Diplom

Sicherheit und Schutz von Wireless Mesh Netzwerken

Jens Mätschke
(Matrikelnummer: 2550063)

Studiengang: Diplom-Informatik an der TU Dresden
Betreuer: Dipl.-Inform. Stephan Groß
Verantwortlicher Hochschullehrer: Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill
Institut: Systemarchitektur

Zeitraum: 15. Juni 2005 bis 15. März 2006

Aufgabenstellung für die Diplomarbeit

Der schon länger anhaltende Erfolg von Wireless Netzwerken beruht auf dem Aufbau einer Funk basierten Architektur, welche das immer größer werdende Verlangen unserer mobilen Computer nach ständiger Internet/ Datenverfügbarkeit von überall aus, erfüllt. Von Wireless Mesh Netzwerken wird erwartet, dass sie eine Schlüsseltechnik der Zukunft werden. Sie sind entscheidend für großräumige Wireless Netzwerke ohne existierende Infrastruktur. Bisher ist jedoch noch unklar, wie ein vertrauensvolles Netzwerk in einer multihop adhoc Umgebung, wie im Wireless Mesh Netzwerk, geschaffen werden kann.

Ziel der Diplomarbeit ist die Entwicklung einer geschützten Wireless Mesh Architektur. Der Schutz betrifft sowohl die Inhalts- als auch Verbindungsdaten der Nutzer sowie die Verfügbarkeit des Netzwerkes für alle Berechtigten zu jeder gewünschten Zeit. Ausgehend von existierenden Mesh Architekturen und Einsatzszenarien soll die Arbeit klären, welche Schutzanforderungen die einzelnen Beteiligten haben und ob bzw. wie diese realisiert werden können. Hierzu sind mögliche Angriffsszenarien zu definieren und entsprechende Schutzmaßnahmen abzuleiten. Von diesen sollen exemplarische Maßnahmen soweit sinnvoll realisierbar in einem Laborversuch demonstriert werden.

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und nur die angegebenen Quellen verwendet habe.

Dresden, den 15. März 2006

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Die Vernetzung mittels Funktechnik boomt in der Computerbranche. Aufbauend auf dem Standard IEEE 801.11 hat es die WLAN-Technik (wireless local area networks) geschafft, sowohl im Geschäfts- als auch im Privatbereich eine der wichtigsten Techniken zur Vernetzung von Computern und anderen meist mobilen Endgeräten zu werden.

Wie eine Erhebung des Research-Unternehmens IDC belegt, sind bereits mehr als 25.000 Hotspots an öffentlichen Orten wie Bahnhöfen, Hotels, Restaurants u.s.w. in Europa installiert. Insgesamt über 50 Provider sollen die Zahl der Hotspots in den kommenden Jahren in Europa auf 110.000 erhöhen [1]. Neben den öffentlichen Zugangstechniken über einen Access Point werden z.B. durch DSL-WLAN-Router immer mehr funkbasierte Netzwerke im Firmen- und Privatbereich eingesetzt. In fast jedem neu verkauften Laptop ist eine WLAN-Netzwerkkarte installiert.

Den Vorteilen von Wireless LAN, wie der einfachen Installation, geringen Kosten im Aufbau, Ortsunabhängigkeit oder leichte Integration in bestehende Netzwerkinfrastruktur, stehen einige schwerwiegende Probleme gegenüber. Die Reichweite des populären WLAN Standards 801.11 beträgt durch die geringe Sende- und Empfangsleistung oft weniger als 100m und eine Sichtverbindung zwischen zwei kommunizierenden Netzgeräten wird meist vorausgesetzt. Die Performance in Wireless-Netzwerken sinkt mit der Anzahl der NetzteilnehmerInnen.

Wireless Mesh Netzwerke wurden entwickelt, um diese Einschränkungen in der Skalierbarkeit und damit in der Performance von Ad-hoc-Netzwerken und WLANs zu beheben. In einem Mesh-Netzwerk fungiert jeder Knoten als Router und leitet Daten anderer Knoten weiter. Die Reichweite des Netzes wird somit viel größer. Neben den Vorteilen von Wireless LANs bieten Mesh-Netzwerke eine erhöhte Robustheit und damit Fehlertoleranz, durch redundante Verbindungen ist ein Lastausgleich möglich und das Netzwerkmanagement beschränkt sich im Gegensatz zu anderen Netzwerktopologien auf sehr wenige Einstellungen (Routingprotokoll, IP-Adressbereich). Die Forschung zu Mesh Netzwerken befindet sich noch am Anfang, jedoch kann auf viele Techniken und Forschungsergebnisse von Ad-hoc-Netzwerken zurück gegriffen werden.

Mesh-Netzwerke versprechen, eine funkbasierte Technologie für eine weite Spannbreite von Applikationen zu sein - z.B. Kommunikation bei Rettungseinsätzen, Verkehrswarnmeldungen, Heimvernetzung, Communitynetzwerke, Gebäudeautomatisierung. Die Technik bekommt Bedeutung für Internet Service Provider, Carrier und andere, welche mit möglichst wenig Investitionskosten ein möglichst großflächige und robuste Breitbandversorgung aufbauen wollen. Mit der Eigenschaft der Selbstorganisierung und Selbstkonfiguration können die Netze nach und nach aufgebaut werden - jeder Knoten erst dann, wenn er benötigt wird. Je mehr Knoten installiert sind, desto ausfallsicherer sind die Verbindungen für die Nutzerinnen und Nutzer. Diese Spannbreite und das enorme Potential von Mesh-Netzwerken wird im dritten Kapitel durch aktuelle Einsatzszenarien belegt.

Für Wireless-Mesh-Networking werden viele Implementierungen im gewerblichen Bereich der Überwachungs- und Kontrollapplikationen erwartet. In einer Prognose des Marktforschungsinstituts VDC werden die Umsätze von Produkten, die Mesh-Networking nutzen, von 6,1 Mio. US-Dollar im Jahr 2004 auf 25,1 Mio. US-Dollar im Jahr 2007 steigen. Das entspräche einer jährlichen Wachstumsrate von 60,2 Prozent [2].

In den letzten Jahren entstand gleichzeitig an verschiedenen Orten der Welt eine Bewegung, die sich zum Ziel gesetzt hat, *freie Netze* aufzubauen. In Anknüpfung an die Idee von Freenets, Bürgernetzen, Mailbox-Szene und digitalen Städten geht es darum, dass Bürgerinnen und Bürger die Vernetzung selbst in die Hand nehmen. In dem Mesh Konzept steckt somit auch ein gesellschaftlicher Aspekt. Die Freifunk-Szene war und ist ein Motor für die Entwicklung einfacher Mesh-Protokolle und erster Großversuche, z.B. des *Berlin Backbone* [3].

Ein Manko gibt es jedoch bei allen aktuellen Mesh-Implementierungen und Forschungsarbeiten: Dem Thema Sicherheit wurde nur wenig Aufmerksamkeit geschenkt. Entweder wurden alle Sicherheitsaspekte auf den Nutzer umgelegt (dieser muss verschlüsseln und nur vertrauenswürdige Geräte benutzen) oder nur einzelne Aspekte, wie die Verfügbarkeit von Netzverbindungen durch Fairness aller teilnehmenden Geräte, wurden betrachtet. Die Entwicklung eines grundlegenden Sicherheitskonzeptes für Mesh-Protokolle, welches die Sicherheitsbedürfnisse in konkreten Szenarien erfüllt, ist noch offen. Das Ziel dieser Diplomarbeit ist die Entwicklung einer geschützten Wireless Mesh-Architektur.

1.2 Aufbau der Diplomarbeit

Nachdem in diesem einleitenden Kapitel schon die Bedeutung von Mesh-Netzwerken unterstrichen worden ist, werden im zweiten Kapitel die verwendeten Begriffe geklärt

und die theoretischen Rahmenbedingungen beschrieben. Auf die Mesh-Architektur und das Ad-hoc-Routing wird ausführlicher eingegangen, da dies die Basis für die weiteren Betrachtungen der Sicherheit darstellt. Einige grundlegende Probleme und Besonderheiten, welche im Allgemeinen alle Mesh-Implementierungen betreffen, werden im Abschluss des Kapitels 2 aufgeführt.

Konkrete Mesh-Einsatzszenarien und die Sicherheitsanforderungen aller beteiligten Personen werden im dritten Kapitel analysiert. Dieses geschieht möglichst konkret, damit in der späteren Architekturentwicklung darauf zurückgegriffen werden kann. Da in verschiedenen Szenarien die Sicherheitsanforderungen unterschiedlich ausfallen, wird im Anschluss noch einmal zusammengefasst, inwieweit sich die Anforderungen gleichen und ob eine einheitliche Sicherheitsstruktur möglich ist.

Im vierten Kapitel werden allgemeine Schutzmethoden für die Sicherheitsanforderungen der Beteiligten aus dem vorhergehenden Kapitel erläutert. Die zugrunde liegenden kryptographischen Algorithmen und Protokolle werden benannt. Anschließend wird aufgezeigt, wo die Knackpunkte in der Sicherheit von Mesh-Protokollen liegen und bei welchen Punkten klassische Verfahren aus kabelgebundenen Netzwerken übernommen werden können.

Eine Analyse recherchierter Forschungsarbeiten und Implementierungen von Mesh-Protokollen erfolgt im fünften Kapitel. Der Fokus liegt auf den Schwachstellen eines Mesh-Netzwerkes. Beurteilt wird ebenfalls das Zusammenspiel verschiedener Schutzmechanismen. Im Anschluss wird anhand der Einsatzszenarien bewertet, in welchem Umfang diese bereits abgesichert werden können und welche Punkte der Sicherheitsbedürfnisse weitere Betrachtung erfordern.

Durch die Feststellung, dass die Entwicklung einer sicheren Mesh-Architektur für alle genannten Einsatzbeispiele den Rahmen dieser Arbeit sprengen würde, wird im sechsten Kapitel eine Sicherheitsarchitektur für Community-Netzwerke entwickelt. Bisher sind keine zufriedenstellenden Protokolle für dieses Szenario vorhanden. Nach einer Klärung der Rahmenbedingungen erfolgt die genaue Zielbeschreibung. Als Grundlage für die Protokollentwicklung wird das OLSR-Protokoll und die Software Unik-OLSR verwendet. Letztere beinhaltet eine Plugin-Funktionalität, welche verwendet wird, um nicht nur allgemein die Sicherheitsprotokolle zu beschreiben, sondern am praktischen Beispiel zu erläutern, wie eine Implementierung der Sicherheitsfunktionen aussehen könnte. Den Abschluss des 6. Kapitels bildet eine Zusammenfassung der entwickelten Komponenten, eine Bewertung, wie die in den Zielen genannten Forderungen umgesetzt werden konnten und welche weiteren Optimierungsmöglichkeiten bestehen.

Eine Zusammenfassung der gesamten Arbeit und ein Ausblick auf offene Fragen und Ideen für zukünftige Arbeiten wird im siebenten Kapitel gegeben.

2 Theoretischer Hintergrund

In diesem Kapitel werden die Grundlagen für die weiteren Sicherheitsbetrachtungen gelegt. Ausgehend von der Begriffsklärung werden die verschiedenen Architekturen eines Mesh-Netzwerks erläutert. Obwohl die Mesh-Technologie unabhängig von einem konkreten funkbasierten Verfahren ist, werden die populärsten genannt. Aus Funktechnik, Antennentechnik und benutzten Sendeleistungen ergeben sich die notwendigen und möglichen Schutzmaßnahmen der Übertragung. Dies betrifft vor allem die Verfügbarkeit des Mesh-Netzwerkes.

Eine Hauptaufgabe von sicheren Mesh-Architekturen ist die Absicherung des Routings. Je nach verwendetem Routing-Verfahren sehen diese Schutzmaßnahmen unterschiedlich aus bzw. bieten besondere Möglichkeiten. Im Unterkapitel Routing (2.3) werden sehr kurz die grundlegenen Verfahren beschrieben.

Den Abschluß dieses Kapitels bildet eine kurze Beschreibung von Problemem, welche alle Mesh-Netzwerke betreffen. Diese wichtigen Merkmale müssen bei einer Implementierung bedacht werden, gerade weil es kein direktes technisches Verfahren gibt, um diese ganz zu vermeiden.

2.1 Charakteristik von Mesh-Netzwerken

2.1.1 Begriffsklärung

Ein *Ad-hoc-Netzwerk* (ad hoc: lat. zu diesem Zweck) bezeichnet in der Informationstechnologie ein Netzwerk zwischen zwei oder mehr mobilen Endgeräten, welches ohne feste Infrastruktur auskommt.

Ein *Mobiles Ad-hoc-Netzwerk (MANET)* ist ein selbstkonfigurierendes Netzwerk, das auf Basis eines Funknetzwerks aufgebaut ist. Innerhalb des Netzwerks können mobile Geräte, wie z.B. Mobiltelefone, Kleincomputer oder Laptops, sofort eine Verbindung zueinander aufbauen, ohne dass eine übergeordnete Infrastuktur (wie z. B. Wireless Access Points) benötigt wird. Jedes Gerät (Knoten) dient hierbei nicht nur als Sende- und Empfangsstation für Sprache oder Datenübertragung, sondern auch als Router für andere TeilnehmerInnen [4].

Ganz allgemein beschreibt ein Ad-hoc-Netzwerk eine Verbindung zwischen Knoten, wogegen in einem mobilen Ad-hoc-Netzwerk nicht nur Verbindungen zwischen Knoten, sondern auch zu anderen Netzen wie dem Internet betrachtet werden.

Ein *Wireless-Mesh-Netzwerk* (mesh: engl. Masche, Netz) ist ein Netz, das komplett in einem Wireless LAN implementiert ist. Ein Mesh basiert auf *multi-hop*[1]-Kommunikation. Die Netzwerkinfrastruktur ist dezentralisiert. Knoten agieren als Repeater um Daten von nahen Knoten zu anderen zu übertragen, die zu weit weg sind um sie zu erreichen. Es resultiert ein Netzwerk, das weite Distanzen überspannen kann, besonders in unebenem oder schwierigem Terrain. Mesh-Netzwerke können feste oder mobile Geräte miteinbeziehen [5].

In vielen wissenschaftlichen Publikationen werden die Begriffe Mesh-Netzwerk und MANET synonym verwendet. Da ein Mesh-Netzwerk jedoch mehr als nur ein mobiles Ad-hoc-Netzwerk ist, finde ich den zweiten Begriff nicht aussagekräftig genug und verwende *Mesh-Netzwerk*.

Ein Wireless Mesh kann als erweitertes Ad-hoc-Netzwerk betrachtet werden. Bei traditionellen Ad-hoc-Netzen werden die Daten auf einem Kanal (Frequenz) übertragen. Bei Mesh-Netzwerken sind verschiedene Kanäle, welche durch Gateways verbunden sind, möglich. Gateways können verschiedene Netzwerke, wie kabelgebundenen Internetzugang und WLAN, verbinden. Im mehrschichtigen Architekturansatz für Wireless-Mesh-Netzwerke werden einzelne Knoten des Mesh-Netzwerkes in unterschiedliche Rollen eingeteilt und übernehmen verschiedene Funktionen. Diese Unterteilung ist bei allgemeinen Ad-hoc-Netzen nicht bekannt.

2.1.2 Selbstorganisierung, -konfigurierung

Ein Mesh-Netzwerk ist dynamisch selbstorganisierend und selbstkonfigurierend, d.h. die Knoten im Netzwerk erkennen automatisch Verbindungen zu anderen Knoten (etablieren oder erweitern das Mesh-Netzwerk). Im einfachsten Fall wird dies durch Versenden von *Hello-Nachrichten* bei Eintritt eines Knoten in das Mesh-Netzwerk erreicht. Alle benachbarten Knoten kennen nun diesen und falls die Hello-Pakete periodisch weiter versandt werden, bleiben auch bei hoher Dynamik alle Knoten auf dem aktuellen Topologiestand.

Die Eigenschaft der Selbstkonfiguration bringt viele Vorteile, z.B. geringe Aufbaukosten, einfache Netzwerkverwaltung und vergrößerte Dienstleistungsgebiete. Das Netz-

[1] Ein Netzwerk, in dem Knoten als Repeater Daten zu anderen Knoten weiterleiten, wird auch als multi-hop-Netzwerk bezeichnet (hop: engl. Sprung).

werk reagiert auf fehlerhafte oder nicht mehr existierende Knoten - die Robustheit und Fehlertoleranz steigt. Es sind keine zentralen Instanzen notwendig - dies spart Investitions- und Wartungskosten.

Der administrative Aufwand beschränkt sich auf die Festlegung der grundsätzlichen Mesh-Parameter (Protokoll, IP-Bereich usw.). Danach ist nur noch eine Überwachung des Netzwerkes notwendig (Verbindungsqualität, Abdeckung). Ein Administrator kann mit diesen erhobenen Daten das Mesh-Netzwerk optimieren, ein Eingriff zur reinen Funktionalität ist nicht notwendig.

2.1.3 Sicherheit, Kooperation und Fairness

Im englischen Sprachgebrauch wird beim Begriff Sicherheit zwischen *Security* und *Safety* unterschieden. Safety beschreibt den Zustand, dass ein System in sich sicher ist, d.h. beim normalen Betrieb des Systems kommt es nicht zu Fehlern. Security beschreibt die Sicherheit gegen gezielte Angriffe auf das System. Diese Arbeit beschäftigt sich mit beiden Aspekten. Sowohl fehlerhafte Knoten in einem Mesh-Netzwerk sollen erkannt werden (Safety), als auch Angriffe von außen (Security).

Im Fall eines Mesh-Netzwerkes bedeutet *Kooperation* das korrekte Weiterleiten von Daten anderer Knoten. Jeder im Mesh-Netzwerk beteiligte Knoten verhält sich protokollkonform und stellt seine Funktion dem Netz zu jeder Zeit und an jedem Ort (innerhalb der Reichweite des Netzes) zu Verfügung. Das ist grundlegend für die Verfügbarkeit des Netzwerkes.

Fairness bedeutet die "gerechte" Aufteilung der Bandbreite unter allen Knoten, die Daten erhalten oder versenden wollen. Eine Position im Mesh-Netzwerk soll nicht in Bezug auf die verfügbare Datenrate einer anderen Position vorgezogen oder benachteiligt werden. Diese Benachteiligung kann auch bei korrektem Verhalten aller Knoten auftreten, wenn im topologischen Aufbau des Netzes Kommunikationsnadelöhre existieren.

Weitere Informationen, in welchem Verhältnis Vertrauen, Kooperation und Sicherheit in einem Ad-hoc-Netzwerk stehen, können in [6] gefunden werden.

2.1.4 Aktueller Forschungsstand

Pragmatisch gesehen ist es nicht schwierig ein Mesh-Netzwerk aufzubauen, da alle benötigten Komponenten, wie Funkhardware und Ad-hoc-Routingprotokolle vorhanden

sind. Die 802.11-WLAN-Netzkomponenten sind kostengünstig und in fast allen neueren Computern verfügbar. Einige Firmen haben in den letzten Jahren das Potential dieser Technik erkannt und bieten Mesh-Produkte an[2].

Sobald die Anzahl der Knoten ansteigt oder eine hohe Datenmenge über die Funkstrecken transportiert werden muss, reichen die bisherigen Ad-hoc-Protokolle nicht mehr aus. Die Bandbreite bricht ein und das Netz ist nicht mehr für alle Knoten funktionsfähig. Neue Protokolle wurden entwickelt. Erste positive Ergebnisse konnten in Feldversuchen von Forschungslaboren an Universitäten oder in Community-Netzwerken erreicht werden (siehe Kapitel 3.7).

Die bisherigen Ergebnisse verbessern die Skalierbarkeit, bieten aber noch lange nicht das Potential, welches Mesh-Netzwerke bieten könnten. Nicht nur das Routing muss optimiert werden, auch das MAC-Protokoll (OSI Layer 2) ist für multi-hop-Netzwerke mit dynamischen Routen sehr ineffizient. Von einigen WissenschaftlerInnen wird die komplette Überarbeitung einer cross-layer-Architektur gefordert.

Internationale Kongresse wie *Mesh Networking Summit*[3], maßgeblich von Microsoft organisiert, *MeshNets*[4] oder *adHocNow*[5] sind Austauschplattformen für WissenschaftlerInnen und Firmen. In Fachzeitschriften werden monatlich neue Forschungsergebnisse veröffentlicht. Hauptfokus liegt dabei auf der Skalierbarkeit der Protokolle.

Industrielle Standardisierungsgruppen arbeiten an neuen Spezifikationen für Mesh-Netzwerke. Mit den Standards IEEE 802.11s, 802.15 und 802.16 werden durch Arbeitsgruppen momentan eigene Mesh-Unterstandards entworfen und diskutiert. 802.11s (Wireless Mesh Network) soll im Mai 2006 verabschiedet werden. Nach der Vorstellung des Mesh Connectivity Layer von Microsoft [7] ist abzuwarten, ob dieser sich zu einem Standard-Mesh-Protokoll entwickeln wird. Die Vergangenheit zeigte, dass Microsoft durch seine Marktposition dazu in der Lage ist.

2.2 Mesh-Architektur

Die Architektur eines Mesh-Netzwerkes ist eine Ansammlung von sich im Raum frei bewegenden Knoten (teilnehmende Computer mit Funknetzwerkkarte), die selbstständig und dynamisch andere Knoten suchen und Verbindung zu ihnen halten. Die Kommunikation erfolgt funkbasiert (siehe Mesh-Definition). Der Status der Verbindungen

[2] siehe im Kapitel 3 genannte Praxisbeispiele
[3] http://research.microsoft.com/meshsummit/
[4] http://www.meshnets.org/
[5] http://fismat.umich.mx/adhocnow/

zwischen diesen Knoten ist zu jeder Zeit abhängig von ihrer Position, Übertragungsstärke ihres Signals, Struktur der Antennenaufteilung im Raum und Interferenzen mit anderen Sendern. Nicht alle Knoten sind untereinander in direkter Kommunikationsreichweite. Die Mobilität der Knoten und die Unsicherheit der anderen Übertragungsfaktoren resultieren in einer schnell wechselnden und unvorhersehbar schwankenden Topologie.

Grundsätzlich unterscheidet man zwei Arten von Mesh-Netzwerkarchitekturen: das in Abbildung 2.1 dargestellte *gleichberechtigte*, einstufige Modell und das in Abbildung 2.2 dargestellte *hierarchische*, mehrstufige Netzwerk. In gleichberechtigten Netzwerken haben alle Knoten die gleichen Aufgaben und das Paketrouting, das hier zum Einsatz kommt, basiert auf peer-to-peer-Verbindungen und ist nur durch die Größe des Netzwerkes beschränkt.

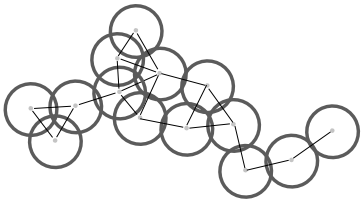

Abbildung 2.1: Einstufiges Mesh

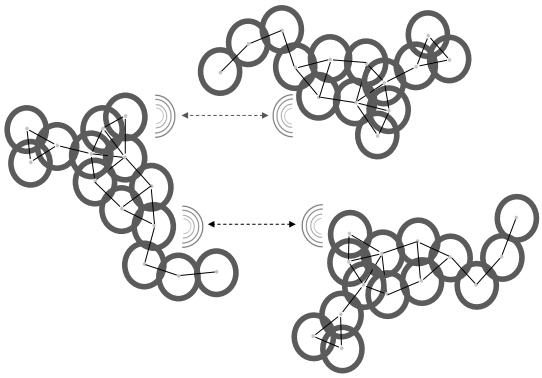

Abbildung 2.2: Zweistufiges Mesh

Bei den *hierarchischen* Netzwerken existieren mindestens zwei Ebenen. Auf der unteren Ebene bilden nahe aneinanderliegende Knoten ein Mesh-Netzwerk, in dem mindestens ein Knoten als Border-Knoten (Grenzknoten) zur höheren Ebene dient. Die Border-Knoten aus den Mesh-Netzwerken stellen, wie in Abbildung 2.2 zu sehen, die zweite Hierarchieschicht dar und benötigen üblicherweise stärkere Übertragungs- und Empfangskapazitäten. Oftmals sind diese Border-Knoten, um eine höhere Datenrate zu erzielen, mit mehr als einer Netzwerkkarte ausgestattet. Das Routing zwischen Knoten desselben Netzwerks in einer unteren Ebene basiert auf einer peer-to-peer-Verbindung, wobei das Routing zwischen verschiedenen Netzwerken unterer Schichten über die Border-Knoten stattfindet. Theoretisch ist es möglich, verschiedene Routingverfahren bzw. -protokolle auf den einzelnen Schichten einzusetzen.

Einzelne Knoten können zusätzliche Dienste wie einen Zugang zu einem anderen Netzwerk mit anderer Funktechnik (WiMAX) oder zum Internet anbieten. Sie besitzen mindestens ein zweites Netzwerkinterface, sind nicht mobil und werden als Gateways bezeichnet. Borderknoten strukturieren das Netzwerk logisch, wogegen in Gateway-Knoten unterschiedliche Netze gekoppelt und ggf. Protokollumwandlungen wie Adressen oder Paketgröße durchgeführt werden.

Knoten und Gateways können normale Computer wie Laptops oder PDAs sein. Um eine bessere Abdeckung in der Fläche zu erreichen, können auch *dedicated computer* (kleine Computer mit Minimalausstattung) installiert sein, welche nur die Aufgabe haben, Daten weiterzuleiten oder Gatewayfunktionen in andere Netze anzubieten.

In einigen wissenschaftlichen Aufsätzen werden die Border-Knoten als *Mesh-Router* und die Knoten auf der untersten Schicht als *Mesh-Clients* bezeichnet. Da in einem Mesh-Netzwerk alle Knoten Routingaufgaben übernehmen - also Router sind, verwende ich diese Bezeichnung nicht.

Die Entscheidung, ob in einem konkreten Fall ein ein- oder mehrstufiges Mesh-Netzwerk eingesetzt werden soll, hängt von der erwarteten Topologie ab.

Ein einstufiges Netzwerk ist die pure Variante eines Mesh-Netzwerkes. Hier werden alle Knoten gleich behandelt, egal wie sie positioniert sind oder über welche Ressourcen sie verfügen. Existieren keine Informationen über den Aufbau und die Dynamik des Mesh-Netzwerkes, ist diese Variante die bessere. Sie ist immer anwendbar.

Sind jedoch Informationen zum Netz vorhanden, kann die Effizienz durch mehrstufige Netze gesteigert werden. In verschiedenen Stufen können unterschiedliche Routingverfahren zum Einsatz kommen. Aufwendige Rechenoperationen können auf leistungsstarke Knoten verlagert werden. Dies ist in einstufigen Netzwerken nicht möglich und falls die Strukturdaten vor Aufbau des Netzes ermittelbar sind, ist dies die Variante mit höherer Performance.

2.2.1 Übertragungsverfahren

Prinzipiell ist der Mesh-Ansatz nicht von einem konkreten Übertragungsverfahren abhängig und könnte auch auf fast alle funkbasierte Verfahren angewendet werden. Zu beachten ist die Kompatibilität und Interoperabilität mit existierenden Netzwerken. Mesh Netzwerke, die mit 802.11-Technologie entworfen werden, sollten mit diesem Standard kompatibel sein. Das bedeutet, dass in einem solchen Netzwerk sowohl Mesh-Knoten als auch konventionelle Wi-Fi Clients unterstützt werden sollten.

Die momentan am meisten beachtete Grundtechnik ist der IEEE 802.11-Standard, aber auch Bluetooth, 802.15 oder 802.16. In diesen Standards wird der Mesh-Ansatz erwähnt bzw. gibt es erste Vorschläge oder Unterstandards, welche auf einem klaren Mesh-Einsatz beruhen.

Bluetooth

Bluetooth ist ein wichtiger Industriestandard für die drahtlose (Funk-)Vernetzung von Geräten über kurze Distanz. Ein Bluetooth-Netzwerk (Piconet) kann bis zu 255 TeilnehmerInnen umfassen, wovon acht Geräte gleichzeitig aktiv sein können. Es besteht aus einem Master und bis zu sieben weiteren TeilnehmerInnen (Slave). Der Master steuert die Kommunikation und vergibt Sendeslots an die Slaves. Ein Bluetooth-Gerät kann in mehreren Piconetzen angemeldet sein, allerdings nur in einem Netz als Master.

In dem Standard können bis zu zehn Piconetze ein Scatternet bilden, wobei die TeilnehmerInnen untereinander über Brückenknoten (bridge nodes) in Kontakt treten können. Leider wurden bisher keine konkreten Scatternet-Protokolle für Bluetooth definiert.

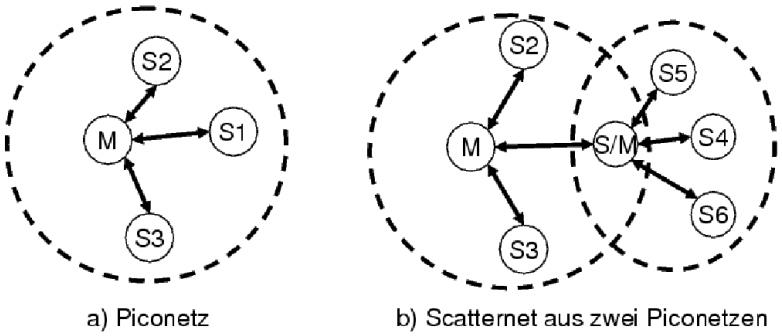

Abbildung 2.3: Bluetooth Pico- und Scatternetz

Der Ansatz eines Scatternet ist dem eines mehrschichtigen Mesh-Netzwerkes ähnlich. Einzelne hierarchisch organisierte Bluetooth-Netze (ein Master und bis zu 7 aktive Slaves) können mit anderen Bluetooth-Netzen in ihrer Reichweite kommunizieren. Die Daten laufen dabei über einzelne Bluetooth-Geräte - eine Art Mesh-Routing. Scatternets sind momentan noch ein reines Forschungs- und Standardisierungsthema. Konkrete Implementierungen gibt es leider noch nicht [8].

