss höchstens wegen gesetzlicher Vorschriften über zugelassene Leistungen erfolgen. Während der Fahrt steht über die leistungsfähige Autobatterie genügend Elektrizität zur Verfügung. Die Stromversorgung stellt somit ebenfalls kein Problem dar. Die Fahrzeuge bewegen sich alle in die gleiche Richtung (pro Spur), und der Ausbreitung des Kommunikationsmediums (z.B. Funk) stehen keine Hindernisse entgegen. Allerdings wirken die Fahrzeuge selbst eventuell als Hindernisse bzw. reflektieren (da aus Metall) Funkwellen.

Ein weiteres Hindernis sind Tunnels. Hier kann es passieren, dass die entstehenden Ad-hoc Netze partitioniert werden.

Die Fahrzeuge auf einer Autobahn bewegen sich im Normalfall sehr schnell. Für LKWs sind bis zu 80 km/h typisch, für langsame PKWs 100-120 km/h und für schnelle PKWs 160-220 km/h und mehr. Ab einer gewissen Geschwindigkeitsdifferenz zwischen zwei Fahrzeugen ist wegen des Dopplereffekts (Frequenzverschiebungen) keine Kommunikation mehr möglich.

Typischerweise bewegen sich die Fahrzeuge je nach Geschwindigkeit auf verschiedenen Spuren der Autobahn. So beanspruchen beispielsweise auf einer dreispurigen Autobahn schnelle PKWs die linke Spur, die mittlere Spur wird von langsamen PKWs und LKWs (beim Überholen) frequentiert, und die rechte Spur wird vorwiegend von LKWs genutzt. Zudem fällt auf, dass auf vielen Autobahnen die LKWs „wie auf einer Schnur aufgereiht“ fahren. Der vorgeschriebene Mindestabstand beträgt hier 50 m. Da sich LKWs über lange Strecken mit annähernd gleicher Geschwindigkeit bewegen besteht hier die Möglichkeit, dass eine Art „Backbone“ für das Ad-hoc-Netz entsteht.

Entscheidend für die Analyse ist auch die Dichte der Fahrzeuge auf der Fahrbahn. Diese variiert meist mit der Tageszeit, allerdings erfolgt die Zunahme/Abnahme der Dichte langsam. Bedingt durch die relativ kleinen Übertragungsreichweiten ist zu erwarten, dass erst ab einer gewissen Verkehrsdichte größere Netze entstehen.

3.2 Authentisierung

Wie bereits in Kapitel 2 angesprochen wurde die Authentisierung für die Sicherheit der Architektur immens wichtig. Die gesamte Sicherheitsarchitektur basiert auf identitätsbezogenen Schlüsselzertifikaten der Knoten. Wird eine unsichere Authentisierung eingesetzt, so kann von einem Schlüsselzertifikat nicht mehr auf die Identität eines Knotens geschlossen werden. Damit ist die Sicherheitsarchitektur hinfällig. Aus diesem Grund wird großer Wert auf eine sichere Authentisierung gelegt.

Es gibt verschiedene Arten der Authentisierung: Bei der *Authentisierung über Zertifikate einer vertrauten Wurzelinstanz* gibt es eine Instanz, der alle Knoten vertrauen. Diese Wurzelinstanz bescheinigt jedem Knoten durch ein Zertifikat seine Identität. Jeder Knoten kann das Zertifikat eines anderen Knoten überprüfen da, die gemeinsame Wurzelinstanz bekannt ist. Bei der Verwendung von Public-Key-Verfahren wird dies z.B. erreicht, indem jeder Knoten

von der Wurzelinstanz neben seinem Zertifikat auch noch den öffentlichen Schlüssel der Wurzelinstanz erhält, mit dem er dann später andere Zertifikate nachprüfen kann.

Eine andere Methode ist die *Authentisierung über Bürgen*, die in der hier betrachteten Sicherheitsarchitektur eingesetzt wird. Dazu wird die Authentisierung eines Knotens auf Bürgen-Knoten verlagert. Diese stellen z.B. mittels physischen Kontakts die Identität eines anderen Knotens fest und erstellen danach ein Bürgen-Zertifikat wie oben beschrieben. In [BAL02] wird eine Methode beschrieben, wie die Authentisierung über einen physischen Kontakt ablaufen kann. Dabei kommt ein so genannter Location-Limited Side Channel (LLSC) zum Einsatz. Dies ist ein physikalischer Übertragungskanal der über spezielle Eigenschaften verfügt, die ihn für die Authentisierung interessant macht:

- Der LLSC ist vom Hauptkommunikationsmedium getrennt.

- Er unterstützt „Demonstrative Identification“, d.h. es ist auf Grund des physikalischen Kontexts klar, wer der Kommunikationspartner ist. Deshalb sind Medien gut geeignet, die durch ihre physikalischen Eigenschaften Einschränkungen bei der Übertragungsreichweite unterliegen (z.B. Infrarot, Schall).

- Weiterhin ist ein Location-Limited Side Channel authentisch, d.h. es ist für einen Angreifer nicht, oder nur schwer, möglich, in den Kanal zu senden bzw. es ist zumindest erkennbar wann ein Angreifer in den Kanal sendet.

Anders als [STA99] setzt die Arbeit [BAL02] nicht voraus, dass der Location-Limited Side Channel Nachrichten geheim überträgt.

Beispiele für Location-Limited Side Channels sind Infrarot und Schall. Infrarot kann sehr einfach abgeschirmt werden. Schon ein Blatt Papier absorbiert Infrarot-Strahlung und macht die Kommunikation unmöglich. Infrarot kann gebündelt abgegeben werden, so dass eine Sichtverbindung zur Kommunikation nötig ist. Damit unterstützt Infrarot auch „Demonstrative Identification“. Audio unterstützt ebenfalls „Demonstrative Identification“, da es einfach ist zu unterscheiden, aus welcher Richtung Schall kommt. Überlagerungen und verschiedene Lautstärken können ebenfalls leicht erkannt werden.

Die Authentisierung unter Verwendung eines LLSC läuft in vier Phasen ab:

1. Zuerst wird eine kurze kryptographische Information, die sich auf den jeweiligen öffentlichen Schlüssel des Knotens bezieht, über den Location-Limited Side Channel übertragen. Diese Information kann z.B. der Hashwert des Schlüssels sein.

2. Anschließend werden in der zweiten Phase die öffentlichen Schlüssel über das Hauptkommunikationsmedium ausgetauscht. Anhand der kryptographischen Information aus Phase eins kann jeder Kommunikationspartner prüfen, ob er auch den richtigen Schlüssel erhalten hat.

3. Erst dann erfolgt die Identifizierung des Gegenübers, der jetzt eindeutig als Kommunikationspartner festgestellt wurde.

4. Abschließend wird das gewünschte Zertifikat im Hauptkommunikationsmedium übermittelt. Es bestätigt die Identität des Gegenübers.

Im Konferenz-Szenario (siehe Kapitel 3, Absatz 1.1) kann Infrarot als Location-Limited Side Channel eingesetzt werden. Die meisten heute gebräuchlichen Laptops, PDAs und Mobiltelefone besitzen eine Infrarotschnittstelle meist nach dem IrDA-Standard. Der Bürge kann seinen Gegenüber entweder über persönliche Bekanntschaft oder anhand eines Ausweises identifizieren.

Im Autobahn-Szenario (siehe Kapitel 3, Absatz 1.2) kann Infrarot nicht eingesetzt werden, denn grelles Tageslicht im Freien stört Infrarot. Im Autobahn-Szenario könnten visuelle Eindrücke als LLSC zum Einsatz kommen. Dazu wird unterhalb des Nummernschilds eine kleine mechanische Zahlenanzeige installiert. Eine Kamera erfasst nun eine dort angezeigte Zahl ebenso wie die Autonummer. Die Zahl ist der Hashwert des öffentlichen Schlüssels.

3.3 Protokollergänzungen

Die Erfahrungen durch die Implementierung der Sicherheitsarchitektur als Simulation haben zu einigen Protokollergänzungen geführt. Diese werden im Folgenden vorgestellt.

3.3.1 Begrenzte Gültigkeit von Zertifikaten

Im ursprünglichen Entwurf der Sicherheitsarchitektur ist eine Begrenzung der Gültigkeitsdauer von Bürgen-Zertifikaten optional möglich. Damit soll eine begrenzte Gültigkeit von Zertifikaten erreicht werden. Eine begrenzte Gültigkeit ist immer dann sinnvoll, wenn ein Bürge nicht für einen längeren Zeitraum für einen Knoten bürgen möchte. Dies kann z.B. der Fall sein bei einem Workshop. Die Teilnehmer kennen sich unter Umständen nicht gegenseitig, möchten aber für die Dauer des Workshops füreinander bürgen. Leider greift der Mechanismus zur Gültigkeitsbegrenzung nicht hundertprozentig. Ein Zertifikat kann zwar mit einer Gültigkeit versehen werden, allerdings erhält ein Knoten nach Anmeldung den symmetrischen Clusterschlüssel des Clusters. Dessen Besitz impliziert die volle Mitgliedschaft im Cluster. Auf dem Besitz des symmetrischen Clusterschlüssels basiert die Sicherheit im lokalen Cluster.

Um trotzdem eine zeitlich begrenzte Mitgliedschaft zu ermöglichen, ist es nötig, den symmetrischen Clusterhead periodisch auszutauschen. Der Clusterhead kennt alle vollen Clustermitglieder seines Clusters. Er sendet in periodischen Abständen neue symmetrische Schlüssel an alle vollen Clustermitglieder. Durch dieses Vorgehen entsteht weiterer Overhead, und während des Schlüsselaustauschs ist keine sichere Kommunikation mit dem symmetrischen Schlüssel möglich Allerdings sind Cluster im Allgemeinen klein und damit hält sich der Kommunikationsoverhead in Grenzen, wenn ein genügend langes Intervall zur Auffrischung gewählt wird. Des Weiteren sind keine beliebigen Gültigkeitszeiten sinnvoll, da der Clusterschlüssel nur in regelmäßigen Abständen ausgetauscht wird. Die Gültigkeit sollte also sinnvollerweise ein Vielfaches des Intervalls sein, in dem der Schlüssel geändert wird. Ist dies nicht der Fall, so wird abgerundet, um sicherzustellen, dass kein Bürge länger für einen Knoten einstehen muss, als ursprünglich beabsichtigt. Die Gültigkeit des identitätsbezogenen Schlüsselzertifikats berechnet sich schließlich aus dem Minimum der Gültigkeitsdauern aller Bürgenzertifikate. Erhält ein Knoten mehrere zeitbegrenzte Bürgenzertifikate, so versucht er,

mehr Zertifikate als eigentlich nötig zu erhalten, um die Zertifikate mit der geringsten Gültigkeitsdauer auszusortieren.

Bei der Erteilung eines Bürgen-Autorisierungs-Zertifikats sollten neben der Autorisierung zur Verteilung von unbegrenzt gültigen Bürgenzertifikaten auch die Möglichkeit bestehen, einem Knoten nur das Recht zu geben, auf einen festen Gültigkeitszeitraum begrenzte Bürgenzertifikate auszugeben. Dadurch kann das CH-Netzwerk die Sicherheitsstufe für seinen gesamten Einflussbereich global festlegen. Ein Clusterhead kann in den Sicherheitsrichtlinien für seinen Cluster festlegen, ob er auch unbegrenzt gültige Bürgenzertifikate akzeptiert.

3.3.2 Entlastung der Clusterheads

Clusterheads nehmen eine besondere Stellung im Cluster ein. Sie organisieren den Cluster und wickeln die Anmeldung neuer Knoten ab. Sie stellen einen integralen Teil der Sicherheitsarchitektur dar und müssen gegen Angriffe geschützt werden. Ein Angreifer könnte versuchen den Clusterhead durch eine große Menge von Zertifizierungsanfragen zu überlasten (Denial-of-Service-Attacke). Da der Clusterhead bei einer Zertifizierungsanfrage die Public-Key-Signaturen einer Anzahl von Bürgen Zertifikaten (Parameter minWarrants) überprüfen muss und dadurch hoher Rechenaufwand entsteht, sollte ein Clusterhead in der Zeitspannen zwischen dem Versenden zweier CH-Beacons maximal zwei Zertifizierungsanfragen eines Knotens zulassen. Damit wird eine Überlastung des CHs verhindert, aber eine missglückte Anmeldung wird dem Knoten zugestanden. Natürlich kann ein Angreifer nun neue Knoten daran hindern, dem Cluster beizutreten, indem er in ihrem Namen zwei Anfragen an den CH schickt. Für die vollen Clustermitglieder bleibt aber ebenso wie für das CH-Netzwerk die Funktionalität des CHs erhalten.

3.3.3 Kurzzeitige Anbindungen an Infrastruktur zur Authentisierung

In manchen Szenarien existieren kurzzeitige Anbindungen an Infrastruktureinrichtungen. So zum Beispiel auch im Konferenz-Szenario (siehe Kapitel 3, Absatz 1.1). Ist ein Cluster mit einem Festnetz verbunden, so besteht nun z.B. die Möglichkeit, Zertifikate einer Wurzelinstanz zu überprüfen (z.B. indem der Public-Key einer Wurzelinstanz besorgt wird).

In Erweiterung der Sicherheitsarchitektur können Gastknoten alternativ zu Bürgen-Zertifikaten und Bürgen-Autorisierungszertifikaten auch Zertifikate einer beliebigen Wurzelinstanz zum Beweis der Identität übermitteln. Es steht Clusterheads frei, statt den Bürgen-Zertifikaten die Zertifikate anderer Instanzen zu verwenden. Die Clusterheads können dies in den Sicherheitsrichtlinien für den eigenen Cluster festlegen. Damit wird optional eine effektive Art der Authentisierung integriert unter Beibehaltung des herkömmlichen Mechanismus. Ist ein Clusterhead an eine Infrastruktur angebunden, so besteht für ihn die Möglichkeit, eine Vertrauensbeziehung mit der Wurzelinstanz einzugehen, indem er sich z.B. deren öffentlichen Schlüssel besorgt.

Bei dem Konferenz-Beispiel ist eine denkbare Wurzelinstanz die Konferenzleitung, die bei Vorlage einer Einladung und eines Personalausweis dem Teilnehmer ein Zertifikat ausstellt und ihren öffentlichen Schlüssel zur Verfügung stellt. Knoten, die über den öffentlichen Schlüssel der Konferenzleitung verfügen, können das Zertifikat überprüfen.

Im Autobahn-Beispiel sind denkbare Wurzelinstanzen die Hersteller der Endgeräte in den Autos, die schon beim Verkauf und Einbau ein Zertifikat im Gerät hinterlegen. Als Infrastruktur kommen Antennen an Brücken in Frage, die eine kurzzeitige Anbindung an das Internet ermöglichen.

3.3.4 Modifikation der Anmeldung

In Kapitel 5, Absatz 4 wird die Aufgabenverteilung in der Sicherheitsarchitektur und die daraus resultierende Belastung der einzelnen Knoten untersucht. Die Belastung der vollen Clustermitglieder und Gateways steigt mit zunehmender Knotenanzahl annähernd linear an. Hier skaliert das Protokoll. Die CHs werden jedoch stark belastet. In Kapitel 5, Absatz 4 wird ebenfalls festgestellt, dass die Anzahl der CHs von der Größe des Simulationsfelds abhängt und nur zu einem geringen Anteil von der Anzahl der Knoten in der Simulation beeinflusst wird. Um eine Überlastung der CHs zu verhindern, ist es deshalb sinnvoll, nur eine begrenzte Anzahl von Knoten in den eigenen Cluster aufzunehmen. Alle weiteren Knoten werden abgewiesen. Kann sich nun ein Knoten bei verschiedenen Clusterheads nicht anmelden, so wird er selbst zum Clusterhead und entlastet damit die schon vorhandenen CHs. Auf diese Art werden künstlich mehr CHs als eigentlich notwendig erzeugt. Dies garantiert aber kleinere Cluster und trägt damit zur Entlastung der gesamten Sicherheitsarchitektur bei.

Es ist darüber hinaus sinnvoll, dass nicht jeder Knoten zum Clusterhead werden kann. Vielmehr sollte auf Grund lokal erhältlicher Informationen geprüft werden, ob der Knoten leistungsfähig genug ist, um die Aufgabe eines Clusterhead zu übernehmen. Dazu kann z.B. ein bereits vorhandener Clusterhead, der einen Knoten abweist, in dem negativen Bescheid eine Empfehlung auf Grund der Größe und Eigenschaften des eigenen CH-Netzwerks aussprechen, welche Eigenschaften ein zukünftiger CH haben sollte. Die Leistungsfähigkeit lässt sich z.B. bemessen an Hand von Hauptspeicher oder Prozessortaktung.

3.3.5 Vereinigung von Clusterhead-Netzwerken

Immer wenn ein Clusterhead ein CH-Beacon eines Clusterheads aus einem anderen CH-Netzwerk empfängt, muss er die Entscheidung treffen, ob er versucht beide CH-Netzwerke zu vereinigen oder nicht. Bei der Vereinigung entsteht großer Aufwand, da alle CHs im neuen Netzwerk in Kooperation einen neuen geheimen Netzwerkschlüssel generieren und alle Signaturen der beiden alten Netzwerke ungültig werden. Knoten mit ungültigen Signaturen müssen ihre öffentlichen Schlüssel neu signieren lassen, wodurch weiterer Overhead entsteht.

Andererseits darf die Entscheidung über Netzvereinigung nicht zu lange hinaus gezögert werden da, sonst gerade in der initialen Entstehungsphase das Gesamtnetz nur sehr langsam wächst. Ein Anhaltspunkt für die Entscheidung ist die Größe der beiden CH-Netzwerke. Trifft ein CH-Netzwerk mit nur einem Mitglied auf ein CH-Netzwerk mit mehreren CHs, so kann in fast allen Fällen eine Vereinigung umgangen werden, indem das CH-Netzwerk mit nur einem Clusterhead sich auflöst, der Clusterhead seine Aufgabe beendet und sich an einem anderen Cluster anmeldet. Um einen zügigen initialen Aufbau zu ermöglichen und für genügend CHs im CH-Netzwerk zu sorgen, sollte dieses Vorgehen nur mit Clusterheads praktiziert werden, die weniger als eine bestimmte Anzahl an Knoten haben.

Abbildung 18 verdeutlicht die Vereinigung von Clusterhead-Netzwerken noch einmal. Die gepunkteten Linien stellen die Verbindungen im CH-Netzwerk dar. Dunkle Kreise symbolisieren Clusterheads, während nicht ausgefüllte Kreise für alle anderen Knoten stehen. In der Abbildung sind drei Fälle dargestellt:

1. Ein CH-Netzwerk mit 4 CHs trifft auf ein CH-Netzwerk, das nur aus einem CH besteht. Das CH-Netzwerk, das nur aus einem CH besteht, löst sich auf und der CH ist nicht länger Clusterhead. Der Aufwand ist in diesem Fall gering, da sich maximal die Knoten des Clusterheads, der seine Tätigkeit eingestellt hat, neu an einem anderen Cluster anmelden müssen.
2. Zwei CH-Netzwerke, die beide aus mehr als einem CH bestehen, treffen aufeinander. Die CH-Netzwerke werden vereinigt. Dadurch entsteht viel Overhead, denn für das neue CH-Netzwerk muss ein neuer Schlüssel konstruiert werden und alle Knoten der alten CH-Netzwerke müssen ihre öffentlichen Schlüssel neu signieren lassen. Allerdings hat ein größeres CH-Netzwerk den Vorteil, dass alle Clusterheads im CH-Netzwerk entlastet werden.
3. Zwei CH-Netzwerke, die nur aus einem CH bestehen, treffen aufeinander. Hier hängt es davon ab, wie viele Knoten zu einem Cluster gehören, ob ein Clusterhead seine Tätigkeit einstellt oder ob ein CH-Netzwerk mit zwei CHs entsteht.

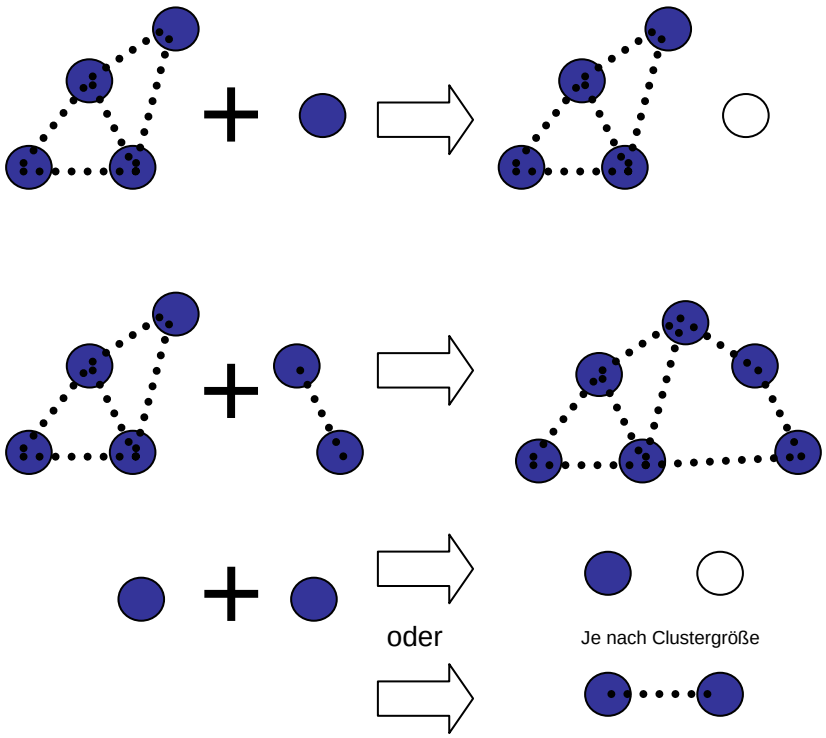

Abbildung 18: Vereinigung von Clusterhead-Netzwerken

3.4 Implementierung der Bewegungsmodelle

Das City-Section-Bewegungsmodell und das Gravity-Modell dienen im Folgenden als Basis für die Entwicklung eigener Bewegungsmodelle für die vorgestellten Anwendungsszenarien.

3.4.1 Ein Bewegungsmodell für das Autobahn-Szenario

Um das Szenario „Autobahn“ (siehe Kapitel 3, Abschnitt 1.2) zu simulieren, kommt ein modifiziertes City-Section-Bewegungsmodell (siehe Kapitel 2, Abschnitt 3.11) zum Einsatz.

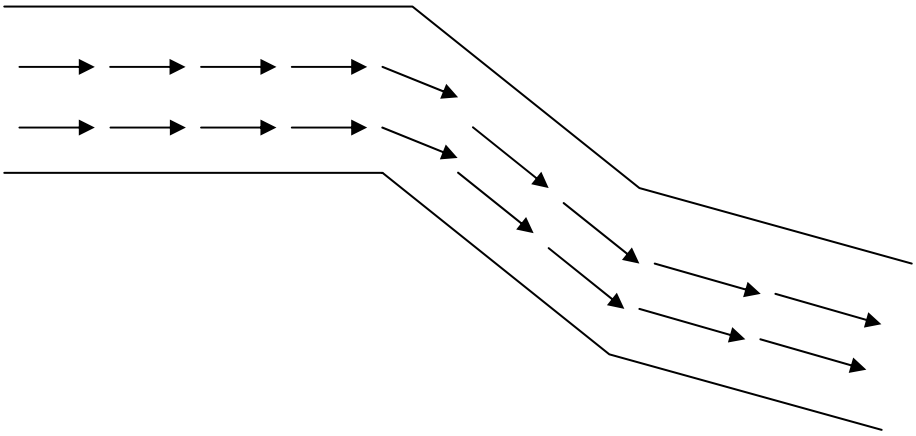

Abbildung 19: Geschwindigkeitsvektoren zur Simulation der Fahrbahn

Bewegung ist in diesem Modell nur in gewissen Regionen möglich („Fahrbahn“). Dort werden Geschwindigkeitsvektoren angegeben, deren Richtung die Bewegungsrichtung darstellt und deren Betrag die „zulässige Höchstgeschwindigkeit“ simuliert (siehe Abbildung 19). Es werden drei unterschiedliche Fahrzeugtypen modelliert: LKWs (Geschwindigkeiten von 40-80 km/h), langsame PKWs (60-120 km/h) und schnelle PKWs (100-220 km/h). Simuliert werden zwei Autobahnstücke von jeweils 2 km Länge mit einem Autobahnkreuz. Die Geschwindigkeit und Richtung eines Teilnehmers wird anhand des Vektors an der aktuellen Position, der aktuellen Geschwindigkeit und den Charakteristika des Teilnehmers bestimmt. Am Autobahnkreuz entscheidet ein Teilnehmer zufällig, ob er abbiegt oder nicht (10% Abbiegewahrscheinlichkeit)

3.4.2 Ein Bewegungsmodell für das Konferenz-Szenario

Das Szenario „Konferenz“ wird durch ein abgewandeltes Gravity-Bewegungsmodell simuliert. Es gibt insgesamt n Teilnehmer. Jeder Teilnehmer hat ein gewisses Interessengebiet, das als Untermenge der möglichen Interessengebiete dargestellt wird:

$$Interesse_i = \{Inter \mid Inter \in Interessengebiet \wedge \ Teilnehmer\ i\ interessiert\ sich\ für\ Inter \ \}$$

Es gibt k Veranstaltungsorte, die durch ihre Koordinaten und durch ihren Radius im Simulationsgebiet beschrieben werden:

$$VOrt_j = (x_j, y_j)$$

$$VRad_j$$

Es gibt l Veranstaltungen. Jede Veranstaltung findet an einem Veranstaltungsort statt. Jede Veranstaltung deckt ein gewisses Interessengebiete ab:

$$VInter_a = \{Inter \mid Inter \in Interessengebiet \wedge die\ Veranstaltung\ deckt\ dieses\ Interesse\ ab \ \}$$

Jede Veranstaltung hat eine Dauer $VDauer_a$, einen Anziehungsfaktor An_a und einen Mobilitätsfaktor α, der angibt, wie mobil die Teilnehmer im Rahmen einer Veranstaltung sind. Bei einer Vorlesung im klassischen Sinn bewegen sich die Teilnehmer nicht, d.h. $\alpha = 0$. Bei einem Stehempfang haben die Teilnehmer eine mittlere Mobilität, während bei einer Ausstellung die Mobilität hoch ist.

Zu Beginn der Simulation werden die Teilnehmer gleichverteilt zufällig auf dem Simulationsgebiet platziert. Dann werden die Veranstaltungsorte und Radien zufällig ermittelt. Danach wird in jeder Zeiteinheit folgendermaßen vorgegangen:

1.) Lösche alle Veranstaltungen, deren Dauer abgelaufen ist.
2.) Bestimme für alle nicht benutzten Veranstaltungsorte zufällig, ob dort eine neue Veranstaltung stattfinden soll.

3.) Für jede neue Veranstaltung lege Dauer und Anziehungsfaktor sowie Mobilitätsfaktor zufällig fest.
4.) Für alle Teilnehmer, die an keiner Veranstaltung teilnehmen und sich auch nicht auf dem Weg zu einer Veranstaltung befinden, stelle fest, ob Veranstaltungen existieren, die einem Interesse des Teilnehmers genügen. Existieren mehrere Veranstaltungen, so wähle anhand des Anziehungsfaktors, der Dauer der Veranstaltung (längere Dauer vorziehen) und der Entfernung eine aus. Der Teilnehmer fühlt sich nun von dieser Veranstaltung angezogen. Wähle zufällig und gleichverteilt einen Punkt innerhalb des Radius der Veranstaltung und bewege den Teilnehmer dort hin mit einer zufälligen, konstanten Geschwindigkeit und einem Bewegungsvektor, der direkt auf den gewählten Punkt am Veranstaltungsort zielt.
5.) Für alle Teilnehmer in einer Veranstaltung lege anhand der Interessengebiete sowie dem Anziehungsfaktor der laufenden Veranstaltung und der anderen Veranstaltungen fest, ob der Teilnehmer die Veranstaltung verlässt. Falls ja, berechne wie unter 4) die Bewegung.
6.) Für alle Teilnehmer in einer Veranstaltung lege anhand des Mobilitätsfaktors der Veranstaltung die Bewegung des Teilnehmers innerhalb der Veranstaltung fest (Random Walk oder ähnliches Bewegungsmodell).