IEEE 802.11

IEEE 802.11 (auch: Wireless LAN, WLAN, WiFi - Wireless Fidelity) bezeichnet einen Industriestandard für drahtlose Netzwerkkommunikation. Ein WLAN kann man in zwei Modi betreiben: dem Infrastruktur-Modus und dem Ad-hoc-Modus. Fast alle bekannten Mesh-Implementierungen beruhen auf dem Ad-hoc-Modus des IEEE 802.11-Standards. Durch einfache Anpassungen in der Treibersoftware der Wirelesskarten kann ein Mesh-Netzwerk aufgebaut werden.

Die Gruppe 802.11s arbeitet an einer Ergänzung des Standards, welcher für etwa 32 Mesh Access Points geeignet ist. Ergänzungen des Protokolls und Erweiterungen des MAC Layers sorgen für eine schnellere Weiterleitung der Datenpakete. Weitere Aspekte werden neue Routing-Algorithmen betreffen, die verschiedene, an WLAN Systeme angepasste Metriken verwenden. Der *call for proposals* für 802.11s endete im Juni 2005 mit 15 eingegangenen Vorschlägen. Bereits im September desselben Jahres wurden die Vorschläge auf 4 Texte zusammengefasst. Erwartet wird, dass alle Arbeiten durch Konsolidierung und Kompromisse zu einem Draft bis Ende 2006 / Anfang 2007 zusammengeführt werden. Dann können die Hersteller beginnen, 802.11s-kompatible Produkte zu entwickeln [9].

IEEE 802.15 - ZigBee und TG5

ZigBee ist eine veröffentlichte Spezifikation eines Kommunikationsprotokolls, welches für kleine digitale Funkgeräte mit schwacher Energieleistung entwickelt wurde. Es basiert auf dem IEEE 802.15.4-Standard für *wireless personal area networks*. Die Verbindung von IEEE 802.15.4 und ZigBee ist ähnlich dem existierenden Standard 802.11 und der Wi-Fi Alliance. Die ZigBee 1.0-Spezifikation wurde am 14. Dezember 2004 ratifiziert und stand anfangs nur den Mitgliedern der Alliance zur Verfügung. Seit dem 13. Juni 2005 ist sie für alle Gruppen öffentlich.

Es gibt drei verschiedene Gerätetypen. In jedem Netzwerk gibt es genau einen *ZigBee coordinator*. Dieses Gerät ist leistungsstark, kann Verbindungen in andere Netze herstellen und stellt den Kopfknoten im Netzwerkbaum. Ein Koordinator kann bis zu 255 aktive Knoten verwalten. *Full function devices* können Daten von anderen Knoten weiterleiten. *Reduced function devices* können nur selbst Daten in das Netzwerk versenden oder empfangen, aber keine Daten für andere Knoten weiterleiten. In den meisten größeren Fällen, besteht das Gesamtnetzwerk aus mehreren Clustern. Es können bis zu 4.000 Geräte starke ZigBee-Netzwerke gebildet werden [10].

Das zugrundeliegende Protokoll versucht automatisch ein Low-speed-Ad-hoc-Netzwerk zu bilden. Daten können über mehrere Wege (dynamisch) zum Ziel gelangen. Dies be-

deutet höhere Robustheit bei Ausfall eines Knoten oder einer Funkverbindung.

Das Marketing von ZigBee spricht von einem Mesh-Netzwerk. Hinter dem Routing steckt jedoch eine hierarchische Routingvariante. Die Rolle der Knoten ist unterschiedlich und die Koordinatoren bilden ein "Nadelöhr". Ich denke, dass der Namengebrauch Mesh überzogene Erwartungen erweckt. Laut Definition ist ZigBee kein Mesh-Netzwerk, auch wenn einzelne Elemente wie Selbstorganisation und redundante Wegewahl berücksichtigt wurden.

Für den Standard IEEE 802.15 existiert seit Anfang 2004 eine *Task Group 5*. Sie hat sich zum Ziel gesetzt, die notwendigen Mechanismen, welche in den PHY- und MAC-Layern von WPANs für ein direktes Meshing vorhanden sein müssen, zu entwickeln. Zwei Varianten - volle und partielle Mesh Topologie - wurden unterschieden. In der vollen Mesh-Topologie verbindet sich jeder Knoten zu allen anderen erreichbaren, in der partiellen Variante nur zu den Knoten, mit denen er die meisten Daten austauscht.

Bis zum 1. September 2004 gab es ein *call for application*. Seit dem scheint die Arbeit in dieser Gruppe zu ruhen. Zumindest sind auf der offiziellen Webseite [11] keine neueren Informationen zu finden. Schade, denn der Ansatz beschäftigte sich mit reinem Meshing - im Gegensatz zu ZigBee.

IEEE 802.16

Der erste 802.16-Standard wurde im Dezember 2001 vorgestellt. Ihm folgten drei Erweiterungen - 802.16a, 802.16b und 802.16c, welche Fragen wie Frequenzspektrum, QoS und inter-operability berücksichtigten. Im September 2003 wurde 802.16REVd, welches unter anderem mit dem europäischen *HiperMAN* harmonisierte und verschiedene Testspezifikationen beinhaltete, veröffentlicht. Schließlich wurden im 802.16-2004-Standardisierungstext alle älteren Dokumente zusammengefasst.

Bisher betraf der Standard nur stationäre Geräte. Im Dezember 2005 ist der Standard 802.16e veröffentlicht worden, in dem auch mobile Komponenten betrachtet werden. Dies bedeutet, Geräte können zwischen verschiedenen Funkzellen wechseln (Roaming). WiMAX (Worldwide Interoperability for Microwave Access) wurde die 802.16-Protokollfamilie von der Industriegruppe WiMAX Forum getauft. WiMAX umfasst nicht alle Dokumente im eigentlichen 802.16-Standard und das Industriekonsortium sieht sich eher als Zertifizierungsinstanz, ähnlich der WiFi Alliance bei WLAN. Das Wimax-Forum hat die ersten Wimax-Komponenten nach dem Standard IEEE 802.16-2004 im Januar 2006 zertifiziert.

Neben dem *Point-to-Multipoint-Modus* führt 802.16a für lizenzfreie Bänder auch

einen optionalen Mesh-Modus ein. Dies bedeutet, dass WiMAX-Knoten gleichzeitig als *subscriber station* und *base station* arbeiten. Ein konkretes Einsatzszenario sind Spielekonsolen. Sony und Microsoft wollen in der nächsten Generation ihrer Konsolen WiMax als ein Feature anbieten. So können SpielerInnen sich in einem Ad-hoc-Netzwerk mit anderen verbinden. Laut Eigenaussagen könnte sich dies zur Killerapplikation entwickeln.

2.2.2 Antennen und Sendeleistung

Die Antennen handelsüblicher 802.11-Endgeräte haben eine Reichweite von 30 bis 150 Meter auf freier Fläche, mit neuester Technik (externe Rundstrahlantennen) teilweise bis zu 300 Meter. Im Normalbetrieb sind nach 802.11a 30 mW und nach 802.11b/g 100 mW Sendeleistung erlaubt. Unter strengen Bedingungen sind teilweise bis zu 1000 mW bei 802.11a vom Gesetzgeber gestattet. Leichtbauwände stellen bereits ein Hindernis dar. Bei Stahl und Betonwänden wird die Reichweite enorm eingeschränkt. Bäume, insbesondere dicht belaubte, sind ebenfalls Hindernisse für WLAN-Verbindungen.

Mit speziellen *Richtfunkantennen* lassen sich bei Sichtkontakt mehrere Kilometer überbrücken. Diese bidirektionalen Verbindungen bieten sich beim Koppeln von weit entfernten Wireless (Mesh)-Netzen an. Es dürfen in Deutschland uneingeschränkt auch selbstgebaute Antennen verwendet werden; hierfür ist keine Amateurfunklizenz notwendig, da die Regulierungsbehörde die entsprechenden Frequenzbereiche in einer Allgemeinzuteilung lizenzfrei gestellt hat.

Es sei noch angemerkt, dass die spektrale Effizienz und somit der Gesamtdurchsatz des Netzes durch Space Division Multiple Access (SDMA) erhöht werden kann, wenn die Knoten ihre Sendestärke so regeln können, dass der Empfangsknoten eines Datenpaketes das Paket gerade noch empfangen kann. Somit könen MANETs auch ein Mittel zur Erhöhung der Leistungsfähigkeit von Funknetzen sein.

Die sogenannten *Software gesteuerten Funkgeräte* (Software-defined radio) führen die gesamte Modulation / Demodulation der Funksignale durch eine Software durch. Es gibt keine Einschränkungen durch fest installierte Elektronikchips, welche einmal hergestellt, keine grundsätzlichen Veränderungen an den Protokollen und Spezifikationen erlauben. Ziel der Technik ist es, eine Funknetzwerkkarte zu entwickeln, welche neue Formen der Datenübertragung sofort durch Verändern der Software bieten kann. Noch wird die Technik nur vereinzelt eingesetzt. Sie könnte in den nächsten Jahren zur dominanten Funktechnologie werden und bietet auch für den Mesh-Einsatz Vorteile, wie angepasste Sendestärke oder Sicherheitsfunktionalität auf dem Netzwerk-Layer [12].

2.3 Routing und Weiterleitung

Eine große Herausforderung von sicheren Mesh-Verfahren ist die Absicherung des Routenfindens und die Weiterleitung der Pakete. Ein Angreifer oder eine Angreiferin darf hier keine Chance haben, z.B. Daten durch kompromittierte Knoten zu schleusen und so die gesamte Sicherheitsarchitektur zu unterlaufen. Ein sicheres Routing ist somit eine wichtige Basis vieler anderer Sicherheitsmethoden.

Auf Grund der speziellen Bedingungen in einem Mesh-Netzwerk können die üblicherweise im Internet eingesetzten Routingalgorithmen nicht verwendet werden. Die wesentlichen Gründe hierfür sind:

- Knoten haben kein Vorwissen über die Topologie des Netzwerkes, sie müssen diese selbst erkunden
- keine zentralen Instanzen zum Speichern von Routinginformationen
- Mobilität der Knoten und damit verbundener ständiger Topologiewechsel
- wechselnde Metrik der Übertragungsstrecken z.B. durch Interferenzen
- (meist) beschränkte Bandbreite gegenüber kabelgebundenen Netzen
- beschränkte Ressourcen der Knoten (z.B. Systemleistung, Energieverbrauch)

Für den Mesh-Einsatz kommt nur eine paketvermittelte Datenübertragung in Frage, da durch das dynamische Verhalten leitungsvermittelte Verbindungen nicht möglich sind.

Aufgabe eines Routingprotokolls ist die Bestimmung eines Weges von einem Quell- zu einem Zielknoten. Je nach verwendeter Metrik sollte dieser möglichst kurz sein oder möglichst gering belastete Regionen des Netzwerks nutzen. Der Einfluss unterschiedlicher Metriken auf die Performance eines statischen multi-hop WLANs auf Grundlage des LQSR-Protokolls wird ausführlich in [13] untersucht.

Weitere Anforderungen an die Protokolle sind eine möglichst kleine Routingtabelle (Speicherplatz), welche ständig, wenn Knoten verschwinden, sich bewegen oder wenn neue erscheinen, aktualisiert werden muss. Die Zeit und die Anzahl der Nachrichten, die zum Auffinden einer Route benötigt werden, sollten möglichst gering sein.

Es gibt mehr als 70 konkurrierende Entwürfe für das Routing der Pakete durch ein Mesh-Netzwerk [5]. Nur ein Teil dieser (theoretischen) Protokolle liegt auch als Linux-Implementierungen vor. Die Protokolle entstehen zumeist an Universitäten und dort kommt bei der Software-Entwicklung überwiegend Linux (Unix) zum Einsatz.

Bei einer funkbasierten Datenübertragung ist zu beachten, dass die kürzesten Pfade (wenigste Hops) nicht unbedingt die schnellste Übertragung bedeuten. Existiert zwischen zwei Knoten eine sehr schlechte Funkverbindung, ist eine Umleitung über einen Zwischenknoten, der zu beiden Stationen eine gute Verbindung hat, meist sinnvoller und schneller, obwohl diese Umleitung einen Sprung (Hop) zusätzlich bedeutet.

Im folgenden Abschnitt werden kurz die für Mesh-Netzwerke wichtigsten Routingmethoden vorgestellt und verbreitete Protokollimplementierungen genannt. Ich beschränke mich auf unicast-Protokolle, da diese in allen Mesh-Szenarien eingesetzt werden und andere Methoden, wie multicast, eher Sonderfälle darstellen. Die Einordnung der Protokolle ist dabei nicht immer eindeutig möglich. Ausführlichere Beschreibungen zu Mesh- bzw. Ad-hoc-Routingprotokollen können in [14], [15] und [16] gefunden werden.

Es existieren zwei grundlegende Protokollkategorien: *proaktive* (Table-Driven) und *reaktive Protokolle* (Source-Initiated bzw. on-demand).

2.3.1 Proaktive Routingprotokolle

Proaktive Routingverfahren bestimmen die zu verwendenden Pfade zwischen zwei Knoten bereits bevor diese tatsächlich benötigt werden. Sollen dann tatsächlich Daten verschickt werden, so muss nicht auf die Bestimmung des Pfads zum Zielknoten gewartet werden. Nachteilig ist dafür jedoch, dass diese Verfahren viele Kontrollpakete verschicken um Pfade zu bestimmen, die womöglich später nicht benötigt werden. Die meisten proaktiven Protokolle basieren auf den Link-State-Algorithmen.

Das *Optimized Link-State Routing* (OLSR) [17] ist ein proaktives Protokoll mit einem Hysterese-Mechanismus und definiert Multi-Point-Relays. Es wird sehr aktiv entwickelt und getestet, unter anderem durch die US-Navy. Implementierungen des *OLSR daemon* liegen für Linux, Windows, OS X, FreeBSD und NetBSD-Systeme vor. Die Unik-OLSR-Software unterstützt ladbare Plugins zum Testen eigener Optimierungen oder Erweiterungen. Dieses Protokoll ist Grundlage für die spätere Entwicklung einer Sicherheitsarchitektur für Community-Netzwerke.

Eines der ersten öffentlichen und speziell für Mesh-Netzwerke entwickelten Routingprotokolle war *Mobilemesh* [18]. Es ist ein OLSR-Protokoll, das seine Routen unter Berücksichtigung aller Netzteilnehmenden berechnet. Die Implementierung besteht aus drei Userspace-Programmen. *mmdiscover* verschickt die Hello-Pakete und kommuniziert mit anderen Knoten. *mmrp* erstellt die Routingtabellen. *mmborder* versucht, Teile des Traffics im Mesh über parallel bestehende schnelle Netzwerkverbindungen (etwa über Fast Ethernet) zu routen, um die Belastungen der Funkstrecken im WLAN zu reduzieren. Mobilemesh erkennt den nächstgelegenen Gateway in ein anderes Netz

(z.B. das Internet), den ein anderer Mesh-Router anbietet, und trägt ihn automatisch in die Routingtabellen ein. *Ipmesh* ist eine Implementierung für Windows.

Das *Topology Broadcast based on Reverse-Path Forwarding*-Routingprotokoll (TBRPF) ist ebenfalls ein Link-State-Protokoll für Wireless Mesh-Netzwerke. Um das Datenvolumen der Routinginformationen einzuschränken, werden nicht gesamte Routinginformationen sondern nur Veränderungen in der Routingtopologie ausgetauscht. Das Protokoll scheint patentgeschützt zu sein, bis es ein IETF-Standard wird.

2.3.2 Reaktive Routingprotokolle

Im Gegensatz zu den proaktiven Verfahren bestimmen reaktive Routingverfahren für mobile Ad-hoc-Netzwerke die benötigten Pfade zwischen zwei Knoten erst, wenn diese tatsächlich benötigt werden. Daraus ergibt sich, dass das erste Datenpaket einer Verbindung erst mit einer geringen Verzögerung versendet werden kann, da zunächst auf den Abschluß der Routenbestimmung gewartet werden muss. Dafür werden allerdings auch nur Kontrollpakete versendet, wenn tatsächlich Daten verschickt werden und dies zur Routenbestimmung notwendig ist. Dies schlägt sich positiv im Energieverbrauch der Knoten nieder.

Das Protokoll *Ad hoc On-Demand Distance Vector* (AODV) hat viel Beachtung gefunden. Implementierungen gibt es verschiedene, u.a. von der Santa Barbara University [19]. AODV unterstütz IPv6 und es existieren Implementierungen für Windows XP aus einer Entwicklungsabteilung von Intel. Der *MeshAP von Locustworld* benutzt ebenfalls eine Linuximplementation des öffentlichen AODV-Protokolls.

Ein weiteres Beispiel für ein verbreitetes reaktives Protokoll ist *Dynamic Source Routing* (DSR) [20]. Routen werden mit einem ähnlichen Verfahren wie bei AODV gesucht. In jedem Datenpaket steht eine Liste aller Zwischenknoten, so dass die weiterleitenden Knoten keine eigenen Routinginformationen haben müssen. Knoten belauschen den Netzwerkverkehr anderer, um an Routinginformationen zu gelangen, welche sie selbst später einmal gebrauchen könnten. Eine Implementierung namens picoNet II kann unter [21] gefunden werden.

Microsoft Research LAB implementierte Ad-hoc-Routing und eine Linkqualitätsmessung in ein Softwaremodul, welches *mesh connectivity layer* (MCL) genannt wurde. Aktuell ist MCL ein ladbarer Windows-Treiber, welcher einen virtuellen Netzwerkadapter simuliert. Das DSR-Protokoll wurde angepasst und *LQSR* genannt [7].

2.3.3 Hierarchische / hybride Routingverfahren

Bei hierarchischen Routingverfahren haben einige Knoten besondere Rollen. Entweder sie stellen als Border-Knoten die Verbindung zu anderen (logischen) Netzwerken her oder fungieren als Clusterhead als eine zentrale Verwaltungsinstanz durch die der gesamte Netzwerkverkehr eines Clusters läuft. Diese Zentralität steht dem dezentralen Anspruch eines Mesh-Netzwerkes nicht im Wege, da diese sich dynamisch verändern kann und kein administrativer Eingriff von außen notwendig ist.

Das *Clusterhead Gateway Switch Routing* (CGSR) Protokoll ist ein zu Clustern zusammengefasstes mobiles Netzwerk, das über heuristische Routingschemata aufgebaut ist. Innerhalb eines Clusters wird ein Knoten als Kopf bestimmt, der die Rahmenbedingungen für die Kommunikation vorgibt. Ein Clusterkopf wird nur verändert, falls er in Reichweite eines anderen kommt oder ein Knoten sich außerhalb der Reichweite aller Clusterköpfe bewegt. DSDV ist das zugrundeliegende Routingprotokoll, mit der Ausnahme, dass alle Datenpakete zuerst an den Clusterkopf gesendet werden und dieser sie über Gatewayknoten an den Clusterkopf des Zielknotens weiterleitet. Gatewayknoten, die zu zwei Clustern gehören, ermöglichen die Kommunikation.

Hybride Verfahren kombinieren proaktive und reaktive Routingverfahren. Dabei soll das Ziel erreicht werden, die Vorteile der beiden Ansätze in einem neuen Routingprotokoll zusammenzufassen. Beispielsweise kann in einem lokal beschränkten Bereich ein proaktives Verfahren eingesetzt werden, während für weiter entfernte Ziele ein reaktives Verfahren eingesetzt wird. Dies vermindert die Belastung des Netzwerks durch Kontrollpakete, die bei einem rein proaktiven Verfahren über das gesamte Netzwerk versendet würden. Trotzdem stehen für lokale Ziele sofort Pfade zur Verfügung, ohne dass, wie bei einem rein reaktiven Verfahren, auf deren Bestimmung gewartet werden müsste.

Im *Zone Routing Protokoll* (ZRP) [22] ist ein solcher hybrider Ansatz umgesetzt. Bis zu einer bestimmten Anzahl von Sprüngen verhält sich ZRP proaktiv (das Intra-Zone-Routing-Protokoll), darüber hinaus generiert es Routen reaktiv (Inter-Zone-Routing-Protokoll). Verfügbare Implementierungen gibt es bislang nur für den Netzwerksimulator NS-2 oder die Quelltexte sind nicht veröffentlicht (closed-source).

Die Scientific Research Corporation in Atlanta hat ZRP um Mechanismen für QoS erweitert - sie nennt ihre Implementierung *Wireless Ad-hoc Routing Protocol* (WARP) und hat sich den Namen Mobilerouter dafür schützen lassen. Warp wurde unter Linux für militärische Zwecke entwickelt und ist nicht öffentlich.

Das *Fisheye State Routing* (FSR) [23] ist entfernt mit diesem Ansatz verwandt. FSR ist ein proaktives Protokoll, welches Topologieänderungen, die nah beim Knoten statt-

finden, mit hoher zeitlicher Auflösung meldet, wohingegen ein Knoten nur ungenau und mit großer zeitlicher Verzögerung Informationen über weit entfernte Änderungen erhält.

2.3.4 Spezielle Routingverfahren

Drei besondere Routingverfahren, welche für Mesh-Netzwerke interessant sind, möchte ich kurz nennen. Auf sie wird in der weiteren Arbeit hin und wieder verwiesen. Eine konkrete Analyse und Implementierung würde aber den Rahmen dieser Diplomarbeit sprengen.

Energiesparendes Routing

Beim Associativity Based Routing (ABR) wird der Batteriepegel einzelner Geräte einbezogen. Das Protokoll versucht bei der Netzwerkpartitionierung zu vermeiden, dass Knoten mit geringem Batteriestand an zentrale Positionen im Routing kommen. Sie werden nur dort als Zwischenknoten verwendet, wo es unvermeidlich ist.

Andere Protokolle versuchen die Sende-/Empfangszeitpunkte in einem Mesh-Netzwerk zu synchronisieren. Die Geräte müssen nicht mehr ständig auf eingehende Daten warten, sondern können sich bzw. die Netzwerkkarte für kurze Zeiträume in einen Ruhezustand versetzen. Weitere interessante Texte zum energiesparenden Routing sind unter [24] oder [25] zu finden.

Signal-stabiles Routen

Es gibt Routingprotokolle, welche die Signalstärke eines Links beachten. Reduziert sich die Signalstärke einer Verbindung zwischen zwei Knoten, so deutet dies darauf hin, dass sich die beiden Knoten voneinander entfernen. Das *Signal-Stability-Based Adaptive Routing*-Protokoll (SSA) versucht in diesem Fall frühzeitig einen alternativen Pfad zu finden.

Quality of Service

Der Protokollentwurf *Quality of Service in OLSR* (QOLSR) [26] erweitert OLSR um Algorithmen für QoS. Jeder Netzknoten misst Bandbreite, Verzögerung der Link-Strecken und den Paketverlust zu den direkten Nachbarn. Auf Basis dieser Informationen werden die Routen kalkuliert und Multi-Point-Relays bestimmt. QOLSR ist zur Zeit noch in einem frühen Stadium der Entwicklung. Die Software steht unter GPL,

unterstützt IPv6 und arbeitet unter Linux.

2.3.5 Auswahlkriterien für den Mesh-Einsatz

Die Entscheidung, ob ein pro- oder reaktives Routingverfahren zu verwenden ist, kann nur für ein konkretes Einsatzszenario getroffen werden. Reaktive Protokolle verbrauchen nur Bandbreite, wenn eine Route gesucht wird. Die Netzwerkbelastung beim Suchen ist durch das Fluten mit Route-Request sehr hoch und bis zum Finden vergeht eine bestimmte Zeit. Bei proaktiven Protokollen entsteht eine beständige Netzwerklast, die Routen sind aber ständig aktuell und verfügbar. Für eine Entscheidung muss also ermittelt werden, wie groß das Netzwerk wird, welche Dynamik die Knoten zeigen, wie hoch das Datenaufkommen im Bezug zur maximalen Bandbreite werden kann oder welche Anforderungen an die Latenzzeit bei einem Verbindungsaufbau gestellt wird.

Ein Vergleich zwischen AODV und OLSR kann unter [27] gefunden werden. Ergebnis dieser Arbeit ist, dass das reaktive AODV-Protokoll bessere Performance in Netzwerken mit statischen Verbindungen und wenigen Kommunikationsverbindungen erreicht. Es benötigt weniger Ressourcen als das proaktive Protokoll OLSR und kann damit auch in ressourcenbeschränkten Geräten eingesetzt werden.

2.4 Mesh-spezifische Probleme

Resultierend aus der Datenübertragung per Funk ergeben sich einige Problematiken, welche in kabelgebundenen Netzwerken nicht bekannt sind. Die bekannten Ad-hoc-Routingprotokolle, die angepasst in Mesh-Netzwerken eingesetzt werden, haben im Allgemeinen ebenfalls diese Probleme, die ich kurz nennen will.

2.4.1 Funkbasierte Datenübertragung

Die Funkübertragung von Daten erfolgt auf bestimmten Kanälen (Frequenzband) und mittels eines Modulationsverfahrens. Bilden verschiedene Computer ein funkbasiertes Netzwerk, kommunizieren (meist) alle Computer auf dem selben Kanal. Die für die einzelnen Knoten zur Verfügung stehende Bandbreite nimmt mit der Anzahl der Knoten ab, da die Maximalbandbreite auf alle sendewilligen Stationen aufgeteilt wird. Eine Implementierung von Quality-of-Service-Garantien (QoS) und somit die Bereitstellung von Breitbanddiensten ist in einem Funknetzwerk besonders aufwendig.

Funkverbindungen haben den Nachteil, dass bei einer Frequenz höher als 100 MHz eine Sichtverbindung (line of sight) ab einer geringen Reichweite notwendig ist. Beim

802.11-Standard können Daten innerhalb von Gebäuden (keine Sichtverbindung) nur bis etwa 25m übertragen werden. Mesh-Netzwerke wurden entwickelt, um diese Einschränkung aufzulösen (Umleitung von Datenpaketen über andere Knoten), jedoch bleibt der grundlegende Fakt, dass zwei Nachbarknoten, die kommunizieren wollen, "sich sehen" müssen.

Eine funkbasierte Netzwerkkarte kann entweder senden oder empfangen. Dies spielt bei Mesh-Netzwerken eine bedeutende Rolle, da hier Knoten Daten weiterleiten. Für eine Datenübertragung zwischen zwei Knoten und einem Zwischenknoten verdoppelt sich mindestens die Übertragungsdauer, da der Zwischenknoten mit einer Antenne erst die Daten empfangen muss und anschließend senden kann.

2.4.2 Hidden-station- / Exposed-station-Problem

Das *hidden-station-Problem* erscheint dann, wenn zwei oder mehrere Stationen (Abbildung 2.4: Stationen A und C), welche ihr Funkübertragungen gegenseitig nicht bemerken, gleichzeitig an eine dritte Station (B), die sie beide erreichen können, senden. Es entsteht an der Position der dritten Station eine Kollision, ohne das dies die beiden sendenden Stationen bemerken.

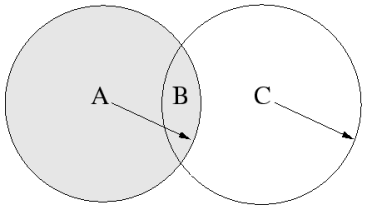

Abbildung 2.4: Hidden-Station-Problem

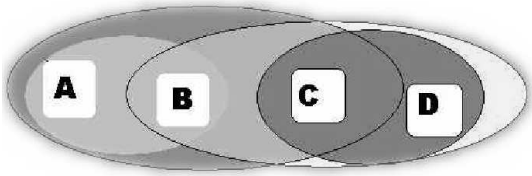

Abbildung 2.5: Exposed-Station-Problem

Zur Vermeidung dieser unbemerkten Kollision wird der *virtual carrier-sensing* Mechanismus, basierend auf dem RTS/CTS (ready to send / clear to send) Verfahren, benutzt. Vor der eigentlichen Übertragung schickt der Sender eine kurze Kontrollnachricht zum Empfänger, dass er Daten zu ihm übertragen will (RTS). Erst nach der Bestätigung duch den Empfänger (CTS), sendet er seine Daten. Empfängt er keine Bestätigung, geht der Sender davon aus, das der Empfänger gerade beschäftigt ist und wartet ein bestimmtes Zeitintervall ab. Sowohl die RTS- als auch die CTS-Kontrollnachricht enthält ein Feld mit der erwarteten Übertragungsdauer. Alle Stationen, die sich in Reichweite dieser beiden Stationen befinden, können diese Zeitspanne lesen und warten mit ihren Übertragungswünschen für diese Dauer.

Das *exposed-station-Problem* bezeichnet eine unnötige Wartezeit einer Sendestation (Abbildung 2.5: Station B will an A senden), welche den Transfer zwischen zwei anderen Stationen (D sendet an C) bemerkt und nicht sendet, da der Kanal aus ihrer Sicht nicht frei ist. Liegt der Empfänger dieser wartenen Station z.B. in der entgegengesetzten Richtung außerhalb der Übertragungsreichweite der anderen beiden Stationen, wäre eine Übertragung möglich (Störbereich von D endet vor der Empfangsstation A). Dieses Verhalten führt somit zu keiner Kollision, verringert jedoch den Datendurchsatz im Netzwerk. Die allgemeine Schwierigkeit besteht darin, zu erkennen, wo Störbereiche enden.

2.4.3 Egoistische Knoten

In herkömmlichen Netzen ist die Infrastruktur meist in eine administrierte Routingumgebung und die normalen Netzteilnehmenden unterteilt. Die Routing-Infrastruktur wird in der Regel durch Passwörter und andere Authentisierungmechanismen geschützt. In Mesh-Netzwerken gibt es dagegen keine Unterscheidung.

Aus dem letzten Punkt ergibt sich auch ein negatives Verhalten von Knoten. Um Energie und Rechenleistung zu sparen, können sie sich weigern, am Routing teilzunehmen. Dieses Verhalten ist in klassischen Netzen irrelevant.

Knoten, welche die Sicherheit und die Funktionsfähigkeit in einem Mesh-Netzwerk beeinträchtigen, können in drei Gruppen unterteilt werden:

- Durch *Fehlerhafte Knoten* werden falsche Topologie- bzw. Routinginformationen verteilt oder Datenpakete nicht weitergeleitet. Das Fehlverhalten kann durch Hard- und Softwareprobleme entstehen. Dieses Verhalten stellt keinen Angriff auf das Netzwerk im klassischen Sinne dar, jedoch sollte eine Sicherheitsstruktur solche Knoten erkennen und sie aus dem Netz ausschließen.
- *Egoistische Knoten* (engl. selfishness nodes) erliegen der Versuchung, Ressourcen zu sparen (Rechenleistung, Bandbreite, Batterie). Sie beteiligen sich nicht oder nur gelegentlich an der Weiterleitung von Daten- und Routingpaketen anderer Knoten. In dem Maße wie die Anzahl der egoistischen Knoten in einem Mesh-Netzwerk zunimmt, nimmt die Gesamtleistung rapide ab. Etwas weniger egoistische Knoten verwerfen vielleicht nicht alle Pakete fremder Knoten, sondern entscheiden aufgrund der verbleibenen Batteriekapazität, ab wann Pakete nicht mehr weitergeleitet werden. Ein Sicherheitssystem muss den Umfang, innerhalb dessen ein solches Verhalten toleriert wird, bestimmen.

- *Böswillige Knoten* (engl. malicious nodes) sind Angreifer des Netzwerkes. Da sie selbst Teilnehmende des Netzwerkes sein können, erlaubt ihnen die Einbindung in die Routing-Struktur viele neue Angriffsverfahren. Im Gegensatz zu egoistischen Knoten, welche die Funktionsfähigkeit des Mesh-Netzwerkes nutzen und erhalten wollen, versuchen böswillige Knoten oft die Funktionsfähigkeit des gesamten Netzes zu zerstören. Auch auf Kosten, dass sie selbst nicht mehr kommunizieren können.

Die Problematik von egoistischen Knoten gibt es in kabelbasierten Netzen nicht, da sich hier alle Routinggeräte kooperativ verhalten, defekt sind oder von einem/einer AngreiferIn kontrolliert werden.

2.4.4 Mindestanforderung an Ressourcen

Es sollen sich möglichst viele Geräte an einem Mesh-Netzwerk beteiligen, um eine redundante und großräumige Abdeckung zu erzielen. Dies betrifft auch Batterie betriebene Geräte wie PDAs, Laptops oder Sensoren in einem Sensornetzwerk. Bei ihnen spielt der Energieverbrauch bzw. energiesparsame Hardware mit geringerer Leistungsfähigkeit eine bedeutende Rolle.

Der Wunsch nach Beteiligung möglichst vieler Geräte steht somit im Spannungsverhältnis zum Wunsch nach Ressourcen-schonenden Protokollen. Ansätze wie energiesparendes Routing oder eine unterschiedliche Klassifikation von Knoten nach leistungsstark oder -schwach, welche beim Routing Beachtung findet, sind notwendig.

2.4.5 Skalierbarkeit und Cross-Layer-Implementierung

Basierend auf den existierenden MAC-, Routing- und Transport-Protokollen ist die Netzwerkperformance (Durchsatz, Ende-zu-Ende-Verzögerung, Fairness) nicht skalierbar mit der Anzahl der Knoten oder der Anzahl der Sprünge (hops) in einem Netzwerk. Dieses Problem kann durch Kapazitätserhöhung der Knoten vermindert werden. Typische Ansätze sind Multi-Kanäle/Frequenzen jedes Knotens oder höhere Übertragungsraten. Diese Ansätze verbessern die Skalierbarkeit nicht wirklich, da sich die relative Performance der erhöhten Übertragungsrate nicht verbessert. Um höhere Skalierbarkeit zu erreichen, ist es notwendig, neue MAC-, Routing- und Transport-Protokolle für Mesh-Netzwerke zu entwickeln [28].

Momentan sind Ad-hoc-Netze im PAN- (Personal Area Network) bis LAN- (Local Area Network) Bereich einsetzbar. Testversuche mit zufriedenstellender Übertragungskapazität wurden mit bis zu 100 Knoten durchgeführt.

Protokollerweiterungen, die nur auf einem OSI-Layer beruhen, können die aktuellen Probleme in Mesh-Netzwerken nicht zufriedenstellend lösen. Die Verbindungsqualität eines Links wird auf der untersten Schicht (Layer 1 und 2) festgestellt. Diese Information sollte bei der Routenermittlung beachtet werden. Das Routing erfolgt jedoch auf Layer 3. Für diese Aufgabe ist demnach ein Layer-übergreifendes Protokoll notwendig [28].

In vielen wissenschaftlichen Arbeiten wird die komplette Neuentwicklung einer cross-layer-Architektur gefordert, damit die Performance von Mesh-Netzwerken signifikant wächst und das Potential der Technik viel besser genutzt werden kann.

2.4.6 IP-Adressierung

Keinesfalls dürfen zwei Teilnehmer in einem Mesh-Netzwerk die gleiche IP-Adresse verwenden. Die automatische Konfiguration über DHCP ist kaum möglich: Solange ein Netzteilnehmer keine eigene IP-Adresse besitzt, funktioniert das Ad-hoc-Routing nicht. Es ist also nicht möglich, dem nächstgelegenen DHCP-Server einen DHCP-Request zu schicken, wenn dieser nicht in unmittelbarer Reichweite ist.

In kleinen Gruppen mögen die gegenseitigen Absprachen und manuelle Zuweisung der Adresse funktionieren - eventuell über eine Webseite, auf der sich die Benutzer registrieren und eine eindeutige Adresse zugewiesen bekommen. Die Vergabe von IP-Adressen in einem größeren Mesh-Netzwerk lässt sich jedoch am einfachsten durch die Verwendung von IPv6-Adressen lösen. Anhand der MAC-Adresse der Netzwerkkarte generiert ein kleines Programm eine eindeutige IP-Adresse, ohne dabei das Risiko einzugehen, dass zwei Teilnehmer die gleiche IP-Adresse verwenden [29].

Die Verwendung von IPv6 hätte den zusätzlichen Vorteil, dass es eine Erweiterung des Protokolls für mobiles drahtloses Internet, genannt MobileIP, beinhaltet. Das Problem bei IPv6 ist die sehr langsame Umstellung auf diese neuere Version. Skeptiker meinen sogar, IPv6 werde sich nie durchsetzen.

Einen anderen Weg geht *WIANA* (Wireless Internet Assigned Numbers Authority) [30], eine kleine gemeinnützige Organisation aus London. WIANA vergibt u.a. an die Nutzer ihrer MeshBox IP-Adressen im Class A Netz, das mit 1 beginnt (1.*.*.*). Dieser Nummernraum war bisher nicht genutzt worden, und Adressen, die mit 1 beginnen, werden von der IANA (offizielle Internetorganisation zur Verwaltung des IP-Nummerraums) nicht als öffentliche IP-Adresse anerkannt und außerhalb des Mesh-Netzwerkes im offiziellen weltweiten Netz nicht gerouted. Unklar ist, ob sich dieses eigenwillige Vorgehen durchsetzen wird.

Ein eigener Adressraum für Mesh-Knoten ist aus Sicht des einfachen Aufbaus von Mesh-Netzwerken sinnvoll. So könnte durch die IP-Adresse bereits deutlich ersichtlich sein, ob der Zielknoten über ein Mesh-Routing oder über Internet-Gateways erreichbar ist.

2.4.7 Physikalischer Zugangsschutz

Aufgrund der Beschaffenheit (Funkwellen) eines drahtlosen Netzwerkes sind die Sicherheitsprobleme, denen ein solches Netzwerk ausgesetzt ist, in keinster Weise mit denen eines leitungsgebundenen Netzwerkes zu vergleichen. Der Datenfluss in einem kabelgebundenen Netzwerk ist durch die fest installierte Infrastrukur unveränderbar vorgegeben. Die Ausbreitung der elektromagnetischen Wellen und der damit verbundene Datenfluss lassen sich dagegen kaum begrenzen. Die Sicherheit eines Mesh-Netzwerkes darf nicht von der Kenntnis der Daten abhängen.

Um in ein kabelgebundenes lokales Netzwerk physikalisch einzudringen, muss der Angreifer in der Regel Zutritt zu einem Gebäude und entsprechenden Zugriff auf zentrale Verbindungsstellen (Hub, Switches oder Netzwerkdosen) haben. Bei einem funkbasierten Netzwerk muss ein Angreifer sich nur in die Nähe einer Funkzelle begeben und sich nicht zwingend innerhalb eines Gebäudes aufhalten.

In Mesh-Netzwerken kommen viele mobile Komponenten zum Einsatz. Im Gegensatz zu fest installierten PCs oder Routern in abgeschlossenen Räumen können diese einfacher gestohlen werden bzw. verloren gehen. Sämtliche Daten können so in unbefugte Hände gelangen (Passwörter oder andere Schlüssel). Ein Sicherheitssystem in einem Mesh-Netzwerk darf nicht auf der physikalischen Sicherheit jeder einzelnen Komponente beruhen. Der Einsatz von einem Netzwerkschlüssel für alle Knoten ist somit nicht möglich.

3 Einsatzszenarien

In der folgenden Analyse werden der Aufbau verschiedener Einsatzszenarien konkret beschrieben und die Sicherheitsanforderungen der beteiligten Personen untersucht. Durch diese Bedürfnisse können im Anschluss bestehende Schutzmethoden ausgewählt oder neue Methoden für konkrete Einsatzszenarien entwickelt werden. Bei gegensätzlichen Interessen ist manchmal nur ein Abwägen der Forderungen möglich.

Anzumerken ist, dass die HerstellerInnen der Hard- und Software ebenfalls zu den Beteiligten bei diesen Szenarien gehören. Geht ihre Bedeutung nicht über das Bereitstellen einer gewünschten Mesh-Architektur hinaus, werden sie nicht explizit erwähnt. Bei Szenarien mit sehr hohen Sicherheitsansprüchen ist darüber hinaus zu beachten, dass Software mittels Entwurfssoftware erstellt wird und diese wiederum auf einem Rechner mit einem Betriebssystem läuft. Geschickte AngreiferInnen könnten hier schon versuchen, Hintertüren in die Software einzubauen, um in der späteren Mesh-Implementierung Zugang zum Mesh-Netzwerk zu erhalten.

3.1 Taktische Netzwerke

Schon seit über 30 Jahren wurden die ersten mobilen Ad-hoc-Netzwerke für militärische Operationen entwickelt. Durch die dynamische Natur einer militärischen Operation konnte auf keine feste Infrastruktur zurückgegriffen werden. Reine Funkverbindungen hatten den Nachteil, dass sich die Funkverbindungen gegenseitig beeinflussten und bei einer Frequenz höher als 100 MHz eine Sichtverbindung (line of sight) notwendig war [15].

Bei der militärischen Kommunikation werden SoldatInnen, Fahrzeuge und weitere Mess-Sensoren mit Minicomputern ausgestattet. Durch die hohe Anzahl der Knoten und die verschiedenen Datenströme wie Sprache oder Bilder ist das Datenaufkommen sehr groß. Die Knoten befinden sich in Bewegung, das Mesh-Netzwerk ist stark dynamisch und Daten müssen über viele Knoten weitergeleitet werden. Die Entfernung zwischen benachbarten Knoten kann einige Kilometer betragen (benachbarte Einheiten).

Durch Einsatz von Repeatern an geografisch günstigen Positionen kann die Verbindungsqualität zwischen verschiedenen Einheiten verbessert werden. Die Spannbreite der verfügbaren Ressourcen in den Knoten ist sehr unterschiedlich. Bei SoldatInnen befinden sich batteriebetriebene leistungsschwache Geräte, wogegen in Fahrzeugen oder am Kommandostützpunkt leistungsfähige Rechner vorhanden sind.

Sicherheitsanforderungen

Die Sicherheitsanforderungen an ein militärisches Mesh-Netzwerk sind besonders hoch. SoldatInnen möchten sicher gehen, dass das Mesh-Netzwerk immer verfügbar ist, wenn sie es brauchen. Dies bedeutet auch, dass die Systemressourcen wie die Batterie geschont werden. Die AbsenderInnen einer Nachricht sollen immer die sein, die sie vorgeben zu sein. Nicht gewollt ist, dass einE FeindIn mithört und dieseR Nachrichten an SoldatInnen senden kann, welche die Systemressourcen unnötig beanspruchen. Wer mit wem kommuniziert (Befehle bekommt), soll dem Gegner und der Gegnerin verborgen bleiben.

Für den Kommandeur oder die Kommandeurin gelten die selben Sicherheitsinteressen: Verfügbarkeit des Netzes, Integrität und Vertraulichkeit der Nachrichten, keine Chance für den Feind oder die Feindin, sich an der Kommunikation zu beteiligen oder Kommunikationsbeziehungen zu erkennen. Das Militär wird die Herstellung der Hard- und Software besonders überwachen und kontrollieren, um Spionage und Schlupflöcher im Mesh-Netzwerk zu vermeiden.

Beteiligte	Sicherheitsinteressen
SoldatIn	Verfügbarkeit der Kommunikation Integrität des Absenders / der Absenderin Echtheit / Integrität der Daten Geheimhaltung der Informationen Unverkettbarkeit der Kommunikationsbeziehung Keine unnötige Belastung der eigenen Ressourcen
Kommandeur / Kommandeurin	Verfügbarkeit der Kommunikation Integrität des Absendenden und der Daten Geheimhaltung der Informationen Unverkettbarkeit der Kommunikationsbeziehung

3.2 Katastrophenschutz

Im Falle einer Katastrophe, wie z.B. einem starken Erdbeben, wird eine Informationsinfrastruktur benötigt, um einen Blick auf die aktuelle Lage zu bekommen, um *Rettungseinsätze* und verschiedene Behörden wie Polizei, Feuerwehr, SanitäterInnen oder Hilfswerke zu koordinieren. Sind alle Fahrzeuge und teilweise Personen mit einem Mesh-Computer ausgestattet, kann dieses Mesh-Netzwerk diese Aufgaben übernehmen. Informationen können per Broadcast an alle Knoten verteilt werden. Eventuell sind eine oder mehrere zentrale Instanzen (Haupteinsatzcomputer) notwendig, um Informationen sicher zu speichern. Diese können sich an einem geschützteren Ort innerhalb des Mesh-Netzwerkes befinden. Neu hinzukommende Rettungskräfte können sofort in das Informationsnetz integriert werden.

Ende Februar 2004 rückten in San Francisco die Rettungskräfte aufgrund eines Katastrophenfalls aus. Beteiligt waren verschiedene Organisationen wie das Marin County Office of Emergency Services, San Francisco Fire Department und weitere Einheiten. All diese Organisationen benutzen eine Vielzahl unterschiedlicher Funksysteme, die normalerweise nicht untereinander kommunizieren können. Doch dieses Mal waren sie mit einer neuen Art von Datenverbindung ausgestattet, mit deren Hilfe sie den Einsatz über Laptops, PDAs und Tablet-PCs koordinieren konnten. Dabei kamen Video-Verbindungen, in Echtzeit generierte Karten zur Verteilung der Einsatzkräfte sowie Multimedia-Messaging zum Einsatz.

Die dreistündige Aktion war ein Probealarm des aus elf kalifornischen Rettungsorganisationen bestehenden Golden Gate Safety Network, in dem eine neu entwickelte Funknetztechnologie namens Mesh getestet werden sollte. Genauer wurde die Technologie von Packethop verwendet, einem Start-up-Unternehmen aus dem Silicon Valley, um ein Ad-hoc-Breitband-Funknetz einzurichten, das die Einsatzkräfte auf der Brücke, zu Lande und zu Wasser miteinander verband, wobei diese unterschiedlichste Standardgeräte benutzten. Es gab keinen zentralen Server und keinen Single Point of Failure und die einzelnen Knoten konnten ihre Netzwerkverbindungen auch außerhalb der Reichweite eines Access-Points aufrechterhalten, weil jeder Knoten gleichzeitig als Repeater und Router für die benachbarten Knoten diente [31].

Die Forscher des Fraunhofer-Instituts für Angewandte Informationstechnik FIT wollen die Arbeit von Feuerwehrleuten bei einem Notfalleinsatz erleichtern, indem die Einsatzkräfte durch tragbare Computer unterstützt werden. Sensoren in der Kleidung sollen den körperlichen Zustand der RetterInnen und Werte der Umgebung erfassen und die Daten zur Zentrale funken. Die elektronischen Helfer werden in die Ausrüstung der Einsatzkräfte integriert (wearable computing). Über Helmdisplays erhalten die Feuerwehrleute Informationen und Befehle.

Zu den zukünftigen Technologien könnten etwa Funkrelais gehören, die kaum größer als eine Münze sind. Die Einsatzkräfte verteilen diese im brennenden Gebäude und es baut sich ein selbst organisierendes Ad-hoc-Funknetz auf, das die Einsatzkräfte bei der Orientierung und Navigation unterstützt. Ob und wie solche Systeme in der Praxis zum Einsatz kommen, wird nicht nur von der technischen Machbarkeit, sondern auch von den permanent knappen Kassen der Gemeinden und Kreisen abhängig sein, die in Deutschland für die Ausrüstung der Feuerwehren zuständig sind [32].

Die Kommunikation von Rettungskräften ist der militärischen Kommunikation ähnlich, hat aber keine so starken Sicherheitsanforderungen. Das multi-hop-Mesh-Netzwerk ist meist nur temporär (Einsatz bei einem Großbrand) und die Größe beschränkt sich durch die Anzahl der beteiligten Einsatzkräfte auf maximal einige Hundert. Die Knoten sind meist in einem beschränkten Gebiet aktiv. Der Datenverkehr ist hoch und die Entfernung zwischen benachbarten Knoten meist gering. Knoten können PDAs oder auch Sensoren sein, welche die Körperfunktionen der Einsatzkräfte überwachen. Gemeinsam ist ihnen, dass die Rechenleistung beschränkt ist und die Geräte durch Batterien ihre Energieversorgung bekommen.

Sicherheitsanforderungen

Neben den Einsatzkräften wie Polizei, Feuerwehr oder technischen Hilfswerken sind teilweise Behörden oder eine Kommandozentrale mit diversen Personen am Mesh beteiligt. Die Einsatzkräfte wollen eine zuverlässige Kommunikation, um eigene (Sensor-) Daten zu übertragen, als auch Befehle und Informationen jederzeit zu erhalten. Sie müssen sicher gehen, dass die Daten wirklich von anderen befugten Einsatzkräften oder von der Kommandozentrale stammen und nicht verfälscht sind. Die Hierarchien bei Einsatzkräften, d.h. wer wem Befehle erteilen kann, ist klar definiert. Je nach Dauer des Einsatzes ist der sparsame Umgang mit den Batterieressourcen notwendig.

In der Kommandozentrale laufen alle Informationen zusammen, das Netz muss zuverlässig funktionieren, die Daten der Einsatzkräfte bzw. der Sensoren sind unverändert und stammen wirklich von den angegebenen AbsenderInnen. Befehle vom Kommandostab sollen nur die adressierten Einsatzkräfte erhalten. Verschiedene Behörden oder "wichtige" Personen wie PolitikerInnen haben ein Interesse, den Einsatz zu beobachten, jedoch nicht direkt in die Kommunikation einzugreifen.

Beteiligte	Sicherheitsinteressen
Einsatz-kräfte	Verfügbarkeit der Kommunikation Integrität des Absenders / der Absenderin Nur Befugte dürfen kommunizieren / Befehle erteilen Nachrichten sind unverändert Geringe Belastung der eigenen Ressourcen
Kommando-zentrale	Ständig verfügbare Kommunikation Integrität der AbsenderInnen Unveränderte Daten Sicherer Nachrichtenversand an einzelne Personen / Gruppen
Behörden (Bürger-meisterIn)	Bei Bedarf verfügbare Kommunikation Daten sind unverändert

3.3 Verkehrsinformationen und -dienstleistungen

Ein neues Transportsystem in Portsmouth bietet ein Echtzeit-Informationssystem der öffentlichen Buslinien. Die Technologie, welche die Mesh-Technik nutzt, verteilt die Informationen an individuelle Empfänger in Bussen und Haltestellen. Dies reduziert die Risiken eines Systemausfalls oder Informations-Flaschenhälse bei zentralen Servern. Insgesamt 36 ausgerüstete Busse bieten nun verschiedene Dienstleistungen an, z.B. Informationen, wo sich die Busse zur Zeit befinden und wie lange die Wartezeit an den Haltestellen voraussichtlich beträgt. Touchscreens in den Bussen bieten darüber hinaus Wetterinformationen, offene Jobs usw. an. Das System ist bedeutend kostengünstiger als ein vergleichbares GPRS-Netzwerk. Allerdings wird GPRS noch in Gebieten benutzt, welche durch das Mesh-Netzwerk nicht abgedeckt sind [33].

Die Sicherheit und die schnellere Fortbewegung von *Fahrzeugen im Straßenverkehr* ist Ziel einer Studie von Donald Wilkins [34]. Grundlage des Szenarios ist ein Mesh-Netzwerk zwischen Ampeln und Fahrzeugen. Der Verkehrsfluß wird auf Grundlage der Position und Geschwindigkeit der Mesh-Fahrzeuge optimiert.

In den sogenannten *vehicular ad hoc networks* (VANETs) verfügt jedes Fahrzeug über eine Möglichkeit, Warnmeldungen per Funk an benachbarte Fahrzeuge zu senden. In einem definierten Bereich, leiten diese die Information an folgende Fahreuge weiter. Das Routing beschränkt sich auf Broadcast in einem bestimmten Radius um den initialisierenden Knoten. Das bedeutet, dass keine expliziten Routen gesucht werden müssen. Da nur kurze Meldungen versandt werden, ist das Datenaufkommen gering. Die Mobilität der Knoten reicht vom Stillstand bis zu Geschwindigkeiten von

über 130 km/h. Sollte sich diese Technik durchsetzen, geht die Anzahl der Knoten in die Millionen. Die Kommunikation erfolgt jedoch immer nur lokal zwischen wenigen Fahrzeugen. Die Ressourcen der Fahrzeugcomputer sind beschränkt.

Neben den Warnmeldungen in einem VANET ist der Austausch weiterer Informationen zwischen Fahrzeugen oder Infrastruktur wie Ampeln sinnvoll. Die zwischen zwei oder mehreren Fahrzeugen ausgetauschten Informationen können das Verkehrsaufkommen oder den Fahrbahnzustand betreffen. Melden Fahrzeuge ihre Geschwindigkeit und Wegewunsch an Ampeln, so können diese den Verkehrsfluss effektiver gestalten. Umgekehrt können Ampeln oder Bahnübergänge den Zustand "Halt" an Fahrzeuge senden. Bei der Kommunikation zwischen Fahrzeugen und Infrastruktur sind weitere Dienste, wie das Finden der nächsten Tankstelle, möglich. Infrastrukturkomponenten beteiligen sich als stationäre Knoten am Mesh-Netzwerk der Fahrzeuge. Inwieweit sich z.B. Ampelanlagen mit anderen Ampeln vernetzen und mit welcher Technik dies realisiert wird, ist je nach Anforderung unterschiedlich.

Ende 2004 haben sich Audi, BMW, DaimlerChrysler, Fiat, Renault und Volkswagen zu einem Konsortium zusammengeschlossen, das einen gemeinsamen Standard für die *Car-to-car-Kommunikation* erarbeiten will. Die Idee ist, Informationen über Witterung, Verkehrsfluss und Straßenzustand von Auto zu Auto weiterzureichen. Der Austausch soll über Ad-hoc-Netze auf Basis der Wireless-LAN-Technologie erfolgen. Ihr Auto wird nicht nur wie heute schon besser bremsen, lenken und schalten, sondern auch die Verkehrslage besser einschätzen können als Sie selbst [35].

Andere *Informationsdienste* stellen Sehenswürdigkeiten in der direkten Umgebung vor oder NutzerInnen können sich zu den gesuchten Lokalitäten (nächste Tankstelle, Restaurant, Werkstatt usw.) durch das Mesh-Netzwerk leiten lassen.

Sicherheitsanforderungen

An VANETs sind die HerstellerInnen der Fahrzeuge bzw. der Warncomputer, die FahrerInnen (BesitzerIn des Autos), staatliche Behörden wie die Polizei oder die Fahrzeugzulassungstellen, AnbieterInnen von Informationsdiensten und die BetreiberInnen von Signalanlagen beteiligt. Der / die FahrerIn will sicher gehen, dass die Informationen, welche sie/er erhält, echt sind. Werden falsche Nachrichten bemerkt (bewusst ausgesandt oder durch ein defektes Fahrzeug verursacht), so muss das verursachende Fahrzeug identifiziert werden. Der/die FahrerIn möchte bestimmte Nachrichten wie Fahrtziel oder Geschwindigkeit nur an befugte Knoten versenden, sowie vermeiden, dass Bewegungsprofile von ihr/ihm erstellt werden und er/sie nicht wegen angeblicher Vergehen von der Polizei bestraft wird, die nicht begangen worden sind.

Die FahrzeugherstellerInnen wollen viele Autos verkaufen und die KundInnen von den Sicherheitsvorteilen von VANETs überzeugen. Sie werden sich deshalb hinter die Schutzinteressen der KundInnen stellen und nur in Fällen wie Diebstahl oder Unfällen mit Behörden zusammenarbeiten. Die Polizei hat ein Interesse daran, einen Unfallhergang zu rekonstruieren und möchte ggf. auf die versandten Nachrichten zugreifen und diese konkreten Fahrzeugen zuordnen (ohne dass der/die FahrerIn den Versand abstreiten kann). Falls der Zugang von Fahrzeugen zu Informationsdiensten abgerechnet werden soll, muss der oder die Anbieterin des Dienstes eine Sicherheit für die korrekte Vergütung haben. Verkehrsleitsysteme, welche die Geschwindigkeit oder den Wegewunsch von Fahrzeugen beachten, können den Verkehrsfluss besser steuern, wenn sie die einzelne Fahrzeuge wiedererkennen.

Beteiligte	Sicherheitsinteressen
Fahrer / Fahrerin	Ständig verfügbare Kommunikation Integrität der erhaltenen Nachrichten Nachweis des Erhalts gefakter Verkehrsinformationen und Möglichkeit des Aufdeckens des/der VersenderIn bei einer zentralen Instanz Kein Versenden von Fahrzeuginformationen an unbefugte Knoten Schutz vor der Erstellung von Bewegungsprofilen Keine unberechtigte Denunzierung durch Polizei / andere FahrerInnen
HerstellerIn der Fahrzeuge	Vertreten Schutzinteressen der FahrzeugkäuferInnen Zusammenarbeit mit Polizei bei Unfällen oder Diebstahl
Polizei	Echtheit der Nachricht Aufdecken des Absenders / der Absenderin von Nachrichten Nachweis des Versandes der Nachricht
Betreibende von Warnanlagen	Echtheit der übermittelten Daten Wiedererkennung von Fahrzeugen im Verkehrsleitsystem
AnbieterIn von Infodiensten	Identifizierung der Nutzenden (KundInnen) Korrekte Abrechnung der Leistungen

3.4 Gebäude- und Geländeüberwachung

In der *Umweltüberwachung* werden z.B. Daten wie Luftdruck, Windrichtung oder -geschwindigkeit an verschiedenen Orten im Umland eines Flughafens aufgezeichnet und gesammelt an die Flugsicherung oder die PilotInnen weiter gegeben. Bei anderen Szenerien werden Meeresströmungen oder Erschütterungen der Erde an vielen Meßpunkten aufgenommen. Die Datenübertragung durch feste Verkabelung ist bei einer

großen Anzahl von Sensoren zu aufwendig und kostspielig. Mesh-Netzwerke bieten sich hier als Kommunikationsnetzwerk an.

Sensornetzwerke können auch zur Überwachung von Gebäuden oder Geländen mittels Videokameras eingesetzt werden. Motorola nutzte die Mesh-Technik in der Hurrikan-Saison 2005 in Florida, als Charley durch das Land fegte. Mit dem Motorola-Mesh-Netz wurden Gebiete überwacht, in denen die Gefahr von Plünderungen bestand. Statt ein Dutzend Polizeifahrzeuge auf einem großen Parkplatz voller Lebensmittel abzustellen, wurden Videokameras installiert, die über ein Mesh-Netzwerk ihre Bilder funkten. Zur Überwachung reichte dann ein einziges Polizeifahrzeug, die restlichen BeamtInnen konnten sich um wichtigere Dinge kümmern [36]. In einem anderen Szenario von Tropos Networks aus dem Jahre 2004 sollen komplette Straßenzüge in New Orleans mittels Videokameras in einem Mesh überwacht werden und so die Sicherheit der BewohnerInnen erhöht werden [37].

Bei der *Gebäudevernetzung* (building automation) werden Daten wie Sonnenintensität, Raumtemperatur oder ein-/ausgeschaltete Beleuchtung gemessen und z.B. die Verdunkelung der Räumlichkeiten gesteuert. Eine Datenweiterleitung per Mesh bietet sich an.

Viele Sensoren messen bestimmte Werte und senden diese periodisch oder nach Aufforderung an einen oder mehrere Auswertungscomputer. Die Auswertungsstationen können untereinander Daten austauschen und reagieren auf die Messergebnisse mit diversen Reaktionen. Neben der Möglichkeit eineN MitarbeiterIn per Alarm zu benachrichtigen, werden oftmals andere Geräte wie Ventile oder Lüftung gesteuert.

Sensornetzwerke mit einigen hundert Sensoren existieren bereits. Die Mobilität und somit die Dynamik spielt in einem solchen Netzwerk keine Rolle. Die Entfernung zwischen benachbarten Sensoren und das Datenaufkommen ist gering. Nur Meßergebnisse müssen über mehrere Knoten zum Auswertungscomputer weitergeleitet werden. Sind Routingpfade am Anfang gefunden, muss nur nach einem Ausfall oder neu hinzugefügten Sensoren ein erneutes Suchen nach Routen durchgeführt werden. Die Sensoren werden teilweise mit Batterien betrieben und die Ressourcen sind auf das Wesentliche beschränkt. Die Auswertungsstationen sind leistungsstärkere Computer.