Einrichtungen eines Kongresszentrums wie z.B. eine Cafeteria können simuliert werden durch einen niederen Anziehungsfaktor, mittleren Mobilitätsfaktor und *VInter=Interessengebiet* sowie unendliche Dauer.

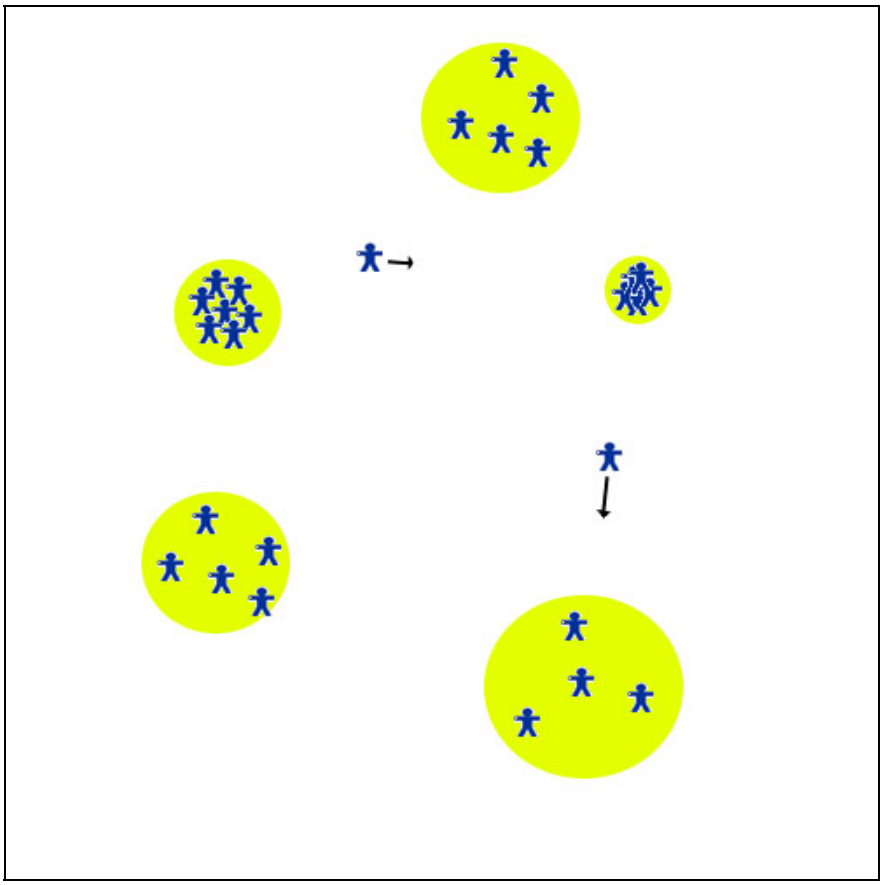

Abbildung 20: Bewegungsmodell Konferenz

Abbildung 20 zeigt beispielhaft eine Konferenz mit 5 Veranstaltungen und 5 Veranstaltungsorten. Im Radius eines Veranstaltungsorts ist die Konzentration von Teilnehmern im Allgemeinen höher als außerhalb der Veranstaltungsorte. Dort kommt es seltener zum Abbruch einer Verbindung, da sich die Teilnehmer relativ wenig bewegen. Haben die Veranstaltungen alle ein gemeinsames Zeitfenster (wie z.B. bei Vorlesungen an einer Universität), so kommt es nach einer Phase der Ruhe nach dem Ende der Veranstaltungen zu großen Änderungen in den Standorten der Clients. Solange sich die Teilnehmer eine neue Veranstaltung suchen, ist die Dynamik der Netztopologie sehr hoch. Am Ende einer Veranstaltung bricht der dortige Cluster in der Regel zusammen.

3.4.3 Random Waypoint

Um eine Vergleichbarkeit mit anderen Arbeiten herzustellen wurde neben den Bewegungsmodellen für die oben beschriebenen Szenarien auch noch das weitverbreitete Random-Waypoint Modell (siehe Abschnitt 3.1.9) implementiert

4. Evaluation

Ein wesentliches Ziel dieser Arbeit ist die Bewertung der Leistungsfähigkeit der betrachteten Sicherheitsarchitektur. Hierzu ist es nötig, die Sicherheitsarchitektur im Rahmen einer Simulation zu implementieren. Die Implementierung wird dann für Messungen herangezogen. In der Simulation kommen Bewegungsmodelle für zwei konkrete Szenarien zum Einsatz: Konferenz und Autobahn. Außerdem wird das Bewegungsmodell Random-Waypoint verwendet, um die Vergleichbarkeit mit anderen Arbeiten herzustellen.

4.1 Grundlagen und Werkzeuge

Im Folgenden werden die grundlegenden Werkzeuge vorgestellt, mit Hilfe derer die Implementierung erstellt wurde.

4.1.1 Omnet++

Für die im Rahmen dieser Diplomarbeit erstellte Simulation wurde Omnet++ in der Version 2.2 verwendet [Var01]. Bei Omnet++[4] handelt es ich um eine objektorientierte Simulationsumgebung in C++. Omnet++ stellt ein Gerüst zur Simulation von Netzen zur Verfügung. Dieses Gerüst schließt verschiedene, zum Teil sehr komplexe, Benutzeroberflächen ebenso mit ein, wie einen Mechanismus zur Aufzeichnung von Ereignissen und eine Protokollfunktoin. Die Struktur des simulierten Netzes wird mit Hilfe von Modulen nachgebildet. Omnet++ unterscheidet zwischen einfachen und komplexen Modulen. Letztere bestehen wieder aus mehreren einfachen oder komplexen Modulen. Dadurch entsteht eine Modulhierarchie. Die Simulation erfolgt in Omnet++ diskret und ereignisgesteuert.

Ein Omnet++-Modell besteht aus hierarchisch verschachtelten Modulen, die über Verbindungen zwischen ihnen mittels Nachrichten kommunizieren. Verbindungen können mit Übertragungseigenschaften, wie z.B. Datenrate, Laufzeit oder Fehlerrate, versehen werden. Nachrichten können beliebige Datenstrukturen transportieren. Jede Nachricht verfügt über einen Typ. Damit können Nachrichten einfach verschiedenen Modulen zugeordnet werden. Benutzer können eigene Typen erstellen. Die Module können parametrisiert werden. Module können in

[4] Objective Modular Network Testbed in C++

einer eigenen Sprache, NED[5], zu Simulationsbeginn oder durch C++-Funktionen während der Laufzeit erzeugt werden. Die Module der unteren Ebenen werden von den Benutzern in C++ realisiert. Sie enthalten die eigentlichen Algorithmen der Simulation. Da die Module als Co-Routinen implementiert sind, scheinen sie bei der Ausführung parallel zu laufen.

Die Omnet++ Bibliothek stellt eine Vielzahl von Funktionen zur Verfügung, mit denen Zufallszahlen mit Hilfe verschiedener Wahrscheinlichkeitsverteilungen gewonnen werden können.

Omnet++ verfügt über verschiedene Benutzerschnittstellen, z.B. zur Fehlersuche, zur Demonstration und zur Ausführung im Rahmen eines Skripts. Während der Simulation können die komplexen Module und die Verbindungen zwischen diesen über eine grafische Benutzeroberfläche inspiziert werden. Die Benutzeroberfläche wird im Anhang erläutert.

Omnet++ verfügt neben der eigentlichen Simulationsumgebung über eine Anzahl von Werkzeugen zur Auswertung der Ergebnisse. Ergebnisvektoren und Ergebnisskalare dienen in Omnet++ zur Speicherung von Messergebnissen. Das mitgelieferte Werkzeug Plove kann aus den Vektoren einfache Grafiken erstellen. Im Rahmen dieser Diplomarbeit wurden allerdings Perl-Skripte verwendet, um den Ergebnisvektor einer Simulation in die einzelnen Teilvektoren der gemessenen Größen zu zerlegen. Anschließend wurden die Teilvektoren in das Format des Statistik-Werkszeugs Origin Pro 7.0 konvertiert und mit Hilfe dieses professionellen Werkzeugs weiterverarbeitet.

4.1.2 Ad-hoc-Simulator

Die hier vorgestellte Simulation basiert auf dem Ad-hoc-Simulator der Arbeit [BAN01]. Der Simulator wurde als Framework in Omnet++ realisiert und bietet vielfältige Erweiterungsmöglichkeiten um eigene Module einbinden zu können. Dabei hat das Framework folgenden Aufbau. Die Knoten und die (dynamischen) Verbindung zwischen diesen werden zentral von einem Air-Modul verwaltet. Alle Knoten haben eine Verbindung von und zu diesem Modul. Das Air- Modul simuliert die Luftschnittstelle einschließlich abbrechender und neu entstehender Verbindungen, Übertragungsfehler etc. Abbildung 21 zeigt die Anbindung der einzel-

[5] NEtwork Description

nen Knoten an das Air-Modul. Dabei werden alle Knoten des Modells bei der Initialisierung erstellt. Ein verzögerter Systemstart ist allerdings möglich, um später hinzukommende Knoten nachzubilden. Separate Konfigurationsdateien enthalten Parameter für die einzelnen mobilen Knoten. Jeder Knoten kann damit über eigene Charakteristiken verfügen. Da auch die jeweilige Übertragungsreichweite festgelegt wird, können mit dem Ad-hoc-Simulator auch asymmetrische Verbindungen simuliert werden. Ein eigenes Programm wurde geschrieben, um Konfigurationsdateien für die einzelnen Knoten zu erzeugen. Damit eignet sich der Ad-hoc-Simulator auch für Testreihen im Batch-Betrieb.

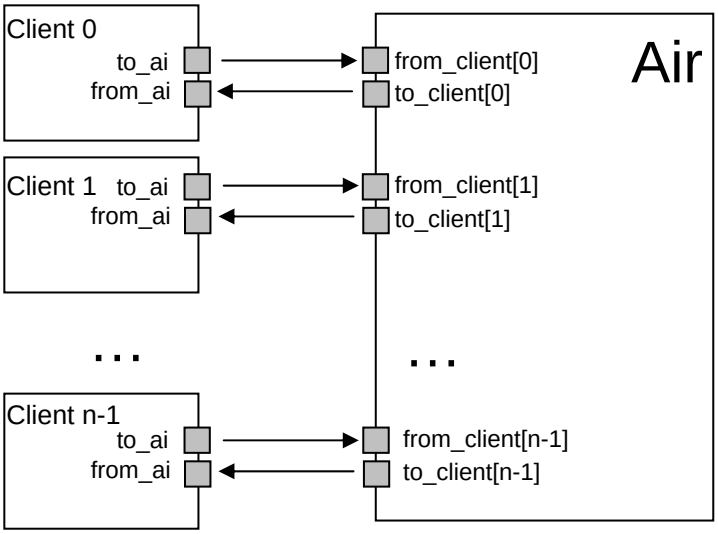

Abbildung 21: Anbindung der Knoten an das Air-Modul

Die einzelnen Knoten sind als komplexe Module in Omnet++ realisiert. Abbildung 22 zeigt den Aufbau eines Client-Moduls. Der Aufbau folgt keinem klaren Schichtenaufbau wie z.B. dem ISO/OSI-Basisreferenzmodell.

Das Transmission-Modul trägt im Knoten die Verantwortung für die Kommunikation mit dem Air-Modul. Jede Nachricht an den Knoten oder vom Knoten durchläuft dieses Modul. Hier wird eine empfangene Nachricht an das entsprechende lokale Modul weitergeleitet. Das Transmissions-Modul bildet also die funktechnische Anbindung in einem realen Gerät nach.

Das Status-Modul verwaltet die Eigenschaften des Knoten und bekommt vom Air-Modul die (beschränkte) Netzsicht des Knotens mitgeteilt. Diese beinhaltet z.B. die entdeckten Nachbarn.

Im Routing-Modul können verschiedene Routing-Algorithmen implementiert werden. Für die vorliegende Arbeit wird auf eine Implementierung des Fisheye-State-Algorithmus [DIN02] zurückgegriffen (siehe Kapitel 4, Abschnitt 1.3). Weitere Implementierungen können problemlos eingebunden werden. Der Ad-hoc-Simulator ist so ausgelegt, dass mehrere verschiedene Routing-Algorithmen in einer einzigen Simulation verwendet werden können.

Das FindRoute-Modul stellt den Zugang zum Routing-Protokoll für das Application-Modul dar. Hier erfolgt auch die Verwaltung des Routings.

Im Application-Modul können eigene Algorithmen implementiert werden. In der Standardimplementierung ist keine Funktionalität ausgeführt.

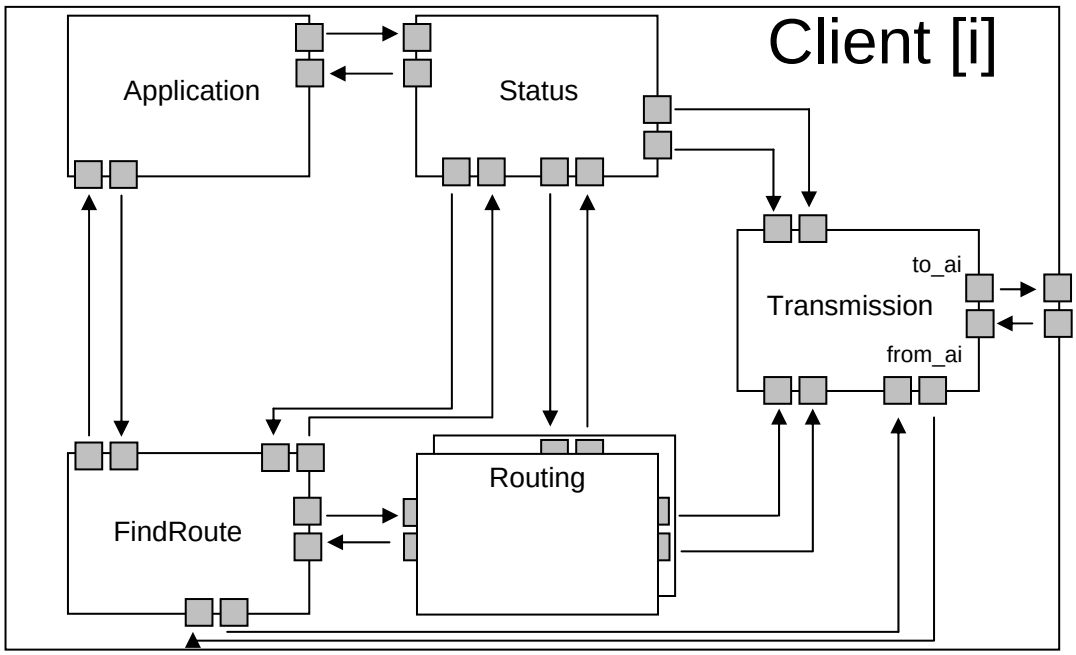

Abbildung 22:Untermodule eines Client-Moduls

4.1.3 Routing

Die im Rahmen dieser Arbeit erstellte Simulation verwendet das Routing-Verfahren *Fisheye-State-Routing.*

Fisheye-State-Routing wird in [GER02] beschrieben. Es handelt sich um ein proaktives Routingverfahren basierend auf einem Link-State-Ansatz. Fisheye-State-Routing nutzt die Fischaugentechnik, um die Anzahl der Routingeinträge innerhalb eines Pakets gering zu halten. Die Fischaugentechnik leitet sich von der Art und Weise ab, wie Fische ihre Umgebung wahrnehmen. Fischaugen sind so konzipiert, dass sie Regionen nahe dem Brennpunkt des Auges besser wahrnehmen. Um diesen Punkt liegende Bereiche werden stärker aufgelöst als weiter entfernte. Übertragen auf den Routing-Algorithmus bedeutet dies, dass über nahe liegende Knoten aktuellere Informationen vorhanden sind als über weiter entfernt liegende Knoten. Abbildung 23 zeigt verschiedene Fischaugenbereiche. Routingnachrichten beim Fisheye-State-Routing enthalten nur Informationen über Knoten, die innerhalb einer bestimmten Distanz zum Sender liegen. Die Nachrichten sind deshalb kurz. Sie werden häufiger an direkte Nachbarn versendet als an weiter entfernte Knoten. Die Nachrichten werden nicht geflutet sondern direkt verschickt. Eine Nachricht, die nur Einträge für direkte Nachbarn enthalten soll, würde z.B. in Abbildung 23 einen Eintrag für jeden Knoten im schwarzen Bereich haben. Die Routingtabelle in jedem Knoten basiert auf den aktuellsten Topologieinformationen. Wegen des Verbreitungsverfahrens sind die Informationen über nahe gelegene Knoten aktueller. Zudem ist es möglich, Informationen ohne zu große Netzbelastung auch in kürzeren Intervallen zu versenden. Über weiter entfernte Knoten können veraltete Informationen vorliegen. Dies stellt jedoch kein Problem dar, da gesendete Datenpakete auf dem Weg zum Ziel immer mehr in Bereiche kommen, in denen die Routinginformationen aktuell sind. Anders als bei Link-State-Routingalgorithmen üblich, wird auf Verbindungsfehler nicht mit Kontrollnachrichten reagiert, da wegfallende Knoten nach der nächsten Aktualisierung ohnehin nicht mehr aufgeführt werden. Sequenznummern und Tabellenauffrischung sorgen im Fisheye-State-Routing-Verfahren dafür, dass immer die aktuellsten Informationen verarbeitet werden und keine Schleifen entstehen.

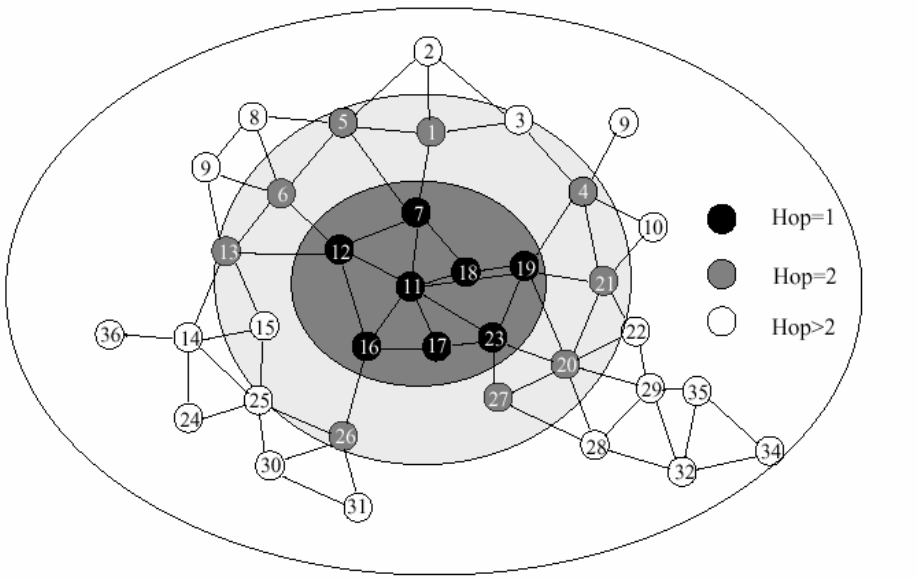

Abbildung 23: Fischaugenbereiche im Fisheye-State-Routing

4.1.4 Grenzen der Simulation

Mit dem Ad-hoc-Simulator kann im Moment leider keine Kollision von Paketen bei der Übertragung simuliert werden. Der eingesetzte Ad-hoc-Simulator bietet lediglich die Möglichkeit eine Fehlerrate bei der Übertragung über die Luft zu definieren. Diese ist aber nicht lastabhängig und damit nicht zur Kollisionssimulation geeignet. Auch bietet der Ad-hoc-Simulator leider keine entfernungsabhängige Verzögerung der Signale.

Für Simulationen mit vielen Knoten eignet sich der Ad-hoc-Simulator nicht, denn er verwendet für viele interne Meldungen das in Omnet++ integrierte Nachrichtensystem. Dieses erwies sich als langsam und sehr speicheraufwändig. So braucht in der vorliegenden Implementierung ein Simulationslauf auf einem Pentium 1,2 GHz unter Windows XP mit 60 Knoten ca. 17 Minuten für 240 Simulationszeit-Sekunden. Es werden ca. 70 MB Hauptspeicher belegt. Daran wird ersichtlich, dass Simulationen mit 1000 Knoten nicht denkbar sind.

4.2 Implementierung

Im Rahmen dieser Arbeit wurde die Sicherheitsarchitektur in Omnet++ mit Hilfe des oben beschriebenen Ad-hoc-Simulators implementiert.

4.2.1 Implementierung der Sicherheitsarchitektur

Am Ad-hoc-Simulator [BAN01] wurden umfangreiche Änderungen vorgenommen, um breitere Anwendungsmöglichkeiten der Implementierung zu schaffen und die Sicherheitsarchitektur angemessen implementieren zu können. So wurden z.B. neue Möglichkeiten zur Einbindung von Bewegungsmodellen geschaffen. Die Verantwortlichkeit für Bewegung liegt nun bei einer eigenen Klasse, MobilityManagement, die verschiedene Mobilitätsprofile verwaltet. Im Rahmen der Simulation wird die Mobilität nicht wie in der Arbeit [BAN01] in den Knoten realisiert, sondern zentral im Air-Modul. Da viele Bewegungsmodelle eine zentrale Sicht über die Position der Knoten benötigen, um zum Beispiel auf die Position anderer Knoten zu reagieren, können diese Modelle nur im Air-Modul realisiert werden, das als einziges Modul auch wirklich über eine netzweite Sicht verfügt. Der alte Mechanismus bleibt aus Gründen der Abwärtskompatibilität weiter bestehen. Beide Implementierungen behindern sich gegenseitig nicht, grundsätzlich ist aber von einer gleichzeitigen Benutzung abzuraten.

Weitere Anpassungen wurden im Bereich der Dynamik vorgenommen. Probleme in Bezug auf Bewegung der Knoten und Rückmeldung der neuen Positionen an die Status-Module der einzelnen Client-Module wurden behoben.

Ein zentrales Problem bei der Implementierung der Sicherheitsarchitektur und beim Ad-hoc-Simulator stellte die Speicherverwaltung der versendeten Nachrichten dar. Um die entstandenen Probleme zufrieden stellend zu lösen wurde eine eigene Klasse, MemoryManagement, erstellt, der die Nachrichtenvernichtung, Entfernung aus dem Zeitplan und Freigabe des Speicherplatz obliegt. Die Implementierung der Sicherheitsarchitektur verwendet diese Klasse durchgehend. Im Ad-hoc-Simulator wird sie eingesetzt, wo immer Probleme entstanden. Durch diesen Mechanismus konnte der Speicherverbrauch der Simulation drastisch gesenkt werden. Die Implementierung der Sicherheitsarchitektur wurde als eigenes Modul realisiert

und ist in den Ad-hoc-Simulator eingebettet. Abbildung 24 zeigt die aktuelle Struktur des Ad-hoc-Simulators. Das Sec-Modul hat eine Verbindung zum Application-Modul. Von dort kommen die Nutzdaten, die im Sec-Modul zur sicheren Übertragung eingekapselt werden. Die Verbindung des Sec-Moduls zum FindRoute-Modul dient zum Versenden von Nachrichten. Das FindRoute-Modul übergibt die Nachrichten zur Beförderung an ein Routing-Modul. Schließlich verfügt das Sec-Modul noch über eine Verbindung zum Status-Modul. Dadurch kann das Sec-Modul auf Daten über den Status des Knotens und seiner Umgebung zurückgreifen.

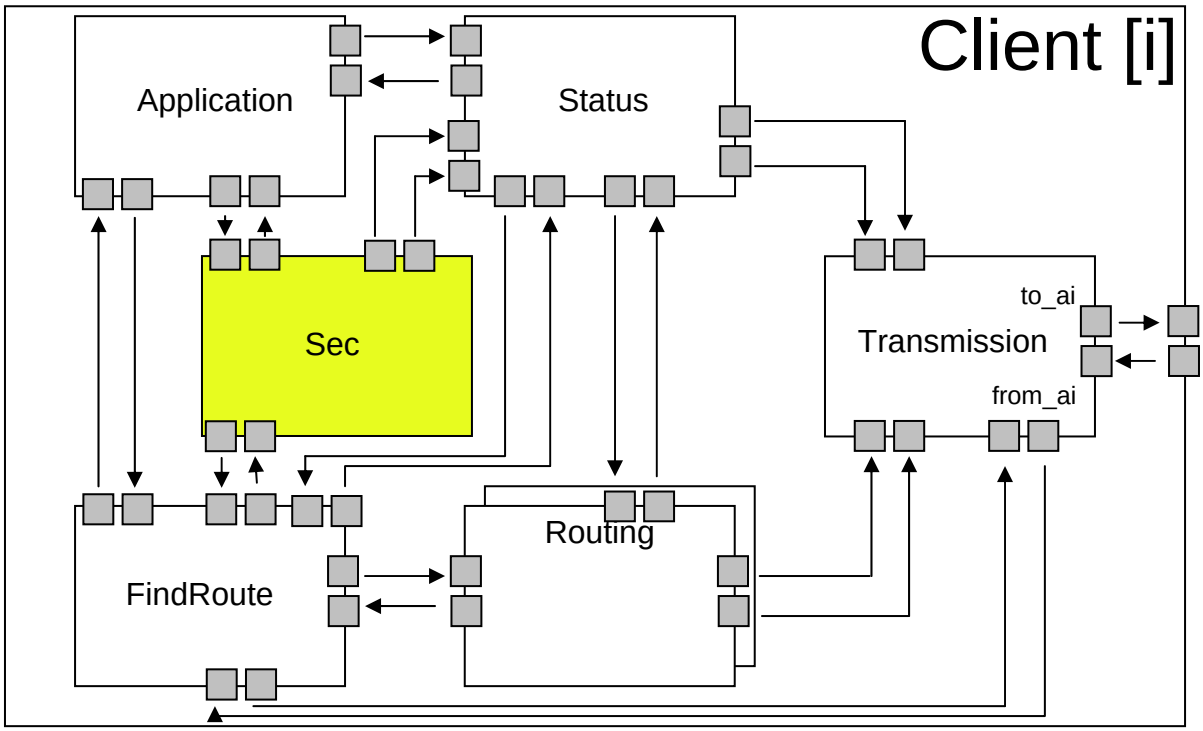

Abbildung 24: Einbettung der Sicherheitsarchitektur

Die folgende Abbildung 25 zeigt beispielhaft den Weg, den eine Nachricht des Security-Moduls durch den Ad-hoc-Simulator nimmt. Das Security-Modul leitet die Nachricht an das FindRoute-Modul weiter. Dieses veranlasst im gewählten Routing-Modul (z.B. Fisheye-State-Routing) eine Weg-Anfrage. Erhält das Routing-Modul eine Weg-Antwort, so wird diese an das FindRoute-Modul weitergeleitet, das anschließend die ursprüngliche Nachricht an das Transmission-Modul weitergibt. Dieses leitet die Nachricht schließlich dem Air-Modul zu.