Die Geländeüberwachung per Videokameras ist der Gebäudevernetzung sehr ähnlich. Die Sensoren sind in diesem Fall Kameras und der Datenverkehr steigt durch die Übertragung von Bildern an. Die Kameras sind nicht mobil. Der Abstand zwischen ihnen kann jedoch mehrere 100 m betragen.

Sicherheitsanforderungen

Die Beteiligten an einer Gebäudevernetzung oder -überwachung sind die NutzerInnen (MieterInnen in dem Gebäude), die Wartungsfirma bzw. einE externeR BetreiberIn und ggf. die von den Videokameras aufgenommenen Personen. EinE BetreiberIn möchte die Geräte eindeutig identifizieren und verhindern, dass fremde Knoten Daten in das Netzwerk liefern und so nicht gewünschte Reaktionen verursachen könnten. Dies bedeutet auch, dass die Daten bei der Übertragung nicht verfälscht werden dürfen. Die Energie von batteriebetriebenen Sensoren ist bei langem Einsatz zu schonen.

Ein Interessengegensatz entsteht bei der Videoüberwachung aus dem Wunsch der gefilmten Personen, nicht beobachtet zu werden und dem Schutz von Geländen und Gebäuden durch die Videoaufzeichnung. Teilweise kann der Persönlichkeitsschutz durch nicht-technische Maßnahmen (Warnhinweis, statt Mitschnitt nur Beobachtung der Bilder) gewährleistet werden.

Beteiligte	Sicherheitsinteressen
Betreiber / Betreiberin der Sensoren	Ständig verfügbare Kommunikation Eindeutige Identifizierung der AbsenderInnen Nur befugte Geräte dürfen kommunizieren Nachrichten sind unverändert Geringe Belastung der Ressourcen von Sensoren
Von Sensoren erfasste Personen	Schutz der eigenen Privatsphäre

3.5 Internet Service Provider

Erste Feldversuche in Großstädten haben lokale Energieversorger unternommen. Sie integrierten die Funknetzwerkkarten in den Masten der Straßenbeleuchtung. Nutzer und Nutzerinnen konnten diese sowohl als APs benutzen, als auch selbst ein Knoten in diesem Mesh-Netzwerk werden. Die Straßenlaternen vernetzten sich untereinander in einem Mesh. Die erforderliche Energieversorgung ist bei Straßenlampen von sich aus vorhanden (von Gaslaternen einmal abgesehen). Der Anbieter Metricom (inzwischen im Konkurs) baute in 21 US-amerikanischen Städten ein solches Mesh-Netzwerk auf. *Microcell Radios* befanden sich auf Straßenlaternen [38].

Cisco hat Ende 2005 seine erste WLAN-Mesh-Lösung vorgestellt, mit der man stadtweite WLAN-Netze errichten kann. Die Cisco-Lösung wird derzeit in zwei US-Städten aufgebaut: in Dayton, Ohio sowie in Lebanon im Bundesstaat Oregon. Sogar Google ist in die drahtlose Mesh-Vernetzung in einigen US-Städten eingestiegen. Als erster

Ort kommt aller Voraussicht nach Googles Heimatstadt Mountain View in den Genuss.

Drahtloses Internet soll den Gästen der Olympischen Spiele 2008 in Peking helfen, sich in der Stadt zurechtzufinden. Das Unternehmen Capinfo aus Hong Kong hat eine Software entwickelt, die es möglich machen soll, mit Smartphones im Internet zu surfen - um Informationen über die Stadt abzurufen. Die Technik basiert auf Wimax, dem Nachfolger von WLAN. Zudem will Capinfo die angeschlossenen Geräte nutzen, um ein so genanntes Mesh-Netzwerk aufzubauen. Dabei wird jedes Gerät selbst zum Mini-Sendemast, an dem wiederum weitere Geräte eine Internet-Verbindung zum nächsten Funkmast erhalten [39].

Eine großflächige und kostengünstige Internetversorgung ist durch eine Architektur mit zwei Ebenen möglich. Mehrere APs bieten NutzerInnen im Infrastrukturmodus einen Zugang an. Die APs vernetzen sich untereinander in einem Mesh, so dass nicht jeder AP einen kabelbasierten Internetzugang braucht. Die APs im Mesh-Netzwerk sind nicht mobil[1] und besitzen einen Stromanschluss (auch über PowerOverEthernet). Die Entfernungen sind meist nicht größer als 100 m. Das Datenaufkommen variiert je nach Nutzungsverhalten von wenig bis sehr hoch. Die Anzahl der Knoten beträgt nicht mehr als einige Dutzend.

Einige BetreiberInnen gehen noch einen Schritt weiter als die einfache Mesh-Backbone-Vernetzung und versuchen, durch Mesh-Netzwerke der NutzerInnen den gesamten Bereich zu vergrößern. Ähnlich dem letzten Absatz, bauen die BetreiberInnen ein Mesh-Netz mittels Knoten auf Laternen, Tankstellen oder ähnlichen Gebäuden auf. Diese werden *Transfer Access Points* (TAPs) genannt. NutzerInnen erhalten die Zugangssoftware mit einer Mesh-Funktionalität, so dass die Nutzenden ihrerseits ein Mesh-Netz aufbauen. Dies ist ein zweistufiges Netz mit geringer Mobilität und mit teilweise sehr vielen Knoten und Datenverkehr. Die Knoten der NutzerInnen sind teilweise leistungsstarke Computer, aber auch tragbare Geräte wie PDAs.

Der *4G Access Cube* des Hamburger Unternehmens 4G Systems beruht auf diesem Verfahren [40]. Als programmierbares LINUX-Gerät bietet der *mesh cube* alles Notwendige, um die heutigen Anforderungen an einen AP zu erfüllen. Er kann als AP und Mesh-Knoten eingesetzt werden und die Elektronik ist in einem kleinen Würfel untergebracht. Der Energieverbrauch beträgt 4 Watt und ein modifiziertes Mobilemesh-Protokoll wird eingesetzt. Die Daten können neben einem TP-Anschluss auch über UMTS oder GPRS weitergeleitet werden.

Ein konkretes Beispiel für den Einsatz der Access Cube ist die kabellose Ticketkon-

[1] Kommen z.B. in Bahnzügen solche Mesh-Netzwerke zum Einsatz, so sind die APs relativ zueinander nicht mobil.

trolle beim VfL Osnabrück. Diverse Einlasskontrollen konnten auf eine zentrale Datenbank zugreifen, ohne das eine spezielle Infrastruktur im Stadion verlegt werden mußte.

Sicherheitsanforderungen

Hauptbeteiligte an dem ISP-Szenario sind die BetreiberInnen der APs / TAPs, welche die Verantwortung für die Mesh-Software tragen, sowie die NutzerInnen des Mesh-Netzwerkes. Die BetreiberInnen möchten eine korrekte Abrechnung der von ihnen erbrachten Dienstleistungen. Es muss sichergestellt werden, dass nur berechtigte NutzerInnen Verbindungen über die APs / TAPs aufbauen können.

Die NutzerInnen wollen eine bestimmte Bandbreite des Netzes zu jeder gewünschten Zeit nutzen. Sie wollen anonym vor anderen Beteiligten im Netzwerk agieren. Ihre versandten Daten sollen nur von der/dem EmpfängerIn entschlüsselt werden können und die Kommunikationsbeziehung ebenfalls vor allen anderen verborgen bleiben. Die Daten auf dem privaten Rechner bzw. die Kontrolle der Funktionsweise will der/die NutzerIn behalten. Dies verlangt beim Einsatz der Mesh-Software, welche direkt auf dem Rechner des Kunden / der Kundin installiert wird, besonderes Vertrauen.

Beteiligte	Sicherheitsinteressen
Betreiber / Betreiberin der APs / TAPs	Ständig verfügbare Kommunikation Identifizierung berechtigter NutzerInnen Korrekte Abrechnung der erbrachten Dienstleistungen Keine Gefährdung der Sicherheit durch Diebstahl eines AP/TAPs
NutzerInnen	Faire Beteiligung an der verfügbaren Bandbreite Anonymität vor anderen Personen (ggf. vor BetreiberIn) Schutz der übertragenen Daten Schutz der Kommunikationsbeziehung Keine Kosten für nicht genutzte Dienstleistungen Schutz der privaten Daten auf dem eigenen Rechner, Vertrauen in korrekte Funktionsweise der Mesh-Software

3.6 Heimvernetzung

Die Heimvernetzung ist ein lokales Mesh-Netzwerk von Sensoren, Computern und Multimedia-Geräten. Die Geräte stehen meist dicht beeinander und sind nicht mobil. Die geringe Anzahl der Knoten und das beschränkte Gebiet ermöglichen ein einfaches Routing. Meist können die Geräte mit einem hop (Sprung) miteinander kommunizieren. Die meisten Geräte verfügen über einen Stromnetzanschluss und sofern erfor-

derlich, können sie mit leistungsstarker Hardware ausgestattet werden. Über einige Knoten können externe Dienste wie Pizzabestellungen genutzt werden.

Die *Heimvernetzung* hat eine ähnliche Funktion, wie der der Gebäudevernetzung. Im sogenannten intelligenten Haus werden Daten gesammelt und in aufbereiteter Form an die BewohnerInnen weitergegeben. Denkbar sind auch automatisierte Reaktionen, z.B. wenn der Kühlschrank als Sensor den Mangel an bestimmten Lebensmitteln meldet und eine Bestellung beim örtlichen Lieferanten ausgelöst wird. Der Türöffner wird durch das Haustelefon oder eine Fernbedienung gesteuert. Die Bilder der Kamera am Eingang können auf einem Fernseher betrachtet werden.

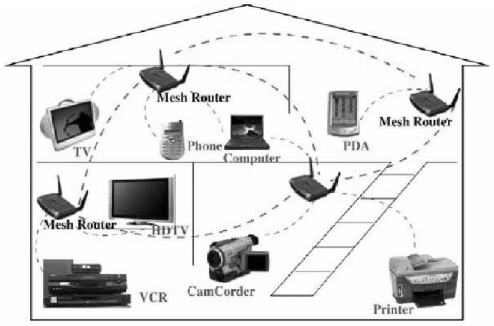

Abbildung 3.1: Mesh für multimediale Heimvernetzung

Sicherheitsanforderungen

Neben der/dem MieterIn in der Wohnung sind diverse HerstellerInnen der Geräte, InstallateurInnen bzw. Wartungsbefugte und ggf. externe DiensteanbieterInnen an der Mesh-Heimvernetzung beteiligt. Die Nutzerin / der Nutzer möchte ein funktionierendes Netzwerk und den Schutz der übertragenen Daten. Weder die NachbarInnen noch die Wartungsfirma soll erfahren, welche Daten übertragen wurden. EinE DiensteanbieterIn möchte eine korrekte und nachweisbare Abrechnung für die erbrachten Leistungen. Der/die NutzerIn möchte für keine Leistungen bezahlen, welche sie/er nicht genutzt hat.

Die GeräteherstellerInnen haben ein Interesse, statistische Daten für Verbesserungen ihrer Produkte oder zur Vermeidung von Fehlern zu erlangen und Softwareupdates in die Geräte einspielen zu können. Bei Multimedia-Geräten spielt Copyright eine besondere Rolle. So könnte ein weiteres Interesse darin liegen, keine Schutzverletzungen zuzulassen. Das könnte bedeuten, dass ein DVD-Player die Daten an einen Fernseher senden kann, aber nicht zum Computer, da hier der Film kopiert werden könnte.

Beteiligte	Sicherheitsinteressen
KäuferIn / NutzerIn	Funktionierende Kommunikation Beteiligung nur berechtigter Geräte Schutz der Inhaltsdaten Keine Ausgaben für nicht erbrachte Leistungen durch Externe (Authentizität der Nachrichten/Bestellungen)
HerstellerIn (Wartungs-firma)	Informationen über Fehler ihrer Geräte und Möglichkeit, Updates einzuspielen Vertretung der Schutzintressen (copyright) der InhaltsanbieterInnen
Dienst-leisterIn	Korrekte Abrechnung erbrachter Leistungen Nutzung nur durch berechtigte NutzerInnen

3.7 Community Mesh Networks

Die Entwicklung militärischer Mesh-Produkte war nicht öffentlich und auf einen begrenzten Wissenschaftsbereich beschränkt. Im Gegensatz dazu wurde ab dem Jahr 2000 durch verschiedene Initiativen versucht, offene und öffentliche Mesh-Funknetzwerke aufzubauen. Als Community-Mesh-Netzwerke verstehe ich gemeinschafliche, nicht kommerzielle Vernetzungsprojekte. Motivation dafür war die Suche nach Alternativen zu teuer gemieteten Standleitungen von ISP und ein gemeinschaftlich organisierter Aufbau von Nachbarschaftsnetzen mit eigenen Inhalten. Dieser Gründungsboom hält bis heute an.

Eines der ersten universitären Netzwerke war das *MIT RoofNet Projekt* in Cambridge (USA) [41], welches Studierenden des MIT einen breitbandigen Internetzugang ermöglichen sollte. Seit 2000 wurden die ersten 802.11b/g Mesh-Knoten aufgebaut und aktuell sind 40 Knoten am Mesh beteiligt. Über drei Gateway-Knoten gelangen die Studierenden in andere Netze.

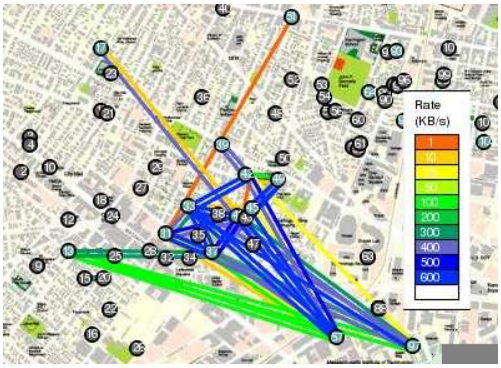

Abbildung 3.2: Roofnet - Auszug aktive Knoten (15.9.2005)

In den letzten Jahren hat sich auf der ganzen Welt eine aktive *Freifunk-Szene* [42] entwickelt. In Deutschland entstanden sie vor allem in größeren Städten, in denen keine breitbandige Internetversorgung vorhanden war (DSL ist aufgrund eines Glasfaseranschlusses nicht

möglich). Als Technik wurden ältere Computer mit 802.11-Interface oder Linux programmierbare Access-Points wie der *Linksys WRT54G* oder *ASUS SpaceLink WL-500g* benutzt. Als Software kamen entweder Eigenentwicklungen oder größere freie Softwarepakete wie *MobileMesh* [18] oder *Locustworld MeshAP* [43] zum Einsatz. Seit Mitte der 90er Jahre wird z.B. in Berlin über ein stadtweites freies Netz auf Grundlage der Mesh-Architektur gegrübelt. Neben dem *Computer Chaos Club* (CCC) sind Initativen wie das *BerlinBackbone* [3] oder der jährliche Kongress *Wizards of OS* treibende Kräfte, die Technik und Software ständig zu verbessern.

Sowohl das *Roofnet* Projekt als auch *Locustworld* bieten Live-CDs (ISO Image) auf Linux-Basis an. Das Mesh-Netzwerk kann dadurch ohne spezielle Installation getestet werden und ältere Computer können energiesparend ohne Festplatte als Mesh-Knoten betrieben werden.

Community-Netze haben das Ziel, eine großflächige Netzbereitstellung zu ermöglichen. Die NutzerInnen sind gleichzeitig die BetreiberInnen und gemeinschaftlich wird ein breitbandiger Internetzugang gemietet. Oft werden in Community-Netzen zusätzliche Dienste wie Telefonie, Radiostream oder eigene Informationen auf Gemeinschaftsservern integriert. Die Anzahl der meist nicht mobilen Knoten reicht aktuell bis 100. Die Entfernung zwischen benachbarten Knoten kann teilweise recht groß werden (verschiedene Häuser), so dass nur durch den Einsatz von Richtantennen eine Verbindung möglich ist.

Sicherheitsanforderungen

Im Prinzip sind alle Beteiligten gleichzeitig NutzerInnen, die ein Interesse an der Funktionsfähigkeit des Netzes, fairen Verteilen der Systemressourcen und dem Schutz sämtlicher Transferdaten als auch der lokalen Daten haben. Die Personen, welche für den Breitbandinternetanschluss verantwortlich sind, möchten wegen gesetzeswidriger Taten Anderer nicht haftbar gemacht werden. Wird der Internetzugang von einer einzelnen Person gestellt und bezahlt, gibt es hier meist das Interesse, jederzeit einen bestimmten Anteil der Bandbreite selbst nutzen zu können oder einen Teil der Kosten für den Internetzugang von den NutzerInnen erstattet zu bekommen.

Beteiligte	Sicherheitsinteressen
NutzerInnen	Faire Beteiligung an der verfügbaren Bandbreite Anonymität vor anderen Personen Schutz der übertragenen Daten Schutz der Kommunikationsbeziehung Keine Kosten für nicht genutzte Dienstleistungen Schutz der privaten Daten auf dem eigen Rechner

3.8 Zusammenfassung

Die im letzten Abschnitt genannten Schutzinteressen lassen sich grob in vier Kategorien einteilen: *Identifizierung* der Knoten, *Inhaltsschutz* der übertragenen Daten, *Geheimhaltung der Kommunikationsbeziehung* und *Verfügbarkeit* der Kommunikation. Darüber hinaus gab es weitere Punkte wie *Copyright*, *korrekte Abrechnung*, *geringe Belastung der Ressourcen* und *Vertrauen* in die eingesetzte Software.

Sehr auffällig ist die unterschiedliche Komplexität der einzelnen Szenarien. Dies betrifft sowohl die Sicherheitsinteressen als auch die Grundstruktur des Mesh-Netzwerkes, in welches die Sicherheitsarchitektur eingebettet werden muss. Folgende Tabelle gibt einen Überblick über diese Struktur. Die Einordnung "+++" bedeutet hoch bzw. stark, "++" mittelmäßig und "+" gering in Relation zu den anderen Szenarien.

	Militär	Rettungs-kräfte	Verkehr	Gebäude	ISP	Heim	Gemein-schafts-netze
Anzahl Knoten	+++	++	+++	++	++	+	++
Entfernung zw. Knoten	+++	+	+	+	++	+	++
Mobilität	+++	+	+++	keine	+	+	+
Routing-anforderung	+++	++	+	+	+++	+	+++
Daten-aufkommen	+++	++	+	+	+++	+++	+++
Knoten-ressourcen	+	+	++	+	++	+++	+++

Dies bedeutet, dass eine Sicherheitsarchitektur, welche auf alle Szenarien angewen-

det werden soll, nur unnötig aufwendig und damit wenig effizient sein kann. Die Entscheidung, ob ich ein pro oder reaktives Routingprotokoll verwende, ergibt sich aus der Netzwerkstruktur. Die Knotenressourcen entscheiden darüber, ob ich aufwändige (asymmetrische) Kryptographie verwenden kann.

Der zweite Punkt, welcher gegen eine einheitliche Sicherheitsarchitektur spricht, sind die unterschiedlichen Sicherheitsanforderungen. Mechanismen, wie die Überwachung, ob sich alle Knoten kooperativ verhalten, sind nicht in allen Szenarien notwendig (z.B. Gebäude- oder Heimvernetzung). Das Routing von Warnmeldungen im VANET erfolgt per Broadcast[2]. Routen müssen nicht gefunden werden, ein sicheres Routingprotokoll ist damit überflüssig.

In der folgenden Arbeit werde ich keine *einheitliche* Sicherheitsstruktur entwickeln, sondern die verschiedenen Klassen von Szenarien einzeln betrachten. Wie sich durch die im nächsten Abschnitt vorgestellten Grundtechniken zeigen wird, können einige Szenarien bereits mit bekannten und "wohl untersuchten" Methoden geschützt werden.

[2] Da Warnmeldungen auch über mehrere hops per broadcast weitergeleitet werden, sehe ich VANETs als Mesh-Netzwerke, auch wenn diese Zuordung grenzwertig ist.

4 Allgemeine Schutzmethoden

In den vergangenen Abschnitten wurden die grundlegenden Anforderungen an eine sichere Mesh-Architektur eines bestimmten Szenarios definiert und einzelne Schutzziele aufgelistet. Im Folgenden werden nun einige grundlegende Algorithmen und Verfahren der Kryptographie kurz vorgestellt, mit deren Hilfe Sicherheitsmechanismen in einem Mesh-Netzwerk implementiert werden können. Desweiteren wird ersichtlich, welche Szenarien bereits mit einfachen Sicherheitsmethoden gesichert werden können. Falls diese nicht ausreichend sind, werden Knackpunkte und die weiteren Anforderungen an eine sichere Mesh-Architektur klar erkennbar.

4.1 Datenverschlüsselung

Vertraulichkeit der Daten wird durch Verschlüsselung erreicht. Grundsätzlich gibt es zwei Verfahren - symmetrische Verschlüsselung mit einem geheimen Schlüssel und asymmetrische Kryptographie mit öffentlichen und geheimen Schlüsselpaar.

Bei der *symmetrischen Kryptographie* wird ein Schlüssel zum Ver- und Entschlüsseln benötigt[1]. Die Sicherheit beruht auf der Geheimhaltung der Schlüssel und nicht auf der Geheimhaltung des Verfahrens. Jeweils zwei KommunikationspartnerInnen brauchen einen geheimen Schlüssel, der vorher auf einem sicheren Kanal ausgetauscht werden muss. Bekannte Verfahren sind DES (bzw. Tripple-DES) oder AES.

1976 beschrieben Whitfield Diffie und Martin Hellman eine neue Klasse von *asymmetrischen Kryptosystemen*, die mit Schlüsselpaaren arbeiten. Zu jedem Schlüsselpaar gehören ein öffentlicher (public key) und ein geheimer Schlüssel (secret key). Nachrichten können nun verschlüsselt werden, indem sich der Sender den öffentlichen Schlüssel des Empfängers / der Empfängerin besorgt (öffentliche Verzeichnisse). Die Aktualität und die Authentizität des öffentlichen Schlüssels muss sichergestellt werden. Bekannte Algorithmen sind RSA oder ElGamal.

Beide Verfahren erlauben unter Beachtung einiger Sicherheitsaspekte eine vertrauliche Kommunikation. Die Stärke der symmetrischen Kryptographie beruht auf der

[1] Dies umfasst auch Verfahren, bei denen sich der eine Schlüssel einfach aus dem anderen ableiten lässt.

sehr hohen Verarbeitungsgeschwindigkeit (Faktor 1000 gegenüber asymmetrischen Verfahren). Für asymmetrische Verfahren werden leistungsstarke Geräte vorausgesetzt. Nachteil des symmetrischen Verfahrens ist der Austausch des geheimen Schlüssels vor Beginn einer Kommunikationsbeziehung über einen sicheren Kanal.

Eine Verbesserung bieten *hybride Systeme*, in denen ein symmetrischer Sitzungsschlüssel zufällig erstellt wird. Dieser Schlüssel wird mit dem öffentlichen Schlüssel der EmpfängerIn verschlüsselt; der restliche Nachrichtentext symmetrisch mit diesem Sitzungsschlüssel. Die populäre Verschlüsselungssoftware PGP/ GnuPGP nutzt das hybride System.

4.1.1 Schlüsselaustausch

Das Problem des Austauschs geheimer symmetrischer Schlüssel kann durch *Schlüsselverteilzentren* gelöst werden. Sie stellen eine vertrauenswürdige zentrale Instanz dar, mit der jedeR beteiligte KommunikationspartnerIn einen geheimen Schlüssel besitzt. Wollen zwei Geräte (Personen) miteinander kommunizieren, stellen sie eine verschlüsselte Anfrage an ein oder mehrere Verteilzentren, welche ihnen - wieder verschlüsselt - einen gemeinsamen und ansonsten geheimen Kommunikationsschlüssel zukommen lassen. Das Schlüsselverteilzentrum stellt natürlich ein attraktives Angriffsziel dar.

Das sogenannte *Diffie-Hellman-Protokoll* [44] stellt ein Verfahren des Schlüsselaustauschs dar, bei dem sämtliche Nachrichten von Dritten mitgelesen werden können und welches auf der Schwierigkeit, diskrete Logarithmen in endlichen Körpern zu berechnen, beruht. Die Authentizität der beiden Beteiligten muss jedoch durch andere Verfahren gesichert werden, weil sonst *man-in-the-middle-Angriffe* möglich sind.

4.2 Datenintegrität, Authentizität

Hashfunktionen können Eingaben beliebiger Länge auf Ausgaben fester Länge abbilden. Sie sind nicht umkehrbar. Sie eignen sich damit zur Sicherung der Integrität von Nachrichten. Von Nachrichten wird ein Hashwert erzeugt, welcher mit der Nachricht übertragen wird. Einem Angreifer oder einer Angreiferin darf es natürlich nicht gelingen, einen neuen Hashwert, welcher die Manipulation verdeckt, zu erzeugen. Der Hashwert muss demzufolge geschützt übertragen werden. Erfolgt dies mit einem geheimen Schlüssel, wird von Message Authentication Codes (MACs) gesprochen.

Bei asymmetrischen Verschlüsselungsverfahren wird nicht von Hashwerten der Schlüssel, sondern von *Fingerprints* gesprochen.

MACs sichern neben der Integrität auch die Authentizität von Nachrichten. Die Prüfsumme wird von der Nachricht und dem geheimen Schlüssel gebildet. Ebenfalls kann der Hashwert der Nachricht mit dem geheimen Schlüssel codiert werden.

Die *digitale Signatur* ist das digitale Äquivalent zur eigenhändigen Unterschrift auf physische Dokumente. Sie wird an eine Nachricht gebunden und soll primär die Authentizität und Integrität einer Nachricht schützen. Eingebettet in eine public-key-Infrastruktur ist auch eine Authentifikation (Identifizierung) des Kommunikationspartners / der Kommunikationspartnerin möglich. Häufig werden sichere Hashfunktionen und asymmetrische Verschlüsselung kombiniert - der Hashwert einer Nachricht wird mit dem privaten (geheimen) Schlüssel der Absenderin oder des Absenders chiffriert. Der/die EmpfängerIn kann nun die Unterschrift und damit Integrität und Authentizität mit dem öffentlichen Schlüssel überprüfen.

Integrität und Inhaltsschutz der Nachrichten ist durch bekannte Methoden in einem Mesh-Netzwerk zu erreichen. Bei großen Netzwerken muss jedoch eine PKI eingesetzt werden. Wie diese realisiert wird, hängt wieder von den im Abschnitt "Identifizierung" beschriebenen Voraussetzungen ab.

4.3 Authentifizierung, Identifizierung

In fast allen Einsatzszenarien werden Sicherheitsanforderungen formuliert, welche auf einer eindeutigen Identifizierung der Knoten beruhen. Beispiele sind: Nur Befugte dürfen kommunizieren, Nachweis des Versandes von Nachrichten, Aufdecken des Senders oder Wiedererkennung von Knoten.

Am Anfang der Betrachtungen stellt sich die Frage, was eine Identität ist. Ist sie einem technischen Gerät, einem Netzwerkinterface oder einer Nutzerin / einem Nutzer zugeordnet? Wird die Identität als dauerhaft und unveränderlich angesehen? Kann einE NutzerIn die Identität selbst erstellen oder werden eine bzw. mehrere zentrale Instanzen benötigt?

4.3.1 Wer oder was wird identifiziert?

Bereits Routingprotokolle müssen eindeutige Routing-Identifier verwenden, da sonst keine konsistente Topologie gebildet werden kann. Oftmals bekommen Knoten eindeutige Adressen zugeordnet, welche jedoch veränderlich sind und somit nicht als eindeutige Identifizierung über einen längeren Zeitraum taugen. Durch Spoofing-Angriffe

können zum Beispiel MAC und IP-Adressen gefälscht werden.

Eine Identifizierung über das Netzwerkinterface ist momentan nicht eindeutig. In der Elektrotechnik gibt es erste Ansätze, eine Funknetzwerkkarte durch einen eindeutigen Fingerprint der Funkcharakteristik zu identifizieren. Diese Technik ist sehr neu, erfordert aufwändige Komponenten und ist (noch) nicht in der Praxis einsetzbar.

Da die technische Grundlage für die Teilnahme an einem Mesh-Netzwerk Computer-Knoten sind, ist eine Identifizierung über die Hardware *immer* möglich. In Einsatzszenarien wie der Gebäudevernetzung stehen hinter den Knoten keine Personen. Verantwortlich ist nur einE BetreiberIn. Hier kann die Identifizierung *nur* über die Hardware erfolgen.

Stehen hinter den Knoten verschiedene Personen oder nutzt eine Person verschiedene Knoten, so ist ein individuelles Erkennen nur durch eine Personenidentifikation möglich. Werden verschiedene externe Dienste genutzt, ist eine Kombination von Hardware- und mehreren Personenidentifikationen aus Datenschutzgründen sinnvoll[2].

4.3.2 Eindeutige Identifizierung

Können Knoten selbst Identitäten erstellen und mit wechselnden, nicht korrelierbaren Identitäten im Netz auftreten, laufen diverse Schutzmaßnahmen ins Leere. Um eine eindeutige Identifizierung zu ermöglichen, muss die Identität eines Knotens von einer oder mehreren externen Parteien (Knoten, Personen oder Institutionen) erstellt oder beglaubigt werden.

Dies bedeutet auch, dass zu einer Identität mehrere *Pseudonyme* existieren können. Sofern diese durch eine Instanz oder durch die Zusammenarbeit mehrerer Instanzen aufgedeckt werden können, erfüllt dies die Anforderung der Eindeutigkeit.

4.3.3 Techniken zur Identifizierung

Grundsätzlich erfolgt die Identifizierung einer Netzteilnehmerin / eines Netzteilnehmers über ein Identifizierungsmerkmal, welches eindeutig (einmalig) ist und in das der/die kommunizierende PartnerIn (Knoten oder Person) Vertrauen hat. Dieses Vertrauen muss nicht absolut sein, sondern es kann je nach Einsatz und Sicherheitsanforderung manchmal ausreichend sein und manchmal nicht.

[2] Zum Beispiel zur Wahrung der Anonymität oder die Unverkettbarkeit von verschiedenen Einkäufen.

In kleinen Netzen kann ein gemeinsamer Schlüssel zwischen zwei Kommunikationsknoten, der vor der eigentlichen Kommunikation auf einem sicheren Kanal ausgetauscht worden ist, eingesetzt werden. Sind mehr als zwei Knoten an dem Mesh-Netzwerk beteiligt, müssen zwischen jeweils zwei Knoten individuelle Schlüssel ausgetauscht werden. Der Aufwand steigt mit der Größe des Netzwerkes und ist ab einer bestimmten Anzahl von Knoten nicht mehr effizient. Durch den geheimen Schlüssel werden nicht einer, sondern immer zwei Knoten untereinander identifiziert.

Public-Key-Infrastruktur

Die meisten gängigen Verfahren zur Identifizierung von Knoten in einem größeren Netzwerk kommen um die Verwendung von asymmetrischer Kryptographie nicht herum. Hierbei besitzt jedeR Teilnehmende ein Schlüsselpaar, welches durch eine oder mehrere vertrauensvolle Instanzen (Trusted Third Party) mit der Identität des/der BenutzerIn verknüpft wird. Die TTP erzeugt ein Zertifikat, welches diese Beziehung zwischen Identität und öffentlichem Schlüssel bestätigt. Der öffentliche Schlüssel der TTP muss allen Teilnehmenden zur Verfügung stehen, damit sie das Zertifikat überprüfen können. Zertifikate haben eine Gültigkeitsdauer und können bei Mißbrauch widerrufen werden.

In Bezug auf die Mesh-Architektur widersprechen die zentralen Instanzen dem Ansatz der Dezentralität. Sie sind ein Kommunikations-Nadelöhr, müssen ständig erreichbar sein und stellen ein stark gefährdetes Angriffziel dar. Ein Ansatz, aus diesem Dilemma heraus zu kommen, ist, die Zertifizierung im Netz möglichst breit zu verteilen.

In der Arbeit [45] stellen die Autoren ein Konzept für eine verteilte Certification Authority (CA) vor. Jeder Knoten im Netz ist gleichzeitig Zertifizierungsserver. Sämtliche Nachrichten werden mit einem Public-Key-Verfahren gesichert. Die Validierung der Schlüssel wird über ein *Schwellwertverfahren* gelöst. Die Sicherheit beruht darauf, dass es wohl einige kompromittierte Knoten geben kann, diese Anzahl aber immer unter der Schwelle bleibt, die nötig wäre, um den Schlüssel der (verteilten) CA zu rekonstruieren. Kommen neue Knoten in das Netzwerk, werden die Teilschlüssel neu generiert und verteilt. Problematisch bei diesem Verfahren ist die anfängliche (manuelle) Initialisierung, der hohe Rechenaufwand für die asymmetrische Kryptographie und das Schwellwertschema sowie der hohe Kommunikationsaufwand.

Einen anderen Weg zur Zertifizierung geht die Software Pretty-Good-Privacy (PGP). Hier gibt es keine zentrale Instanz, sondern jedeR TeilnehmerIn kann andere Schlüssel signieren. Eine Signatur sagt lediglich aus, das ein Knoten der Meinung ist, dass ein Knoten zu einer Identität gehört. In wie weit sich ein dritter Knoten auf diese Aussage verläßt, hängt von seinem Vertrauen in den signierenden Knoten ab. Durch ein Geflecht von gegenseitigen Signaturen ergibt sich ein Vertrauensnetz, welches große

Netzwerke abdecken kann. Problematisch ist, dass bereits ein falsches Urteil zu einem Bruch in vielen Vertrauensketten führen kann.

PGP verwendet zum Speichern der Zertifikate zentrale Schlüsselserver. Ein Verfahren, welches das PGP-Konzept umsetzt, aber keine zentralen Schlüsselserver benötigt, wurde in [46] entwickelt. Jeder Knoten speichert die vom ihm erstellten und eine Auswahl weiterer Zertifikate. BenutzerInnen und Zertifikate werden als gerichteter Graph angesehen. Ergibt sich durch die Vereinigung verschiedener Graphen ein Weg zwischen SenderIn und EmpfängerIn, so ist ein transitiver Vertrauensweg über Zwischenknoten gefunden (Shortcut-Hunter-Algorithmus). Eine Garantie für das Finden einer transitiven Vertrauensverbindung gibt es gerade beim Aufbau eines Mesh-Netzwerkes oder in stark dynamischen Umgebungen nicht. Darüber hinaus sagt die Signatur nicht nur aus, dass die Identität zu der vorgegebenen Person gehört, sondern auch, dass die Person ihrerseits gewissenhaft Signaturen von Dritten erstellen kann. Bei PGP waren diese beiden Ansätze voneinander getrennt.

Zusammenfassend läßt sich sagen, dass in Mesh-Netzwerken, bei denen keine Personen hinter den Knoten stehen (Sensoren), einE BetreiberIn über ein manuelles Verfahren den Knoten eindeutige Identitäten zuordnen kann und keine weitere CA braucht. In nicht-dynamischen kleineren Netzen mit recht leistungsstarken Knoten können die Knoten über eine verteilte CA identifiziert werden. Sind die Netze jedoch sehr groß und dynamisch, kommen aus momentaner Sicht nur verteilte aber zentrale Schlüsselserver für die Signierung und Speicherung der Knotenschlüssel in Frage. Dies steht im Spannungsverhältnis zur Dezentralität des Mesh-Ansatzes.

Identitätsbasierte Kryptographie

Identitätsbasierte Kryptosysteme ermöglichen es, aus der Identität einer/eines BenutzerIn direkt deren öffentlichen Schlüssel eines Public-Key-Systems abzuleiten. Damit entfällt die Signatur eines öffentlichen Schlüssels durch eine vertrauenswürdige Instanz als auch die Verteilung durch öffentliche Schlüsselserver.

Damit das System funktioniert, muss der/die EmpfängerIn die Nachrichten wieder entschlüsseln können. Ein geheimer Schlüssel passend zum öffentlichen muss hierzu von einer vertrauensvollen Instanz (private key generator) mittels eines Masterkeys erstellt werden. EinE NutzerIn hat allerdings das Problem, seine/ihre Identität bei Kompromittierung des geheimen Schlüssels zu ändern. Ein bedingter Ausweg ist die Bindung eines öffentlichen Schlüssels an ein Ablaufdatum.

Sichere und zugleich praktikable identitätsbasierte Systeme lassen sich nur schwer entwickeln. Für den Mesh-Einsatz sind mir keine Protokolle bekannt.

4.3.4 Verbindlichkeit

Der Nachweis des Versandes von Nachrichten, kann durch asymmetrische Verfahren gewährleistet werden (solange ein Schlüssel nicht kompromittiert ist). Wenn die Bestellung einer Pizza mit dem geheimen Schlüssel des/der eindeutig identifizierten NutzerIn signiert wird, kann der Lieferservice dies damit nachweisen. Ein aktueller Zeitstempel muss mit signiert werden, damit Replay-Angriffe (Mithören und erneutes Versenden der Bestellung) vermieden werden können. Kann der Service keine signierte Auftragsbestellung vorweisen, können keine Personen ungerechtfertigt "zur Kasse" gebeten werden.

Im Kontext der Mesh-Netzwerke läßt sich also Verbindlichkeit durch digitale Unterschriften lösen, sofern die KommunikationspartnerInnen Vertrauen in die Echtheit der gegenseitigen Schlüssel haben. Dies kann durch eine Kontrolle (Vorlage des Ausweises) oder durch vertrauensvolle dritte Instanzen geschehen.

4.3.5 Anonymität

Der Wunsch des Absenders / der Absenderin einer Nachricht nach Anonymität bzw. der Unmöglichkeit, Bewegungsprofile seines/ihres Knoten zu erstellen, steht im Gegensatz zur eindeutigen Identifizierung. Der Ansatz von Pseudonymen kann einen geringen Schutz für den/die AbsenderIn bedeuteten. Eine oder mehrere zentralen Vertrauensinstanzen signieren nicht nur eine Hauptidentität, sondern viele Pseudonyme, welche ggf. periodisch erneuert werden. Die AbsenderIn wechselt nun in regelmäßigen Abständen und bei Benutzung verschiedener Dienste seine/ihre Pseudonyme. Soll die Hauptidentität eines Knotens aufgedeckt werden, so kann dies z.B. durch einen Gerichtsbeschluss durch die Vertrauensinstanz geschehen.

Sind die HerausgeberInnen der Hauptidentität und der Pseudonyme voneinander unabhängige DienstleisterInnen, erhöht dies die Sicherheit. Durch die Methode des *Identitätsmanagements* für anonyme beglaubigte Identitäten bürgt eine andere Institution, dass einzelne Knoten für bestimmte Dienste berechtigt sind, ohne die Hauptidentität zu veröffentlichen.

Wie deutlich ersichtlich, muss der/die AnwenderIn Vertrauen in die Signierungsinstanz haben. Arbeitet diese mit AngreiferInnen, welche Bewegungsprofile erstellen wollen, zusammen, ist kein Schutz möglich.

In einem Mesh-Netzwerk, in dem die NutzerInnen anonym agieren wollen, können sie dies durch ständiges Erstellen unterschiedlicher Identitäten erreichen. Sobald zusätzliche Interessen, die auf eindeutiger Identifizierung der Knoten beruhen, hinzukommen, kann eine Anonymität nicht mehr erreicht werden. Einige oder viele zentrale Instanzen können eine Hürde schaffen, dass die Identitäten erst bei einem schweren Vergehen aufgedeckt werden können und ein Angriff auf die Pseudonymität erschwert wird.

4.4 Geheimhaltung der Kommunikationsbeziehung

Bei der Geheimhaltung einer Kommunikationsbeziehung geht es darum, dass Außenstehende nicht erkennen können, wer mit wem kommuniziert. Die beiden Kommunikationsknoten sehen sich jedoch bzw. sind voreinander nicht anonym. Es gibt Verfahren, die den Schutz des Senders bzw. der Senderin und des Empfängers / der Empfängerin ermöglichen.

Das einfachste Verfahren ist, ständig verschlüsselte Daten an alle Beteiligten zu versenden. Die Daten sind entweder richtige Nachrichten oder *Dummy-Traffic*. Eine außenstehende Person kann nicht entscheiden, wann und an wen die wichtigen Nachrichten gesandt werden. Das Verfahren verursacht ein enormes Datenaufkommen und jeder Knoten muss alle Daten entschlüsseln, um zu entscheiden, ob sie sinnvollen Inhalt enthalten. Zwei effektivere Verfahren, die von anonymen Proxies und Mixen, werden im Folgenden vorgestellt.

Proxies agieren als zwischengeschaltete Einzelcomputer in einer Kommunikationsverbindung. Möchte ein Knoten zu einem anderen Knoten Kontakt aufnehmen, so leitet er verschlüsselt alle Anfragen an den Proxy weiter (SSL). Dieser entschlüsselt die Anfrage und leitet sie an die Empfängerin bzw. den eigentlichen Empfänger im Klartext oder neu verschlüsselt weiter und agiert als AbsenderIn der Anfrage. Außenstehende die entweder den/die EmpfängerIn oder den/die SenderIn überwachen, bemerken nur eine Komunikation zum Proxy-Server.

Ein Knoten muss Vertrauen in die/den BetreiberIn eines Proxies haben, da dieser die beiden PartnerInnen natürlich kennt. Darüber hinaus kann einE AngreiferIn, welcheR alle Ein- und Ausgänge des Proxies überwacht, die ein- und ausgehenden Daten korrelieren und die Anonymität aufdecken. Der Einsatz von Proxies bietet also nur einen geringen Schutz.

Sofern sich ein Proxy im Mesh-Netzwerk befindet, garantiert er keinen Schutz der Kommunikationsbeziehung, da durch die Funkübertragung einE AngreiferIn in der Nähe des Proxies den gesamten Netzwerkverkehr belauschen kann. Ein Proxy macht so

nur Sinn, wenn er sich in einer geschützten Umgebung oder in einem kabelbasierten Netz befindet.

Die Weiterentwicklung, welche die eben genannten Probleme beseitigt, sind *Mix-Kaskaden*. Wie in [47] ausführlich beschrieben, garantiert dieses Konzept den Schutz der Kommunikationsbeziehung (bis zur Erweiterung der SenderInnen und EmpfängerInnenanonymität), solange viele NutzerInnen den Dienst nutzen und nicht alle benutzen Mix-Computer von einem/einer AngreiferIn beherrscht werden. Die erforderlichen Rechenoperationen sind sehr anspruchsvoll, da Daten mehrmals verschlüsselt bzw. entschlüsselt werden müssen. Die Latenzzeiten steigen ebenfalls an und Mix-Kaskaden sind nur eingeschränkt für Echtzeitanwendungen geeignet. Implementierungen wie die Java Anon Proxy (JAP) [48] nutzen feste Kaskaden, wogegen TOR [49] jeder/jedem NutzerIn ermöglicht, einen Mix zu betreiben.

Sind die beteiligten Mix-Knoten im Mesh-Netzwerk ungünstig verteilt, werden während der Übertragung eines Datenpaketes die Daten "hin und her" übertragen. Der Netzwerktraffic kann sich durch Mixe vervielfachen.

4.5 Verfügbarkeit der Kommunikation

Damit das Mesh-Netzwerk verfügbar ist und die gewünschte Funktionalität erbringt, sind mehrere Voraussetzungen zu erfüllen. Prinzipiell muss eine Kommunikation über Funk möglich sein - benachbarte Knoten müssen in ihren Reichweiten liegen und die benutzten Frequenzbänder dürfen nicht durch starke Störungen belegt sein.

Ist die Funkübertragung möglich, müssen die versandten Daten korrekt und in einem maximalen Zeitintervall weitergeleitet werden. Dies erfordert ein Routingprotokoll, welches das Finden der Routen ermöglicht und jedem Knoten eine bestimmte Bandbreite zur Weiterleitung bietet.

4.5.1 Sichere Funkübertragung

Ist ein Frequenzband gestört, so kann auf ein anderes Frequenzband umgeschaltet werden. Die Mesh-Implementierung muss dabei das Erkennen der Störung und die Umschaltung (die bei mehreren kommunizierenden Knoten erfolgen muss) bewerkstelligen.

Durch fehlertolerierende Datenmodulation und das Übertragen von Daten auf sehr breiten Frequenzbändern kann die Wirkung von Störungen oder DoS-Angriffen eingeschränkt werden. Die Datenrate sinkt, jedoch ist eine Kommunikation trotz Störung

möglich.

Prinzipiell gibt es keine Garantie auf der Implementierungsebene, welche eine Funkübertragung garantiert. Gezielte Störungen einer Übertragungsfrequenz (DoS-Angriffe) stellen eine ernste Gefahr für jedes Funknetz dar.

4.5.2 Sicheres Routing

Wie schon im Kapitel 2 festgestellt, genügen die Routing-Protokolle für kabelbasierte Netze nicht dem Einsatz in einem Mesh-Netzwerk. Durch die andere Mesh-Struktur ergeben sich folgende Ziele für ein sicheres Routing in einem Mesh-Netzwerk [50]:

- Eindeutige Netzwerkkarten-Identifizierung der Knoten (kein Spoofing möglich)
- Außenstehende können keine Routingnachrichten in das Netz einschleusen
- Die Routingnachrichten können nicht unberechtigt verändert werden
- Ein Knoten kann keine Routingschleifen oder unnötig lange Routen erzeugen
- Sicherstellung, dass die Route authentisch und aktuell ist

Zusätzliche Forderungen:

- Nicht authentisierte Knoten können am Routing nicht teilnehmen
- Möglichst geringer Overhead (Netzwerkverkehr, Rechenaufwand in den Knoten)

Anforderungen, wie die Identifizierung oder Integritätsschutz der Routingdaten, überschneiden sich mit den allgemeinen Anforderungen für alle Nachrichten - wie in den letzten Abschnitten beschrieben. Die Ziele eines sicheren Routings gehen aber weit darüber hinaus und die Entwicklung von Mesh-Routingprotokollen gehört zu den Mesh-Knackpunkten und ist eines der am meisten betrachteten Forschungspunkte in diesem Gebiet. Einige bekannte Arbeiten werden im nächsten Kapitel ausführlicher vorgestellt und bewertet.

Eine weitere offene Frage bleibt, was passieren soll, wenn sich *egoistische Knoten* nicht an das Routingprotokoll halten wollen? Ignorieren? Egoistisches Verhalten kann durch Bestrafen von unkooperativem oder durch Belohnung von kooperativem Verhalten vermieden werden.

Werden Veränderungen der Routingdaten durch fehlerhafte oder böswillige Knoten erkannt, müssen diese aus dem Netzwerk ausgesperrt werden. Dies gelingt jedoch nur,

wenn sich möglichst viele (nicht-böswillige) Knoten beteiligen. Eine Variante, dies umfassend zu tun, ist der Einsatz eines Intrusion-Detection-Systems (IDS).

Sowohl Kooperationsprotokolle als auch IDS werden im Kapitel 5 ausführlich erläutert.

4.5.3 Faires Verteilen der Bandbreite

In Mesh-Netzwerken, in denen sich alle Knoten korrekt am Routing beteiligen, kann es bedingt durch die Netzwerktopologie zu unfairer Verteilung der Bandbreite kommen.

Ein Knoten, der mehrere Datenpakete bekommt, muss entscheiden, in welcher Reihenfolge er diese weiterleitet (load balancing). Er kann dabei nicht überprüfen, ob der benachbarte Knoten der Absender der ursprünglichen Daten ist oder ob dieser viele Daten für andere Knoten weiterleitet und damit zu bevorzugen ist. Im Extremfall kann sich ein Knoten zu dem Internet-Gateway verbinden und bekommt dieselbe Bandbreite geboten, wie ein Knoten, der für viele andere Knoten die Daten weiterleitet. Einzeln betrachtet haben diese vielen, weiter entfernten Knoten nur eine sehr geringe Bandbreite. Dieser Effekt der Unfairness tritt erst bei hohem Datenverkehr im Mesh-Netzwerk auf.

In [51] untersuchen die AutorInnen die per-TAP (Transfer Access Point)-Fairness und end-to-end-Performance in Mesh-Netzwerken. Sie entwickeln einen inter-TAP-Fairnessalgorithmus, der ohne Veränderungen am TCP-Protokoll die per-TAP-Fairness erreicht. Diese Arbeit betrachtet nur das Einsatzszenario ISP und bietet keine per-client-Fairness für die NutzerInnen.

Auf der letzten Arbeit aufbauend, entwickeln die Autoren in [52] für das ISP-Szenario einen Algorithmus, der per-client-Fairness garantiert und die Bandbreitennutzung optimiert. Der Arbeit liegt die Annahme zugrunde, dass der meiste Netzwerkverkehr zwischen Clients und Internet-Gateway fließt. Der Gatewayknoten arbeitet als zentrale Instanz und verteilt nach einer Initialisierungsphase Übermittlungs- bzw. Senderechte auf die TAPs, welche sich nach der Anzahl verbundener Clients richten.

Die vorliegenden Arbeiten bieten für Mesh-Netzwerke mit unterschiedlichen Ebenen ein faires Verteilen der Bandbreite. Allerdings wird eine zentrale Instanz, welche Senderrechte vergibt, benötigt.

Sollen die gleichen Daten an viele Knoten gesendet werden, sollte über den Einsatz eines multicast-Routingprotokolls nachgedacht werden, anstatt die Daten dupliziert an

alle zu versenden. Das spart viel Bandbreite.

4.6 Sonstige Schutzmethoden

4.6.1 Geringstmögliche Belastung eigener Ressourcen

Ist die Hardware von Knoten nicht leistungsfähig, haben sie nur einen geringen Speicher und Energieressourcen, so muss die Implementierung des Mesh-Netzwerkes besondere Rücksicht darauf nehmen. Dies betrifft besonders das Routing, die Rechenoperationen für die Kryptographie und die verlangte Verfügbarkeit der Knoten im Netzwerk.

Eine Einbeziehung von verschiedenen Knotenarten (schwacher und leistungsfähiger Knoten) im Routingmodell kann so aussehen, dass nur schwache Knoten in Routen benutzt werden, wenn es keine Alternative über leistungsfähige Knoten gibt.

Neben dem Einsatz von energiesparenden Routingmethoden muss ein Knoten möglichst schnell erkennen können, ob eine Nachricht für ihn bestimmt ist und ob er diese weiterleiten soll. Das sollte geschehen, bevor der Knoten aufwendige Operationen zum Entschlüsseln des Dateninhalts vornimmt. Die Ver- und Entschlüsselungsalgorithmen müssen für den Mesh-Einsatz optimiert werden und - falls möglich - in Hardware-Rechenbausteine implementiert werden.

4.6.2 Schutz eigener Rechner und installierter Mesh-Software

In heutigen Zeiten hoch komplexer Software gibt es keine Garantien für das korrekte Verhalten und den Nachweis, dass keine Schadens- oder Spionageelemente enthalten sind. Die NutzerInnen müssen sich entscheiden, ob sie genug Vertrauen in die HerstellerInnen der Software haben und ob sie diese einsetzen. Dieses Vertrauen kann erhöht werden, wenn die Software von verschiedenen (kommerziell und politisch unabhängigen) Instanzen kontrolliert wird. Noch besser ist der Ansatz, die Quellcodes der Software zu veröffentlichen und somit potenziell jeder/jedem die Chance zu geben, die korrekte Implementierung zu untersuchen.

Jede Software hat Fehler, welche früher oder später entdeckt werden. Geschieht dies, muss über ein Update der Fehler beseitigt werden. Da einer/einem AnwenderIn nicht eine beständige Kontrolle, ob es neuere Softwareversionen gibt, auferlegt werden kann, ist eine integrierte Updatemöglichkeit zu bevorzugen.

4.6.3 Anonyme Kostenabrechnung

Bei einigen Einsatzszenarien muss eine sichere Abrechnung von Dienstleistungen erbracht werden. Möchte der/die NutzerIn Schutzziele wie Unbeobachtbarkeit, Anonymität und Unverkettbarkeit der gesendeten Nachrichten erhalten, können Abrechnungssysteme wie nicht manipulierbare Zähler für die Netzbenutzung oder anonyme digitale Zahlungssysteme eingesetzt werden.

Nicht manipulierbare Zähler werden im Netzanschluss einer Teilnehmerin/eines Teilnehmers untergebracht. Erfolgt das Auslesen des genutzten Traffics in großen Abständen, geben die Zähler nur minimal Informationen über die Verbindungen der Nutzenden preis. Eine Einzelaufstellung der Verbindungen darf der Abrechnungsfirma natürlich nicht vom Zähler zur Verfügung stehen. Der/die NutzerIn ist gegenüber der/dem NetzbetreiberIn nicht anonym, allerdings werden die Informationen über die Verbindungen nicht benötigt (Inhalte, Kommunikationsverbindung) und können durch andere Methoden geschützt werden.

Eine Abrechnung eines Netzanschlusses kann natürlich auch pauschal durch eine NutzerInnen-Gemeinschaft erfolgen. Die individuelle Identifizierung aller Beteiligten gegenüber dem/der NetzbetreiberIn ist nicht mehr nötig.

Protokolle für ein *anonymes digitales Zahlungssystem* werden unter anderem in [53] oder [47] entwickelt. Die Verfahren beruhen auf digitalen Beträgen, bei denen die reale Einlösung bei einer Bank durch deren Signatur gewährleistet wird. Die Protokolle sind recht komplex, geben aber keinen der beteiligten Personen die Chance zu betrügen. Das Sicherheitsverfahren kann bei geringeren Beträgen vereinfacht werden (digitale Briefmarke), da hier ein Missbrauch zwar möglich ist, aber entdeckt werden kann und die Strafe für den/die AngreiferIn gegenüber dem Nutzen zu hoch ist.

4.7 Zusammenfassung

Die Knackpunkte in der Entwicklung einer sicheren Mesh-Architektur für die meisten Einsatzszenarien sind das *sichere Routing*, die *Verhinderung von egoistischem Verhalten* und die *Erkennung von böswilligen Knoten*. Sind böswillige oder egoistische Knoten erkannt, müssen sie aus dem Netzwerk ausgesperrt werden.

Erkennbar ist auch, dass "einfache" Szenarien wie das der Gebäudeüberwachung nicht von den genannten Knackpunkten betroffen sind. Hier kann davon ausgegangen werden, dass alle Geräte mit einer eindeutigen Identität und geheimen symmetrischen Schlüsseln programmiert werden und sich alle Knoten an das Routingprotokoll halten.

Dieses Szenario kann somit ohne spezielle Sicherheitsprotokolle mit bekannten Verfahren abgesichert werden.

Prinzipiell läßt sich sagen, dass alle Schutzmethoden, welche für die Szenarien benötigt werden, vorhanden sind. Die Herausforderung besteht vielmehr darin, sie sinnvoll zu verknüpfen, die Rahmenbedingungen konkret zu umreißen und ein umfassendes Konzept zu entwickeln. Wie dies existierende Forschungsarbeiten leisten, wird im nächsten Kapitel analysiert.

5 Analyse aktueller Forschungsarbeiten

Etwa seit dem Jahr 2000 gibt es verschiedene Arbeiten, die sich mit der Absicherung von Ad-hoc-Netzen (und damit zu großen Teilen auch Mesh-Netzwerken) beschäftigen. Die verschiedenen Projekte betrachten fast immer einzelne Aspekte der Themen *Identifizierung*, *sicheres Routing* und *Verhinderung von Egoismus* (unfairem Verhalten der Knoten). Weitere Aspekte, wie die Verhinderung der *Erstellung von Bewegungsprofilen*, wurden seit 2003 ebenfalls analysiert.

In diesem Kapitel werden verschiedene Forschungsarbeiten analysiert. Sie sind grob in drei Themenfelder Routing, egoistisches Verhalten und IDS eingeteilt. Bis auf die theoretische Arbeit von Frank Kargl [50] werden immer nur Einzelaspekte betrachtet. Im Anschluss an die Analysen wird nochmals bewertet, wie die im Kapitel 3 beschriebenen Szenarien durch die hier vorgestellten Verfahren gesichert werden können.

Im Gegensatz zu akademischen Arbeiten gibt es nur wenige Implementierungen von Protokollen. Oftmals sind diese auch nicht öffentlich dokumentiert (propritär). Eine Untersuchung dieser Implementierungen ist mir nicht möglich.

5.1 Sicheres Routing

Im folgenden Abschnitt werden ausgesuchte Routingprotokolle vorgestellt und analysiert. Die Auswahl wurde so getroffen, dass möglichst viele der aktuell diskutierten Sicherheitstechniken aufgezeigt werden.

5.1.1 Secure AODV

SAODV [54] wurde von Nokia Research entwickelt. Es ist eine Weiterentwicklung des reaktiven AODV-Protokolls mit dem Ziel, verschiedene Angriffe gegen das Routingprotokoll zu verhindern. Grundlage ist eine funktionsfähige PKI und eine sichere Verteilung von Schlüsseln an alle Knoten. Die statischen Teile der Managementnachrichten des Findens von Routen werden signiert. Durch einen Hashchain wird erreicht, dass die

variablen Anteile, wie der *hop-count*, nur vergrößert werden können. Darüber hinaus wird dieser Wert von jedem Zwischenknoten überprüft und vor dem Weiterversand angepasst. Durch dieses Verfahren werden Route-Request und Route-Reply abgesichert. Darüber hinaus signiert der Zielknoten das Route-Reply-Paket.

Durch sogenanntes *Route-Caching* kann normalerweise das Routenfinden verkürzt werden, indem ein Knoten, welcher nicht der Zielknoten ist, auf eine Routenanfrage antwortet, sofern er eine aktuelle Route kennt. Da hier der Zielknoten nicht beteiligt ist und das Route-Reply nicht von ihm signiert werden kann, kommt bei SAODV die *Double Signature Extension* zum Einsatz. Der Zwischenknoten, der eine Route kennt, verschickt zum einen die Originalsignatur des Zielknotens, die er bei Erhalt der Route bekommen hat, zum anderen fügt er eine neue Gültigkeitsdauer für die Route hinzu, die er selbst signiert.

Darüber hinaus gibt es ein spezielles Verfahren bei Route-Errors und der Integration neuer Knoten.

Bewertung

Das Key-Management wird als Voraussetzung genannt, aber nicht genauer bestimmt. Unklar bleibt, wie die Knoten an Schlüssel kommen, ohne dass das Routing funktioniert. Jeder Knoten muss bei Erhalt einer Routingnachricht eine Signatur überprüfen, egal ob er auf der gewünschten Route liegt. Das Route-Caching erlaubt es einer/einem AngreiferIn, alte Topologie-Informationen zu verbreiten. Die Daten werden bei SAODV nicht gesichert und die Schutzmechanismen beziehen sich nur auf die Sicherheit des Routings.

Der Hashchain verhindert, dass der hop-count vermindert werden kann und somit günstigere Routen durch einen kompromittierten Knoten verlaufen. Der hop-count kann jedoch beliebig erhöht werden. Dies ist das Ziel von egoistischen Knoten, welche keine Daten weiterleiten wollen. Dieses Verhalten wird somit nicht verhindert.

Der Schutz des SAODV-Protokolls ist gering. Die Verwendung von signierten Routingnachrichten kann als Standardverfahren angesehen werden. Der Einsatz des Hashchains zum Schutz des TTL-Feldes ist jedoch neu und erfüllt den Zweck, dass Routen nicht durch angreifende Knoten verkürzt werden können.

5.1.2 Ariadne

Ähnlich wie SAODV arbeitet das reaktive Ariadne-Protokoll [55] mit signierten Routing-Nachrichten und Hashchains als Sicherung des TTL-Feldes. Es wird mit Source-Routen

ähnlich DSR gearbeitet. Das Protokoll bietet drei Betriebsarten zur Authentisierung, wobei TESLA für Mesh-Netzwerke die interessanteste ist. Voraussetzung für TESLA ist ein geheimer Schlüssel zwischen zwei kommunizierenden Knoten. Ein authentischer TESLA-Schlüssel pro Knoten im Netzwerk muss bekannt sein.