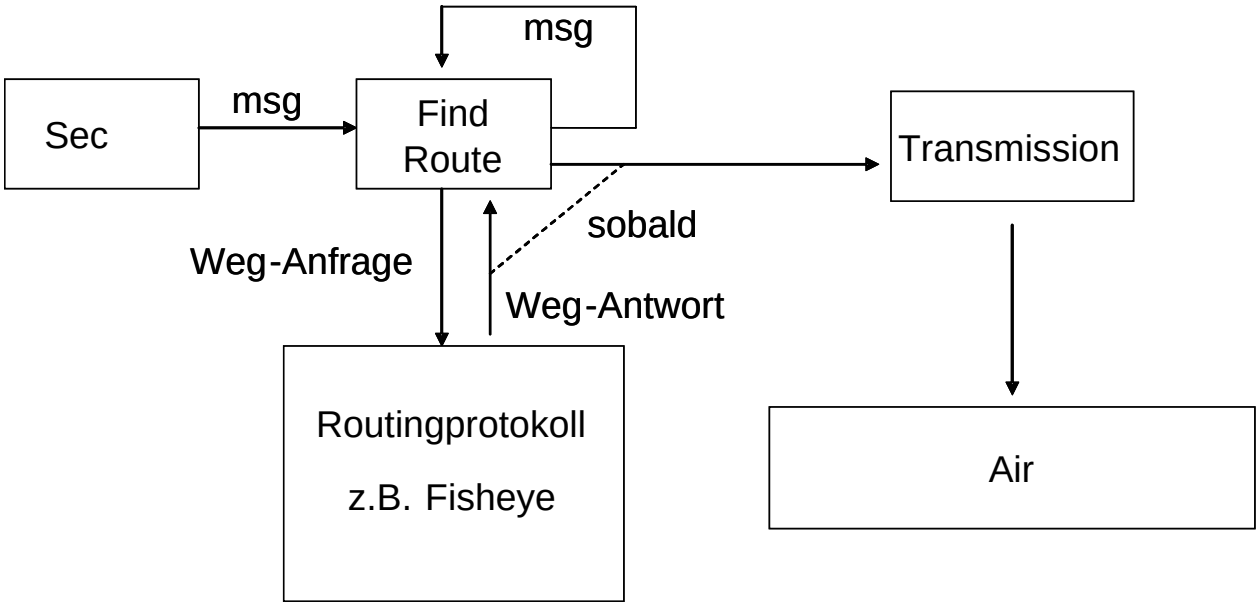

Abbildung 25: Weg einer SEC-Nachricht durch den Ad-hoc-Simulator

4.2.2 Protokollablauf

Die Funktionalität der Sicherheitsarchitektur wird über den Austausch von Nachrichten realisiert. Im Anhang A werden alle Nachrichten der Implementierung samt ihrer Funktion aufgeführt. Der Protokollautomat findet sich noch einmal in Abbildung 26, Abbildung 27 und Abbildung 28.

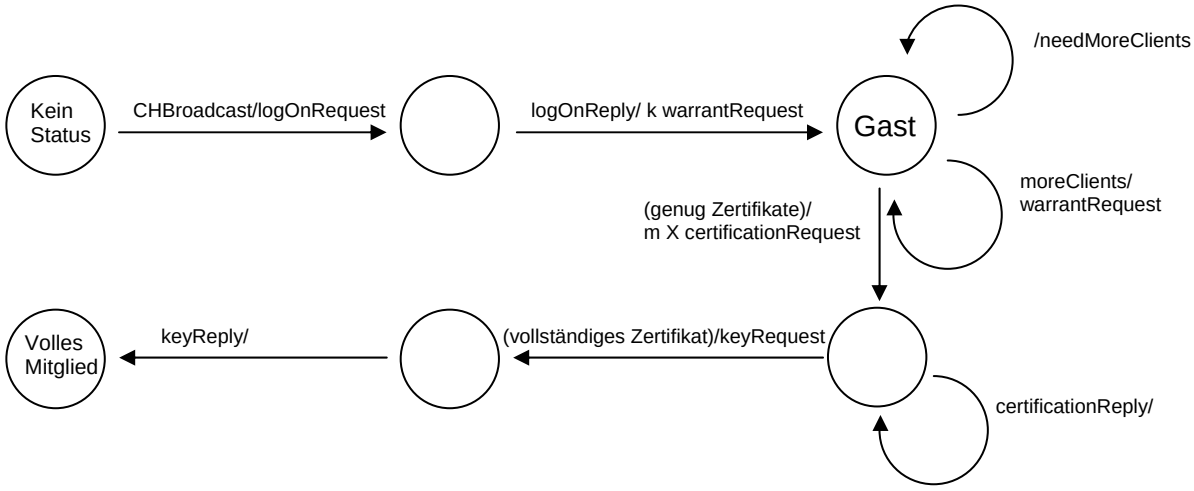

Abbildung 26: Protokollautomat Anmeldung

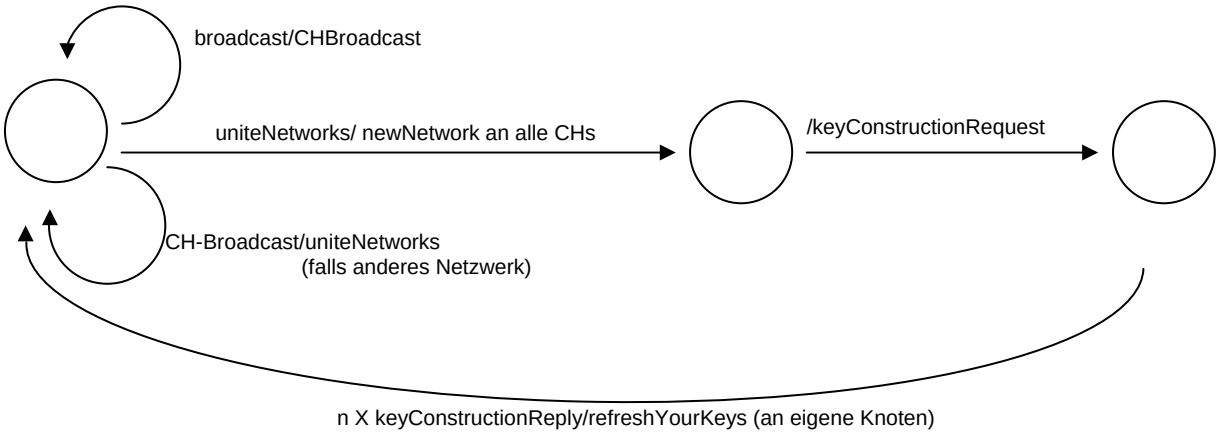

Abbildung 27: Protokollautomat für Clusterhead

Zu Beginn startet ein Knoten den Zeitgeber „noCHfound“. Nun wartet er auf die Nachricht „CHBroadcast“ von einem Clusterhead. Erhält der Knoten „CHBroadcast“, so stoppt er den Zeitgeber „noCHfound“ und entscheidet je nach gewählter Clustergröße, ob er „CHBroadcast“ an alle Nachbarn flutet oder nicht. Die Nachricht „CHBroadcast“ ist mit einer eindeutigen Sequenznummer versehen, anhand derer Duplikate und veraltete Nachrichten erkannt werden. Nun startet der Knoten die Zeitgeber „lostCH“ und „logOnTimeout“. Anschließend sendet er die Nachricht „logOnRequest“ an den Clusterhead, von welchem die Nachricht „CHBeacon“ stammte, die zuvor empfangen wurde. Damit beginnt der Anmeldevorgang des Knotens.

Abbildung 28: Protokollautomat

Der Knoten wartet dabei auf die Nachricht „logOnReply“. Empfängt er diese, so überprüft er die mitgesendeten Sicherheitsrichtlinien des Clusters. Kann der Knoten die Sicherheitsrichtlinien einhalten, so liest er aus der Nachricht „logOnReply“ eine Liste mit potentiellen Bürgen aus. Diese potentiellen Bürgen sind alle Knoten im Cluster, die bürgen dürfen. Diese Bürgen bittet der Knoten mit der Nachricht „warrantRequest“ um ein Bürgenzertifikat. Enthält die Liste nicht genug Bürgen, so ermittelt der Knoten entweder anhand der Nachricht „CHBroadcast“ oder durch den Empfang einer Nachricht „GWBeacon“ eine Liste von Gateways und bittet über diese Gateways die Clusterheads der benachbarten Cluster mit der Nachricht „needMoreClients“ um eine Liste mit zusätzlichen potentiellen Bürgen. Die Clusterheads antworten auf „needMoreClients“ mit der Nachricht „moreClients“.

Empfängt ein Knoten die Nachricht „warrantReply“, so speichert er das darin enthaltene Bürgenzertifikat und das Bürgen-Autorisierungszertifikat und testet, ob er bereits genügend Zertifikate gesammelt hat. Hat der Knoten genügend Bürgenzertifikate und zugehörige Bürgen-Autorisierungszertifikate gesammelt sendet er die Nachricht „certificationRequest“ an eine Anzahl von bekannten Clusterheads. Diese werden wieder wie bei „needMoreClients“ beschrieben ermittelt. Jeder Nachricht „certificationRequest“ werden alle Bürgenzertifikate und Bürgen-Autorisierungszertifikate mitgegeben.

Die Clusterheads antworten schließlich mit der Nachricht „certificationReply“. Enthalten ist ein Teil des identitätsbezogenen Schlüsselzertifikats. Der Knoten speichert das Teilzertifikat zwischen und wartet, bis er mindestens so viele gesammelt hat, wie der Schwellwert angibt. Der Schwellwert wurde zuvor über „CHBroadcast“ bekannt gemacht. Dann setzt er die Teilzertifikate zum vollständigen Schlüsselzertifikat zusammen und sendet das Schlüsselzertifikat mit der Nachricht „keyRequest“ an seinen Clusterhead. Als Antwort erhält der Knoten den symmetrischen Clusterschlüssel in der Antwort „keyReply“. Der Knoten stoppt jetzt den Zeitgeber „logOnTimeout“. Damit ist der Anmeldevorgang für den Knoten abgeschlossen. Der Knoten ermittelt die Gültigkeit seines Schlüsselzertifikats und startet entsprechend den Zeitgeber „IDCertsExpired“, um sich nach Ablauf des Zeitgebers ein neues Zertifikat zu beschaffen. Abbildung 26 zeigt den gesamten Anmeldevorgang noch einmal im Überblick. Die Zeitgebernachrichten wurden weggelassen.

Erhält ein Knoten die Nachricht „CHBeacon“, so überprüft er zuerst, ob es sich um eine Nachricht seines Clusterheads handelt. Ist dies der Fall, so löscht der Knoten den Zeitgeber „lostCH“ und startet ihn neu. Anschließend wird der Inhalt der Nachricht gesichert. Ist ein Knoten noch kein Clustermitglied, dann startet er einen Anmeldeversuch wie oben beschrieben. Ist ein Knoten bereits in einem Cluster volles Mitglied und empfängt „CHBeacon“ von einem anderen Clusterhead, dann wird der Knoten zum Gateway. Er startet einen Zeitgeber „broadcastGW“ und versucht, sich wie oben beschrieben am neuen Clusterhead anzumelden. Erhält ein Clusterhead die Nachricht „CHBroadcast“ von einem anderen Clusterhead, dann überprüft er, ob der Clusterhead dem eigenen CH-Netzwerk angehört oder nicht. Handelt es sich um ein fremdes Netz, dann sendet er dem Clusterhead die Nachricht „uniteNetworks“ einschließlich Informationen über das eigene CH-Netzwerk und fordert ihn damit auf, eine Entscheidung über die Vereinigung der beiden CH-Netzwerke zu treffen. Empfängt ein Clusterhead „uniteNetworks“, dann prüft er (wie in Kapitel 3, Abschnitt 3.4 beschrieben), ob die Netze vereinigt werden sollen. Soll eine Vereinigung stattfinden, dann sendet der Clusterhead an alle anderen Clusterheads die Nachricht „newNetwork“.

Empfängt ein Clusterhead die Nachricht „newNetwork“, so tritt er dem neuen CH-Netzwerk bei und beteiligt sich mit „keyConstructionRequest“ an der Konstruktion des privaten geheimen Netzschlüssels. Als Antwort erhält er von den anderen CHs „keyConstructionReply“-Nachrichten. Schließlich fordert der Clusterhead mit der Nachricht „refreshYourKey“ alle bei ihm angemeldeten Clustermitglieder auf, ihren Schlüssel neu signieren zu lassen. Abbildung 27 zeigt den Protokollautomaten für einen Clusterhead. Auf diverse Zeitgebernachrichten wurde der Übersichtlichkeit wegen verzichtet. Beschließt ein Clusterhead (z.B. aufgrund der Entscheidung bei der Clustervereinigung) seine CH-Tätigkeit aufzugeben, so signalisiert er dies allen Knoten in seinem Cluster mit der Nachricht „noLongerCH“. Die Knoten versuchen daraufhin, sich bei anderen Clusterheads anzumelden.

Erhält ein Knoten eine Nachricht „GWBroadcast“ von einem Gateway, dann startet er für dieses Gateway den Zeitgeber „lostGW“. Hat der Knoten das Gateway schon zuvor empfangen, so wird der alte Zeitgeber „lostGW“ zuerst gestoppt. Läuft bei einem Gateway der Zeitgeber „broadcastGW“ ab, so flutet es die Nachricht „GWBroadcast“ und startet den Zeitgeber neu.

Wird ein Knoten Clusterhead infolge eines Timeouts „noCHfound“ oder wegen der Ablehnung durch einen Clusterhead (durch die Nachricht „logOnReply“), so startet der Knoten den Zeitgeber „broadcast“. Immer wenn dieser Zeitgeber abgelaufen ist, sendet der Clusterhead einen „CHBroadcast“ und startet den Zeitgeber neu. Läuft der Zeitgeber „lostCH“ beim Knoten ab, so hat er seinen zugeordneten Clusterhead verloren. Er sucht sich einen neuen Clusterhead bzw. wird selbst zum CH. Die Nachricht „EncapsulatedSec“ dient zur Kapslung von Inhalten höherer Schichten.

4.2.3 Kommunikationstechnologien

Für die Simulation der Szenarien kommen wie oben beschrieben IEEE 802.11x (Anhang A) oder Bluetooth (siehe Anhang A) zum Einsatz. Die Luftschnittstelle wird durch das Air-Modul simuliert. Dabei werden die Verbindungen vom Air-Modul zu den Clients verwendet um die Charakteristika der eingesetzten Kommunikationstechnologien zu simulieren. Daneben verfügt jeder Client über eine maximale Kommunikationsreichweite, die ebenfalls als Parameter im Ad-hoc-Simulator vorgegeben wird. Die Datenraten der einzelnen Standards können zu Vergleichen mit den Datenraten in der Simulation herangezogen werden. Auf Grundlage des Anhang A gegebenen Zahlenmaterials aus [CIS02] , [COM02] und [SIE02] werden in der Simulation die in Tabelle 1 angegebenen Werte verwendet.

Charakteristik	802.11b	Bluetooth
Max. Reichweite	180m/40m [6]	10 m
Erreichbare Datenrate	5 MBit/s	200 kBit/s
Laufzeitverzögerung[7]	100 ms	20 ms

Tabelle 1: Physikalische Parameter der Simulation

In Kapitel 4 Absatz 1.4 wurden bereits die Grenzen der Simulation angesprochen. Der Ad-hoc-Simulator verfügt nicht über die Möglichkeit, Kollisionen von Paketen zu simulieren.

[6] Im Freien / in Gebäuden
[7] Da im Ad-hoc-Simulator die Laufzeit nicht anhand der Entfernung der Knoten berechnet wird kann hier nur ein Mittelwert verwendet werden, der auf der maximalen Reichweite beruht.

Deshalb ist in der Simulation auch nicht das typische Verhalten von 802.11 oder Bluetooth zu erwarten.

5. Messergebnisse

Im Folgenden wird die Implementierung zu Messungen herangezogen. Soweit nichts anderes erwähnt wird kommt der Routing-Algorithmus Fisheye State zum Einsatz. Beim Einsatz der Bewegungsmodelle Random-Waypoint und Konferenz ist das Simulationsgebiet 600m auf 600m groß. Beim Autobahnbewegungsmodell ist es 2 km auf 2 km. Gemessen wurden die Log-On-Zeit, Overhead, Verfügbarkeit und die Belastung von Clusterheads, Gateways und vollen Mitgliedern. Eine genaue Definition der gemessenen Größen findet sich im entsprechenden Abschnitt.

5.1 Log-On-Zeit

Mit *Log-On-Zeit* wird die Zeitspanne zwischen Empfang des ersten CH-Beacons und der Zuteilung des symmetrischen Cluster-Schlüssels durch den CH bezeichnet. In diesem Intervall sucht der Knoten Zeugen für seine Identität, sammelt von den einzelnen CHs im CH-Netzwerk Teile des identitätsbezogenen Schlüsselzertifikats (IdZert), setzt dieses zusammen und fordert damit den symmetrischen Clusterschlüssel vom eigenen Clusterhead an. Durch die Zuteilung des symmetrischen Cluster-Schlüssel wird ein Knoten volles Mitglied im Cluster.

Die Log-On-Zeit soll idealerweise klein sein, um den Knoten schnellen Zugang zur Sicherheitsarchitektur zu garantieren. Die Dynamik in Ad-hoc-Netzwerken erschwert die Anmeldung in kurzer Zeit, denn besonders bei schneller Bewegung besteht die Gefahr, dass die Verbindung zwischen einem Knoten seinem Clusterhead während der Anmeldung abbricht. Die Anmeldung scheitert dann garantiert. Ebenso kann es einem Knoten passieren, dass er sich in ein Gebiet bewegt, in dem er nicht genug Bürgen erreichen kann. Dadurch wird seine Anmeldung verzögert. Es ist nicht zu erwarten, dass sich alle Knoten innerhalb kürzester Zeit anmelden können. Jedoch ist es wünschenswert, dass zumindest möglichst viele Knoten in kurzer Zeit zu vollen Clustermitgliedern werden. Aus diesem Grund werden im Weiteren Wahr-

scheinlichkeitsverteilungen über Log-On-Zeiten betrachtet. In den folgenden Abbildungen wurden die Wahrscheinlichkeiten für Log-On-Zeiten in 5-Sekunden-Intervallen zusammengefasst.

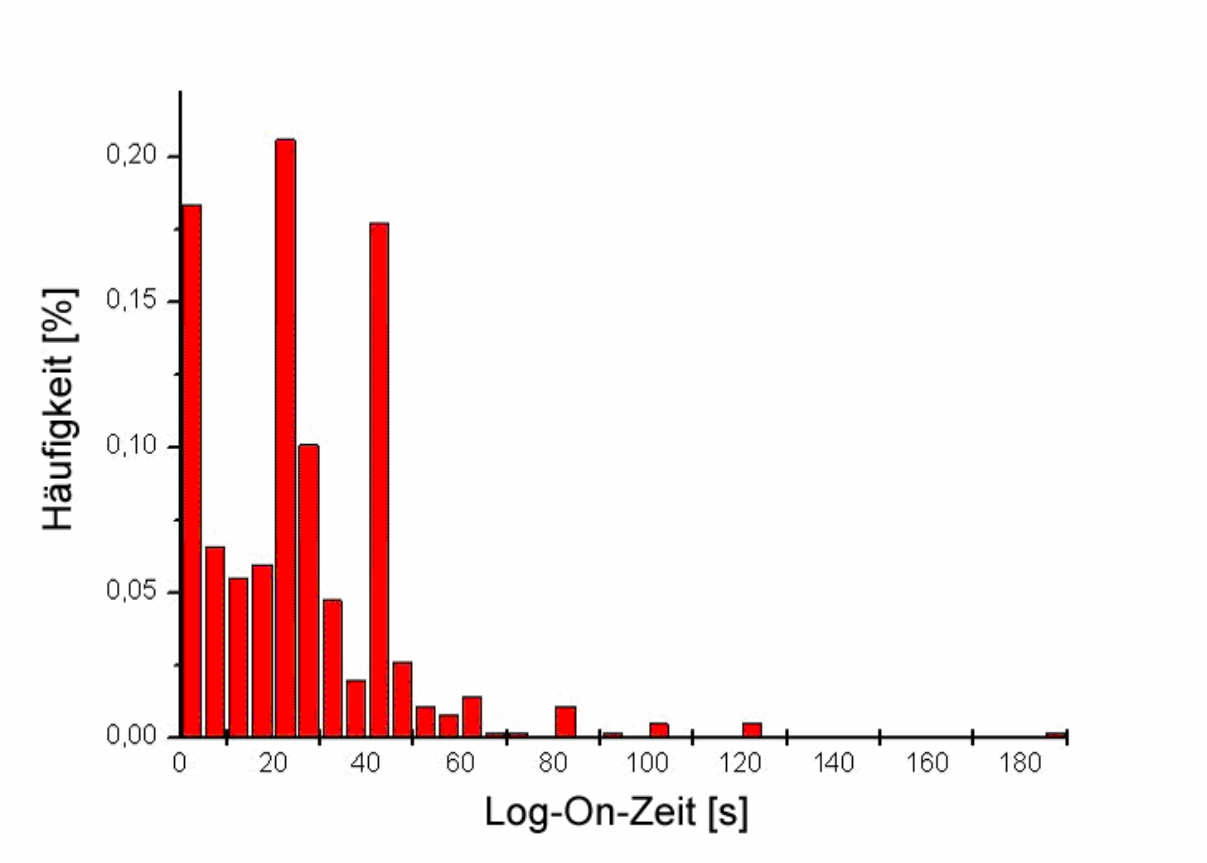

Abbildung 29: Häufigkeit verschiedener Log-On-Zeiten

Abbildung 29 zeigt die Häufigkeitsverteilung der Log-On-Zeit. Ermittelt wurden die Zahlen durch 50 Simulationsläufe mit 15 Knoten unter Einsatz des Random-Waypoint-Modells, wobei bereits eine Sicherheitsstruktur vorgegeben wurde. Im Durchschnitt braucht ein Knoten 24,9 Sekunden für die Anmeldung am Netz. Ca 25 % der Knoten können sich allerdings innerhalb der ersten 10 Sekunden anmelden. Besonders die hohen Werte treten im Abstand der Zeitspanne zwischen den CH-Broadcasts auf (in den Messungen 20 Sekunden).

Die Anmeldezeit hängt wesentlich vom Intervall ab, nach dem eine Zeitüberschreitung bei der Anmeldung erkannt wird. Erhöht man den Wert von 30 auf 80 Sekunden, so steigt die Anmeldezeit um 25,3% auf 31,2 Sekunden. Abbildung 30 zeigt die Häufigkeit bei erhöhtem Timeout (80 Sekunden). Deutlich zeigt sich auch eine ähnliche Häufigkeitsverteilung, die aller-

dings bei den hohen Werten dichter besetzt ist. Zusammenfassend weist dies darauf hin, dass es vielen Knoten nicht möglich ist, sich beim ersten empfangenen CH anzumelden. Weiterhin fällt auf, dass auch hier die hohen Werte im Abstand der Zeitspanne zwischen den CH-Broadcasts gehäuft auftreten. Dies ist ein Indiz dafür, dass den Knoten, die sich nach dem ersten CH-Broadcast nicht anmelden konnten, nach dem Empfang eines weiteren CH-Broadcast die Anmeldung relativ schnell gelingt. Es zeigt sich der gleiche Verlauf der Verteilung wie am Anfang. In Abbildung 30 ähnelt z.B. der Verlauf der Verteilung zwischen t=80 s und t=110 s dem Verlauf zwischen t=0 s und t=30 s.

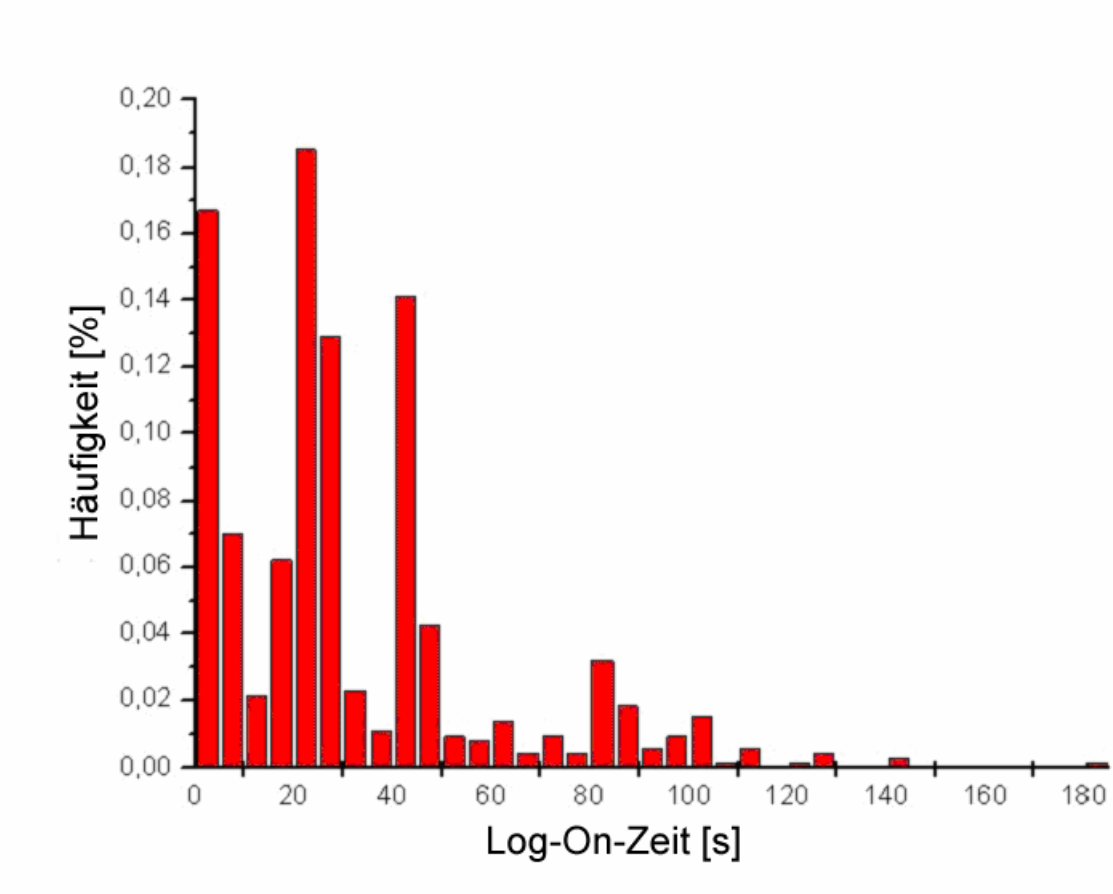

Abbildung 30: Log-On-Zeiten bei erhöhtem LogOn Timeout

Wird die Anzahl der Knoten in der Simulation erhöht, so steigt die Zeit, die für den Log-On nötig ist. Durch die höhere Knotenanzahl verfügt jeder Knoten im Durchschnitt über mehr Nachbarn. Dadurch werden die entstehenden Ad-hoc-Netze größer und es kommt am Anfang häufiger zu Clustereingliederungen und Clustervereinigungen. Dies wirkt sich negativ auf die Log-On-Zeit aus. Bei einer Netzvereinigung werden unter Umständen alle Netzschlüssel der betroffenen Netze ungültig, d.h. die Zahl der potentiellen Bürgen sinkt. Natürlich ist eine hö-

here Anzahl an Knoten dann günstiger, wenn viele Bürgen nötig sind, um von den Clusterheads ein Teil des identitätsbezogenen Schlüsselzertifikats IdZert zu erhalten. Abbildung 31 zeigt die Häufigkeitsverteilung einer Simulation mit 50 Durchläufen und 30 Knoten.