Beim TESLA-Broadcasting-Authentifizierungsprotokoll berechnet der sendende Knoten über eine Nachricht und einen geheimen Schlüssel k_i einen MAC. k_i ist ausgehend von einem zufälligen Initialschlüssel K ein Hashchain. In einem festen Zeitintervall wird ein Hashchain-Schlüssel je Knoten verwendet und erst nach dieser Zeitspanne wird dieser Schlüssel veröffentlicht. Die Knoten können nun die Authentizität der Nachrichten überprüfen. Die Sicherheit beruht darauf, dass bei Veröffentlichung eines Schlüssels bereits alle damit verschlüsselten Nachrichten die Empfänger erreicht haben müssen. Darüber hinaus müssen die Knoten über hinreichend genau synchronisierte Uhren verfügen.

Der Absender eines RREQ authentifiziert sich gegenüber dem Empfänger mittels MAC über eindeutige Daten (Timestamp) und dem vorausgesetzten geheimen Schlüssel. Bevor die Zwischenknoten einen REQ weiterleiten, fügen sie sich als Source-Route hinzu und berechnen mit ihrem aktuellen TESLA-Schlüssel einen MAC. Das Ziel bestätigt im RREP durch eigenen MAC plus geheimen Schlüssel, dass alle Sicherheitsvorschriften eingehalten wurden - jeder Source-Knoten hat korrekt die Weiterleitung und Authentizität durch Veröffentlichung seines TESLA-Schlüssels bestätigt. Bei jedem hop wird ein Hashwert aus Knoten und Request gebildet. So können keine Knoten aus der Source-Route gelöscht werden.

Weitere Aspekte von Ariadne sind die Routenpflege mit Route-Error-Paketen und ein einfacher Schutz vor DoS durch Fluten des Netzes mit gefälschten RREQ-Paketen.

Bewertung

Bei der Routenfindung ist es nicht möglich, Knoten einzufügen oder andere zu löschen. Alle Knoten werden zuverlässig authentifiziert. Gute Performance-Werte liefert der Verzicht auf Public-Key-Kryptographie.

Die Schwächen des Protokolls sind die geheimen Schlüsselpaare zwischen kommunizierenden Knoten. Woher diese kommen, wird im Protokoll nicht beschrieben. Bei der Verwendung von TESLA müssen am Anfang die Hashchains verteilt werden. Wie dies geschieht ist ebenfalls unklar. Die Autoren verlangen ebenfalls lose synchronisierte Uhren - sonst muss mit der Veröffentlichung der TESLA-Schlüssel zu lange gewartet werden. Es wurde kein Mechanismus zum Abgleich der Uhrzeit benannt. Diese Wartezeit führt auch zu einer erhöhten Wartezeit (Delay) bei Versand der Daten.

5.1.3 Authenticated Routing for Ad hoc Networks (ARAN)

Das ARAN-Protokoll [56] ist ein reaktives Protokoll, welches einen TTP-Zertifikatsserver benötigt. Die Knoten brauchen ein Zertifikat von der CA und alle versandten Nachrichten werden mit diesem signiert. Das Zertifikat beinhaltet die IP-Adresse, den öffentlichen Schlüssel und Zeitstempel. Die versandten Nachrichten enthalten neben dem Zertifikat eine Zufallszahl und einen Zeitstempel. Alle Zwischenknoten entfernen die letzte Signatur und hängen ihre eigene an.

Bewertung

Durch ARAN lassen sich viele Modifikationen von Nachrichten erkennen oder teilweise verhindern. Da die Zertifikate vorher erstellt und signiert werden, ist der Zugriff auf die CA nicht mehr notwendig. Einen Rückruf von Schlüsseln gibt es nicht. Zertifikate sind nur durch die Gültigkeitsdauer beschränkt.

Jeder Knoten muss Daten (Zufallszahl, Timestamp, öffentlicher Schlüssel) jedes weitergeleiteten Paketes speichern. Dies bedeutet ein enormer Speicherbedarf. Es müssen eine große Menge an digitalen Signaturen erstellt werden. Jeder Knoten muss pro empfangenem Route-Request zwei Signaturen prüfen und eine neue generieren. Dies stellt hohe Anforderungen an die Hardware der Knoten. Betroffen sind fast alle Knoten, da RREQ im gesamten Netzwerk geflutet werden. Egoistische Knoten werden nicht erkannt.

5.1.4 Secure Routing Protocol (SRP)

Das reaktive SRP [57], welches auf DSR beruht, geht davon aus, dass zwischen Sender und Empfänger ein geheimer Schlüssel vorhanden ist. RREQ werden mittels eines MAC unter Benutzung des geheimen Schlüssels authentifiziert. Dies geschieht ebenfalls mit dem Route-Reply-Paket. Duch den Einsatz einer Sequenznummer und eines zufälligen *Query Identifier* schützt sich das System vor Replays und Routingschleifen.

Die Arbeit beschäftigt sich weiterhin mit Route-Errors und, falls zusätzliche Gruppenschlüssel vorhanden sind, mit gecachten Routen.

Bewertung

SRP ist ein schlankes Routingprotokoll - nur MACs werden in den Endknoten berechnet und der Routing-Overhead ist minimal. Nachteile sind die fehlende Authentifizierung der Zwischenknoten, zudem gibt es kein Verfahren für die Verteilung der geheimen Schlüssel ohne funktionierendes Routing.

Das Protokoll kann bestimmte Angriffe verhindern, jedoch können Routen verlängert und umgeleitet werden. Man-in-the-middle-Angriffe sind so möglich. Darüber hinaus kann sich ein Zwischenknoten in der Routenfindung nicht in die Source-Route eintragen und trotzdem die Pakete weiterleiten. Der Knoten, welcher den RREQ ausgelöst hat, erhält nun eine Route, die nicht existiert.

5.1.5 Secure Dynamic Source Routing (SDSR)

Das SDSR-Protokoll [50] basiert auf dem reaktiven DSR-Protokoll, welches mit Source-Routen arbeitet und ist in ein Framework namens SAM integriert. SAM leistet die Schlüsselverwaltung, Erzeugung von Identitäten oder Bestrafung und Ausschluss von egoistischen oder bösartigen Knoten. SDSR arbeitet mit Diffie-Hellman Schlüsseln. Die globalen Parameter n und z sind fest vorgegeben.

Der Route-Discovery-Prozess besteht - wie meist üblich - aus einer RREQ- und RREP-Phase. Die festen Bestandteile des RREQ-Paketes (Typbezeichner, Quelle, Ziel, eindeutige Route-Request-ID und Diffie-Hellman-Schlüssel) werden durch den Sender und seine *MANET-ID* signiert. Zusätzlich befindet sich die Source-Route und eine Zufallszahl (nonce[1]) im RREQ. Zwischenknoten überprüfen, ob sie schon ein RREQ mit gleichem Absender und eine Route-Request-ID weitergeleitet haben. Falls nicht, ergänzen sie sich in der Source-Route und generieren eine neue *Nonce* mittels symmetrischer Verschlüsselung von der alten Nonce und einem zufällig gewählten geheimen Schlüssel k_i. Erreicht RREQ das Ziel, wird die Signatur überprüft und das RREP eingeleitet.

Im RREP signiert das Ziel die Source-Route und Signatur vom Sender mittels eigener MANET-ID. Die Route kann nun nicht mehr unbemerkt verändert werden. Ebenfalls wird ein neuer Diffie-Hellman-Schlüssel angefügt. Der Zielknoten kann nun den geheimen DH-Schlüssel k_{sd} berechnen und fügt dem RREP-Paket einen signierten Hashwert des gemeinsamen Schlüssels an. Der Initiator der Routensuche kann nun sicher gehen, dass der DH-Schlüssel wirklich vom Zielknoten kommt. Alle Zwischenknoten überprüfen die Signaturen, erstellen eigene DH-Schlüssel und fügen diese samt signiertem Hashwert des gemeinsamen DH-Schlüssels an das RREP-Paket an. Außerdem muss der alte Nonce wieder hergestellt werden. Dies kann nur der Zwischenknoten des Hinweges, da er den geheimen Wert k_i kennt. Das RREP muss den selben Weg wie das RREQ nehmen.

Der Sender kann nun sicher sein, dass er alle Zwischenknoten authentifizieren kann (verwendete MANET-ID), dass die Route nicht manipuliert wurde und er zu allen

[1] nonce - Number used Once

Zwischenknoten gemeinsame DH-Schlüssel besitzt. Eine Erweiterung des Protokolls ermöglicht es dem Ziel der Route, alle Knoten zu authentifizieren und geheime DH-Schlüssel auszutauschen.

Die Schlüsselverteilung ist sehr einfach - Schlüssel können ungesichert im Netz übertragen werden, da sich die Adresse eines Knotens aus dem öffentlichen Schlüssel berechnen läßt. Jeder Knoten kann direkt ermitteln, ob eine Adresse zu einem angegebenen öffentlichen Schlüssel gehört. Über Signatur und Verifikator kann ferner ermittelt werden, ob es sich um eine gültige MANET-ID handelt.

Bewertung

Alle am Routing beteiligten Knoten werden authentifiziert und Änderungen an einer Route werden erkannt. Die Schlüsselverteilung ist integriert und das Routing stellt mäßige Anforderungen (im Vergleich zu den anderen vorgestellten Protokollen) an Hardware und Speicherplatz. Asymmetrische Kryptooperationen finden nur in der RREP-Phase statt, wo weniger Knoten als in der RREQ-Phase betroffen sind.

Das Protokoll verwendet mit Ausnahme einer CA zur Verwaltung der MANET-IDs keine zentralen Komponenten. Die Knoten müssen sich in regelmäßigen Abständen von der Gültigkeit der MANET-IDs überzeugen und somit Kontakt zu dieser zentralen CA aufnehmen. Neue Knoten werden erst akzeptiert, wenn sie dort verzeichnet sind. Egoistische Knoten, die nicht an der Routenfindung teilnehmen, werden nicht erkannt. Dies wird extra von einem IDS im SAM-Framework gewährleistet.

Ein Nachteil des Protokolls ist die zentrale Zertifizierungsinstanz für alle Mesh-Geräte. Für VANETs würde dies eine weltweite Instanz bedeuten. Unrealistisch sind ebenfalls die vorausgesetzten manipulationssicheren Kryptoeinheiten.

5.1.6 Secure OLSR

Secure OLSR ist eine Erweiterung der UniK-OLSR-Implementierung von A. Tønnesen. Alle berechtigten Knoten im Netzwerk besitzen einen geheimen symmetrischen Schlüssel, mit dem sie eine Signatur an jede Nachricht im Netzwerk anhängen. Dies sichert die Integrität der Daten und nur Knoten mit Kenntnis des Schlüssels können am Mesh-Netzwerk teilnehmen.

Die Signatur wird bei jeder Weiterleitung erneut erstellt. So müssen Veränderungen des hop-counts und des TTL-Feldes nicht beachtet werden. Wenn mehrere Nachrichten in einem OLSR-Paket versandt werden, ist nur eine Signatur notwendig. Die Signatur wird wahlweise mit MD5 oder SHA-1 erstellt und ist immer die letzte Nachricht im

Paket.

Um Replay-Angriffen vorzubeugen, werden *Zeitstempel (timestamps)* benutzt. Nur Nachrichten, die einen aktuellen Zeitstempel haben, werden akzeptiert. Da die Knoten jedoch über keine synchronisierten Uhren verfügen und geringe Abweichungen durchaus normal sind, erfolgt ein Abgleich der Uhrzeiten zweier Knoten über einen 3-Wege-Zeitstempel-Austausch. Diese Zeitdifferenz wird gespeichert und später ständig aktualisiert.

Wenn Knoten A eine signierte Nachricht von Knoten B erhält, er aber über keine gespeicherte Zeitdifferenz von B verfügt, initialisiert er den Zeitstempel-Austausch. Er sendet eine *challenge message* mit einer 32-bit Zufallszahl (nonce) und einer Signatur über die Nachricht mit dem geheimen Schlüssel. Knoten B beantwortet mit einer *challenge response message* - neue Zufallszahl, Zeitstempel, Signatur über IP von B und Zufallszahl von A und Signatur über gesamte Nachricht. Knoten A bestätigt nun wiederum eine *response-response message* mit Zeitstempel von A, Signatur über IP von A und Zufallszahl von B, sowie wie immer eine Signatur über die gesamte Nachricht.

Um DoS-Angriffe mittels des Zeitstempel-Austausch-Verfahrens zu minimieren, werden die Zeitstempel-Anfragen eines Knotens in einem bestimmten Intervall nur einmal beantwortet. Weitere Anfragen werden verworfen.

Bewertung

Die Signatur sagt nur aus, dass der Knoten, von dem die Nachricht zuletzt weitergeleitet wurde, dem davor sendenen Knoten vertraut. Es ist keine end-to-end-Signatur. Geheime Schlüssel werden vorausgesetzt und es ist kein Mechanismus der Schlüsselverwaltung bekannt. Da alle Knoten denselben Schlüssel benutzen, ist bereits nach einem komprimittierten Knoten das gesamte Verfahren gebrochen.

5.1.7 Zusammenfassung

Die Voraussetzungen der Protokolle reichen von einer funktionierenden PKI, vorhandenen geheimen Schlüsseln und synchronisierten Uhren bis zu sehr hoher Rechenleistung der Knoten. Je nach Mesh-Einsatzszenario können nur bestimmte Routingverfahren eingesetzt werden. Die vorgestellten Mesh-Protokolle kombinieren teilweise verschiedene Sicherheitsaspekte wie das Sichern der Routen als auch der Inhaltsdaten. Dies bietet den Vorteil, dass in einem Protokoll auf den unteren Ebenen der Schutz des Netzwerkes erhöht wird.

SDSR geht noch einen Schritt weiter und stellt das Routing-Protokoll in ein Sicherheitsframework, welches neben dem Routing Identifizierung und ein IDS umfasst. Das zuletzt genannte Protokoll ist das einzige, welches eine klare Identifizierung aller Knoten verlangt.

Die vorgestellten Routingprotokolle basieren auf unterschiedlichen Grundverfahren. Einige Verfahren bieten durch ihr Grundkonzept Vorteile für die Implementierung von Sicherheitsmethoden. Beim Source-Routing werden explizit alle Zwischenknoten ermittelt und können so authentifiziert werden. Bei einem Link-State-Protokoll werden die gesamten Topologieinformationen verteilt. Dies bietet noch mehr Informationen über die Routen und beteiligte Knoten, bedeutet aber auch einen enorm höheren Overhead.

Ein weiterer Punkt ist der Speicherbedarf der einzelnen Knoten. Bei AODV müssen die Zwischenknoten alle RREQ speichern, bei DSR jedoch nicht. Eine Entscheidung für ein Routingprotokoll muss also auf Grundlage der gewünschten Sicherheitsimplementierung getroffen werden.

5.2 Egoistische Knoten / Kooperation

Egoistisches Verhalten kann durch Bestrafen von unkooperativem oder durch Belohnung von kooperativem Verhalten vermieden werden. Im Folgenden werden kurz die grundsätzlichen Verfahren vorgestellt und Protokollimplementierungen genannt.

5.2.1 Nuglets System

Der motivationsbasierte Ansatz versucht, Knoten zur aktiven Teilnahme zu veranlassen. So werden virtuelle Währungen eingesetzt, die sich ein Knoten durch aktive Teilnahme am Netz erwerben kann und diese Währung im Gegenzug einsetzen kann, wenn der Knoten selbst Verkehr generiert. Im *Nuglets System* [58] sind Nuglets eine virtuelle Währung, die zur Verrechnung verschiedener Dienste im Mesh-Netzwerk eingesetzt wird. Um Manipulation an der virtuellen Währung zu vermeiden, setzen die AutorInnen manipulationssichere Hardware voraus, die so gestaltet sein muss, dass zwar das Nuglets System, nicht jedoch die BenutzerInnen die Kontostände verändern können.

Bewertung

Beim Nuglets System bleiben einige grundlegende Fragen unbeantwortet. Zunächst ist der Ansatz der manipulationssicheren Hardware unrealistisch. Ein angreifender

Knoten kann einem anderen Knoten durch Zusenden vieler Pakete alle Nuglets entziehen. Liegt ein Knoten am Rande des Mesh-Netzwerkes, kann er kaum Nuglets durch Weiterleiten verdienen und hat keine Möglichkeit, seinen eigenen Datenverkehr zu bezahlen. Auch erkennt das System keine angreifenden Knoten.

5.2.2 Watchdog und Overhearing

Wie können nun egoistische und böswillige Knoten erkannt werden? Eine Überwachung der korrekten Weiterleitung ist durch die Watchdog-Methode mit *Overhearing* oder dem Senden von Testnachrichten möglich. Beim Overhearing sendet ein Knoten ein Datenpaket, schaltet in den *Promiscuous Mode* und hört nun den gesamten Datenverkehr ab, ob die Daten weitergeleitet werden. Bemerkt der Knoten nach einer bestimmten Zeitspanne keine Weiterleitung, schließt er auf ein Fehlverhalten des Weiterleitungsknotens und benachrichtigt den ursprünglichen Quellknoten des Datenpaketes.

Bewertung

Bei diesem Ansatz des Promiscuous Overhearing kann es zu Fehleinschätzungen kommen, wenn alle Knoten korrekt arbeiten, aber durch Empfangsprobleme des Watchdogs die Weiterleitung nicht bemerkt wird. Egoistische Knoten können die Sendeleistung anpassen, um eine Weiterleitung vorzutäuschen, die der Watchdog bemerkt, aber das eigentliche Ziel nicht erreicht. Kooperieren mehrere egoistische/ angreifende Knoten, werden diese nicht erkannt (zwei benachbarte Knoten leiten keine Daten weiter und "verpfeifen" sich nicht gegenseitig).

5.2.3 Eindeutiges iteratives Probing

In [50] wird das *eindeutige iterative Probing* vorgestellt. Das Verfahren kombiniert das Senden von verschlüsselten Testnachrichten und Overhearing. Eine Testnachricht besitzt ein Kommandofeld, welches von den weiterleitenden Knoten entschlüsselt wird. Hier kann eine Aufforderung stehen, den nächsten weiterleitenden Knoten mittels Overhearing zu überwachen und das Ergebnis an den Quellknoten zu senden. Durch sequenzielles Durchprobieren einer Route, welche einen oder mehrere egoistische/ böswillige Knoten enthält, können diese (zumindest der erste nach der Quelle) zuverlässig gefunden werden. Die Testnachrichten beziehen sich sowohl auf Routen-Suchnachrichten, als auch auf normale Datenpakete.

Bewertung

Mit dem Verfahren des eindeutigen iterativen Probing können egoistische und böswillige Knoten in einem Mesh-Netzwerk erkannt werden, wenn eine PKI verfügbar ist und die Knoten ein eindeutiges Identifizierungsmerkmal haben. Bisher wird das Fehlverhalten nur von der Quelle der Testpakete festgestellt. Um einen oder mehrere Knoten aus einem Netzwerk isolieren zu können, müssen die ermittelten Fehlverhalten bewertet und an alle Knoten verteilt werden. Diese Aufgabe wird mit wachsender Größe des Netzwerkes aufwendiger und läßt sich am Besten mit einem *Intrusion Detection System* (IDS) realisieren.

Wird das Verfahren eingesetzt, müssen die Knoten bei jedem Datenpaket das Kommandofeld entschlüsseln, um zu entscheiden, ob sie den nächsten Knoten überprüfen sollen. Dies bedeutet höhere Rechenlast für alle Knoten.

5.3 Intrusion Detection System (IDS)

Durch ein IDS können die gewonnenen Daten des Routing und der Kooperationstests bewertet, verteilt und Reaktionen, wie der Ausschluss von einzelnen Knoten aus dem Netzwerk, ausgelöst werden. In einigen Arbeiten wird statt des IDS der Begriff Reputationssystem verwendet. Das Verfahren ist jedoch ähnlich - Sensoren / Monitore erfassen Daten, welche durch ein Reputationssystem bewertet werden und dies Auswirkung auf das Routing (Aussperren einiger Knoten) hat.

Aufgabe der Sensoren ist es, ein Fehlverhalten eines Knotens in einem Mesh-Netzwerk festzustellen (eindeutiges iteratives Probing). Hat ein Knoten ein Fehlverhalten festgestellt, wertet er lokal die Vertrauenswürdigkeit des fehlerhaften Knotens herab. Verschieden Knoten gleichen ihre lokalen Bewertungen untereinander ab. Die Verteilung kann per Broadcast an die benachbarten Knoten ggf. über mehrere Hops erfolgen (Anpassen des TTL-Wertes). Diese Nachrichten sind mit einem Zeitstempel versehen und vom absendenen Knoten signiert, um Fälschungen zu vermeiden. Empfängt ein Knoten eine neuere Liste, so überprüft er die Signatur und speichert sie.

Durch Listen mehrerer NachbarInnen kann der Knoten neben seiner eigenen Sicht eine globale Sicht auf viele Knoten, vornehmlich in seinem lokalem Umfeld, aufbauen. Ein Knoten sollte nur dann einen anderen Knoten bewerten, wenn eigene Erfahrungen oder mehrere Bewertungen von unterschiedlichen Knoten vorliegen. Dies vermeidet, dass Knoten durch eine einzelne (böswillige) Bewertung global abgewertet werden können.

Unterschreitet die Bewertung eines Knotens einen bestimmten Wert, so kann dieser durch schlichtes Ignorieren ausgeschlossen werden. Ist sich die Nachbarschaft eines ne-

gativ bewerteten Knotens einig, so wird er effektiv isoliert. Allerdings hat er noch die Möglichkeit, nach Ablauf einer gewissen Zeit und Erholung seiner globalen Bewertung wieder kooperativ am Netz teilzunehmen und seine Bewertung zu verbessern. In der Arbeit [50] kann sogar ein globaler Ausschluss erfolgen, da ein Knoten seine eindeutige Identität (öffentlicher Schlüssel) durch Widerruf bei der zentralen CA verlieren kann. Eine Wiederteilnahme ist dann nicht mehr möglich.

Der Einsatz eines IDS in einem Mesh-Netzwerk verbessert die Kooperation und damit die Verfügbarkeites des Mesh-Netzwerkes enorm. Grundlage ist eine eindeutige Identifizierung der Knoten und eine PKI. Knoten müssen teilweise aufwendige Berechnungen bewältigen und Bewertungs- und Schlüsseltabellen speichern.

5.4 Schutz der beschriebenen Einsatzszenarien

Nachdem im letzten Abschnitt Techniken vorgestellt wurden, mit denen die Sicherheitsinteressen der Beteiligten unterschiedlicher Szenarien unter bestimmten Annahmen erfüllt werden können, werden nun für die einzelnen Szenarien Gesamtschutzkonzepte entwickelt. Bei Interessensgegensätzen der Beteiligten wird versucht, einen möglichst großen Schutz aller Beteiligter zu gewähren, jedoch die Grenzen klar aufzuzeigen. Können einzelne Sicherheitsinteressen nicht zur vollen Zufriedenheit beschrieben werden, erfolgt eine Definition der zusätzlichen Anforderungen für weitere Entwicklungen.

Militärische Kommunikation

Für das militärische Szenario fällt es schwer, eine gute Bewertung der Schutzmöglichkeiten zu treffen. Zu viele Fragen der Struktur eines solchen Netzes sind offen bzw. werden nicht öffentlich publiziert.

Die *Identifizierung* der Geräte und Personen muss eindeutig sein und voneinander getrennt erfolgen. Nur authentifizierte Geräte dürfen am Mesh-Netzwerk teilnehmen. Geräte können verloren oder von einem Feind / einer Feindin erobert werden. Die Benutzung muss daher von einem weiteren Schlüssel abhängig sein, welcher bei Beginn eines Einsatzes und periodisch eingegeben werden muss. Falls nicht jede Person im Mesh-Netzwerk über ein persönliches Gerät verfügt, sollte zur eindeutigen Identifizierung dieser Schlüssel je Person eindeutig sein.

Die *Schlüssel* müssen vor Beginn eines Einsatzes durch eine zentrale Instanz vergeben werden. Jedes Gerät und jede Person besitzen ein asymmetrisches Schlüsselpaar. Der öffentliche Schlüssel ist von einer TTP signiert. Vor Beginn einer Kommunika-

tionsverbindung tauschen alle am Routing beteiligten Schlüssel einen symmetrischen Sitzungsschlüssel aus und authentifizieren sich mit ihren asymmetrischen Schlüssel und 3-Wege-Verfahren mit verschlüsselter Zufallszahl. Um Angriffe auf die Verschlüsselung zu erschweren, sollten die Sitzungsschlüssel in bestimmten Abständen (nach 24 Stunden) neu erstellt und ausgetauscht werden.

Der Ansatz von Mixen zum *Schutz der Kommunikationsverbindungen* ist durch die großen Entfernungen zwischen den Knoten, der damit verbundenen Verzögerung und dem hohen Rechenaufwand nicht anwendbar. Proxies bieten in der abhörbaren Funkumgebung keinen Schutz.

Periodischer *Dummy-Traffic* kann teilweise einen Schutz gewährleisten. Ein Gegner oder eine Gegnerin kann nachvollziehen, wer mit wem kommuniziert (dies ist in militärischen Strukturen meist vorher klar), jedoch nicht erkennen, zu welchem Zeitpunkt die wichtige Kommunikation erfolgt. Das periodische Senden von Daten kann auch als "Herzschlag" der Knoten aufgefasst werden. Es ist erkennbar, ob der Knoten funktioniert und im Mesh-Netzwerk erreichbar ist.

Der Ansatz des Dummy-Traffics ist dennoch unbefriedigend, da er eine hohe Netzlast erzeugt und die Energieressourcen der Knoten nicht schont. Der Schutz der Kommunikationsbeziehung wird auch nicht erreicht.

Die *Verfügbarkeit* des Mesh-Netzwerkes ist grundlegend von der funktionierenden Funkübertragung abhängig. Je nach Entfernung benachbarter und kommunizierender Knoten muss die Sendeleistung variabel sein und ggf. erhöht werden. Durch mögliche Wechsel oder gleichzeitiges Senden auf verschiedenen Frequenzen können Störsignale umgangen werden. Daten müssen mit einem besonders fehlerkorrigierenden Code moduliert werden.

Mit dem theoretischen Protokoll SDSR kann der Schutz des eigentlichen *Routings* erreicht werden. Das Protokoll arbeitet jedoch proaktiv. Ich halte für die stark dynamische Umgebung und die große Anzahl von Knoten im militärischen Szenario, wo ein Knoten nur mit wenigen anderen kommuniziert, reaktive oder hybride (fish-eye) Verfahren für die effektiveren Routingmethoden.

Eine Kontrolle, ob die Knoten sich *kooperativ* verhalten und der Einsatz eines IDS sind nicht notwendig, da alle Beteiligten ein Interesse am kooperativem Verhalten haben.

Durch die hohe Anzahl der Knoten und den teilweise großen Übertragungsstrecken kann es zu *unfairem Verteilen* der Bandbreite kommen. Bedingt durch die hohe Dynamik können keine bisher beschriebenen Protokolle angewendet werden. Wird ein

periodisches Senden angewendet, kann das Zeitintervall dynamisch an die verfügbare Bandbreite angepasst werden und somit ein QoS erreicht werden. Genügend Reserve vorausgesetzt, können dynamische Spitzen durch neue Knoten im Mesh abgefangen und die Intervalle angepasst werden.

Zusammenfassung

Ein umfassendes Sicherheitskonzept für militärische Kommunikation kann nicht beschrieben werden. Diese Arbeit kann jedoch die Knackpunkte aufzeigen.

Die bisher erforschten Verfahren des Schutzes der Kommunikationsbeziehung und die gleichzeitige Anforderung eines möglichst Ressourcen schonenden Protokolls stehen im Widerspruch. Wie in einem stark dynamischen Szenario die faire Verteilung der Bandbreiten gesichert werden kann, ist ein weiteres offenes Forschungsfeld.

Bei den bisher betrachteten Routingprotokollen gibt es kein reaktives oder hybrides System, welches die Sicherheitsanforderungen der militärischen Kommunikation erfüllen kann. Mesh-Routing-Verfahren gehen von einer potentiellen Kommunikation zwischen jedem möglichen Knoten aus. Im militärischen Bereich existieren jedoch feste Kommunikationswege und -hierarchien. Erst durch Kenntnis dieser Strukturen kann ein entsprechendes Routingprotokoll entwickelt werden, welches mit sehr vielen Knoten effizient und sicher arbeitet.

Kommunikation von Rettungskräften

Bei einem Rettungseinsatz sind Personen und teilweise Sensoren im Einsatz. Durch eine zentrale CA können vor dem Einsatz alle Geräte und Personen eine eindeutige *Identifizierung* bekommen. Um eine eindeutige Identifizierung in unterschiedlichen Zusammensetzungen von Einheiten zu ermöglichen, ist ein asymmetrisches Verfahren unabdingbar. Werden die öffentlichen Schlüssel durch eine TTP signiert, können sich die Knoten am Anfang des Einsatzes untereinander authentifizieren und geheime symmetrische Sitzungsschlüssel austauschen.

Der *Inhaltsschutz* der zu übertragenden Daten kann, aber muss nicht verschlüsselt stattfinden. Ähnlich dem noch analogen Funk, können Befehle über Abkürzungen oder Codes übermittelt werden. Dies schont die Ressourcen der Mesh-Geräte. Die *Integrität* der Nachrichten kann durch einen signierten MAC-Code oder durch in der Initialisierungsphase ausgetauschte symmetrische Sitzungsschlüssel erfolgen.

Um die *Verfügbarkeit* des Mesh-Netzes bei Störungen auf einigen Frequenzen zu erhöhen, muss am Beginn oder beim Auftreten von Störungen auf andere Frequenzen

umgeschaltet werden können. Da eine Störung aller Frequenzen fast vollkommen ausgeschlossen werden kann, sind keine speziellen Modulationsverfahren nötig. Darüber hinaus kann die Bandbreite erhöht werden, wenn die Kommunikation (Sprache) und Datenübertragung der Sensoren auf unterschiedlichen Frequenzen erfolgt.

Je nach räumlicher Größe des Einsatzes reicht ein einfaches Broadcast-*Routing* aus. Ist die Fläche größer, müssen Routen abgesichert gefunden werden und alle am Routing beteiligten Knoten identifiziert werden. Das beschriebene Protokoll SDSR bietet diese Funktionalität. Ein Absichern der Kooperation, ein Einsatz eines IDS oder eines Fairness-Protokolls ist durch die ähnlichen Sicherheitsinteressen und die geringe Ausdehnung des Netzwerkes nicht notwendig.