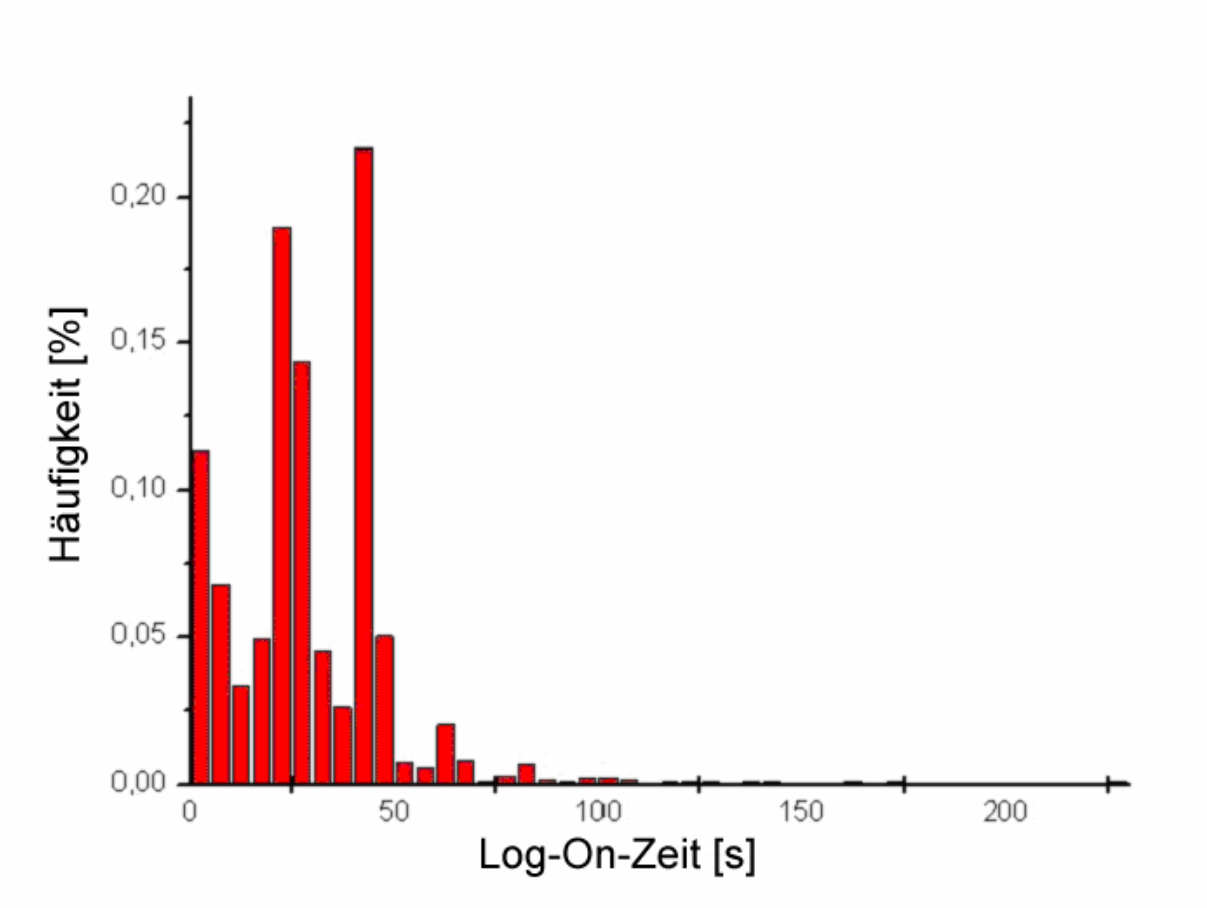

Abbildung 31: Log-On Zeiten bei erhöhter Knotenanzahl

Wie bereits oben besprochen, deuten die bisher gemessenen Verteilungen darauf hin, dass sich viele Knoten nicht sofort beim ersten Cluster anmelden können, den sie entdecken. Knoten können sich zum Beispiel aus folgenden Gründen nicht anmelden:

- weil es ihnen nicht gelingt, in der vorgegebenen Zeit genug Bürgenzertifikate zu sammeln,
- weil der Clusterhead nach dem ersten Kontakt nicht mehr mit dem Knoten kommunizieren kann,
- weil es in der Zeit der Anmeldung zu einer Vereinigung von Clusterhead-Netzwerken kommt in Folge derer alle (oder auch nur einige) Bürgenzertifikate ungültig werden.

Untersucht man nicht die Zeit zwischen dem ersten Kontakt zu einem Clusterhead und dem Empfang eines symmetrischen Clusterschlüssels sondern die Zeit zwischen dem Empfang des Clusterheads, zu dessen Cluster der Knoten später gehört, und der Zuteilung des symmetrischen Clusterschlüssels, so ergibt sich bei einer Simulation mit 50 Durchläufen, 15 Knoten (davon 2 CHs und 6 volle Clustermitglieder) und Random-Waypoint eine durchschnittliche reine LogOn-Zeit von ca. 2 Sekunden. Abbildung 32 zeigt die zugehörige Häufigkeitsverteilung.

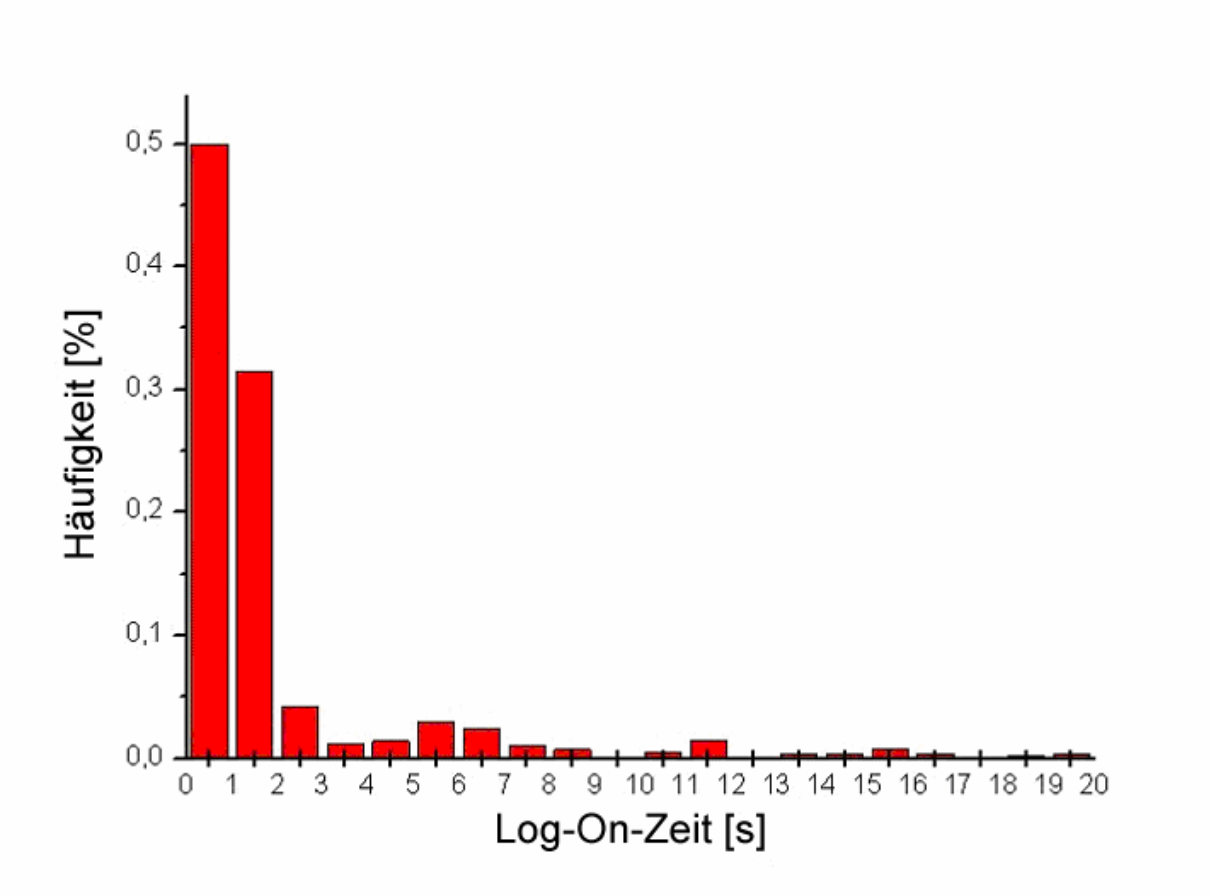

Abbildung 32: Häufigkeit der Zeit für den reinen Log-On

Kommt statt Random-Waypoint das Konferenz-Bewegungsmodell zum Einsatz, so erhöht sich die Zeit für den Log-On im Vergleich zu obigen Messungen leicht. Bei einer Simulation

mit 15 Knoten und 50 Durchläufen ergab sich eine durchschnittliche Log-On-Zeit von 27 Sekunden, eine Steigerung um 2 Sekunden. Die in Abbildung 33 wiedergegebene Verteilung ähnelt der Verteilung bei Random-Waypoint. Die leicht erhöhte Log-On-Zeit im Vergleich zum Random-Waypoint Modell hängt mit der sehr ungleichmäßigen Dichteverteilung im Simulationsgebiet beim Konferenzbewegungsmodell zusammen. An den Veranstaltungsorten ist die Knotendichte hoch während sich im übrigen Gebiet („auf den Gängen“) nur sehr wenige Knoten befinden. Nehmen nun an einer Veranstaltung nicht genügend Knoten teil, so hat ein Knoten dort nicht genügend Bürgen zur Verfügung. Durch die geringe Knotendichte außerhalb der Veranstaltungsorte gibt es unter Umständen keine Verbindung zwischen zwei Veranstaltungsorten und damit auch nicht genügend potentielle Bürgen. Dadurch dauert dann der Log-On-Vorgang länger.

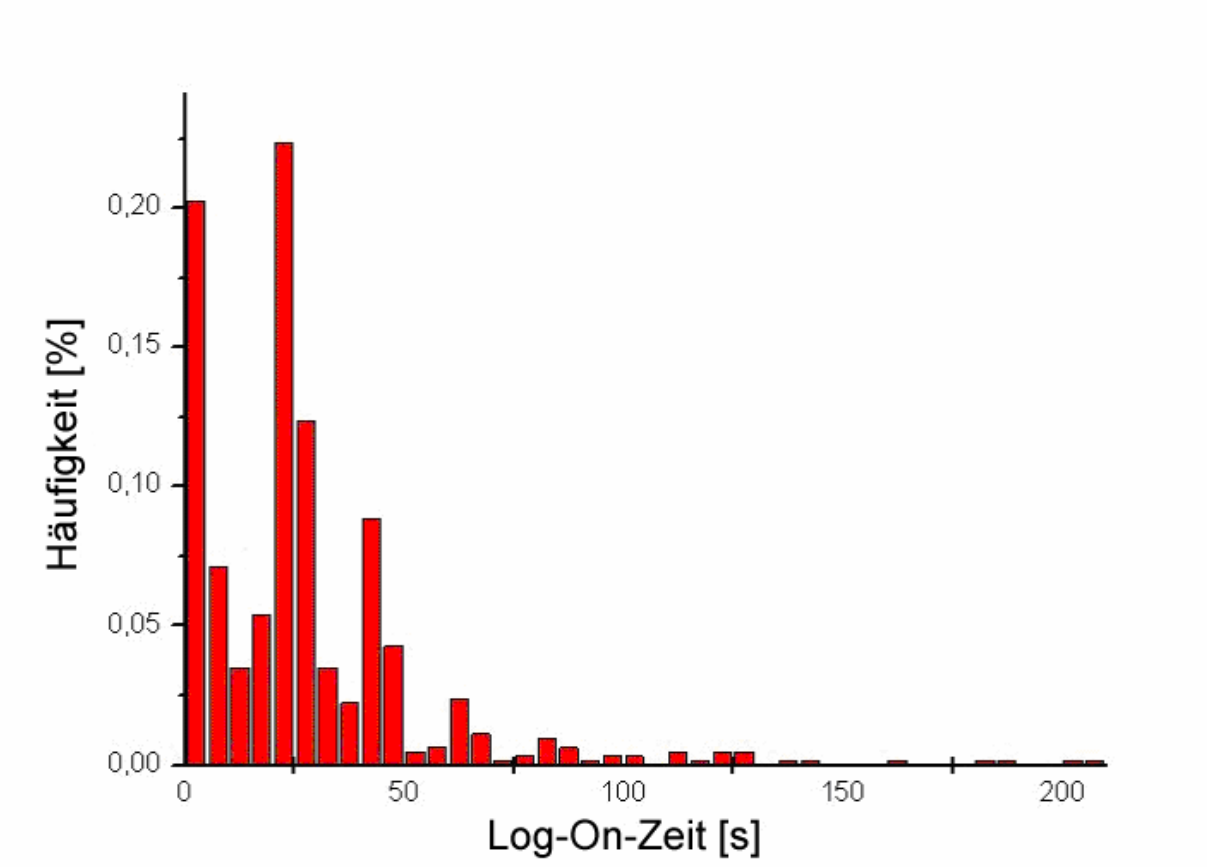

Abbildung 33: Log-On-Zeiten [Bewegungsmodell Konferenz]

Auch die reine Log-On-Zeit ähnelt bei Verwendung des Konferenz-Bewegungsmodells der von Random-Waypoint. Im Gegensatz zu Random-Waypoint sinkt allerdings die Log-On-Zeit, wenn man die Anzahl der Knoten in der Simulation erhöht. Bei 50 Läufen und 30 Knoten sinkt der Wert um 1 s auf 26 s. Abbildung 34 zeigt die Häufigkeitsverteilung. Anders als

im Random-Waypoint-Modell steigt durch die größere Anzahl von Knoten die Knotendichte zwischen den Veranstaltungsorten. Somit besteht eine höhere Wahrscheinlichkeit, dass die einzelnen Veranstaltungsorte demselben CH-Netzwerk angehören. Daraus resultiert aber auch eine höhere Anzahl an potentiellen Bürgen und damit auch ein schnellerer Log-On-Vorgang. Daraus wird auch ersichtlich, dass sich beim Konferenzbewegungsmodell nicht die Clustervereinigungen (wie bei Random-Waypoint) sondern der Mangel an potentiellen Bürgen negativ auf die Log-On-Zeit auswirkt. Hier macht sich ein Unterschied zwischen dem Random-Waypoint und dem Konferenz-Bewegungsmodell bemerkbar: während bei Random-Waypoint alle Knoten die meiste Zeit in Bewegung sind (und damit die Dynamik im entstehenden Netz sehr hoch ist) gibt es beim Konferenz-Bewegungsmodell bestimmte Bereiche des Simulationsgebiets, in denen keine oder nur sehr wenig Bewegung herrscht (die Veranstaltungsorte). Dies begünstigt schon sehr früh die Bildung eines (über gewisse Zeit) stabilen Netzes und führt somit zu weniger Clustereingliederungen oder Clustervereinigungen.

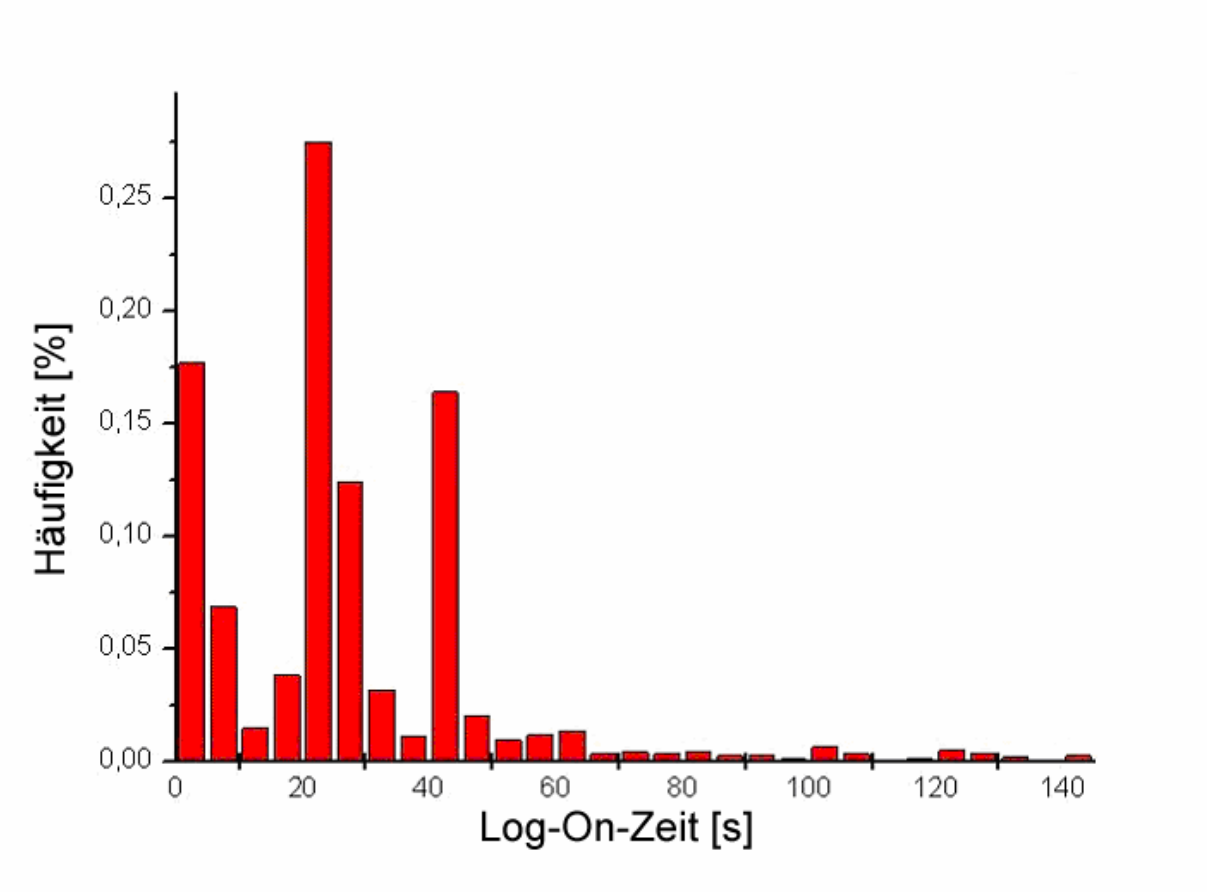

Abbildung 34: Log-On-Zeiten bei erhöhter Knotenanzahl

Die Sicherheitsarchitektur schneidet bei der Häufigkeitsverteilung der Log-On-Zeit deutlich schlechter ab, wenn das Bewegungsmodell „Autobahn“ zum Einsatz kommt. Die durchschnittliche Log-On-Zeit beträgt bei 50 Simulationsläufen mit 60 Knoten 41,5 Sekunden. Auch die Wahrscheinlichkeitsverteilung ist deutlich schlechter, wie aus Abbildung 35 ersichtlich wird. Nur ca. 11 % der Knoten können sich in den ersten 10 Sekunden anmelden. Auch im Vergleich der reinen Log-On-Zeit schneidet die Sicherheitsarchitektur in einer Simulation mit dem Bewegungsmodell Autobahn schlechter ab als unter Random-Waypoint und dem Konferenz Bewegungsmodell. Vergleicht man die in Abbildung 36 (Autobahn) und Abbildung 32 (Random-Waypoint) dargestellten Log-On-Zeiten, so fällt auf, dass sich beim Autobahnbewegungsmodell mehr Knoten innerhalb der ersten Sekunde anmelden können. Allerdings sind danach höhere Zeiten auch wahrscheinlicher als bei Random-Waypoint.

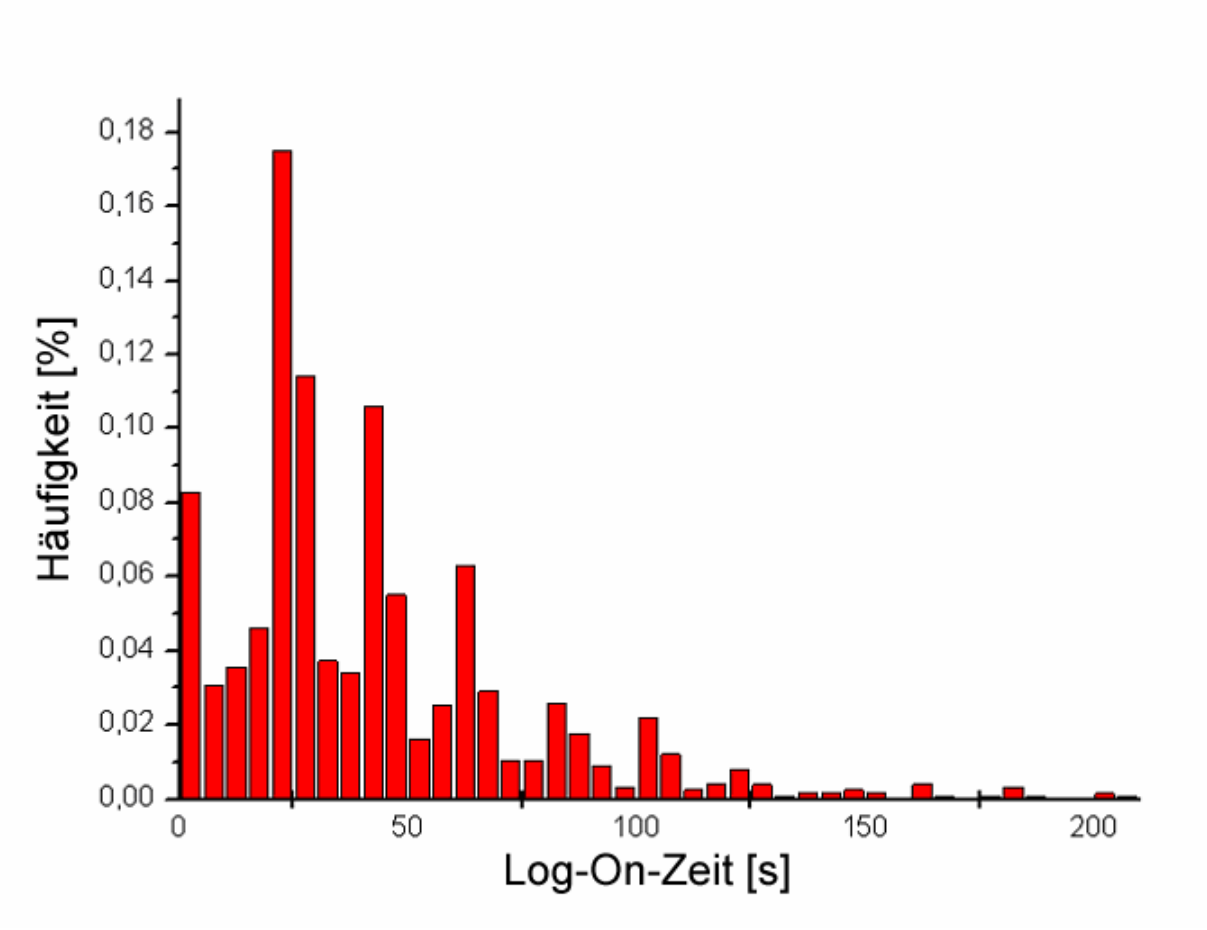

Abbildung 35: Log-On-Zeiten [Bewegungsmodell Autobahn]

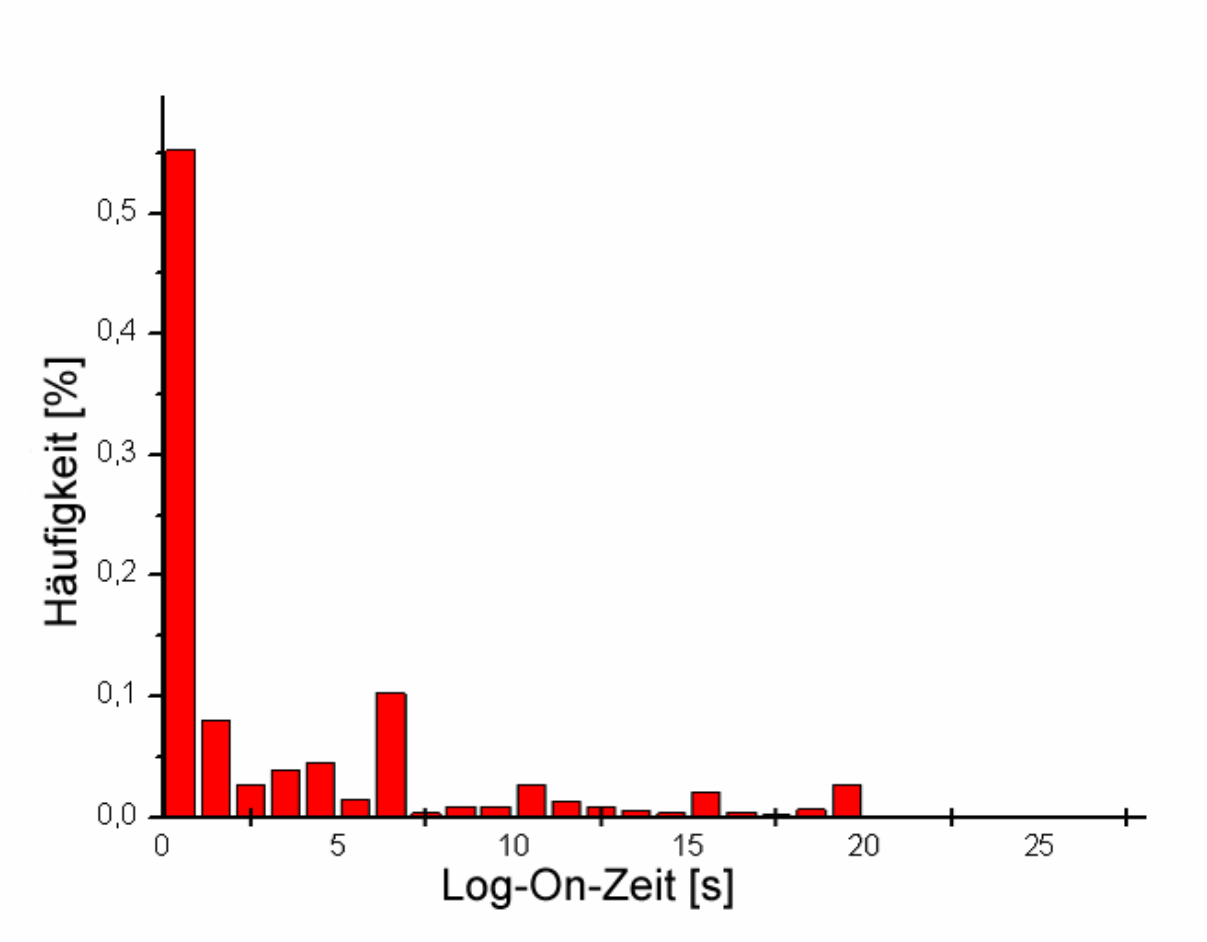

Abbildung 36: Reine Log-On-Zeit [Bewegungsmodell Autobahn]

5.2 Verfügbarkeit

Mit Verfügbarkeit wird jene Zeitspanne bezeichnet, in der ein Knoten sicher kommunizieren kann. Sichere Kommunikation wird nach einer erfolgreichen Anmeldung am Clusterhead möglich und wird aufrechterhalten, solange das Zertifikat des Knotens gültig ist. Die Verfügbarkeit im gesamten Netz sinkt, wenn zwei Netze vereinigt werden, da durch die Vereinigung der geheime Netzschlüssel neu erstellt werden muss. Alle Knoten, deren Schlüssel zuvor mit dem alten geheimen Netzschlüssel zertifiziert wurden, müssen sich erneut um ein Zertifikat bemühen.

Abbildung 37 zeigt die durchschnittliche Verfügbarkeit im Rahmen einer Messung mit 50 Simulationsläufen und 15 Knoten bei Verwendung des Random-Waypoint-Modells. Deutlich erkennbar sind zwei Phasen: In der ersten Phase, die in Abbildung 37 bis ca. t=55 s geht, baut sich die Struktur der Sicherheitsarchitektur langsam auf. Noch können nicht alle Knoten si-

cher kommunizieren. Die durchschnittliche Verfügbarkeit bricht immer wieder ein weil Netzvereinigungen stattfinden. Im Versuch lang die durchschnittliche Verfügbarkeit in dieser Phase bei ca. 58%. In der zweiten Phase ab ca. t=56 s kann ein Großteil der Knoten sicher kommunizieren. Es gibt immer wieder Einbrüche in der Verfügbarkeit, allerdings liegt sie größtenteils über 90%. Im Versuch ergab sich eine durchschnittliche Verfügbarkeit von 91,6% in dieser Phase. Zu beachten ist dabei, dass Knoten eventuell nur eine gewisse Zeitspanne nicht sicher kommunizieren können. Deshalb bedeutet das Ergebnis nicht, dass 8,4% der Benutzung der Sicherheitsarchitektur dauerhaft nicht zu sicherer Kommunikation fähig sind. Vielmehr sind 8,4 „wechselnde“ Prozent der Knoten für kurze Zeit nicht fähig sicher zu kommunizieren.

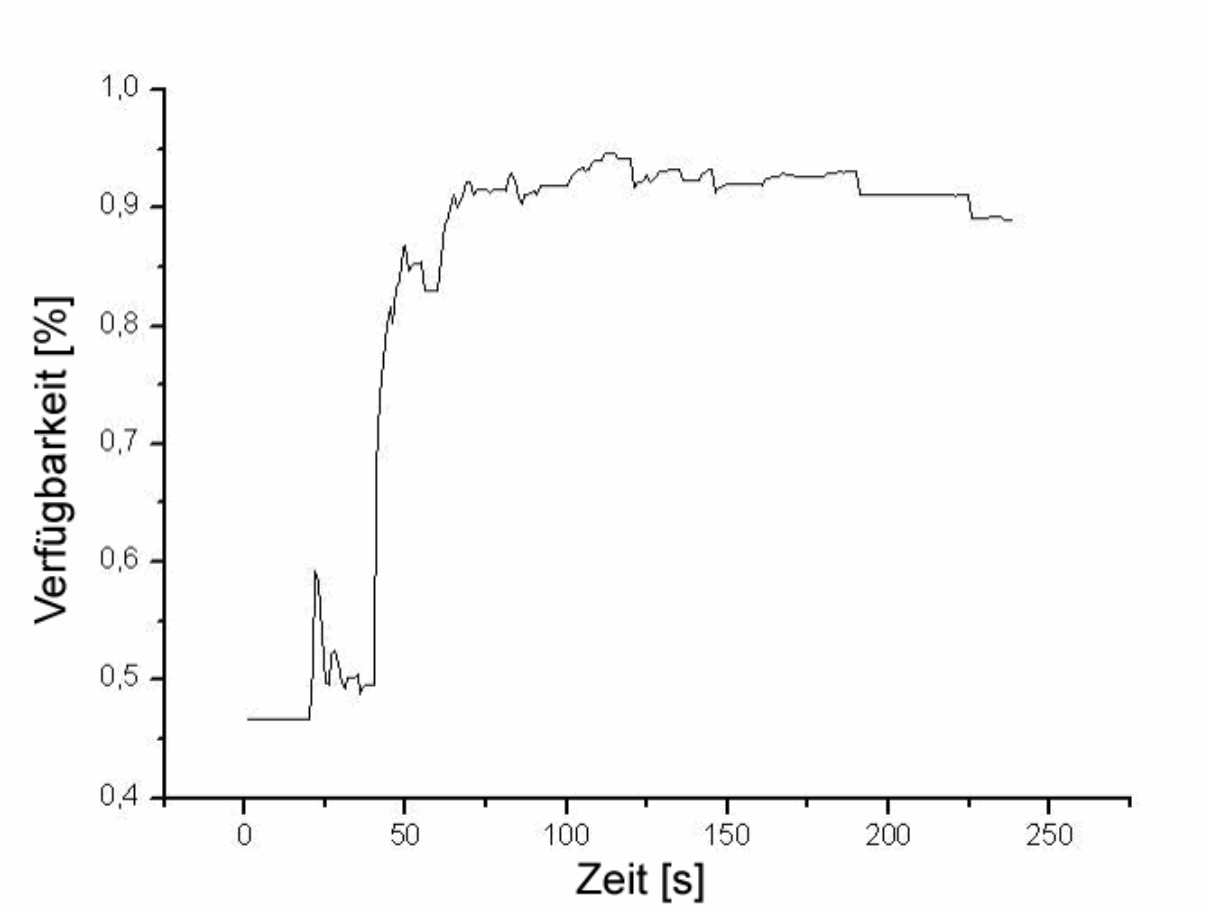

Abbildung 37: Verfügbarkeit der Sicherheitsarchitektur

Abbildung 38 zeigt die Verfügbarkeit bei einer Simulation mit den gleichen Parametern unter Einsatz des Bewegungsmodells „Konferenz“. Hier dauert die erste Phase etwas länger (dies

deutete sich schon durch die längere durchschnittliche LogOn-Zeit an), und die durchschnittliche Verfügbarkeit in Phase 2 ist mit 90,4% geringfügig niedriger.

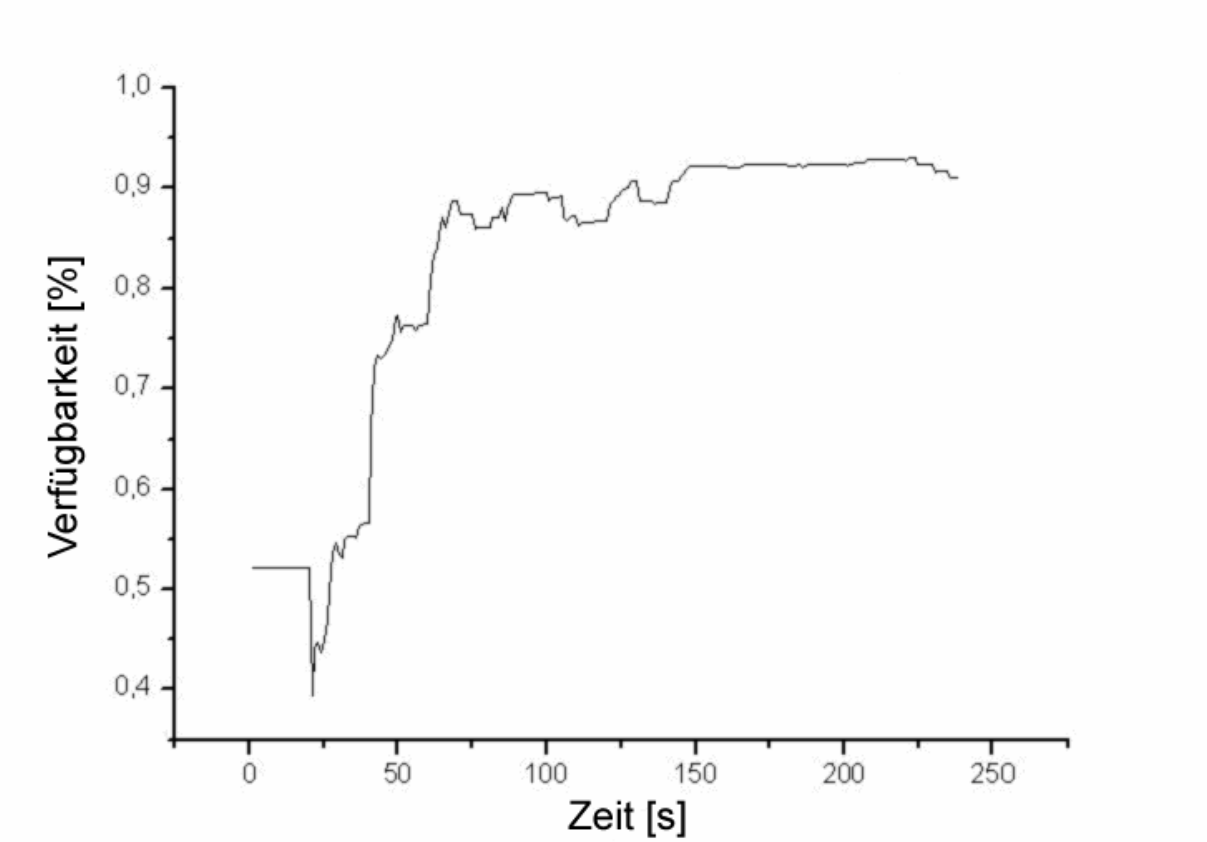

Abbildung 38: Verfügbarkeit der Sicherheitsarchitektur [Konferenzmodell]

Eine gänzlich andere Kurve zeigt eine Simulation mit 50 Läufen und 30 Knoten unter Verwendung des Bewegungsmodells „Autobahn“. Aus Abbildung 39 wird ersichtlich, dass beim Bewegungsmodell „Autobahn“ die Verfügbarkeit nicht sprunghaft ansteigt sondern gemäßigt über einen Zeitraum von ca. 125 Sekunden. Danach liegt die Verfügbarkeit wie bei den Modellen „Random Waypoint“ und „Konferenz“ über 90%. Eine Simulation mit gleichen Parametern und 60 Knoten ergibt einen sehr ähnlichen Kurvenverlauf. Auch dieses Ergebnis deutete sich bereits durch die höhere Log-On-Zeit und die breitere Verteilung der reinen Log-On-Zeit an.

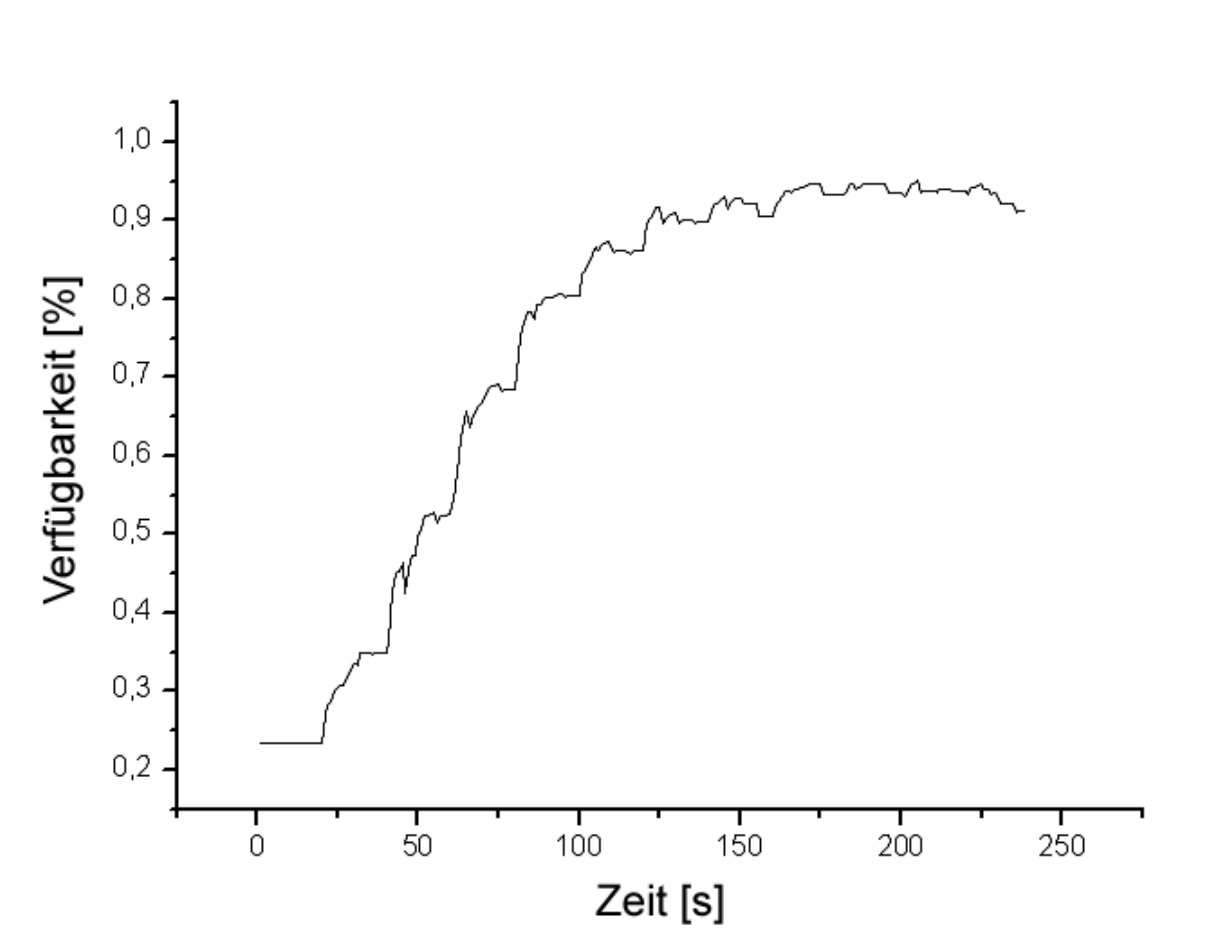

Abbildung 39: Verfügbarkeit [Autobahn]

Die durchschnittliche Verfügbarkeit ist sehr stark abhängig von der Periodendauer, in der CH-Beacons versendet werden. In Abbildung 40 wird die Verfügbarkeit für CH-Periodendauern von 10, 20, 30 und 40 Sekunden dargestellt (in einer Simulation mit 30 Knoten und unter Verwendung des Random-Waypoint-Modells). Deutlich erkennbar wird mit einer CH-Periodendauer von 20 Sekunden das Optimum erreicht. Die einzelnen Verfügbarkeiten unterscheiden sich um bis zu 10%. Die Wahl einer geeignenten CH-Beacon-Periodendauer ist also für die Leistung des Gesamtsystems maßgeblich.

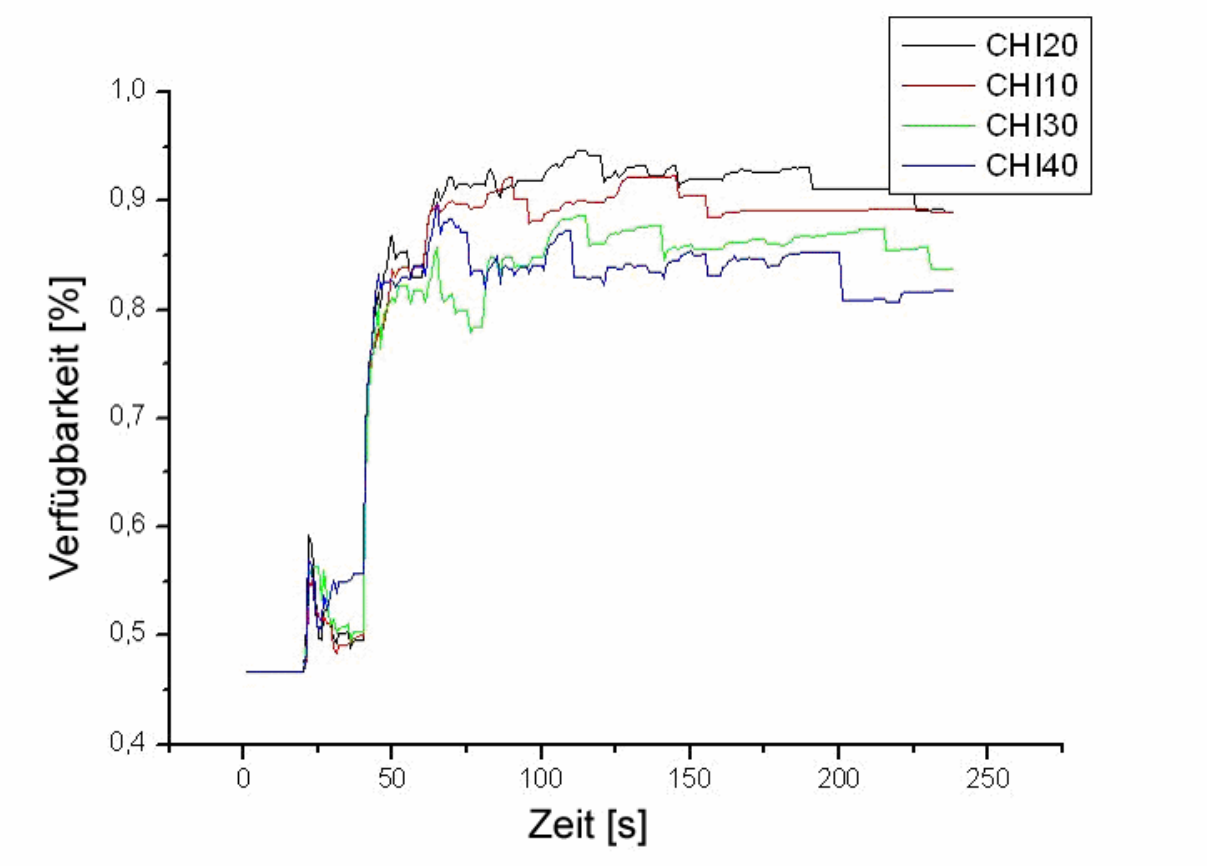

Abbildung 40: Verfügbarkeit für verschiedene CH-Intervalldauer

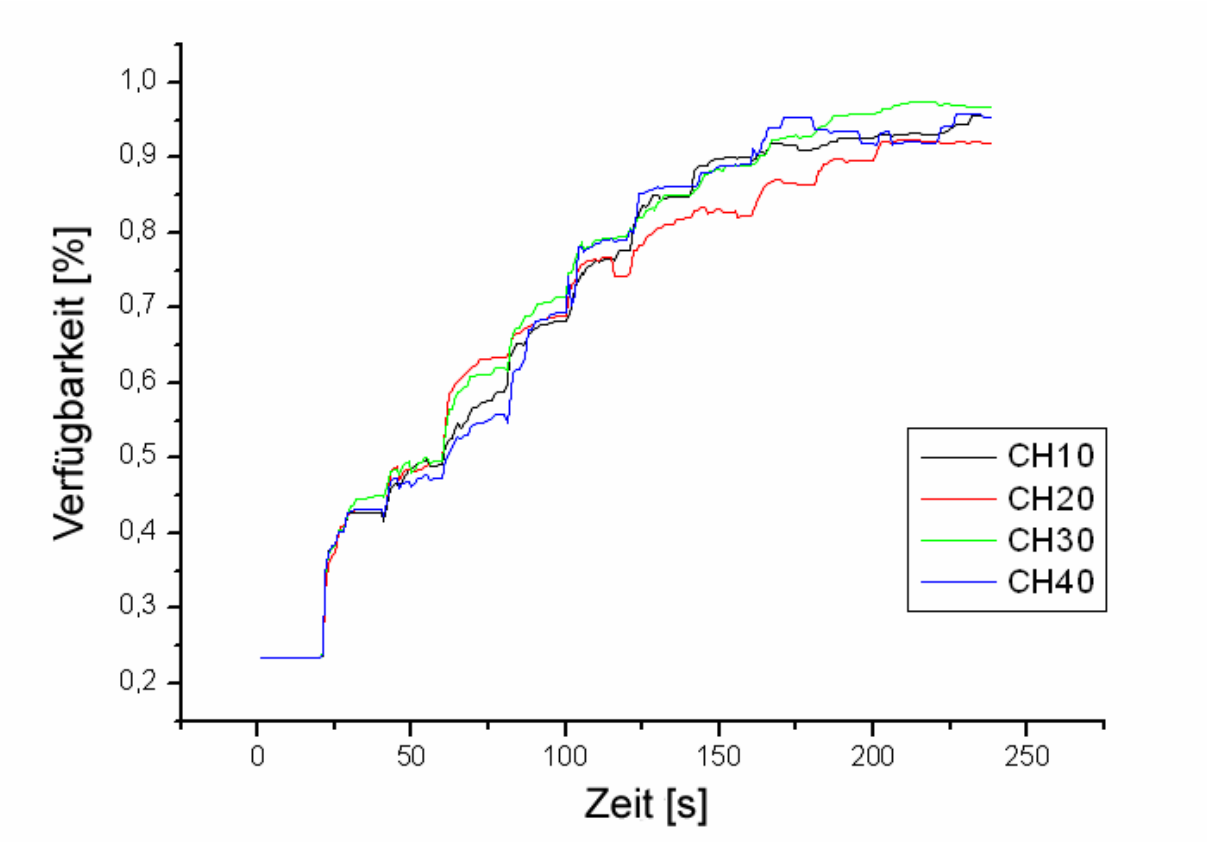

Abbildung 41: Verfügbarkeit für verschiedene CH Intervalldauer [Autobahn]

Eine Simulation mit 30 Knoten und dem Bewegungsmodell „Autobahn“ liefert andere Ergebnisse. Aus Abbildung 41 geht ein CH-Periodendauer von 30 Sekunden als Optimum hervor. Die Periodendauer 20 Sekunden, die im vorherigen Versuch das Optimum darstellte, liefert hier das schlechteste Ergebnis. Allerdings sind beim Bewegungsmodell „Autobahn“ die Unterschiede in der Verfügbarkeit mit 5 % nur halb so groß wie im vorherigen Versuch mit Random-Waypoint.

5.3 Overhead

Im letzten Abschnitt wurde bereits gezeigt, wie wichtig die richtige Wahl eines geeigneten CH-Broadcast-Intervalls ist. In diesem Kapitel wird untersucht, wie sich die Wahl auf den Overhead auswirkt, der durch die Sicherheitsarchitektur entsteht. Da die CH-Broadcasts geflutet werden, löst ein einzelner CH-Broadcast schon sehr viel Verkehr aus. Zudem regt ein erhaltenes CH-Beacon bei den Knoten verschiedene Aktionen an, die weiteren Verkehr nach sich ziehen. So kann sich ein Knoten beispielsweise nach dem Empfang des CH-Beacons einem Clusterhead zuordnen, oder ein CH erkennt einen anderen CH und stößt eine Netzvereinigung an. Es spricht also einiges dagegen, die CH-Beacon-Periodendauer zu klein zu wählen. Andererseits bewirkt ein zu langes CH-Intervall, dass Knoten unter Umständen den Verlust eines CHs zu spät erkennen und durch den misslingenden Anmeldeversuch unnötig Overhead entsteht. Ganz umsonst ist dieser Verkehr aber nicht, denn einmal erhaltene Bürgen-Zertifikate können unter Umständen für spätere Anmeldungen am selben Clusterhead-Netzwerk wieder verwendet werden. Abbildung 42 zeigt den erzeugten Overhead in Paketen pro Sekunde für die Werte 10, 20, 30 und 40 Sekunden CH-Intervalldauer. Die Werte wurden durch 50 Simulationsläufe mit dem Random-Waypoint-Bewegungsmodell und 15 Knoten gewonnen. Der Unterschied zwischen den einzelnen Kurven liegt größtenteils zwischen 10 und 50 Paketen/Sekunde. Abbildung 43 zeigt dieselbe Messung unter Verwendung des Bewegungsmodells „Autobahn“.

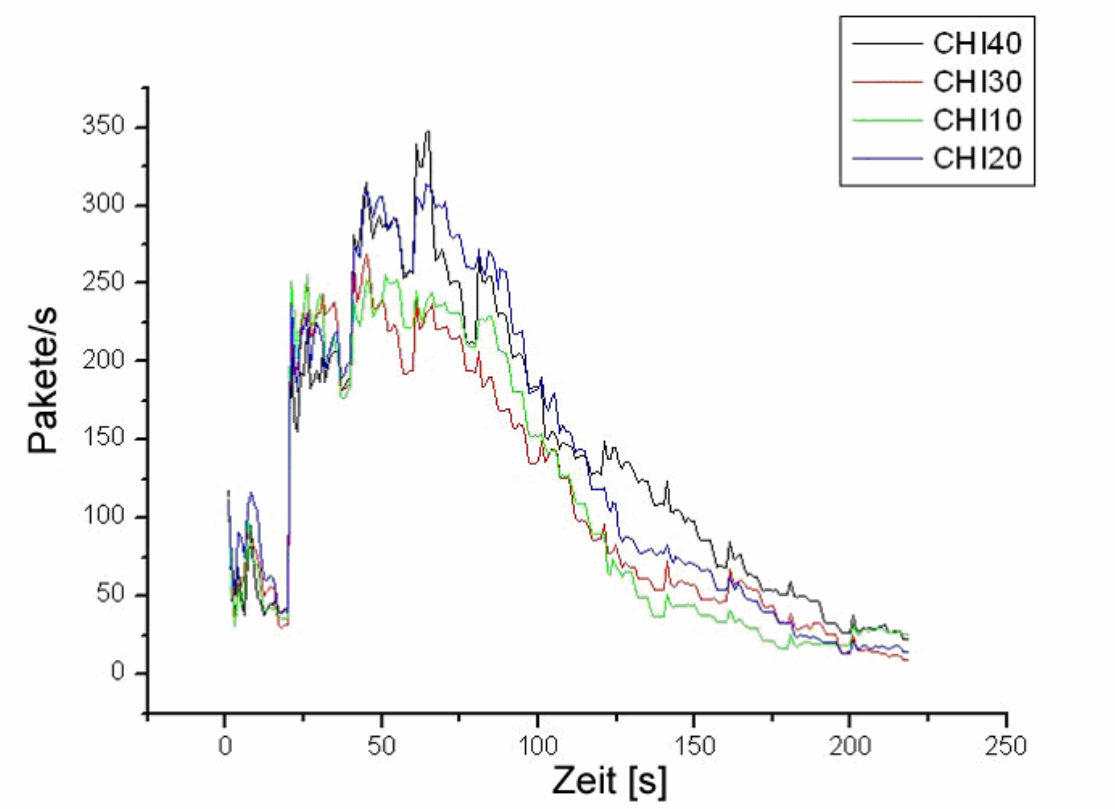

Abbildung 42: Overhead für verschiedene CH Intervalldauer [Random-Waypoint]

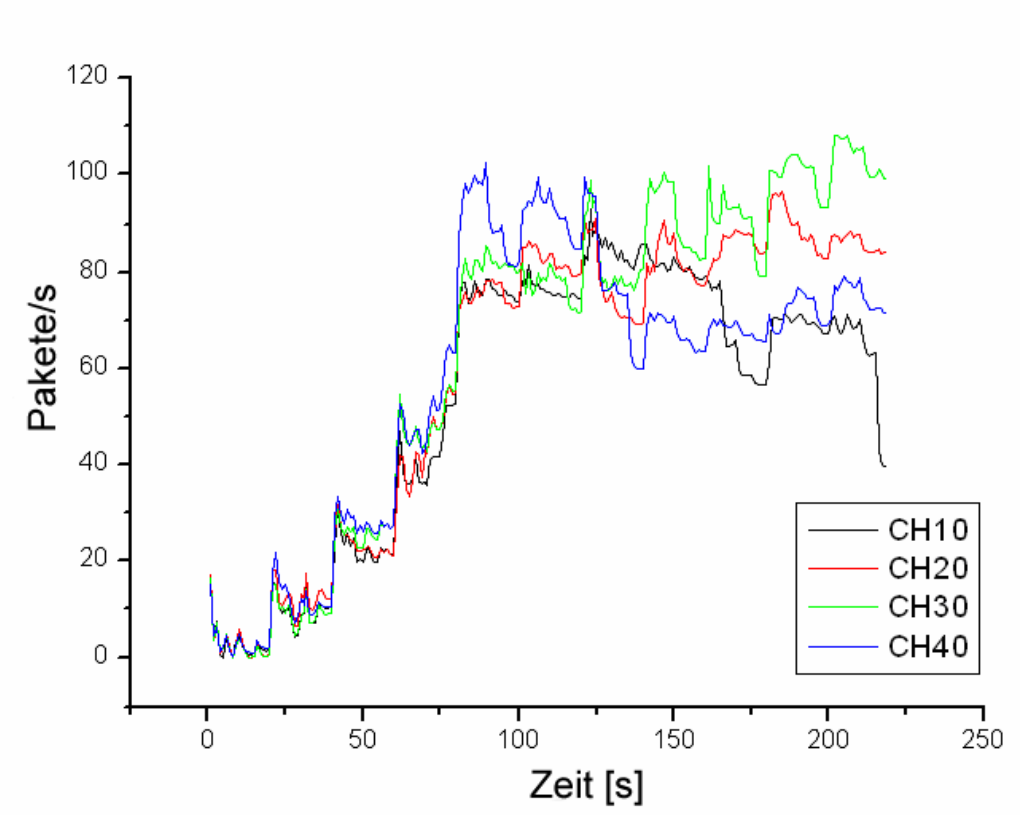

Abbildung 43: Overhead für verschiedene CH Intervalldauer [Autobahn]