Die *Ressourcen* der Knoten können durch ein energiesparsames Protokoll (Taktung der Kommunikation) erreicht werden. Die Dauer des Einsatzes ist begrenzt und Akkus können ggf. ausgetauscht werden.

Zusammenfassung

Unter den getroffenen Annahmen der Struktur des Mesh-Netzwerkes und den geforderten Sicherheitsinteressen bieten aktuelle Methoden eine zufriedenstellende und umfassende Sicherheitsstruktur.

Eine weitere Optimierung bietet die Analyse der Kommunikationsströme. Zu vermuten ist, dass die Einsatzkräfte und Sensoren hauptsächlich mit einer zentralen Kommandostelle kommunizieren. Werden die Datenströme fast immer zentral gebündelt, ist der Einsatz geografischer (Daten zur Zentrale) und Multicast-Protokolle (von Informationen der Zentrale an viele Einsatzkräfte) vorteilhaft.

Verkehrsmeldungen im VANET

In einem VANET können Fahrzeuge oder FahrerInnen *identifiziert* werden. Warn- und Informationsmeldungen (Straßen- und Verkehrszustand) gehen immer von einem konkreten Wagen aus. Um die Architektur möglichst einfach zu halten, ist eine Identifizierung ähnlich der Wagenseriennummer notwendig.

Werden dagegen zusätzliche (kommerzielle) Dienste genutzt, die an sich unabhängig vom eigentlichen Auto sind, ist die Identifizierung von Personen, welche die Leistungen vergüten, unabdingbar. Ein Auto besitzt kein Bankkonto. Methoden des anonymen Micro-Payments, welche keine eindeutige Identifizierung benötigen, sind ebenfalls anwendbar. Je nach benutztem Dienst sind die Personenidentitäten verschieden, um die Privatsphäre der NutzerInnen zu schützen.

Die Fahrzeugidentitäten können bereits im Herstellungsprozeß eindeutig festgelegt und in geschützter Hardware im Fahrzeug gespeichert werden. Die Fahrzeughersteller verwalten Datenbanken mit allen Hauptidentitäten, damit diese überhaupt aufgedeckt werden können. Aus dem Schutzinteresse des/der FahrerIn, keinen Überwachungsaktivitäten (Bewegungsprofile) ausgesetzt zu sein, muss die Hauptidentität geschützt werden und statt dessen mit Pseudonymen gearbeitet werden. Aus dem augenscheinlich gegensätzlichen Interesse, bei heftigen Verkehrvergehen oder Diebstahl die Identitäten eines Fahrzeuges aufzudecken, müssen die Pseudonyme mit der Hauptidentität korreliert werden.

Ein Ansatz für einen größtmöglichen Schutz der FahrerInnen ist die gemeinschaftliche Erstellung von *Pseudonymen* durch politisch und kommerziell unabhängige Institutionen. In Deutschland könnten dies z.B. FahrzeugherstellerInnen, eine öffentliche Behörde wie die Fahrzeugzulassungsstelle und der ADAC sein. Nach Erstverkauf eines Fahrzeuges und beim jährlichen Check des Autos in der Werkstatt werden über ein Schwellwertschema neue Pseudonyme von den Institutionen angefordert, welche danach in zufälliger Reihenfolge und für maximale Zeitintervalle im Fahrzeug benutzt werden. Die Hauptidentität, welche erst das eigentliche Fahrzeug identifiziert, kann nur durch alle Institutionen nach festen Regeln ermittelt werden. Eine Regel könnte ein richterlicher Beschluss sein, falls nach einer Unfallflucht oder Diebstahl der/die EigentümerIn eines Autos ermittelt werden muss.

Bei einigen Diensten, wie Verkehrsleitsystemen, ist eine eindeutige Identifzierung der Fahrzeuge überhaupt nicht notwendig. Hier können die NutzerInnen mit zufälligen Identitäten arbeiten. Natürlich müssen sich die Knoten von Verkehrsleitsystemen bzw. Dienstleistungen durch eindeutige Signaturen vertrauenswürdiger Instanzen als solche ausweisen.

Die Fahrzeugidentifizierungen sind Pseudonyme, welche von verschiedenen vertrauenswürdigen Instanzen signiert sind. In Fahrzeugen sind die öffentlichen Schlüssel dieser Instanzen gespeichert, somit können sie die Signaturen unter Meldungen anderer Fahrzeuge überprüfen. Eine Ungültigkeitserklärung für Schlüssel gibt es nicht.

Werden Warnmeldungen versandt, so muss der Inhalt der Nachricht nicht verschlüsselt werden. Die *Integrität der Nachricht* wird durch einen signierten MAC geschützt. Bei der Kommunikation zu anderen Dienstleistenden ist der Schutz des Inhalts und die Integrität der Nachricht durch den öffentlichen Schlüssel der DienstleisterInnen und private Signierung der FahrerInnen möglich.

Die Sicherstellung der *Verfügbarkeit des Netzes* kann mit zusätzlichen Techniken nicht erhöht werden. Durch die große Anzahl von Knoten und die Dringlichkeit, dass

Warnmeldungen sofort empfangen werden, muss das VANET auf einer Frequenz arbeiten. Sollte diese Frequenz gestört sein, funktionieren zwar die Warnmeldungen nicht, der/die FahrerIn kommt aber in keine schlechtere Situation als er/sie heute hat.

Ein spezielles *Routing* ist nicht notwendig, da alle Daten als Broadcast ausgesandt werden. Fahrzeuge entscheiden je nach Meldung, ob sie die Nachricht noch einen hop weiter per Broadcast aussenden. So sind auch keine Protokolle, die Fairness oder Kooperation sicher stellen, notwendig. Werden externe Dienste genutzt, so erfolgt die Datenübermittlung ebenfalls per Broadcast bis zum Knoten des Dienstes. Wie die Daten dahinter weitergeleitet werden, ist nicht Aufgabe des VANETs.

Zusammenfassung

Durch die große aber einfache Struktur des VANETs können durch aktuelle Schutzmethoden die Interessen aller Beteiligten gewahrt bleiben. Offen bleibt die Frage nach der Zertifizierungsstelle. Fahrzeuge werden überall auf der Welt produziert und eingesetzt. Ist das Ziel, den gesamten Straßenverkehr sicherer zu gestalten, kann dies nur über eine oder mehrere weltweite Instanz geregelt werden.

Kritisch ist das Vertrauen, welches der/die FahrerIn in die HerstellerInnen des Auto-Mesh-Gerätes und die Service-Werkstatt haben müssen, da hier die Daten korrelierbar sind, einer Person zugeordnet werden können und in der Hardware gespeichert werden. Die Auswahl der Partner der CA muss wohl überlegt sein, damit die Anonymität nur im Extremfall aufgehoben werden kann. Was in Deutschland noch recht einfach vorstellbar ist, kann in anderen Ländern bedeutend schwerer sein.

Ein weiterer Anspruch ist die Optimierung der Algorithmen, da in Fahrzeugen die Rechen- und Speicherleistungen beschränkt sind.

Gebäudevernetzung und -überwachung

Die Sensoren in der Gebäudevernetzung können durch eine eindeutige Seriennummer *identifiziert* werden. In großen Netzen können darüber hinaus mit einem asymmetrischen Verfahren individuelle Schlüssel in den Geräten gespeichert werden, welche eine eindeutige Identifizierung ermöglichen. In kleineren Netzen können durch physikalische Programmierung symmetrische Schlüssel verteilt werden. Während der Initialisierungsphase bzw. bei Veränderungen in der Topologie sind ebenfalls Schlüsselaustausch-Protokolle wie von Diffie-Hellman denkbar, sofern gemeinsame Grundparameter in allen Mesh-Geräten vorhanden sind.

Die *Integrität* der Nachrichten wird durch einen verschlüsselten MAC geschützt. Der

Inhalt der Daten muss nicht geschützt werden.

Die benutzten Frequenzen sollten vor der Installation des Sensornetzwerkes überprüft und im Bedarfsfall gewechselt werden. In kleinen Netzen kann das Routing durch einfaches Broadcast erfolgen. Der Datenverkehr ist gering und Daten sollten auch über mehrere Hops ihr Ziel erreichen. Da alle Geräte von einer Partei installiert werden, müssen keine weiteren Schutzmechanismen für die Verfügbarkeit implementiert werden.

Um die *Energieressourcen* der Sensoren zu schonen, müssen energiesparende Protokolle eingesetzt werden, d.h. synchronisiertes Senden; während der restlichen Zeit können die Knoten in den Ruhezustand gehen. Da das Senden von Daten immer mehr Energie verbraucht als das Empfangen, kann das Auslesen der Sensordaten durch zentrale Auswertungscomputer erfolgen. Erst nach Aufforderung zum Auslesen oder Weitersenden eines Paketes schaltet ein Knoten in den Sendemodus.

Der *Schutz der Privatsphäre* der durch Sensoren erfassten Personen ist auf technischer Ebene kaum möglich - z.B. wenn Personen per Video überwacht werden. Durch öffentliche Bekanntmachung, was erfasst wird und wie die Daten weiterverarbeitet oder gespeichert werden, wird die Überwachung allen Beteiligten transparent.

Zusammenfassung

Sind die Sensornetzwerke nicht zu groß, bieten heutige Techniken einen umfangreichen Schutz. Durch Einsatz symmetrischer Verfahren und energiesparender Protokolle ohne aufwendiges Routing werden die Anforderungen erfüllt. In größeren Netzen, in denen ein Routenfinden notwendig wird oder das Datenaufkommen ansteigt, sind "schlanke" Routingprotokolle wie SRP geeignet, müssen jedoch noch an einen möglichst energiesparenden Einsatz angepasst werden.

Internet Service Provider

Durch die zwei Mesh-Ebenen in diesem Szenario wird die Fragestellung, wer sich wie vor wem *identifiziert*, etwas umfangreicher. Die Mesh-APs bzw. TAPs können durch die betreibende Firma vor der Installation manuell mit öffentlichen Schlüsseln versehen werden. Innerhalb des TAP-Netzwerkes identifizieren sie sich darüber. Die NutzerInnen müssen sich durch eine Identität als Berechtigte gegenüber dem/der DiensteanbieterIn ausgeben. Diese Identität (symmetrischer Schlüssel oder Schlüsselpaar bei asymmetrischer Kryptographie) kann von der Firma erstellt und der/dem NutzerIn per Datenträger oder Chipkarte zugesandt werden. Die NutzerInnen wollen innerhalb des NutzerInnen-Mesh-Netzwerkes anonym bleiben und nutzen hierzu eine zufällig erstellte Identität (dem Betreiber ist die Haupt- und zufällige Identität bekannt). AP/TAPs

müssen natürlich eine den NutzerInnen bekannte Identität haben, damit sich diese nicht zu Fake-APs verbinden und ihre Berechtigungen übersenden.

Da teilweise die AP/TAPs nur mit geringen physikalischen Mitteln geschützt werden können, darf nach einem Diebstahl die Sicherheit des gesamten Netzwerkes nicht von diesen abhängen. Sowohl die Identitäten der AP/TAPs als auch die der NutzerInnen müssen nach einer Komprimittierung gesperrt werden können. Da NutzerInnen sich an verschiedenen Punkten über diverse APs mit dem BetreiberInnen-Netzwerk verbinden wollen, sollte die Überprüfung der Berechtigung nicht durch die APs sondern durch einen zentralen, physikalisch gut geschützten, im kabelbasierten Netz befindlichen Server geschehen.

Die für den/die NutzerIn sicherste Methode, einen eigenen *Schlüssel* gegenüber dem Betreiber zu bekommen, ist der Weg, dass er/sie sich selbst ein Schlüsselpaar mit einem asymmetrischen Verfahren generiert und den öffentlichen Schlüssel von der Betreiberin bzw. vom Betreiber bei einem Besuch in der Niederlassung und unter Vorlage seines/ihres Ausweises signieren lässt. Bei diesem Besuch bekommt sie einen Schlüsselbund der öffentlichen Schlüssel der APs, über die er/sie sich in das BetreiberInnennetz einwählen kann, sowie den öffentlichen Schlüssel des Berechtigungsservers. Später muss in periodischen Abständen die Liste der verfügbaren APs aktualisiert werden.

Etwas umfangreicher sind Protokolle, in denen der/die NutzerIn ähnlich den Telefonkarten Berechtigungskarten kauft und so vollkommen anonym für einen bestimmten Zeitraum im Netz surfen kann.

Die *Integrität* der Daten wird zwischen AP und NutzerIn durch die Signierung des MAC gesichert. Bei der Anmeldung werden ebenfalls symmetrische Schlüssel zwischen diesen beiden Parteien ausgetauscht, so dass Inhaltsdaten geschützt werden. Dasselbe Verfahren gilt zwischen einzelnen TAPs. Ist das Ziel der Daten ein Computer im Internet, so reicht der Schutz nur bis zum Gateway des Betreibers / der Betreiberin.

Die *Kommunikationsbeziehung* der NutzerInnen kann durch Nutzung der beschriebenen Systeme JAP oder TOR unter den genannten Voraussetzungen erreicht werden. Zu beachten ist, dass dann jeder Datentransfer über das Internet abläuft, auch wenn der Zielcomputer sich im selben NutzerInnen-Mesh befindet. Dies bedeutet höhere Verzögerung beim Datenversand und mehr Kosten für die Internet-Nutzung, welche eventuell gar nicht notwendig ist.

Innerhalb des TAP-Mesh können speziell angepasste Routing-Verfahren mit sicherem *Routing* und Fairness-Garantien wie in [52] beschrieben, angewendet werden. Im NutzerInnen-Mesh können Routingprotokolle wie SDSR zum Einsatz kommen. Da die Mesh-Teilnetze nicht zu groß werden und die Dynamik eher gering ist, sind proaktive

Protokolle zu bevorzugen.

Um darüber hinaus eine *faire Verteilung* der Bandbreite zu garantieren, können durch die APs Senderechte (maximale Bandbreite je NutzerIn) vergeben werden. Egoistische oder angreifende Knoten können durch aufwändigere Verfahren, wie im Sicherheitsframework SAM beschrieben, erkannt werden. Wenn diese Knoten dem Betreiber / der Betreiberin gemeldet werden, kann ihnen die Nutzung des BetreiberInnennetzes gesperrt werden. Sollen auch andere NutzerInnen in Zukunft diese Knoten ignorieren, ist hierzu ein Verteilen der Information notwendig.

Wie in der allgemeinen Beschreibung der Schutzmethoden schon genannt, muss der/die NutzerIn Vertrauen in die korrekte Arbeitsweise der Software haben, welche im Allgemeinen vom Betreiber ausgeliefert wird.

Zusammenfassung

Die Implementierung eines Mesh-Netzwerkes in mehreren Stufen stellt eine hohe Herausforderung an die Sicherheitsmechanismen dar. Würde nur von einem großen Mesh-Netz ausgegangen werden, wären die Schutzmaßnahmen einfacher zu formulieren, jedoch gehen dann die Vorteile einer individuellen Betrachtung verloren (TAPs werden vom Betreiber initialisiert und verhalten sich kooperativ usw.).

Innerhalb des TAP-Netzwerkes ist mit bekannten Protokollen eine sichere Umgebung einfach zu implementieren. Im NutzerInnen Mesh ebenfalls - bis auf die Frage der Anonymität und Ausschluß von Knoten bei nicht kooperativem Verhalten. Völlige Anonymität würde bedeuten, dass auch der oder die BetreiberIn der TAPs niemanden identifizieren kann und ein Aussperren von Knoten nicht möglich ist.

Heimvernetzung

Die Geräte in der Heimvernetzung erhalten eine *eindeutige Seriennummer*. Die KäuferInnen identifizieren sich gegenüber den Geräten mit einem mitgelieferten Code und programmieren diese auf einen gemeinsamen Heim-Netzwerkschlüssel. Nutzen die KäuferInnen die Geräte, um externe Dienstleistungen in Anspruch zu nehmen und dafür zu bezahlen, muss sich der/die NutzerIn vor dem Dienstleistenden identifizieren. Dies erfolgt über eine Signatur, welche öffentlich beglaubigt ist. Wird anonymes Micro-Payment eingesetzt, ist diese Identifizierung nicht sinnvoll. Die NutzerIn sollte ihren privaten Schlüssel nur in den Heimgeräten speichern, wenn diese durch eine Passphrase geschützt sind. Geräte können schließlich verkauft oder gestohlen werden.

Der *Inhaltsschutz und die Integrität* der Nachrichten kann innerhalb eines Haushaltes durch die symmetrischen Schlüssel erreicht werden. Da Multimedia-Anwendungen einen hohen Datenstrom erzeugen, sollte die Verschlüsselung in der Hardware implementiert sein.

Wird die *Übertragungsfrequenz gestört*, so kann der/die NutzerIn den benutzten Kanal manuell an den Geräten umstellen. Da die Fläche eines Haushalts eher klein ist, können sich fast alle Geräte per Broadcast erreichen. Ist dies nicht der Fall, so muss im Extremfall ein Knoten die Daten erneut per Broadcast weiterleiten. Durch dieses einfache Routing sind keine weiteren Verfügbarkeitsmechanismen notwendig.

Das *Installieren von Updates* sollte aus Schutzgründen nicht automatisch, sondern manuell durch die NutzerInnen erfolgen. Bietet der/die HerstellerIn Updates auf der Homepage an, laden NutzerInnen diese am Computer herunter und installieren diese von diesem direkt auf den Mesh-Geräten. Die Mesh-Geräte selbst sollten keine Möglichkeit haben, Kontakt zu ihrem/ihrer HerstellerIn aufzunehmen. Sollten die HerstellerInnen statistische Daten und Fehleranalysen erstellen wollen, so muss dies explizit von den NutzerInnen freigegeben werden. Diese wissen dann um dieses Sicherheitsrisiko.

Der Schutz von Inhaltsdaten, welche einem *Copyright* unterliegen, ist auf den unteren Ebenen der Mesh-Struktur kaum möglich. Hier werden Nachrichten, egal welchen Inhaltes, übertragen. Denkbar ist ein Versand an Knoten mit freigeschalteten Identitäten. Dies erfordert eine zentrale Instanz, welche die Zertifikate für berechtigte Geräte verwaltet. Dieser Ansatz hat zwei Hauptnachteile. Einerseits könnten inhaltsgeschützte Daten nur über Geräte gelesen werden, welche selbst zertifiziert sind und sich an die Sicherheitsanforderungen halten. Für eineN NutzerIn ist es sicher nicht nachvollziehbar, warum sie/er sich zum Abspielen einer DVD ein spezielles Gerät kaufen muss und nicht den alten DVD-Player oder den Computer nutzen kann.

Eine weitere Erfahrung ist, dass Sicherheitsmethoden, welche auf einem geheimen Schlüssel beruhen (Identifizierte Geräte), früher oder später durch Knacken des Schlüssels gebrochen werden. Ein zuverlässiger Schutz ist somit nur temporär.

Zusammenfassung

Mesh-Netzwerke sind fit für die Heimvernetzung. Hauptproblem bleibt die verfügbare Bandbreite, welche sich mehrere Multimediageräte teilen. Falls in Mehrfamilienhäusern verschiedene Mietparteien Heim-Meshnetzwerke einsetzen, kann es zum Mangel an überlappungsfreien Frequenzen kommen.

Community-Netze

In einem Community-Mesh ist nur eine *Identifizierung* der NutzerInnen eindeutig, da Hardwareplattformen gewechselt werden können und die Geräte meist handelsübliche Computer sind, welche jedeR NutzerIn selbst anschafft und installiert.

Um egoistische und böswillige Knoten erkennen und ausschließen zu können, dürfen die NutzerInnen nicht anonym im Netzwerk agieren. Somit kommt nur asymmetrische Kryptographie in Frage, da durch diese einzelnen Personen individuelle Schlüssel zugeordnet werden können. Durch einen verteilten Schlüsselspeicher sind die Schlüssel allen NutzerInnen zugänglich. Offen ist die Authentisierung der öffentlichen Schlüssel, da es keine zentrale vertrauenswürdige Instanz gibt.

Durch Austausch von symmetrischen Schlüsseln in der Initialisierungsphase (erster Kontakt zwischen zwei Knoten) kann ein *Inhaltsschutz* der Daten erfolgen. Die *Integrität* wird durch eine Signatur gesichert.

Durch meist leistungsstarke Computer der NutzerInnen ist der Aufbau eines MIX-Netzes zum Schutz der *Kommunikationsverbindung* möglich. Erfolgen fast ausschließlich Verbindungen ins Internet über einen Gateway, können MIXe im Internet benutzt werden. Der Gatewayknoten könnte dabei schon die Rolle eines Mixes übernehmen. Aus Performancegründen ist eine Installation von MIXen innerhalb des Mesh-Netzwerkes wenig sinnvoll. Falls sie doch zum Einsatz kommen, ist die räumliche Topologie zu beachten, so dass Daten nicht mehrmals auf ein und derselben Strecke übermittelt werden.

Bei Community-Netzen wird fast ausschließlich billige Standardhardware zum Einsatz kommen. Eine Anpassung der Frequenzen bei *Störungen* ist somit nur im Rahmen der Spezifikationen möglich. Ein Wechsel muss auch allen Beteiligten bekannt gemacht werden. Als Routingprotokoll kann SDSR mit seinen Kooperations- und IDS-Methoden eingesetzt werden.

Eine *faire Verteilung der Bandbreite* ist ohne Kenntnis der Topologie schwer zu implementieren. Hier sollten Verkehrsanalysen durchgeführt werden und ggf. innerhalb "wichtiger" Knoten ein Load-Balancing stattfinden.

Der weiteren Sicherheitsanforderung nach *Schutz des Rechners / Vertrauen in die Software* kann nur allgemein durch Offenlegung der Quellcodes entsprochen werden. Sollen Kosten durch den Internetzugang von den NutzerInnen getragen werden, empfielt sich ein Spendenmodell. Ausgaben und Einnahmen werden transparent auf einer Webseite veröffentlicht. Werden Verfahren im Mesh-Protokoll implementiert, die eine personengebundene Abrechnung nach Datenaufkommen berechnen, würde dies dem

Ansatz der Anonymität zuwiderlaufen und dem gemeinschaftlichen Ansatz - jedeR soll mitmachen, egal welche finanziellen Möglichkeiter er/sie hat - widersprechen.

Zusammenfassung

Offen sind die Fragen nach einem Vertrauensmodell ohne zentrale Instanz und einer Möglichkeit, egoistische und böswillige Knoten aus dem Netzwerk sicher auszuschließen.

Technisch gesehen stellt die Skalierbarkeit des Protokolls und die erzielte Bandbreite für einzelne NutzerInnen die größte Herausforderung dar. Darüber hinaus müssen die vorgeschlagenen Methoden des Inhaltsschutzes, der Integrität und die Authentifizierung berechtigter Knoten miteinander harmonisieren.

5.5 Zusammenfassung

Durch Kenntnis des Grundaufbaus eines Netzwerkes (Anzahl Knoten, Dynamik) und der erwarteten Topologie, können bereits in der Entwicklung eines Sicherheitsprotokolls sinnvolle Annahmen getroffen und Optimierungsmöglichkeiten für Sicherheitsmethoden ausgewählt werden. Dies wird nur im Fall des VANETs, welches z.B. kein Absichern des Routings erfordert, da alle Nachrichten per Broadcast verteilt werden, berücksichtigt.

Eine allgemeine Mesh-Schutzstruktur für alle Szenarien exisiert nicht. Zu unterschiedlich sind die Voraussetzungen und die Schutzbedürfnisse. Manche Dinge, wie Identifizierung, lassen sich nur für Spezialfälle lösen und bleiben ansonsten ein benanntes Spannungsfeld.

Weiterhin werden in verschiedenen Arbeiten Annahmen getroffen, die alleine stehend wenig Sinn machen. Zum Beispiel setzen einige “sichere Routingverfahren“ das Vorhandensein gegenseitiger geheimer Schlüssel vorraus. Wie diese Schlüssel im Vorfeld (ohne funktionierendes Routing) ausgetauscht werden, bleibt ungeklärt.

Nur einzelne Arbeiten, wie [50], beschäftigen sich mit einem übergreifenden Sicherheitsansatz. In dieser Arbeit wird jedoch wieder von einer allgemeinen Mesh-Struktur ohne konkreten Einsatzbezug ausgegangen und die Forderung nach einer weltweiten Zertifizierungsinstanz für alle Mesh-Geräte, welche im übrigen noch auf einer “sicheren“ Kryptoeinheit beruhen müssen, hinterläßt ein unbefriedigendes Gefühl, da die Umsetzung momentan sehr unrealistisch scheint.

Ich denke, dass für die Szenarien VANET, Gebäude- und Heimvernetzung als auch für die Kommunikation von Rettungskräften ausgereifte Sicherheitsmethoden existieren, welche nur in einem übergreifenden Schutzzusammenhang auf ein konkretes Szenario angepasst werden müssen. Zu beachten ist dabei, dass Fragen wie Schlüsselmanagement nicht ausgeklammert werden dürfen.

Im nächsten Kapitel sollen die Überlegungen für ein sicheres Community-Netzwerk fortgesetzt werden und in einer Beschreibung einer umfassenden Sicherheitsstruktur münden. Die Wahl fiel auf dieses Szenario, weil genügend Erfahrungen über den Aufbau vorliegen und die Ausarbeitung später veröffentlicht und praktisch eingesetzt werden kann.

6 Sichere Mesh-Architektur für Community-Netze

In diesem Kapitel wird eine sichere Mesh-Architektur für Community-Netze entwickelt. Grundlage sind die Anforderungen der Beteiligten.

Als Routingprotokoll wurde das proaktive Routingprotokoll OLSR ausgewählt. Es bietet einige Vorteile für die Sicherheitsarchitektur - zum Beispiel kennen Knoten die Topologie des Netzwerkes. Ein weiterer Grund für die Wahl ist die Unik-OLSR-Software. Durch die RFC-kompatible Implementierung mit Plugin-Funktionalität kann die hier entworfende Architektur später einmal direkt umgesetzt werden.

Das bereits existierende Secure-Plugin dieser Software beinhaltet eine Integritätssicherung aller Nachrichten. Gültige Signaturen können alle Knoten, welche einen geheimen symmetrischen Netzwerkschlüssel kennen, erzeugen. Der große Nachteil dieses Ansatzes ist der gleiche Schlüssel für alle Knoten. In der folgenden Ausarbeitung soll dieses Plugin in verbesserter Form benutzt werden.

Nach einer klaren Zielbeschreibung und Klärung der Rahmenbedingungen im folgenden Abschnitt werden die OLSR-Grundfunktionen erklärt. Die weiteren Betrachtungen beschäftigen sich mit der Frage, wie diese Funktionen und weitere Sicherheitsinteressen geschützt werden können.

Die Basis fast aller Sicherheitsfunktionen ist die Identifizierung und Schlüsselverwaltung. Deshalb wird diese zuerst erläutert, bevor auf die konkreten Schutzziele im einzelnen eingegangen wird. Nach Beschreibung der Protokolle werden Angriffsmöglichkeiten untersucht.

Eine besondere Stellung nehmen DoS-Angriffe ein, welche Protokollfunktionen ausnutzen. Da dies alle Teilaspekte der weiteren Arbeit betrifft, wird in einem eigenen Abschnitt auf die Problematik eingegangen. Den Abschluss des Kapitels bildet eine Zusammenfassung, ob und in welcher Form die Architektur für Community-Netzwerke geschützt werden konnte.

6.1 Zielbeschreibung

Ziel der Ausarbeitung ist die Entwicklung einer sicheren Mesh-Architektur für Community-Netze. Die Sicherheitsinteressen, welche im Kapitel 3 erläutert wurden, werden nun in konkrete Teilziele umformuliert.

Beteiligte	Sicherheitsinteressen
NutzerInnen	Faire Beteiligung an der verfügbaren Bandbreite Anonymität vor anderen Personen Schutz der übertragenen Daten Schutz der Kommunikationsbeziehung Keine Kosten für nicht genutzte Dienstleistungen Schutz der privaten Daten auf dem eigen Rechner

Die letzten beiden Punkte der Tabelle werden nicht betrachtet, da für diese nur allgemeine Hinweise gegeben werden können (siehe Kapitel 4.6). Ein technischer Schutz liegt außerhalb einer Mesh-Implementierung.

Folgende Teilziele sollen in dieser Arbeit erreicht werden:

1. Der Inhalt aller Daten soll verschlüsselt werden (end-to-end).
2. Die Integrität der versandten Daten wird sowohl hop-to-hop als auch end-to-end durch eine Signatur gesichert.
3. Nur vertrauenswürdige Knoten können sich am Netzwerk beteiligen (Routing und Datenversand).
4. Angriffe auf das Routingprotokoll und Manipulationen werden erkannt und veränderte Daten verworfen.
5. Die Kommunikationsbeziehung wird soweit wie möglich verschleiert. Dies umfasst auch den Wunsch nach Anonymität.
6. Egoistische und böswillige Knoten werden erkannt und können aus dem Netzwerk isoliert werden.
7. Die Bandbreite wird möglichst fair unter allen berechtigten Knoten aufgeteilt.

Durch das proaktiven OLSR-Protokolls werden bestimmte Grenzen gesetzt. Jeder Knoten muss eine globale und aktuelle Sicht auf die Topologie des Mesh-Netzwerkes haben. Einzelne Knoten können nicht ständig ihre Identität wechseln, da sonst kein Routing mehr möglich ist. Wie stark sich diese Punkte zum Beispiel auf die Anonymität auswirken, wird im konkreten Abschnitt beschrieben.

Ebenfalls keine Betrachtung findet der Schutz gegen Störsender. Das sogenannte *jamming* stellt allgemein eine offene Frage in funkbasierten Netzen dar. Wichtig ist nur, dass durch DoS-Angriffe andere Sicherheitsmechanismen nicht unterlaufen werden können.

Im Vergleich mit den im letzten Kapitel beschriebenen Forschungsarbeiten diese Arbeit eine umfassende Sicherheitsarchitektur aufzubauen. Neben den Grundprotokollen werden auch die Schlüsselverwaltung und der Schutz der Kommunikationsbeziehung beachtet.