5.4 Aufgabenverteilung

Ein Knoten kann im Rahmen der Sicherheitsarchitektur verschiedene Funktionen übernehmen. So kann er z.B. Clusterhead, Gateway oder volles Clustermitglied werden. Daneben gibt es aber immer noch Knoten, die nicht an der Sicherheitsarchitektur teilnehmen. Abbildung 44 zeigt die durchschnittliche Aufgabenverteilung für 50 Simulationsläufe mit 15 Knoten und Bewegung nach Random-Waypoint. Mit steigender Knotenanzahl ändert sich die Aufgabenverteilung. Dabei steigt die absolute Anzahl von CHs nur leicht an. Bei 15 Knoten (Abbildung 44) wurden durchschnittlich 2,7 Knoten zu CHs, bei 30 Knoten (Abbildung 45) 3,25 und bei 45 Knoten (Abbildung 46) 3,23 Knoten. Die Anzahl an CHs und damit auch die maximal mögliche Größe eines CH-Netzwerks in der Simulation scheint bei genügend Knoten nur von der Größe des Simulationsgebiets abzuhängen. Es liegt daher die Vermutung nahe, dass die durchschnittliche Anzahl von CHs sich bei quadratischem Simulationsgebiet annähernd berechnen lässt durch folgende Formel:

$$\frac{\text{Größe des Simulationsgebiets}}{\text{maximale Übertragungsreichweite.}}$$

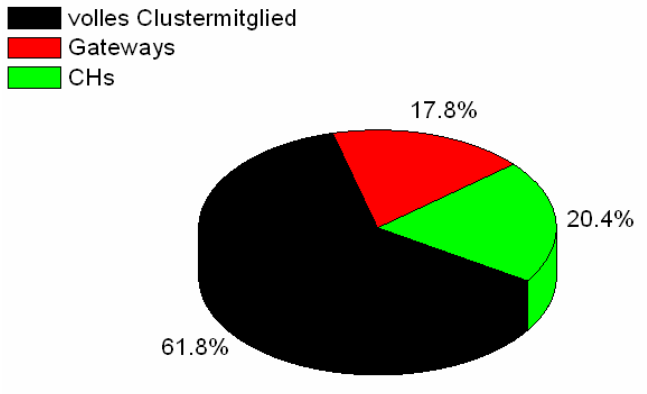

Abbildung 44: Aufgabenverteilung bei 15 Knoten

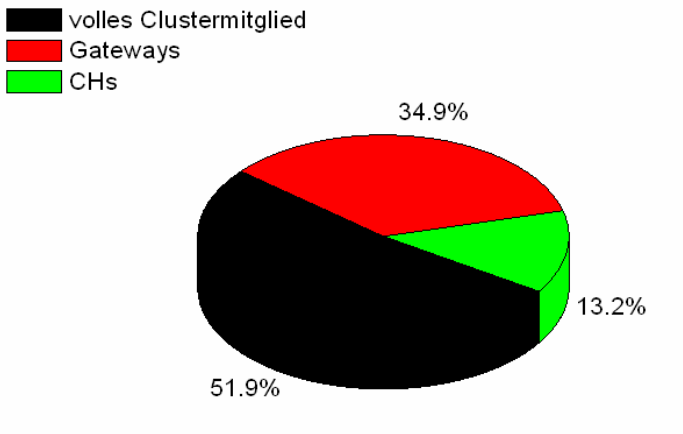

Abbildung 45: Aufgabenverteilung bei 30 Knoten

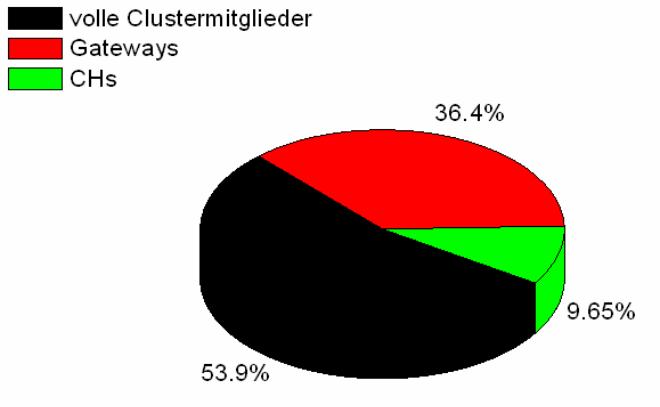

Abbildung 46: Aufgabenverteilung bei 45 Knoten

Beim Bewegungsmodell „Autobahn“ fällt auf, dass im Gegensatz zum Random-Waypoint-Bewegungsmodell wesentlich weniger Gateways vorhanden sind. Dies hängt mit der Verteilung der Knoten im Simulationsgebiet zusammen. Da sich Knoten nur in bestimmten Regionen bewegen können („Fahrbahn“) und Gateways nur in den Schnittbereichen von zwei Clustern auftreten ist die Wahrscheinlichkeit niedriger, dass ein Knoten Gateway wird. Abbildung 47 zeigt die Aufgabenverteilung bei einer Simulation mit 50 Läufen und 30 Knoten.

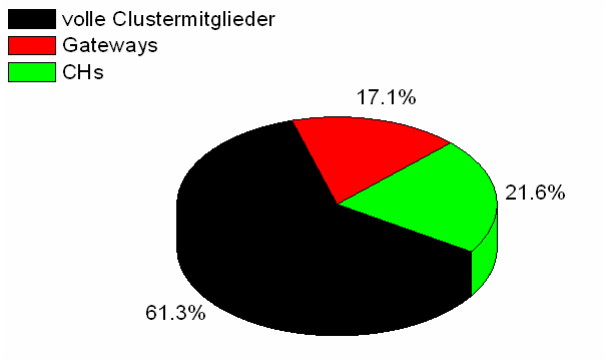

Abbildung 47: Aufgabenverteilung bei 30 Knoten [Autobahn]

5.5 Belastung herausgehobener Knoten

Für die Nutzer der Sicherheitsarchitektur ist interessant, wie stark ihre Geräte durch die Teilnahme an der dezentral ausgelegten Sicherheitsarchitektur zusätzlich belastet werden. Ein Knoten kann besondere Aufgaben übernehmen wie z.B. volles Clustermitglied, Gateway oder Clusterhead. Diese Aufgaben bringend zusätzliche Belastung mit sich.

Clusterheads müssen den Cluster organisieren. Dazu senden sie regelmäßige CH-Broadcasts. Jeder neue Knoten im Cluster nimmt zuerst Kontakt mit dem Clusterhead auf. Der Clusterhead versorgt die Knoten mit einer Liste der Gateways und mit einer Liste potenzieller Bürgen. Schließlich erstellt der Clusterhead einen Teil des Schlüsselzertifikats. Der Clusterhead tauscht sich in regelmäßigen Abständen mit anderen Clusterheads des CH-Netzwerks aus. Durch alle diese Aufgaben entsteht dem Clusterhead zusätzlicher Verkehr, der dessen Ressourcen belastet.

Gateways leiten den Verkehr der Teilnehmer weiter. Diese Aufgabe wird nicht primär durch die Sicherheitsarchitektur bestimmt, denn auch ohne die Architektur würden die Knoten auf Grund ihrer günstigen Lage zwischen mehreren Clustern Pakete weiterleiten. Interessant ist jedoch, wie viele zusätzliche Pakete durch die Sicherheitsarchitektur weitergeleitet werden müssen.

Volle Clustermitglieder dienen als Bürgen für potentielle neue Mitglieder des Clusters. Versendet ein Knoten Bürgenzertifikate, so entsteht ihm dadurch ebenfalls zusätzlicher Verkehr. Die folgende Abbildung 48 zeigt die relative Verteilung für den Verkehr von vollen Clustermitgliedern (Members), Gateways und Clusterheads (CHs) bei einer Simulation mit 50 Läu-

fen und 15 Knoten unter Verwendung des Random-Waypoint-Modells. Abbildung 45 zeigt ebenfalls die relative Verteilung, allerdings für 30 Knoten. Der Verkehr über die CHs nimmt hier zu. Die durchschnittliche Anzahl der CHs steigt nur unwesentlich von 2,7 auf 3,25 die Knotenanzahl verdoppelt sich allerdings, so dass mehr Anmeldungen pro CH geschehen. Dies ist der Grund für den erhöhten Verkehr über die Clusterheads.

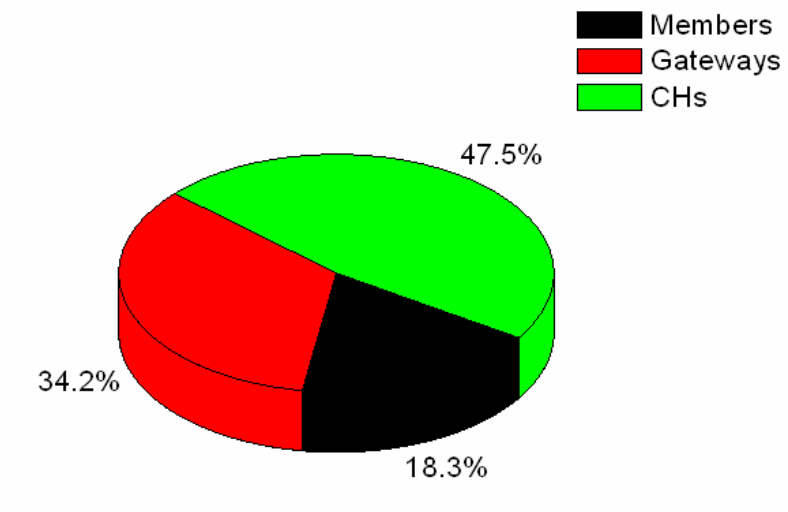

Abbildung 48: Verteilung des Verkehrs auf CHs, GWs und Members bei 15 Knoten

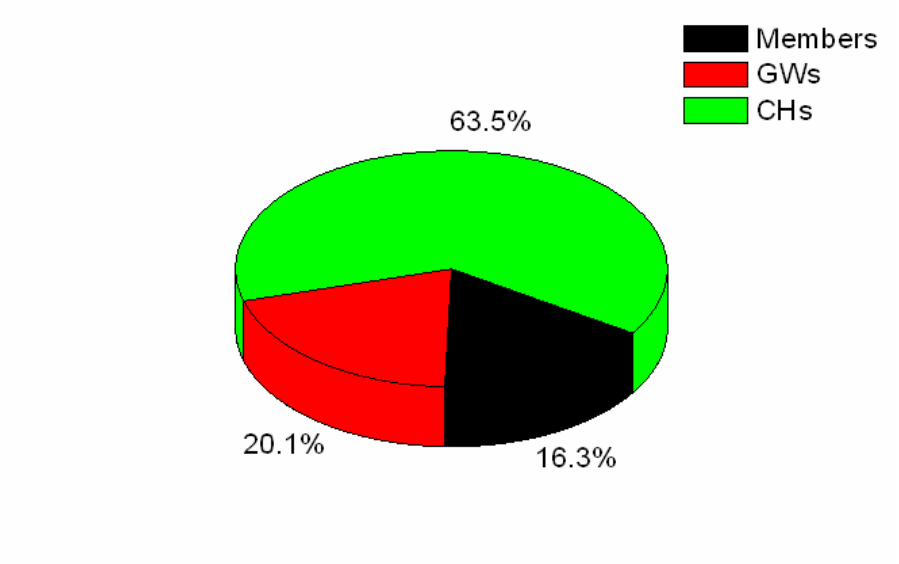

Abbildung 49: Verteilung des Verkehrs auf CHs, GWs und Members bei 30 Knoten

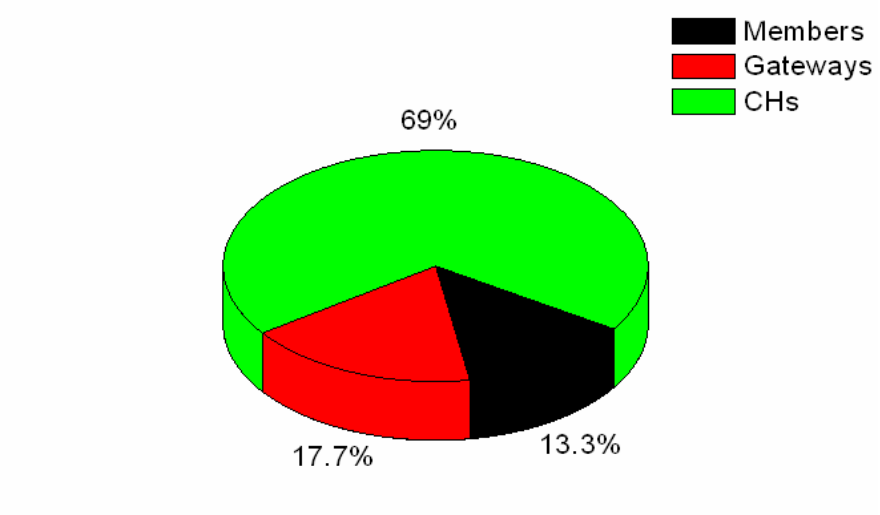

Abbildung 50: Verteilung des Verkehrs auf CHs, GWs und Members bei 45 Knoten

Abbildung 51 schließlich zeigt, wie sich die Belastung der einzelnen Knoten bei steigender Knotenanzahl beim Random-Waypoint-Modell entwickelt. Die Belastung von Gateways und vollen Clustermitgliedern steigt nur mittelmäßig an während Clusterheads deutlich mehr belastet werden, je mehr Knoten an der Simulation teilnehmen. Dies erklärt sich durch die wichtige Rolle, die sie bei der Anmeldung von Knoten wahrnehmen, und durch die nur langsam wachsende Anzahl von CHs. Sowohl beim Konferenz-Modell als auch beim Autobahn-Bewegungsmodell zeigt sich eine ähnliche Kurve. Abbildung 52 zeigt die steigende Belastung im Autobahn-Bewegungsmodell.

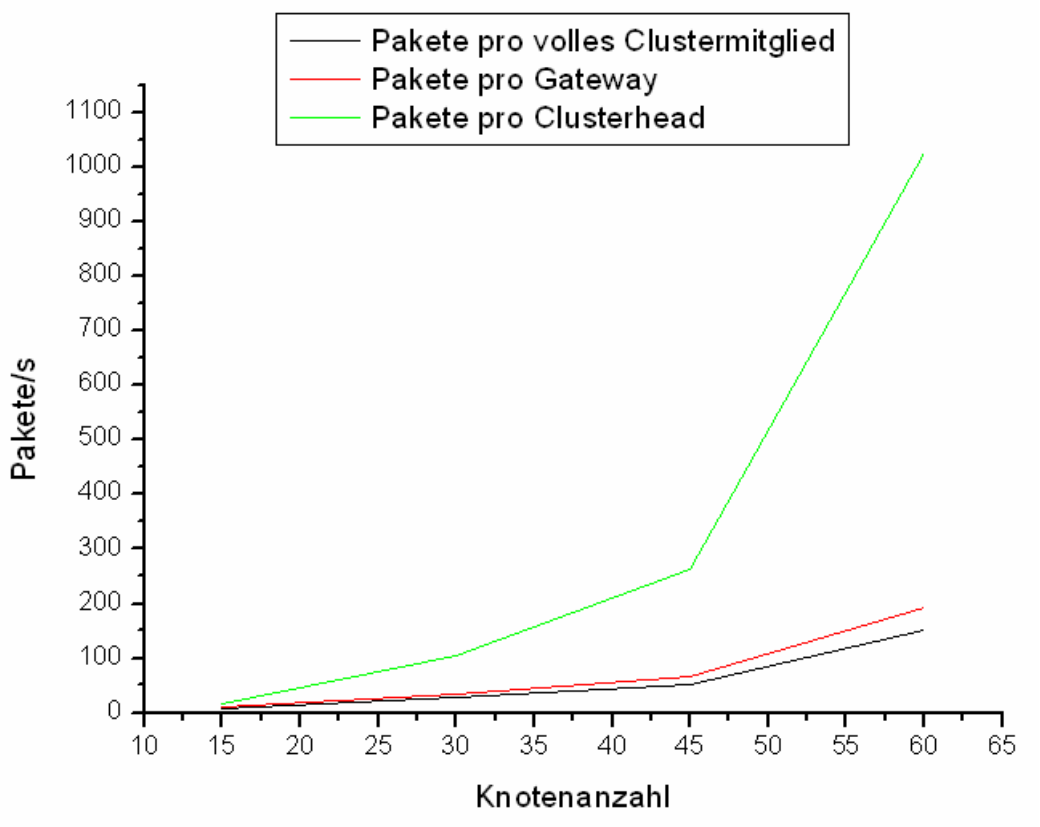

Abbildung 51: Belastung bei steigender Knotenanzahl [Random-Waypoint]

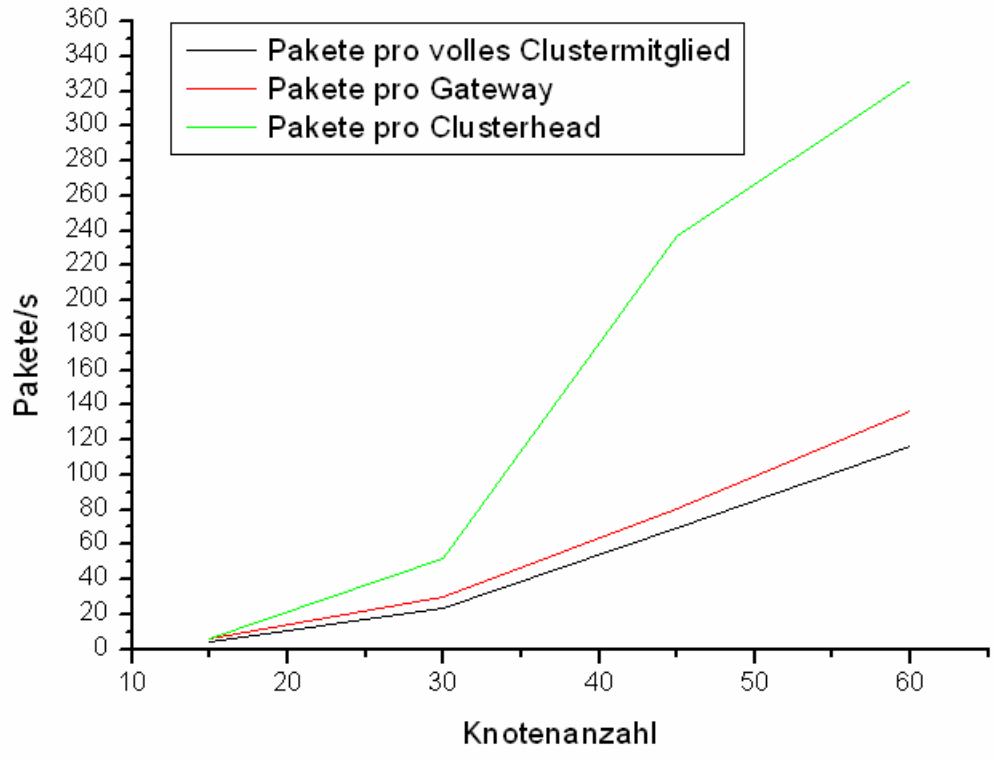

Abbildung 52: Belastung bei steigender Knotenanzahl [Autobahn]

5.6 Clustergröße

Die Clustergröße kann als Parameter gewählt werden. Sie bestimmt nicht, wie groß ein Cluster absolut ist (in Metern), sondern vielmehr, wie viele Hops ein CH-Broadcast weitergeflutet wird, und damit nur indirekt die absolute Größe. Abbildung 53 zeigt, wie sich die Clustergröße auf den Overhead auswirkt. Der Overhead bleibt kleiner, wenn die CH-Beacons nur wenige Hops geflutet werden. Dies hängt aber nicht nur mit dem direkt entstehenden Verkehr zusammen, sondern auch damit, dass sich die Cluster bei höherer Clustergröße schneller gegenseitig entdecken. Dadurch finden schneller Clustervereinigungen statt, welche wiederum eine Neuanmeldung einiger Knoten nach sich zieht.

Auch auf die Verfügbarkeit hat die Clustergröße Einfluss. Ist der Cluster größer, so bleiben bewegte Knoten länger in einem Cluster. Knoten, die in diesen Cluster kommen, stehen dann mehr Bürgen zur Verfügung. Neuanmeldungen z.B. nach Clustereingliederungen laufen damit auch schneller ab. In Abbildung 54 wird die Verfügbarkeit für drei Clustergrößen dargestellt.

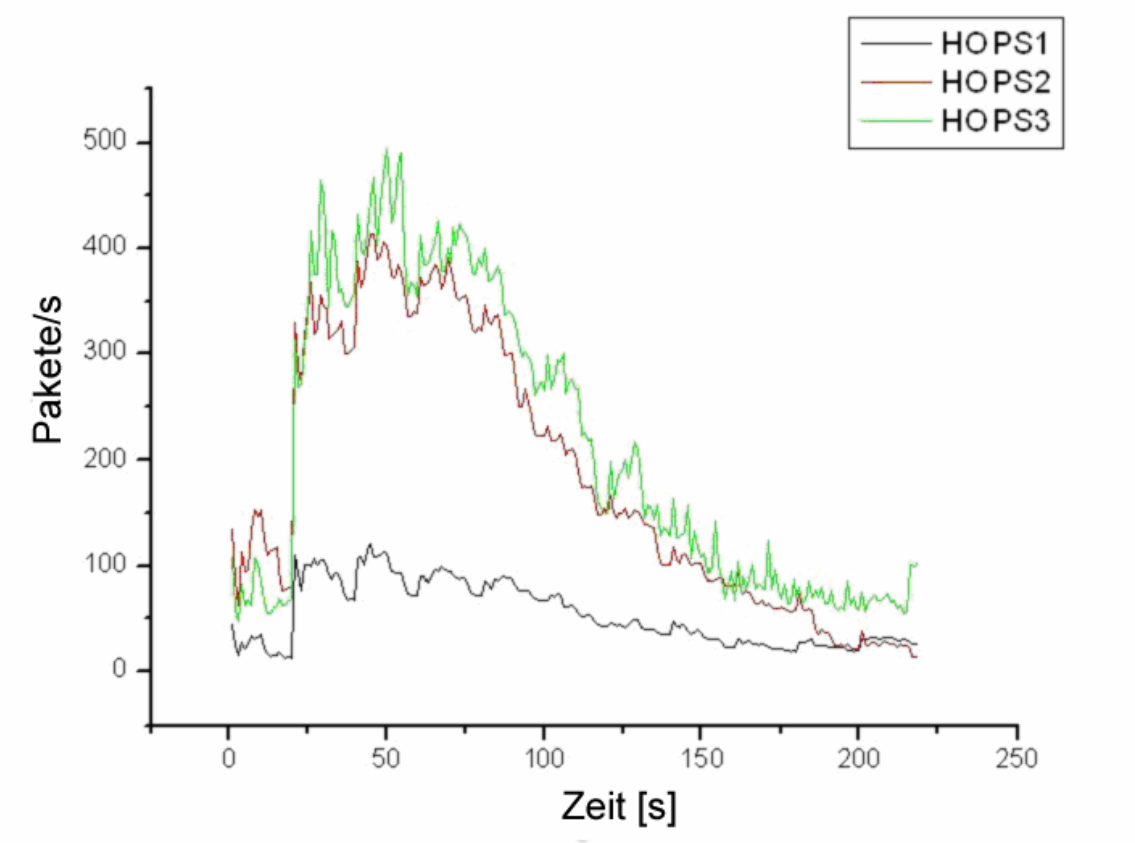

Abbildung 53: Overhead bei verschiedenen Clustergrößen

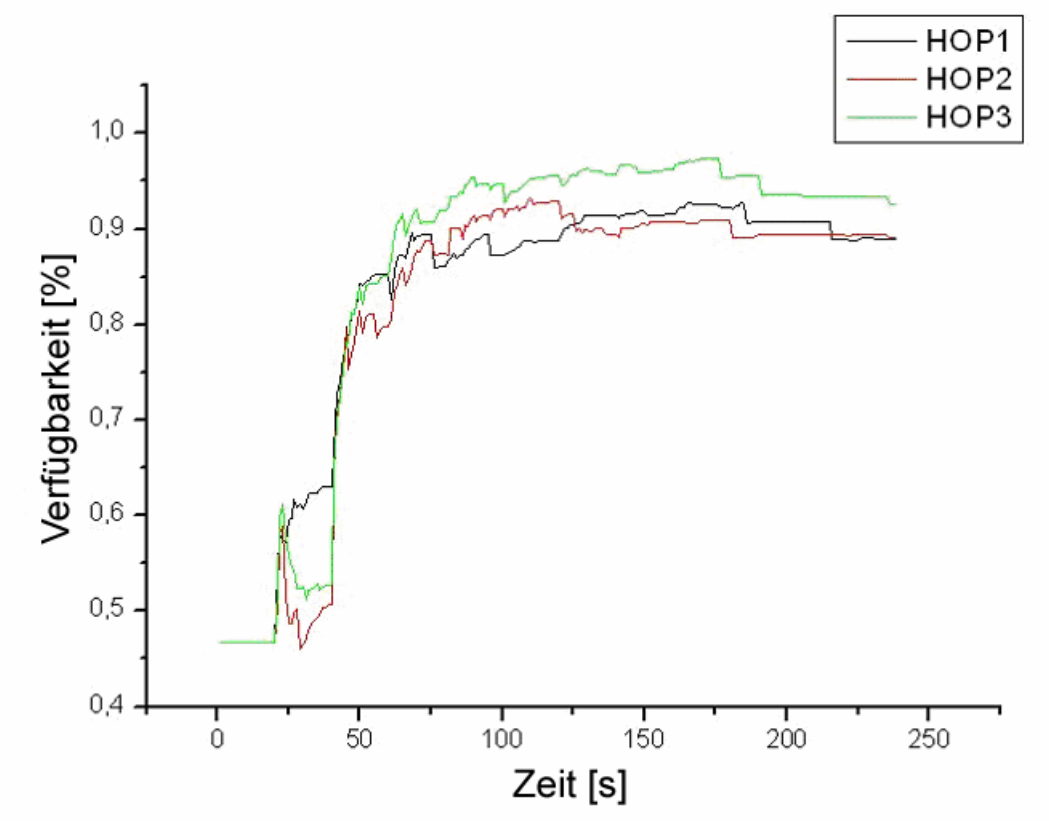

Abbildung 54: Verfügbarkeit bei verschiedenen Clustergrößen

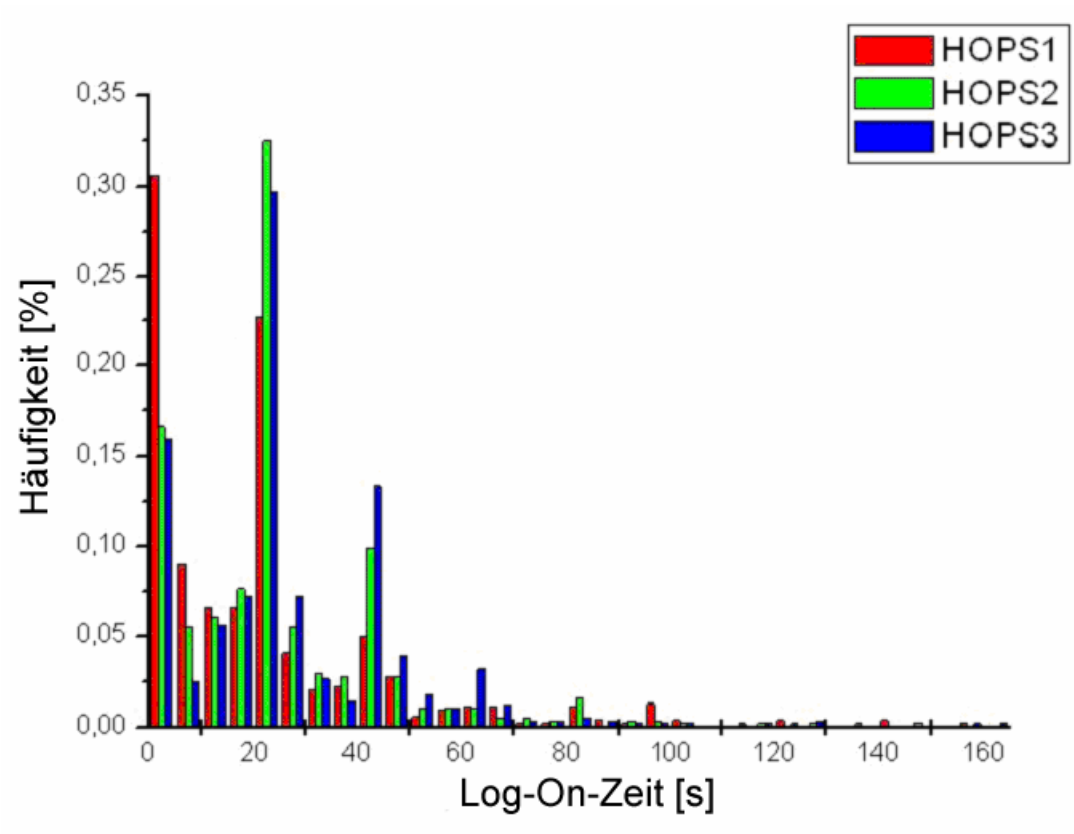

Abbildung 55: Log-On-Zeiten bei verschiedenen Clustergrößen

Auch auf die Log-On-Zeit hat die Clustergröße Einfluss, denn bei kleiner Clustergröße finden Clustervereinigungen später oder gar nicht statt. Die reine Log-On-Zeit ändert sich jedoch nur geringfügig. In Abbildung 55 werden für verschiedene Clustergrößen die Log-On-Zeiten dargestellt.

5.7 Bewertung

Random-Waypoint und das Konferenz-Modell auf der einen Seite und das Autobahn-Modell auf der anderen Seite sind sehr unterschiedliche Szenarien mit völlig verschiedenen Charakteristiken. Die Messungen haben gezeigt, dass sich die Sicherheitsarchitektur für alle drei Szenarien eignet. So liegt z.B. die Verfügbarkeit der Sicherheitsarchitektur in allen drei Modellen nach einer Aufbauphase bei über 90%. Auch für den Overhead lassen sich eindeutige Obergrenzen erkennen, die die Leistungsfähigkeit bestätigen. Die überwiegende Mehrheit der Knoten kann sich am Cluster innerhalb kurzer Zeit anmelden.

Deutlich wurde aber auch, wie wichtig die Wahl der Parameter ist, z.B. des CH-Broadcast Intervalls. In diesem Bereich sind Optimierungen vorstellbar. Ein Vergleich der Messergebnisse verschiedener Szenarien (wie z.B. zwischen Abbildung 40 und Abbildung 41) zeigt jedoch, dass Optimierungen nur auf ein gewisses Szenario bezogen möglich sind und keine globale Optimierung möglich ist. In Kapitel 6, Abschnitt 1 werden die Parameter und deren Auswirkung auf die Leistungsfähigkeit der betrachteten Sicherheitsarchitektur beschrieben. Dort wird auch ein Verfahren zur dynamischen Parameteroptimierung angesprochen.

Die vorgestellten Anwendungsszenarien sind im Bezug auf Dynamik und Verteilung der Knotendichte sehr unterschiedlich. Trotzdem ist die Verfügbarkeit bei beiden Szenarien nach einer kurzen Aufbauphase hoch. Die Messung des Overheads zeigte in beiden Szenarien, dass die betrachtete Sicherheitsarchitektur sparsam mit den zur Verfügung stehenden Ressourcen umgeht. Bei der Anmeldung wird in der Realität die meiste Zeit durch die Identifizierung durch einen Menschen, z.B. mittels Personalausweis, benötigt, so dass die gemessene Log-On-Zeit tolerierbar ist. Zusammenfassend lässt sich sagen, dass sich die Sicherheitsarchitektur

in ganz unterschiedlichen Szenarien mit verschiedenen Bewegungen als stabil erwiesen hat, und somit global einsatzfähig ist.

Anhand einer kleinen Anzahl von Parametern wie z.B. Clustergröße und CH-Beacon-Intervall können je nach Einsatzgebiet lokale Optimierungen vorgenommen werden, welche die Performance der Sicherheitsarchitektur in kleinem Ausmaß noch optimieren können.

6. Optimierungen

Die Messungen haben gezeigt, wie viel Einfluss eine gute Parameterwahl auf die Leistungsfähigkeit der entwickelten Sicherheitsarchitektur für mobile Ad-hoc-Netze hat. In diesem Kapitel wird deshalb auf den Aspekt der Optimierung genauer eingegangen.

6.1 Optimale Parameterwahl

In der Simulationsimplementierung der Sicherheitsarchitektur steht eine Anzahl von Parametern zur Konfiguration zur Verfügung. Nachfolgend werden diese Parameter und Optimierungsmöglichkeiten beschrieben:

- *CH_Wait:*

  Dieser Parameter bestimmt die Dauer, die ein neu hinzukommender Knoten wartet bevor er sich selbst zum Clusterhead erklärt, wenn er vorher nicht bereits ein CH-Beacon empfangen hat. CH_Wait sollte mindestens so groß sein wie CH_Intervall, sonst besteht die Gefahr, dass sich ein Knoten selbst zum CH erklärt, obwohl er eigentlich einen anderen Clusterhead hören könnte. Die Performance der Sicherheitsarchitektur wird durch diesen Parameter nur in der initialen Entstehungsphase des Gesamtnetzes wesentlich beeinflusst. Ansonsten spielt CH_Wait eher eine untergeordnete Rolle. In den Simulationsläufen hat sich CH_Wait=CH_Intervall bewährt.

- *CH_Wait_Rand:*

  CH_Wait_Rand gibt ein Intervall an, aus dem zufällig ein Wert zu CH_Wait addiert wird, um die tatsächliche Zeit zu ermitteln, die ein Knoten auf einen CH-Broadcast wartet. Der Wert sollte nicht zu klein gewählt werden, da durch ihn sichergestellt wird, dass sich nicht mehrere Knoten gleichzeitig zum CH ernennen, obwohl ein CH genug wäre. Besonders beim initialen Aufbau des Gesamtnetzes spielt dieser Parameter eine wichtige Rolle. In den Simulationsläufen hat sich CH_Wait_Rand=10 s bewährt.

- *CH_Intervall:*

Dieser Parameter gibt an, wie oft CH-Beacons versendet werden. Wie wichtig eine geeignete Wahl von CH_Intervall ist, wurde bereits in Kapitel 4 erwähnt. Wird der Parameter CH_Intervall zu klein gewählt, so kommt es auch schneller zu Cluster-Vereinigungen, welche einen hohen Overhead auslösen. Wählt man hingegen den Parameter CH_Intervall zu groß, so benötigen Knoten länger für die Anmeldung und bemerken erst später den Verlust ihres Clusterheads.

- *CH_lost:*

CH_lost gibt die Zeitspanne an, die ein Knoten nach dem Empfang eines CH-Beacons auf ein weiteres CH-Beacon wartet, bevor er annimmt, dass er seinen Clusterhead verloren hat. Besonderen Einfluss hat dieser Parameter während der Anmeldung auf die Leistungsfähigkeit der Sicherheitsarchitektur. Bewegt sich ein Knoten während der Anmeldung aus dem Cluster des angefragten CHs heraus, so erkennt der Knoten dies unter Umständen erst nach CH_lost. Es bietet sich an, CH_lost=2*CH_Intervall zu wählen, um ein versäumtes CH-Beacon zu tolerieren. Die dadurch entstehenden Nachteile bei der Anmeldung können durch einen geschickt gewählten Parameter logOnTimeout ausgeglichen werden (siehe dort).

- *GW_Intervall* und *GW_lost:*

Wie CH_Intervall und CH_lost, allerdings für Gateways. Diese Parameter spielen für die Performance der Sicherheitsarchitektur eine untergeordnete Rolle.

- *minWarrants:*

Mit diesem Parameter wird die minimale Anzahl an nötigen Bürgen im Cluster festgelegt. Der Clusterhead kann einen beliebigen Wert festsetzen. MinWarrants sollte nicht zu groß gewählt werden, da die Anforderung von Bürgenzertifikaten Zeit und Ressourcen kostet. Aber auch ein zu kleiner Wert ist gefährlich, da die Sicherheit der Architektur auf dem Bürgenprinzip beruht. In der Simulation bewährte es sich, minWarrants auf 40% der An-

zahl der vollen Mitgliedern zu setzen. Wird dieser Parameter dynamisch bestimmt, dann ist es sinnvoll eine untere Grenze festzulegen, die nicht unterschritten werden darf.

- *Schwellwert:*

Der Schwellwert gibt an, wie viele Teilzertifikate ein Knoten von den Clusterheads braucht, bevor er sein identitätsbezogenes Schlüsselzertifikat zusammensetzen kann. Für jedes Teilzertifikat muss ein anderer Clusterhead des gleichen CH-Netzwerks angefragt werden. Ein zu hoher Schwellwert erhöht also den Overhead, der über mehrere Cluster geleitet werden muss. Natürlich muss der Schwellwert unter der maximalen Anzahl von Clusterheads bleiben. Zudem zeigte Abschnitt 4.3, dass die Anzahl der Clusterheads in der Simulation sehr begrenzt ist. In der Simulation hat es sich bewährt, einen Wert von 50% der CHs im Clusterhead-Netzwerk zu wählen.

- *logOnTimeout:*

Dieser Parameter legt die Zeitdauer fest, die eine Anmeldung am Clusterhead maximal dauern darf. Der Parameter ist maßgeblich für die Anmeldezeit. Er sollte nicht zu klein gewählt werden, da sich sonst unter Umständen Knoten nicht anmelden können, die nur wenige Bürgen in unmittelbarer Umgebung zur Verfügung haben und deshalb erst mühsam Bürgen suchen müssen. Wird logOnTimeout jedoch zu groß gewählt, so bricht ein Knoten einen aussichtslosen Anmeldeversuch zu spät ab.

- *maxHops:*

Dieser Parameter legt die Clustergröße durch die Anzahl der Hops, die ein CH-Beacon geflutet wird, fest. Bereits in Kapitel 5 wurde dieser Parameter genauer untersucht. MaxHops bietet dem Clusterhead die Möglichkeit, bei Überlastung Knoten auszuschließen um seine Last zu verringern. Dazu verringert er die Anzahl maxHops. Ein Knoten erkennt den Verlust seines Clusterheads spätestens nach CH_lost (siehe dort). War der Knoten bereits angemeldet, so bleibt sein Zertifikat im CH-Netzwerk weiter bestehen und er sucht sich einen neuen Clusterhead.

- *min_Integrate:*

  Bei der Clustervereinigung spielt dieser Parameter eine wichtige Rolle. Durch min_Integrate wird festgelegt, wie viele Knoten ein Clusterhead in seinem Cluster bereits haben muss, damit er ins Clusterhead-Netzwerk integriert wird. Hat ein CH weniger Knoten im Cluster, so gibt er seine CH-Tätigkeit auf (siehe Kapitel 3, Abschnitt 3.5). In der Simulation hat es sich bewährt, minWarrants und min_Integrate gleich zu wählen. Dadurch werden nur Clusterheads zur Aufgabe ihrer Tätigkeit gezwungen, falls der zugehörige Cluster nicht genug Mitglieder hat, um eine Authentisierung im lokalen Cluster zu ermöglichen.

Globale Aussagen zur idealen Parameterwahl sind nicht möglich. Dies hat Abschnitt 4.3 deutlich gezeigt. Ist allerdings das zu optimierende Szenario bekannt und liegt ein Bewegungs- oder sogar ein Spurmodell vor, so kann die Sicherheitsarchitektur durch die oben beschriebenen Parameter schnell an die gegebenen Umstände angepasst und dafür optimiert werden. Alternativ ist es auch möglich, die Parameter dynamisch an die aktuellen Netz-Charakteristiken anzupassen. Die dazu nötige Kommunikation kann zeitgleich mit der Schlüsselauffrischung im CH-Netzwerk geschehen. In den Schlüsselauffrischungsnachrichten teilen die CHs den anderen CHs im Clusterhead-Netzwerk folgende Informationen aus ihrem Cluster mit:

$n_i =$ Anzahl von direkt benachbarten Clustern. Diese können über die erhaltenen GW-Beacons ermittelt werden.

$g_i =$ Anzahl von Gateways zu direkt benachbarten Clustern ($n_i \leq g_i$). Die Werte werden ebenfalls über die erhaltenen GW-Beacons ermittelt.

$c_i =$ Anzahl Clients im eigenen Netz ($c_i \geq g_i$). Der Clusterhead verfügt über eine Liste von Knoten im eigenen Cluster. Diese Liste propagiert er auch im CH-Beacon.

Jeder Clusterhead erhält von den anderen CHs im Clusterhead-Netzwerk die oben genannten Werte und berechnet daraus Durchschnittswerte:

$\overline{n} = \frac{\sum_{i=1}^{n} n_i}{n}$ durchschnittliche Anzahl von Nachbarn,

$\overline{g} = \frac{\sum_{i=1}^{n} g_i}{n}$ durchschnittliche Anzahl von Gateways,

$\overline{c} = \frac{\sum_{i=1}^{n} c_i}{n}$ durchschnittliche Anzahl von Clients pro CH.

Daneben ist natürlich auch die Zahl der Clusterheads im CH-Netzwerk bekannt (n).

Mit diesen Durchschnittswerten können verschiedene Werte berechnet werden, anhand derer die Parameter zu optimieren sind. So gibt zum Beispiel $\frac{\overline{c}}{\overline{g}}$ die Gatewaybelastung an. Je stärker die Gateways belastet werden, desto wahrscheinlicher sind Probleme bei der Weiterleitung von Paketen. Ist die Gatewaybelastung groß, muss es das Ziel sein, möglichst viel Verkehr innerhalb des eigenen Clusters zu halten. Dies kann z.B. dadurch erreicht werden, dass der Wert minWarrant (siehe oben) an die Zahl der Knoten (und damit potentiellen Bürgen) im eigenen Cluster angepasst wird bzw. mehr Bürgen-Autorisierungszertifikate ausgegeben werden.

Einen Hinweis auf den Zusammenhalt des Netzes gibt der Zusammenhaltindikator $\frac{\overline{g}}{\overline{n}}$. Ist dieser Wert nahe eins, d.h. viele Cluster sind nur über ein Gateway zu erreichen, so ist wegen der Dynamik in Ad-hoc-Netzen die Wahrscheinlichkeit einer Netzpartitionierung in naher Zukunft hoch. Empfängt ein Clusterhead in dieser Situation einen noch unbekannten CH zum ersten Mal, dann ist es sinnvoll, mit einer Clusterhead-Integration oder CH–Vereinigung erst noch zu warten um unnötigen Aufwand zu vermeiden.

Die Sichtweite $\frac{\overline{n}}{n}$ gibt an, wie viele Nachbarcluster über nur einen Hop zu erreichen sind. Ist dieser Wert groß, so kann minWarrant und der Schwellwert hoch eingestellt werden, da ja viele Cluster bereits in wenigen Hops erreichbar sind. Ist der Wert klein, so handelt es sich um ein eher lang gestrecktes CH-Netzwerk. Abbildung 56 zeigt Beispiele für CH-Netzwerke mit hoher und niedriger Sichtweite

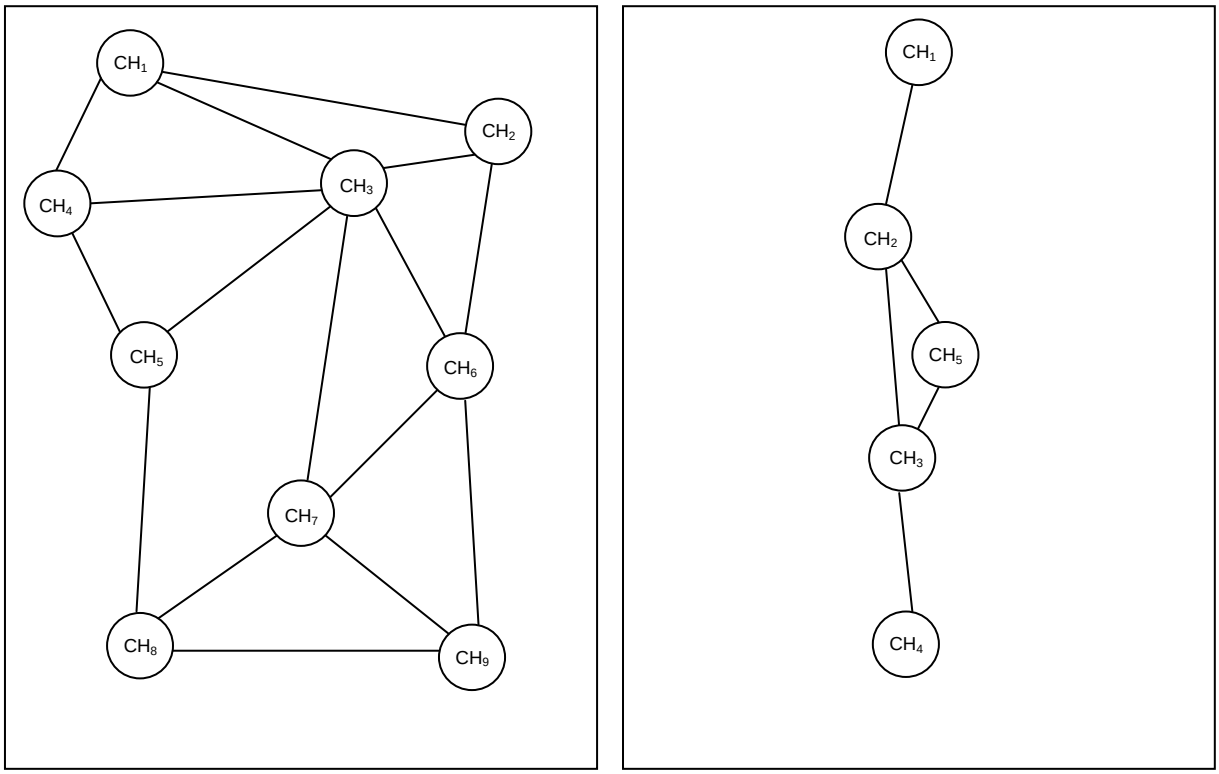

Abbildung 56:links hohe Sichtweite rechts niedere Sichtweite

6.2 Routing

Routing beschäftigt sich allgemein damit, Pakete in einem Netz von einem Startpunkt zum Ziel über eine ideale oder annähernd ideale Route zu transportieren. Was eine ideale Route ist wird durch eine Metrik festgelegt. Routing in Ad-hoc-Netzen stellt ein großes Problem dar, bedingt durch den dynamischen Charakter des Netzes. Noch schwieriger ist es, mobile Routing-Verfahren auch sicher gegen Angriffe zu gestalten. In der hier vorliegenden Arbeit wird das Fisheye-State-Routing-Protokoll ([PEI00]) eingesetzt. Dieses ist nicht gegen Angriffe abgesichert. Es bietet sich an, in dieser Sicherheitsarchitektur auf andere, sichere, Routingver-

fahren zurückzugreifen. Beispiele für sichere Routingprotokolle sind Ariadne ([HU01]) und SEAD (Secure Efficient Distance Vector Routing [HU02]).

Bei Routing Verfahren unterscheidet man grundsätzlich zwischen reaktiven Verfahren und proaktiven Verfahren. Das oben genannte SEAD ist ein proaktives Verfahren während Ariadne reaktiv arbeitet.

In der hier vorgestellten Sicherheitsarchitektur ist bereits durch die Knoten des Clusterhead-Netzwerks eine Cluster-Struktur vorgegeben. Diese Struktur soll beachtet und genutzt werden. Es bietet sich also an, ein clusterbasiertes Routing-Verfahren einzusetzen.

Aber nicht nur die Clusterstruktur kann ausgenutzt werden. Auch der aktuelle Status in der Sicherheitsarchitektur (kein Mitglied, Gast, volles Mitglied, Gateway oder Clusterhead) kann zum Routing herangezogen werden. Gateways können anhand des Status eines Knotens entscheiden, ob sie alle Pakete, nur bestimmte Pakete oder keine Pakete des Senders weiterleiten. Der Status der Knoten im eigenen Cluster wird im CH-Broadcast regelmäßig mitgeteilt und ist somit clusterweit bekannt. Zum Schutz der Sicherheitsarchitektur sollten alle teilnehmenden Knoten nur den Anmeldeverkehr von Gastmitgliedern weiterleiten, den Verkehr von Nichtmitgliedern blocken und den Verkehr von CHs (im Rahmen der Kommunikation im CH-Netzwerk) bevorzugen. Gateways auf dem Weg einer Nachricht kennzeichnen das weitergeleitete Paket entsprechend, um auch späteren, nicht zum Cluster gehörigen Knoten auf der Route eine Einordnung des Pakets zu ermöglichen.

Zusammenfassend ist es sinnvoll, die Sicherheitsarchitektur um eine Routing-Komponente zu erweitern. Derzeit wird in dem vorgestellten Sicherheitskonzept auf bereits vorhandene Verfahren aufgesetzt, ohne die durch die Sicherheitsarchitektur gegebene Struktur auszunutzen. Intelligentere Gateways mit Filterregeln können die Sicherheit der Architektur nachhaltig verstärken.

7. Zusammenfassung und Ausblick

In dieser Arbeit wird eine Sicherheitsarchitektur für mobile Ad-hoc-Netze näher untersucht. Zwei typische Anwendungsszenarien werden gefunden, die sich grundlegend voneinander unterscheiden und versuchen, möglichst viele Problemfelder abzudecken. Das Autobahn-Szenario zeichnet sich durch die verschiedenen Geschwindigkeitsprofile der Teilnehmer (langsame PKWs, schnelle PKWs und LKWs) aus, während im Konferenz-Szenario die unterschiedliche Teilnehmerdichte im Simulationsgebiet eine Herausforderung darstellt. Die Arbeit stellt Erweiterungen und Klarstellungen des Entwurfs der betrachteten Sicherheitsarchitektur vor. So wird ein Verfahren zur Authentisierung vorgeschlagen, und der Ablauf der Clusterhead-Netzwerk-Vereinigung konkretisiert. Die betrachtete Sicherheitsarchitektur für mobile Ad-hoc-Netze wurde im Rahmen dieser Arbeit in der Simulations-Umgebung Omnet++ mit Hilfe eines Ad-hoc-Simulators implementiert. Dabei wurde besonderer Wert auf realistische Bewegungen der Knoten gelegt. Um die Anwendungsszenarien zu implementieren, wurden einige gebräuchliche Bewegungsmodelle vorgestellt. Aus diesen wurden Bewegungsmodelle für die Anwendungsszenarien abgeleitet. Das Random-Waypoint-Modell kommt zusätzlich zum Einsatz, um die Arbeit mit anderen Arbeiten vergleichbar zu machen.

Anhand der Simulation und der Szenarien testete diese Arbeit die Implementierung der Sicherheitsarchitektur und arbeitete Charakteristiken heraus. In allen Versuchen gliedert sich die Verfügbarkeit der Sicherheitsarchitektur in eine Aufbauphase und eine Phase der Funktionalität. Während der Funktionalitätsphase wurde in allen Versuchen eine sehr hohe Verfügbarkeit von über 90% festgestellt. Den Knoten in der Simulation ist es möglich, sich in kurze Zeit am Netz anzumelden. Die Ergebnisse unterscheiden sich nur gering bei Verwendung der verschiedenen Bewegungsmodelle. Auch der Overhead, der durch die betrachtete Sicherheitsarchitektur erzeugt wird, hält sich in Grenzen. Die Architektur geht sparsam mit den zur Verfügung stehenden Ressourcen um. Aus all diesen Ergebnissen wird deutlich, dass die betrachtete Sicherheitsarchitektur global einsatzfähig ist. Ebenso wird die Leistungsfähigkeit gezeigt. Allerdings zeigten die Versuche, dass durch geschickte Parameterwahl noch mehr Leistung erzielt werden kann. Die Größe eines Clusters beeinflusst sie Leistung ebenso wie das Intervall zwischen zwei CH-Beacons. Es wirde deutlich, dass eine optimale Parameterwahl nur lokal, auf ein Szenario bezogen, möglich ist. Die Messergebnisse dient als Basis für Optimie-

rungsvorschläge. Hier wurde der Aspekt der Parameteroptimierung genauer betrachtet und eine dynamische Anpassung der Parameter auf Basis des aktuellen Netzzustands vorgeschlagen.

In der Arbeit wird deutlich, dass in die Sicherheitsarchitektur noch ein sicherer Routing-Mechanismus integriert werden sollte, der die Besonderheiten der Sicherheitsarchitektur sinnvoll nutzt. Es bietet sich im Rahmen einer weiteren Arbeite an, einen solchen sicheren Routing Mechanismus in die Sicherheitsarchitektur zu integrieren und die Leistungsfähigkeit anhand einer Testimplementierung zu untersuchen. Sinnvoll wäre auch eine Erweiterung des Ad-hoc-Simulators, die Kollisionen von Paketen realisiert. Nur so können realistisch Funkverbindungen simuliert werden.

Anhang A: Begriffserklärungen und Details

Die folgenden Begriffserklärungen und Details hätten den Rahmen der vorherigen Kapitel gesprengt und werden deshalb am Ende der Arbeit kurz zusammengefasst.

A.1 Nachrichten der Simulation

Die Implementierung der Sicherheitsarchitektur verwendet eine Reihe von Nachrichten, um den Protokollablauf auszuführen. Im Folgenden werden diese Nachrichten und ihre Bedeutung aufgeführt:

CHBroadcast:	Das CH-Beacon. Diese Nachricht wird von den Clusterheads in regelmäßigen Abständen gesendet und enthält wichtige Informationen über den Cluster sowie Informationen zur Anmeldung im Cluster.
noCHfound:	Zeitgeber-Nachricht des Knotens. Es wurde kein CH-Beacon eines Clusterheads in angemessener Zeit empfangen.
broadcast:	Zeitgeber-Nachricht eines Clusterheads. Sie fordert den Knoten dazu auf, wieder ein CH-Beacon zu versenden.
lostCH:	Zeitgeber-Nachricht eines Knotens, der sich einem Clusterhead zugeordnet hatte. Diese Nachricht tritt auf, wenn der Knoten über eine längere Zeitspanne keine CH-Beacons von seinem Clusterhead empfangen hat. Der Knoten hat seinen Clusterhead verloren.
broadcastGW:	Zeitgeber-Nachricht eines Gateways. Der Knoten wird aufgefordert ein GW-Beacon zu versenden.
GWBroadcast:	GW-Beacon. Diese Nachricht wird periodisch von allen Gateways versendet. Sie enthält Informationen über den Zustand des Gateways und die möglichen Weiterleitungen über dieses Gateway.

logOnRequest: Diese Nachricht wird von einem Knoten zu einem Clusterhead geschickt. Der Knoten bittet um Aufnahme in den Cluster.

logOnReply: Die Antwort des Clusterheads auf eine logOnRequest-Nachricht.

warrantRequest: Diese Nachricht wird von einem Knoten an potentielle Bürgen geschickt. Der Knoten bittet um Authentisierung durch den Bürgen.

warrantReply: Die Antwort eines Bürgen auf warrantRequest. Diese Nachricht beinhaltet im Normalfall ein BürgZert und das BürgAutoZert des Bürgen.

certificationRequest: Diese Nachricht wird von einem Knoten an seinen Clusterhead geschickt. Der Knoten bittet um Erteilung eines Zertifikats für seinen öffentlichen Schlüssel und übergibt dem Clusterhead in der Nachricht auch die zuvor gesammelten Bürgen-Zertifikate.

certificationReply: Die Antwort des Clusterhead auf eine certificationRequest-Nachricht. Sie enthält bei erfolgreicher Überprüfung der Bürgen-Zertifikate und Bürgen-Autorisierungszertifikate ein Identitätszertifikat.

keyRequest: Diese Nachricht wird von einem Knoten an seinen Clusterhead geschickt, wenn der Knoten über einen zertifizierten öffentlichen Schlüssel verfügt. Der Knoten fordert mit dieser Nachricht den symmetrischen Clusterschlüssel vom Clusterhead an.

keyReply: Die Antwort des Clusterhead auf eine keyRequest-Nachricht. Sie enthält den symmetrischen Clusterschlüssel.

logOnTimeout: Zeitgeber-Nachricht eines Knotens. Sie zeigt an, dass für den Anmeldevorgang zuviel Zeit vergangen ist und alle Werte wieder zurück gesetzt werden sollen.

notYetLoggedOn: Zeitgeber-Nachricht eines Knotens. Sie reaktiviert den Anmeldevorgang, wenn der Log-On zu lange dauert.

needMoreClients: Diese Nachricht verschickt ein Knoten an einen Clusterhead. Er fordert damit eine Liste von potentiellen Bürgen an, die der Knoten abfragen kann um Bürgen-Zertifikate zu erhalten.

moreClients: Diese Nachricht ist die Antwort auf „needMoreClients“. Wie der Name schon sagt, enthält die Nachricht eine Liste mit weiteren potentiellen Bürgen.

lostGW: Zeitgeber-Nachricht eines Knotens, der zuvor ein Gateway empfangen hatte. Diese Nachricht tritt auf wenn der Knoten über eine längere Zeitspanne keine GW-Beacons von dem Gateway empfangen hat. Die Wahrscheinlichkeit ist dann hoch, dass er das Gateway verloren hat.

uniteNetworks: Diese Nachricht ist eine Aufforderung zwei CH-Netzwerke miteinander zu verschmelzen.

newNetwork Durch diese Nachricht wird allen betroffenen Clusterheads nach einer Vereinigung zweier Clusterhead-Netzwerke das neue Netzwerk mitgeteilt.

refreshCHPrivateKey: Zeitgeber-Nachricht eines Clusterheads die die Schlüsselauffrischung im eigenen CH-Netzwerk anstößt.

keyConstructionRequest: Diese Nachricht fordert zur gemeinsamen Schlüsselgenerierung bzw -auffrischung im Clusterhead-Netzwerk auf.

keyConstructionReply: Ein Clusterhead antwortet mit diese Nachricht auf eine keyConstructionRequest-Nachricht.

IDCertsExpired: Zeitgeber-Nachricht eines vollen Clustermitglieds. Mit dieser Nachricht wird ein abgelaufenes Zertifikat angedeutet.

EncapsulatedSec: Bei dieser Nachricht handelt es sich um Nutzdaten des Application-Moduls, die gesichert übertragen werden.

noLongerCH: Mit dieser Nachricht teilt ein Clusterhead den Knoten in seinem Cluster mit, dass er seine Tätigkeit als Clusterhead einstellt. Dies kann z.B. bei einer Eingliederung in ein CH-Netzwerk geschehen.

refreshYourKey: Zeitgeber-Nachricht eines Knotens. Der Knoten muss seinen Schlüssel neu signieren lassen, da die letzte Signatur nur von begrenzter Gültigkeit war.

A.2 IEEE 802.11x

IEEE 802.11x ist der derzeit am weitesten verbreitete Standard für drahtlose LANs. Es handelt sich dabei um einen Standard für einfache, robuste WLANs (Wireless Local Area Network). Der Standard ermöglicht sowohl asynchrone als auch zeitbeschränkte Dienste.

IEEE 802.11x sieht sowohl infrastrukturbasierte wie auch Ad-Hoc-Netze vor. Es gibt verschiedene Weiterentwicklungen des ursprünglichen IEEE 802.11 Standards:

- IEEE 802.11 realisiert Datenraten von 1 oder 2 MBit/s und ist der ursprüngliche Standard von dem alle weiteren hier erwähnten Standards abgeleitet sind.

- Der IEEE 802.11b-Standard sieht Datenraten von 5,5 bis 11 MBit/s vor. In der Realität werden aber meist nur Datenraten zwischen 4 und 5 MBit/s erreicht. Ebenso wie 802.11 findet die Kommunikation im Frequenzband um 2.4 GHz statt.

- IEEE 802.11a beherrscht nach Definition des Standards Datenraten zwischen 24 und 54 MBit/s. In der Praxis kann man davon ausgehen, dass die Datenrate sich bei 40 % der theoretisch möglichen Rate bewegt. Damit wären also Raten zwischen 10 und 22 MBit/s auch praktisch realisierbar. Diese hohen Datenraten werden unter anderem

durch den Einsatz eines anderen Frequenzband bei 5 GHz ermöglicht. 802.11 und 802.11b setzen im Gegensatz dazu das Frequenzband bei 2,4 GHz ein.

Den IEEE 802.11x-Standards ist allen gemeinsam, dass sich typische Sendereichweiten in der Praxis zwischen 15 und 150 m in Gebäuden und bis zu 300 m im Freien bewegen. Dabei hängt diese Entfernung von der eingesetzten Sendeleistung ab. In den USA sind vom Gesetzgeber maximal 1 W Sendeleistung erlaubt, während in Europa nur 100 mW eingesetzt werden dürfen.

Die Hersteller von Wireless-LAN-Karten geben verschiedene Werte für ihre Produkte an. Die Firma Compaq gibt in [COM02] beispielhaft für ihre Produkt die in Tabelle 2 abgebildeten Reichweiten an. Die Werte sind bezogen auf Karten, die den Standard 802.11b implementieren. Die gleichen Angaben finden sich unter [ORI02].

	11 MBit/s	5,5 MBit/s	2 MBit/s	1 MBit/s
Offene Halle/im Freien	160 m	270 m	400 m	550 m
Halboffenes Büro	50 m	70 m	90 m	115 m
Geschlossener Raum	25 m	35 m	40 m	50 m

Tabelle 2: Reichweite von Compaq-Wireless-LAN-Produkten

Auf der Webpage von Siemens [SIE02] finden sich die in Tabelle 3 abgebildeten Werte für WLAN-Karten. Besonders interessant sind hier die Angaben zu der Reichweite im Ad-hoc-Modus. Auch diese Angaben beziehen sich wieder auf Produkte, die 802.11b implementieren.

	Ad-hoc-Modus	Mit Access Point
Im Freien	100 m	60 Meter Durchmesser pro Accesspoint, durch spezielle Antennen noch zu verbessern
In Gebäuden	30 m	60 Meter Durchmesser pro Accesspoint, durch spezielle Antennen noch zu verbessern

Tabelle 3: Reichweite von Siemens Produkten

Auch CISCO gibt in den Datenblättern zu seinen WLAN Produkten entsprechende Reichweiten an. Die in [CIS02] aufgeführten Werte zeigt Tabelle 4 für IEEE 802.11b-Produkte und Tabelle 5 für IEEE 802.11a-Produkte .

	11 MBit/s	1 MBit/s
Im Freien	244 m	610 m
In Gebäuden	40 m	107 m

Tabelle 4: Reichweite 802.11b-Karten von Cisco

	54 MBit/s	18 MBit/s	6 MBit/s
In Gebäuden	18 – 21 m	40 – 45 m	52-61 m
Im Freien	30 – 36 m	183 – 213 m	304 – 355 m

Tabelle 5: Reichweite 802.11a-Karten von Cisco

Alle in diesem Kapitel gemachten Angaben sind als obere Grenze zu verstehen.

A.3 Bluetooth

Bluetooth ist ein Kommunikationsstandard, der besonders für batteriegetriebene Geräte mit geringer Leistung entwickelt wurde. Das Hauptaugenmerk beim Entwurf lag auf der Entwicklung eines preisgünstigen Chips, der in mobilen Geräten eingesetzt werden kann. Bluetooth setzt auf Ad-hoc-Netze und ist nicht wie IEEE 802.11 sowohl für infrastrukturbasierte wie auch Ad-hoc-Netze ausgelegt.

Bei Bluetooth werden 1600 Frequenzsprünge pro Sekunde realisiert, um eine geringe Störanfälligkeit zu erreichen. Dadurch wird aber die maximale Blocklänge begrenzt, und der erzeugte Overhead kann die Datenübertragung bremsen.

Bluetooth-Geräte bilden Piconetze mit jeweils maximal acht Teilnehmern. Sogenannte Scatternets entstehen als Zusammenschluss mehrerer Piconetze. In Piconetzen kann ein Knoten

einen Datenstrom mit 1 MBit/s senden, der von allen Knoten empfangen wird. Jeder andere Datenverkehr läuft über den Master des jeweiligen Netzes. Der Master ist ein herausgehobener Knoten, der sein Piconetz organisiert. Er kann synchrone oder asynchrone Verbindungen aufbauen. Bei asynchronen Verbindungen sind Datenraten von 721 KBit/s (Rückrichtung 57 KBit/s) möglich. Synchrone Verbindungen erlauben 432,6 KBit/s. Allerdings bewegen sich realistische Datenübertragungsraten im Feldeinsatz zwischen 70 und 400 KBit/s.

Bei Bluetooth werden meist nur sehr kleine Sendeleistungen im Bereich von 1 mW eingesetzt. Dadurch ist die maximale Reichweite auf ca. 10 m begrenzt. Der Bluetooth-Standard sieht allerdings Sendeleistungen bis 100 mW vor, mit denen Kommunikation im Umkreis von bis zu 100 m zumindest theoretisch möglich ist.

A.4 Benutzerschnittstellen

In diesem Kapitel wird die Benutzerschnittstelle der Simulation beschrieben. Die Implementierung greift dabei auf die Standard-Benutzerschnittstellen von Omnet++ zurück. Diese ermöglicht eine Kommandozeilen-Version für den Batch-Betrieb und eine grafische Simulation zu Präsentationszwecken.

Batch-Betrieb: Die Anweisung #define DEBUG im Quellcode schaltet die Ausgabe für den Batchbetrieb an. Dies kann für jedes Modul einzeln aktiviert oder deaktiviert werden. In der endgültigen Fassung ist die Ausgabe für alle Module aktiviert. Zur Fehlersuche empfiehlt es sich, die Ausgabe nur eines Moduls zu aktivieren um nicht vom wesentlichen abgelenkt zu werden.

GUI: Die Standard-Omnet++-GUI bietet eine Vielzahl von Optionen. Abbildung 57 zeigt das Kontrollfenster der Simulation. Im unteren Teil findet die textuelle Ausgabe statt. Schaltfläche 6 aktiviert die grafische Ausgabe. Ein Screenshot wird in Abbildung 58 gezeigt.

Simulationen können in verschiedenen Geschwindigkeiten ausgeführt werden (Schaltflächen 1-4). Schaltfläche 5 ermöglicht es, eine Simulation bis zu einem bestimmten Zeitpunkt auszuführen. In der Netzvisualisierung (aktiviert durch Schaltfläche 6, Abbildung 58) können die Module eines Knotens durch Doppelklicken inspiziert werden. Eine ausführlichere Beschreibung findet sich in [VAR01].

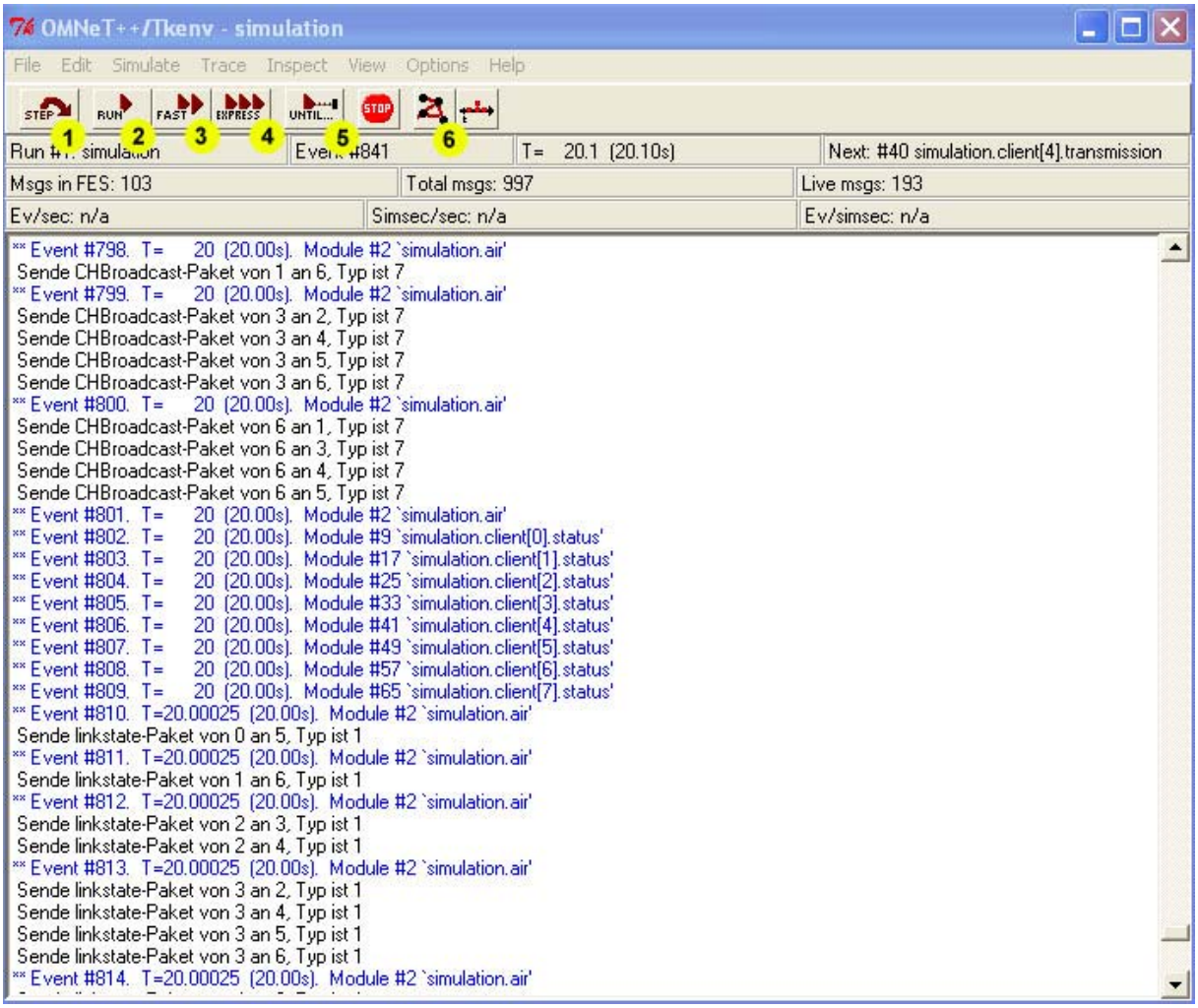

Abbildung 57: Screenshot Simulation

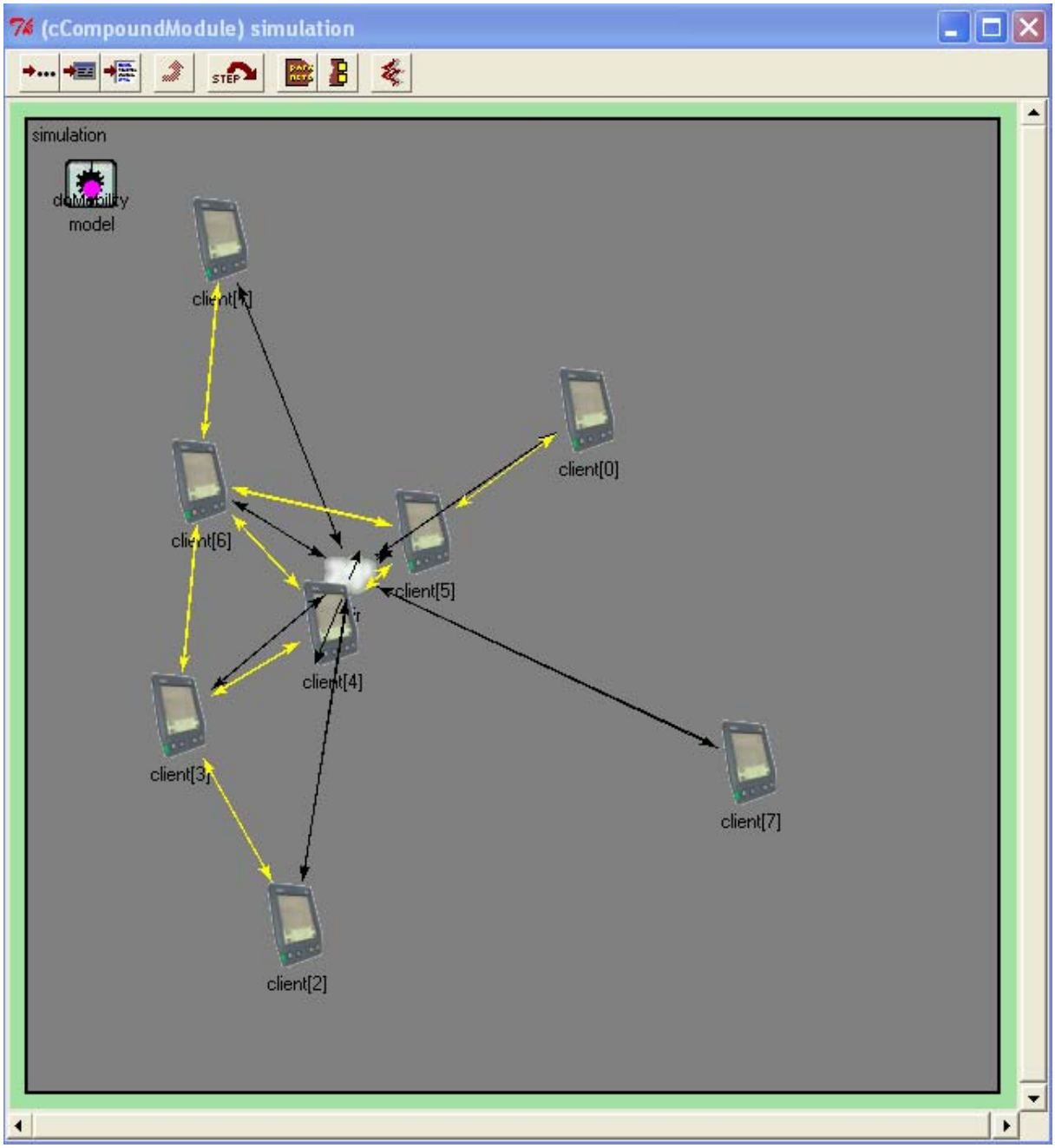

Abbildung 58: Screenshot Netzvisualisierung

In der Netzvisualisierung (Abbildung 58) werden folgende Symbole für die Knoten eingesetzt:

Mobiler Knoten, der noch nicht für die Simulation aktiviert ist (z.B. Knoten mit verzögertem Systemstart).

Mobiler Gastknoten. Dieser Knoten hat sich noch nicht angemeldet bzw. befindet sich zur Zeit noch im Anmeldevorgang.

Dieser Knoten ist volles Clustermitglied. Der kleine grüne Schlüssel unter dem Knoten deutet an, dass der Knoten im Besitz des symmetrischen Clusterschlüssels ist.

Der Knoten arbeitet als Gateway und leitet Datenpakete weiter.

Der Knoten arbeitet als Gateway und ist in mindestens einem Cluster volles Mitglied. Der kleine grüne Schlüssel erinnert wieder daran, dass dieser Knoten im Besitz des symmetrischen Clusterschlüssels ist.

Der Knoten arbeitet als Clusterhead (CH).

Der Knoten arbeitet als Clusterhead (CH) und gleichzeitig als Gateway (GW)

Zwischen den Knoten können Verbindungen bestehen. Dabei geben die schwarzen Verbindungen in Abbildung 58 den Aufbau der Simulation wieder (d.h. sie verbinden jeweils das Air-Modul mit einem Knoten), während die gelben Verbindungen die physikalischen Verbin-

dungen in der simulierten Umgebung darstellen. Farblich wird also das entstehende Netz durch die Aneinanderreihung der gelben Pfeile visualisiert.

Nachrichten werden in der Simulation als kleine Kreise dargestellt, die auf den Verbindungen vom Sender zum Empfänger bewegt werden. Unterhalb des Kreises wird jeweils der Name der Nachricht angegeben. Nachrichten werden immer auf den schwarzen Verbindungen bewegt, niemals auf den gelben Verbindungen, denn die schwarzen Verbindungen spiegeln den Aufbau in Omnet++ wieder, während die gelben Verbindungen in Omnet++ keine Funktionalität haben. Der Kreis jeder Nachricht hat eine andere Farbe, die den Typ der Nachricht symbolisiert.

grün	=	Nachricht vom Routing-Modul
schwarz	=	Nachricht vom Sec-Modul
lila	=	Nachricht vom Bewegungsmodul

Anhang B: Abbildungs- und Tabellenverzeichnis

Anhang C: Literaturverzeichnis

[BAL02] Dirk Balfanz, D. K. Smetters, Paul Stewart und H. Chi Wong: „Talking To Strangers: Authentication in Ad-Hoc Wireless Networks“, Symposium on Network and Distributed Systems Security (NDSS'02), 2002, Xerox Palo Alto Research Center

[BAN01] Jörg Banholzer: „Entwurf und Implementierung einer Simulationsumgebung für mobile Ad-hoc-Netzwerke“, Diplomarbeit am Institut für Telematik, 2001, Universität Karlsruhe (TH)

[BAR95] A. Bar-Noy, I. Kessler, and M. Sidi: „ „Mobile Users: To Update or Not to Update?“, Wireless Networks Journal Vol. 1, 1995

[CAP02] Srdjan Capkuny, Levente Buttyan and Jean-Pierre Hubaux: “Self-Organized Public-Key Management for Mobile Ad Hoc Networks”, Technical Report ,2002, Laboratory for Computer Communications and Applications (LCA) Faculty of Informatics and Communication (I&C) und Swiss Federal Institute of Technology Lausanne (EPFL)

[CHI98] C.-C. Chiang: „Wireless Network Multicasting“, Dissertation am Department of Computer Science, 1998, University of California Los Angeles,

[CIS02] Cisco, http://www.cisco.com/en/US/products/hw/wireless/ps4555/products_data _sheets_list.html, 2002

[COM02] Compaq, http://www.compaq.de/produkte/wireless/infos/reichweite.htm, 2002

[DAV00] V. Davies: „Evaluating mobility models within an ad hoc network“, Diplomarbeit , 2000, Colorado School of Mines

[DIN02] Frank Dinies : „Implementierung des Fisheye-State-Routings zur Simulation von Ad-hoc-Netzen“, Studienarbeit am Institut für Telematik, 2002, Universität Karlsruhe (TH)

[FRO94] V. S. Frost und B. Melamed: „Traffic Modeling for Telecommunications Networks“, IEEE Communications Magazine Vol.32 No.3, 1994

[GER02] Mario Gerla, Xiaoyan Hong, Guangyu Pei: „Fisheye State Routing Protocol (FSR) for Ad Hoc Networks“, Internet Draft, 2002, http://www.ietf .org/inter net- drafts/ draft-ietf-manet-fsr-03.txt

[HAA97] Z.J. Haas : „A new routing protocol for the reconfigurable wireless networks“, IEEE 6th International Conference on Universal Personal Communications, 1997, Cornell University Ithaca

[HAA99] Z. Haas und B. Liang: „Predictive distance-based mobility management for PCS networks“,Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 1999, Cornell University Ithaca

[HAA00] Z.Haas: „The Zone Routing Protokoll (ZRP) for Ad Hoc Networks“, Internet Draft, 2000, Cornell University Ithaca

[HAU01] Achim Hauck: „Entwurf eines Sicherheitskonzepts für mobile Ad-hoc-Netzwerke“, Diplomarbeit am Institut für Telematik , 2001, Universität Karlsruhe (TH)

[HER96] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk: „Proaktive Public Key and Signature Systems“, Internet Draft, 1996, IBM Research Haifa

[HON99] X. Hong, M. Gerla, G. Pei, und C. Chiang: „A group mobility model for ad hoc wireless networks“, In Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), 1999, IBM Research Haifa

[HON01] Xiaoyan Hong, Taek Jin Kwon, Mario Gerla, Daniel Lihui Gu, Guangyu Pei: „Mobility Framework for Ad Hoc Wireless Networks“, Lecture Notes in Computer Science Vol. 1987, 2001, University of California, Los Angeles

[HU01] Yih-Chun Hu, Adrian Perrig und David B. Johnson: „Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks“, Technical Report am Department of Computer Science, 2001, Rice University

[HU02] Yih-Chun Hu, David B. Johnson, Adrian Perrig : „SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks“, Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), 2002, Rice University

[JOH96] D. B. Johnson, D. A. Maltz: „Dynamic Source Routing in ad hoc wireless networks“ , Mobile Computing, 1996, Kluwer Academic Publishers

[KWO99] Taek Jin Kwon und Mario Gerla: „Clustering with Power Control“, in IEEE MILCOM, 1999, Computer Science Department University of California, Los Angeles

[LEU94] Leung, K. K., W.A. Massey, und W. Witt: „Traffic Models for Wireless Communication Networks“, IEEE Journal on Selected Areas in Communication Vol. 12 No.8, 1994

[MAR97] J. Markoulidakis, G. Lyberopoulos, D. Tsirkas, und E. Sykas: “Mobility Modeling in Third-Generation Mobile Telecommunications Systems“, IEEE Personal Communications 4(4), 1997

[ORI02] Orinoco, ftp://ftp.orinocowireless.com/pub/docs/ORINOCO/BROCHURES/Proxim/World%20PC%20Card%20US.pdf, 2002

[PAR97] Vincent D. Park and M. Scott Corson: „A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks“, In Proceedings of 1997 IEEE Conference on Computer Communications (Infocom'97), 1997

[PEI00] Guangyu Pei, Mario Gerla, Tsu-Wei Chen: „Fisheye State Routing in Mobile Ad Hoc Networks“, In Proceedings of the 2000 ICDCS Workshops, 2000, Taipei (Taiwan)

[PER94] Charles E. Perkins und Pravin Bhagwat: „Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers“, In Proceedings of the SIGCOMM ´94 Conference on Communications Architectures, Protocols and Applications, 1994

[ROY01] E. Royer, P. M. Melliar-Smith, und L. Moser: „An analysis of the optimum node density for ad hoc mobile networks“, In Proceedings of the IEEE International Conference on Communications (ICC), 2001, Helsinki (Finland)

[SAN02] Miguel Sanchez : „Mobility Models“, , http://www.disca.upv.es/ msan/mob model.htm , 2002

[SHA79] A. Shamir: „How to share a Secret“, Communication of the ACM Volume 22, Number 11, 1979

[SIE02] Siemens, http://www.mysimens.com/MySiemens/ CDA/Index/0,2730,HQ_en_0_product%253AHO%252FNW%252FNWG%252FIGTE11MPCCARD%252F tech,FF.html,2002

[STA99] F. Stajano und R. J. Anderson : „The resurrecting duckling: Security issues for ad-hoc wireless networks“, In Proceedings of the 7th International Workshop on Security Protocols, Lecture Notes in Computer Science , 1999, Springer Verlag, Berlin

[THO88] Thomas, R.,H. Gilbert, and G. Mazziotto: „Influence of the Moving of the Mobile stations on the Performance of a Radio Mobile Cellular Network“, In Proceedings of 3rd Nordic Seminar, 1988, Copenhagen

[TOL99] V. Tolety : „Load reduction in ad hoc networks using mobile servers“, Diplomarbeit , 1999, Colorado School of Mines

[VAR01] András Varga: „Omnet++, Discrete Event Simulation System“, Dissertation am Department of Telecommunications, 2001, Technical University of Budapest

[YAS02] Alec Yasinsac, James A. Davis: “Modeling Protocols for Secure Group Communication in Ad Hoc Networks (Extended Abstract)”, Technical Report ,2002, Florida State University

[ZAR02] Magda El Zarki, Sharad Mehrotra, Gene Tsudik und Nalini Venkatasubramanian: „Security Issues in a Future Vehicular Network“, EuroWireless 2002, 2002, University of California Irvine

[ZHU02] Feng Zhu: „Security for Ad hoc Networks“, 2002, http://www.ccs.neu.edu/home/zhufeng/security_manet.html