Der Ansatz einer Schlüsselverwaltung mit einer lokalen und globalen Vertrauenssicht, welche auf der Funktionsweise des OLSR-Protokolls aufbaut, wurde in wissenschaftlichen Arbeiten - meines Wissens - noch nicht betrachtet.

6.2 Rahmenbedingungen

Grundlage der Betrachtungen ist das Szenario eines Community-Mesh-Netzwerkes. Ich gehe davon aus, dass sich etwa 20 bis 100 Knoten im Netzwerk befinden. Die Entfernungen zwischen ihnen betragen innerhalb eines Hauses nur wenige Meter. Verbindungen zwischen verschiedenen Häusern können durchaus mehrere Kilometer betragen (Richtfunkantennen).

Die Knoten sind kaum oder gar nicht mobil. Die Dynamik aufgrund der Position von Knoten ist gering. Ein Teil der Knoten, welche wichtige Verbindungen zwischen den Häusern herstellen, sind immer verfügbar (sofern funktionsfähig). Das Datenaufkommen ist hoch und kann zu Spitzenzeiten die maximale Bandbreite der Funkübertragung erreichen (11 MBit bei regulärem 802.11b). Der Datenverkehr verläuft hauptsächlich zwischen den einzelnen Knoten zu wenigen Community-Servern und Internet-Gateways.

Die Geräte der NutzerInnen sind meisten aktuelle Desktop-PCs oder Laptops, welche über viel Rechenleistung und Speicher verfügen. Die Knoten, welche nur die Daten zwischen verschiedenen Häusern routen, bestehen aus Computern älterer Generation. Durch den Einsatz von Linux und die Beschränkung auf das alleinige Routing ohne weitere Software, können auch hier kryptographisch anspruchsvolle Rechenoperationen geleistet werden. Alle Geräte, die länger online bleiben, verfügen über einen Stromanschluss.

Für die Funktion des Mesh-Netzwerkes bedarf es keiner zentralen technischen Elemente. Die Gateways zum Internet bilden jedoch eine Daten-Nadelöhr und werden

extra betrachtet.

Für die Vergabe einer eindeutigen Identifizierung bzw. den Ausschluss egoistischer oder böswilliger Knoten (Personen) bedarf es in irgendeiner Form einer Verwaltungsinstanz, welche Identitäten generiert, speichert und sperren kann. Dies kann durch ein wöchentlich stattfindendes Plenum aller Beteiligten geleistet werden. Community-Netze sind meist überschaubar und die Empathie für das Projekt ist so groß, dass dieses zentrale Treffen die Rolle einer Verwaltung übernehmen kann.

Bei diesen Treffen können sich neue Personen vorstellen und, falls sie das Vertrauen der Anwesenden bekommen, ins Netzwerk aufgenommen werden. Dazu generieren sich die neuen Personen selbst ein Schlüsselpaar mit einem asymmetrischen Verfahren. Der öffentliche Schlüssel gilt als *Identität*. Zu dieser Identität haben verschiedene Mitglieder des Mesh-Netzwerkes ein bestimmtes Vertrauensverhältnis. Die konkrete Form der Schlüsselsignierung, Vertrauen anderer Personen in die Echtheit und die Verteilung wird später im Abschnitt Schlüsselverwaltung beschrieben.

Jeder Knoten verfügt über eine lokale und globale Sicht auf die Vertrauenswürdigkeit aller anderen Knoten. Lokal meint die Vertrauensstufen, welche ein Knoten zu anderen hat. Eine globale Bewertung kann durch die Verknüpfung der Vertrauenslisten aller Knoten bestimmt werden.

Der Begriff *Vertrauen* wird in den letzten Jahren oft verwendet und in vielfältigen Zusammenhängen gebraucht. Ich benutze den Begriff eher intuitiv, ohne dem eine wissenschaftliche Bedeutung zugrunde zu legen.

Bemerkt ein Knoten einen egoistischen oder böswilligen Knoten, speichert er die Daten, welche als Beweis gelten, ab und versendet eine Information an alle Nachbarn des verdächtigen Knotens. Der Knoten wird sofort von diesem einzelnen Knoten ignorieren. Auf dem nächsten zentralen Plenum kann einem Knoten bzw. der dahinter stehenden Person ein globales Misstrauen ausgesprochen und sie aus dem Netzwerk ausgeschlossen werden.

Im Rahmen dieser Arbeit kann nur das Mesh-Netzwerk betrachtet werden. Werden Daten über Gateways in andere Netze versandt, kann der Schutz nicht vom Protokoll geleistet werden.

6.3 OLSR-Grundfunktionen

Die Anforderungen an das OLSR-Protokoll wurden im Oktober 2003 von der *Internet Engineering Task Force* (IETF) als RFC3626 definiert [59]. Als zukünftiger internationaler Standard soll es Anwendungen und Geräte für Mesh-Netzwerke ermöglichen, welche unabhängig vom Hersteller / von der Herstellerin zusammenarbeiten können.

OLSR ist ein proaktives link-state Protokoll. Es besteht aus drei Hauptkomponenten:

- Finden von Nachbarknoten und Verbindungen
- optimiertes Fluten und Weiterleiten (MultiPoint Relaying)
- Verteilung der Routingtabellen (link-state messaging) und Routenbewertung

Nachbarknoten werden durch *HELLO-Nachrichten* gefunden, welche von allen Knoten in regelmäßigen Abständen ausgesandt werden. In der Hello-Nachricht befindet sich eine Liste aller aktiven Verbindungen zu direkten Nachbarknoten und ein Feld, welches aussagt, ob der Knoten als *Multipoint Relay* (MPR) agieren will.

Durch die Hello-Nachricht eines Nachbarknotens kann der empfangende Knoten entscheiden, ob er in dessen Nachbarliste vermerkt ist und somit vom anderen gesehen wird. Dies bedeutet eine symmetrische Verbindung und die Knoten können als MPR benutzt werden. Möglich sind auch asymmetrische Links, bei denen nur ein Knoten den Nachbarn entdeckt. Ursache kann eine unterschiedliche Sendeleistung oder Empfangsempfindlichkeit sein.

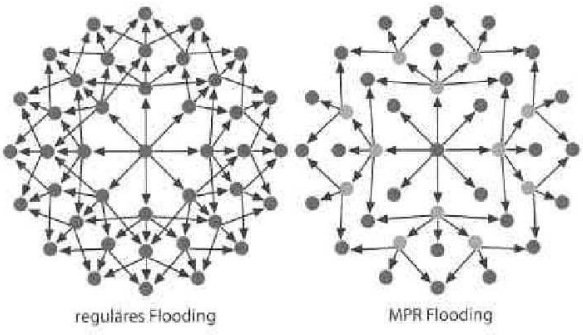

Abbildung 6.1: Reines Broadcast- und MPR-Fluten

Durch das *Multipoint Relaying* wird die Anzahl der mehrmals gesendeten Broadcast-Pakete verringert. Nicht alle, nur ein ausgewählter Teil der Knoten sendet Broadcast-

Pakete weiter. Alle Knoten wählen und verwalten ihre eigenen MPRs. Die Grundregel besagt: für alle Nachbarn, die 2 Hops entfernt sind, muss es nur einen MPR geben, der diese erreicht. Aus Redundanzgründen kann das Protokoll erweitert werden, so dass alle 2-hop Knoten über 2 MPR erreichbar sind. Dies hebt den Vorteil der geringeren Anzahl von Paketen etwas auf.

In einem klassischen *link-state-Schema* fluten alle Knoten ihre Verbindungsstatus-Informationen im Netzwerk. Bei OLSR versenden nur MPRs diese Informationen. In der Verbindungsnachricht werden auch nur MPRs aufgeführt. Dies bedeutet weniger und kleinere Verbindungsnachrichten (siehe Abbildung 6.1).

In der Verbindungsnachricht - *topology control message* (TC) - sind Identifikation des absendenden Knotens, alle seine direkten Nachbarschaftsknoten, welche als MPR ausgewählt wurden und eine eindeutige Sequenznummer enthalten. Die Sequenznummer wird bei jeder Aktualisierung der TC-Daten inkrementiert und zeigt somit die Aktualität an. Neue TC-Nachrichten werden dann gesendet, wenn sich in der Topologie etwas verändert hat oder ein festes Zeitintervall ohne Veränderungen vergangen ist.

Neben diesen Basismechanismen, im RFC als *core functionality* bezeichnet, gibt es weitere Features (*auxiliary functionality*):

- Protokoll kann auf multi-homed-Knoten (mehrere Netzwerkkarten) mittels MID-Nachrichten laufen; einem Knoten werden eine Hauptadresse und mehrere Unteradressen zugeordnet
- Signalisierung von externen Netzwerkverbindungen (z.B. Internet) durch HNA-Nachrichten; Knoten können als Gateways benutzt werden
- Knoten entscheiden, ob und in welchem Maße (willingness) sie MPR sein wollen
- Hysterese-Algorithmus zur Vermeidung von instabilen Verbindungen
- optimiertes MPR-Auswahlverfahren und Inhalt der *Topology Control Messages*
- Beachtung der Linkqualität bei Routenfindung

Jeder Knoten erstellt für sich eine Routingtabelle, um für alle Knoten im Netzwerk den kürzesten Pfad zu ermitteln. Meist wird eine Variante des Dijkstra's Algorithmus angewendet. In der Routingtabelle steht zu jedem Ziel der nächste Nachbarknoten. Sind Routen nicht mehr erreichbar, werden Error-Nachrichten versandt, welche eine neue Berechnung der Routen auslösen.

6.3.1 Nachrichtenformat und Pakete

Alle OLSR-Daten werden in Paketen, welche aus mehreren Nachrichten bestehen können, versandt. Diese Pakete bestehen aus einem *Header* und einem *Body*, wie in der Abbildung 6.2 zu erkennen ist.

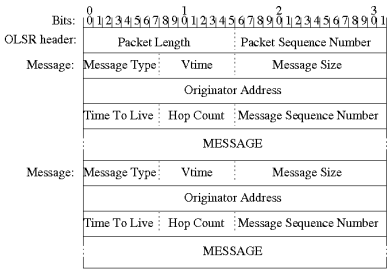

Abbildung 6.2: Standard OLSR-Nachrichtenpaket

Der Paketkopf (Header) besteht aus folgenden Feldern:

- **Packet Length** - Länge in Bytes des gesamten Paketes inklusive header
- **Packet Sequence Number** - Sequenznummer, welche jedes Mal, wenn eine neue OLSR-Nachricht von diesem Knoten versandt wird, erhöht wird.

Der Nachrichtenteil (Body) besteht aus einer oder mehreren OLSR-Nachrichten. Eine Nachricht muss folgende Elemente enthalten:

- **Message type** - Ein Integer-Wert, der den Typ festlegt. Die Nummern 0-127 sind für das eigentliche OLSR-Protokoll reserviert. 128-255 können frei für Erweiterungen des Protokolls benutzt werden.
- **Vtime** - Dieses Feld gibt an, für wie lange nach Erhalt der Nachricht ein Knoten die Informationen als aktuell betrachten soll.
- **Message Size** - Nachrichtengröße inklusive Kopf in Bytes
- **Originator Address** - Hauptadresse des absendenden Knotens
- **Time To Live** - Maximale Anzahl der hops, die die Nachricht weitergeleitet werden kann. Durch dieses Feld kann die Ausbreitung einer Broadcastnachricht begrenzt werden.
- **Hop Count** - Gibt an, wieviel Mal die Nachricht weitergeleitet wurde.

- **Message Sequence Number** - Sequenznummer, welche jedes mal erhöht wird, wenn ein Knoten eine neue Nachricht generiert.

Im Grundprotokoll werden nur vier Nachrichtentypen, welche im Abschnitt OLSR-Grundfunktionen beschrieben wurden, definiert.

Wichtig ist, dass Nachrichten, auch wenn sie im gesamten Netzwerk geflutet werden sollen, nur zum Nachbarknoten übertragen werden. Dort erfolgt ein Auspacken des Inhaltes und erst ein neu zusammengestelltes Paket wird - falls erforderlich - weiter versandt. Das allgemeine Datenformat erlaubt es, nicht nur das Broadcast-Paket, sondern auch weitere Nachrichten (z.B. Hello) als *piggybacked* an das gleiche Paket anzuhängen.

6.3.2 Unik-OLSR-Implementierung

Das Ziel der Unik-OLSR-Implementierung [17] von Andreas Tønnesen ist die Entwicklung einer RFC3626-kompatiblen Protokollimplementierung und der Suche nach nützlichen Plugins. Start des Projektes war Frühling 2003 und mittlerweile liegt die Software in der Version 0.4.10[1] für verschiedene Plattformen vor und wird im RFC als *experimental* eingeordnet.

Die Software ist unter der *General Public License* veröffentlicht worden. Daneben kommt OLSR als Embedded-Software auf dem Access Cube der 4G Systeme GmbH und den AP WRT54G von Linksys mit OpenWRT als Betriebssystem zum Einsatz.

Die Software besteht momentan aus den grundlegenden OLSR-Funktionen, einer Erweiterung eines Linkgütemessverfahrens, welches nicht RFC-kompatibel ist und etwa 7 Plugins, welche das Protokoll um einzelne Funktionen erweitern. Der OLSR-Switch ist eine Testumgebung, welche mehrere virtuelle OLSR-Instanzen erlaubt und zum Testen eigener Implementierungen dient.

6.3.3 Plugins

OLSR unterstützt das dynamische Laden von Plugins, welche eigene Datentypen generieren und verarbeiten können. Dabei müssen keine Veränderungen am eigentlichen OLSR-Daemon implementiert werden. Die Plugins können in jeder beliebigen Programmiersprache geschrieben werden, solange sie sich als dynamische Bibliothek kompilieren lassen.

[1] Stand Februar 2006

Das Plugin muss nicht auf allen Knoten im Netzwerk laufen. Durch eine Standard-Weiterleitung der OLSR-Software von unbekannten Typen von Datenpakten werden sie zuverlässig im Netzwerk verteilt.

Die gesamte OLSR-Software ist modular aufgebaut. In der Abbildung 6.3 sind die einzelnen Komponenten erkennbar. Plugins haben die Möglichkeit, an verschiedenen Punkten die Daten zu bearbeiten bzw. wieder zurück an die OLSR-Software zu leiten.

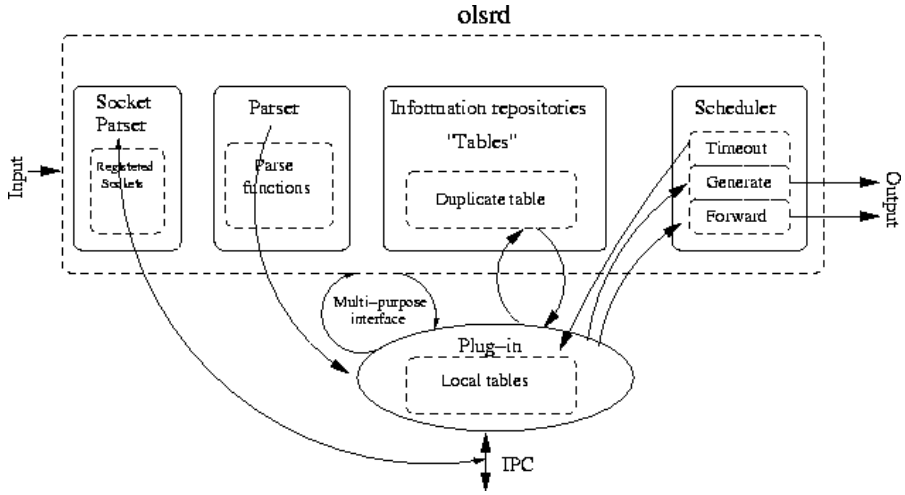

Abbildung 6.3: Ladbare Plugins im OLSR-Daemon

Folgende Plugins sind in der OLSR-Version 0.4.10 verfügbar:

- *HttpInfo* - durch einen kleinen HTTP-Server können Informationen des OLSR-Routing-Programms per Webinterface abgerufen werden
- *Mini* - Demonstrationsplugin, welches nur Funktionsweise verdeutlicht und bei Start und Beenden eine Nachricht ausgibt
- *Nameservice* - einfacher DNS-Ersatz für OLSR-Netzwerke. Jeder Knoten im Netz kann seine eigene oder die IP-Adresse von direkten HNA-Nachbarn an einen Namen binden. Diese Bindungen werden verteilt und von allen Knoten gespeichert. Durch DNS-Masquerade kann ein Knoten die Namen von DNS-Internet-Server in OLSR-DNS-Namen umsetzen.
- *Dynamic Internet Gateway* - dynamisches Aktualisieren von HNA-Nachrichten, falls der Knoten über eine Internetverbindung verfügt, welche nicht immer besteht. Periodisches pingen von Internetadressen, um die Verfügbarkeit zu testen.

- *Dot draw* - Ausgabe der Netzwerktopologie im dot-format
- *Secure OLSR* - Schutz der Datenintegrität durch Signatur über alle Nachrichten. Nur Knoten mit einem symmetrischen geheimen Schlüssel können am Routing teilnehmen und Signaturen erzeugen.
- *Power plugin* - Informationen über den Energiestatus der Knoten werden über MPR-Fluten verteilt. Dieses Plugin dient hauptsächlich als Beispiel für weitere Plugin-Implementierungen.

6.3.4 OLSR-Switch-Netzwerksimulator

Der OLSR-Switch ist eine Anwendung, welche Netzwerkverkehr zwischen verschiedenen OLSR-Instanzen über TCP mittels eines loopback-Interface routet. Zwei Datenbanken werden benutzt - eine zum Verwalten der virtuellen Knoten und eine zum Verwalten der Links. Durch die Links wird eine Verbindungsqualität gesetzt. Der Wert 100 bedeutet, dass beide Knoten sich sehen und alle Daten weitergeleitet werden. Im Gegensatz dazu steht der Wert 0 für keine Verbindung. Abstufungen in diesem Intervall geben dann verschieden Qualitätsstufen an.

In einer Shell können verschiedene Werte gesetzt werden und Informationen des OLSR-Protokolls abgerufen werden. Durch Erhöhung der Anzahl der Knoten sind einfache Performance-Abschätzungen des Protokolls samt Plugins möglich.

6.4 Schlüsselverwaltung und Vertrauensbewertung

Jeder Knoten in einem Netzwerk wird durch seinen öffentlichen Schlüssel eines asymmetrischen Schlüsselpaars identifiziert. Der Schlüssel ist selbst-signiert.

Den Nachweis, dass ein Datenpaket wirklich von einem bestimmten Knoten kommt, bzw. zu einer entsprechenden Identität gehört, wird durch die Signatur erbracht. Am Anfang jeder Nachricht steht die Identität als kurzer Fingerprint und am Ende befindet sich die Signatur über die gesamte Nachricht.

Ähnlich dem Web-of-Trust-Ansatz, wird einem Schlüssel durch andere Knoten ein bestimmtes Vertrauen ausgesprochen. Dies geschieht nicht absolut, sondern durch verschiedene Vertrauensstufen. Folgende Vertrauensstufen kann ein Knoten zu einem anderen haben:

1. unknown - Der Schlüssel ist unbekannt.

2. marginal - Einem Schlüssel wird wenig Vertrauen geschenkt. Vergleichbar ist diese Bewertung mit der Situation, in der ich eine Person nur oberflächlich kenne, ihr aber trotzdem ein vorsichtiges Vertrauen entgegenbringe.

3. trusted - Das Vertrauen ist hoch. Ich bin vom korrekten Verhalten einer Person überzeugt.

4. untrusted - Einem Knoten wird nicht vertraut. Er wird als egoistisch oder böswillig eingestuft.

Alle Knoten im Netzwerk speichern ihre Vertrauensstufe zu allen anderen Knoten (Trust-Liste). Durch das proaktive Protokoll sind ihnen alle Knoten im Netzwerk bekannt. Die eigene Sicht des Vertrauens in andere Knoten wird ähnlich der TC-Nachrichten im Netzwerk verteilt. Das MPR-Fluten kann angewendet werden. Ein Unterschied zu TC-Nachrichten ist, dass nicht nur die Nachbarknoten in der Nachricht aufgezählt werden, sondern die Vertrauensbeziehungen zu allen Knoten versendet werden.

Jeder Knoten kann die Sicht des Vertrauens auf andere Knoten ändern. Die Veränderungen werden im Netzwerk verteilt. Alle Knoten im Netzwerk speichern diese Vertrauensbewertungen und kennen somit alle Vertrauensbeziehungen. In dem Fall, dass sich eine Vertrauenssicht auf *untrusted* verändert, wird die Routingtabelle in diesem Knoten neu berechnet und der Link zum nicht vertrauensvollen Knoten als nicht existent angesehen.

Ein Knoten kommunziert nur mit einem anderen, wenn er dessen öffentlichen Schlüssel (seine Identität) als gültig einstuft. Dies bedeutet sowohl die Weiterleitung eigener Daten über diesen Knoten, als auch die Annahme von Nachrichten dieses Knotens.

Eine Identität ist dann gültig, wenn

- das Vertrauen zwischen beiden kommunizierenden Knoten *trusted* ist oder eine transitive Vertrauensverbindung *trusted* zwischen beiden Knoten existiert (der eine Knoten vertraut einem Knoten, der dem Zielknoten vertraut usw.).

- sie mindestens 10% aller Knoten im Netzwerk bekannt ist. Dies müssen mindestens 2 Knoten sein.

Die Identität eines Knotens ist immer ungültig, wenn mehr als 20% aller Knoten, die die Identität bewertet haben, den Zielknoten als *untrusted* sehen. Dies müssen ebenfalls mindestens 2 Knoten sein. Die positive Formulierung für diese Ausschlussgrenze kann lauten: “Solange mir 80% der Knoten vertrauen, die mich bewertet haben, kann ich im Mesh-Netzwerk teilnehmen“.

Wie in den Rahmenbedingungen beschrieben (Kapitel 6.2), gehe ich von einem Mesh-Netzwerk mit 20-100 Knoten aus. Je nach Größe muss ein Knoten somit mindestens 2 bzw. 10 Knoten bekannt sein (10%). Dies kann durch das Plenum oder persönliche Kontakte erreicht werden. Die Mindestschwelle von 2 Knoten soll verhindern, dass eine Person allein neue Knoten in einem kleinen Netzwerk integrieren kann.

Der zweite Schwellwert von 20% als Ausschlusskriterium bezieht sich nur auf Knoten, welche schon eine Vertrauensbewertung zu einem verdächtigtem Knoten haben. Die Mindestanzahl von 2 Knoten soll wiederum verhindern, dass eine Person allein einen anderen Knoten ausschließen kann. Besteht das Mesh-Netzwerk aus 20 Knoten, müssen zwischen 2 und 4 Knoten ihr Misstrauen aussprechen. Bei 100 Knoten ist die Spannbreite, welche auf der Bekanntheit des Knotens beruht, zwischen 2 und 20.

Die Werte wurden so gewählt, dass sowohl mit geringem Aufwand ein Knoten in das Netzwerk integriert werden kann, als auch ein Ausschluss möglich ist. Der eventuell fälschliche Ausschluss wird immer schwerer, je mehr Knoten eine Identität kennen. Dies soll das gegenseitige Einschätzen und Bewerten motivieren.

Für eine gegenseitige Bewertung durch mehr Knoten als der Mindestbekanntheitswert fordert, spricht auch, dass je mehr Knoten neu ins Netzwerk kommen, mehr Knoten für die 10% Bekanntheit notwendig sind.

In Abbildung 6.4 wird ersichtlich, wann ein Knoten eine Identität als gültig ansieht und Daten von oder zu diesem Knoten weiterleitet.

Die Prozentzahlen können verwendet werden, da durch das OLSR-Grundprotokoll jeder Knoten jeden anderen kennt (Gesamtzahl) und durch die später beschriebenen Trust-Nachrichten die Vertrauensbewertungen aller Knoten an alle anderen versandt werden. Differenzen zwischen der tatsächlichen globalen Vertrauenssicht und der vom Knoten berechneten Werte sind nur innerhalb kleiner Zeitschranken möglich. Die Zeitspanne ist die Periode, in der die TC- und Trust-Nachrichten verschickt werden und führt nur dazu, dass Knoten *etwas später* ins Netzwerk integriert oder ausgesperrt werden. Dies halte ich für vernachlässigbar.

Die Prozentzahlen für die Gültigkeit der Identität bilden einen *Schwellwert* für die Teilnahme eines Knotens am Netzwerk. Die Verwendung einer dynamischen Bezugsgröße bietet den Vorteil, dass das System unabhängig von der Gesamtanzahl aller Knoten arbeitet. Bei festen Größen ("5 Knoten müssen mir vertrauen und ich kann teilnehmen") müssen die Schwellwerte immer dann angepasst werden, wenn viele neue Knoten ins Netzwerk kommen oder eine große Anzahl dieses verlässt.

Anzumerken ist, dass ein absendender Knoten nicht die Route bestimmen kann,

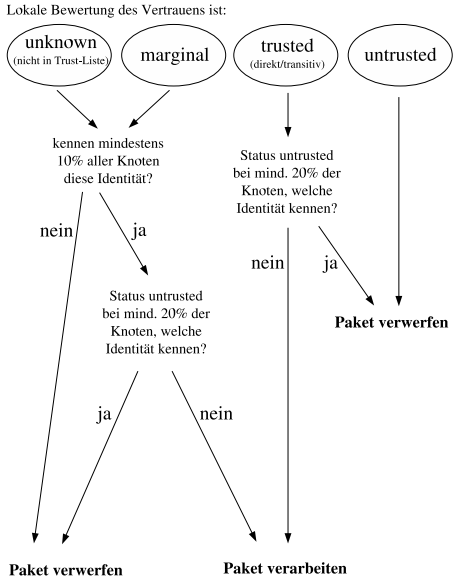

Abbildung 6.4: Entscheidung über die Verarbeitung von Daten

auf der die Daten zum Ziel gelangen. Die Knoten entscheiden individuell, ob sie dem nächsten Hop vertrauen. Auf der Route können somit auch Knoten liegen, welchen der absendende Knoten nicht vertraut. Dies ist nicht weiter tragisch, da der Knoten die Sicherheitsstruktur nicht beeinträchtigen kann und im schlimmsten Fall die entsprechenden Datenpakete nicht weiter versendet. Dieses Verhalten kann, wie später beschrieben, durch die benachbarten Knoten entdeckt werden. Diese stufen den Knoten als *untrusted* ein und er wird nach einer kurzen Zeitspanne nicht mehr benutzt.

Ein weiterer lokaler Effekt ist, dass benachbarte Knoten, welche sich vertrauen (Status trusted), aber keinen weiteren Knoten (unter 10%) im Netzwerk bekannt sind, trotzdem miteinander kommunizieren können.

6.4.1 Neue Knoten und Initialisierung

Will eine Person, dass ihr Computer ein Knoten im Mesh-Netzwerk wird, so generiert sie sich selbst ein Schlüsselpaar mit dem vereinbarten asymmetrischen Kryptoverfahren. Der öffentliche Schlüssel ist die Identität und wird von der Person selbst signiert. Die Teilnahme hängt nun davon ab, dass mindestens 10% aller Knoten diese Identität als *trusted* oder *marginal* einstufen. Die neue Person kann sich auf dem Plenum vor-

stellen und dort um Vertrauen werben. Falls sie genügend TeilnehmerInnen persönlich kennt, kann dies natürlich privat und ohne Plenum erfolgen.

Vertrauen bisherige TeilnehmerInnen der neuen Person, tragen sie die Identität (öffentlicher Schlüssel oder Fingerprint) der neuen Person in die Trust-Liste ihrer Mesh-Software ein und setzen den Status auf *marginal* oder *trusted*. Diese Information wird automatisch durch das Protokoll im Netzwerk verteilt. Sobald dies genügend Personen getan haben, kann der neue Knoten im Mesh-Netzwerk teilnehmen.

Wird das Mesh-Netzwerk erstmalig initialisiert, so sieht der Ablauf ähnlich wie mit einer einzelnen Person aus. Alle Mesh-InitiatorInnen erstellen sich einen Schlüssel und tragen die der anderen manuell in ihre Trust-Liste ein.

6.4.2 Ausschluss von Knoten

Ein Knoten ist dann vom Netzwerk ausgeschlossen, wenn 20% aller Knoten, welche die Identität bewertet haben, ihn als *untrusted* einstufen.

Der individuelle Ausschluss kann durch Setzen der Vertrauensstufe auf *untrusted* bewirkt werden. Ein individueller Ausschluss hat jedoch nur eine begrenzte Wirkung auf den verdächtigen Knoten und wird diesen nicht dazu bringen, sich protokollkonform zu verhalten. Deshalb werden die Nachbarknoten des verdächtigten Knotens aufgefordert, ihn zu überwachen und bei Festellung eines Fehlverhaltens ebenfalls als *untrusted* zu bewerten. In kleinen Netzwerken kann so ein globaler Ausschluss erreicht werden.

In größeren Netzen kann die 20%-Schwelle oftmals nicht erreicht werden. Um trotzdem egoistisches Verhalten zu bestrafen, kann auf dem zentralen Plenum das Fehlverhalten erläutert werden. Sind genügend Personen überzeugt (mindestens 20%) und setzen ihre Vertrauenssicht auf *untrusted*, wird der Knoten ausgeschlossen.

Die soziale Kontrolle auf einem Plenum soll eine Diffamierung verhindern, auch wenn ein Mißbrauch nicht gänzlich ausgeschlossen werden kann. Auf dem Plenum sollte der verdächtigten Person die Gelegenheit gegeben werden, zu den Vorwürfen Stellung zu beziehen.

Bemerkt eine Person, dass ihr Schlüssel kompromittiert ist, kann sie natürlich selbst auf dem Plenum die Sperrung des eigenen Schlüssels fordern.

6.4.3 Angriffsmöglichkeiten auf die Schlüsselverwaltung

Angriffe auf das Vertrauensmodell können in zwei Kategorien unterteilt werden. Auf der einen Seite kann ein Knoten versuchen, unberechtigt in ein Netzwerk aufgenommen zu werden - das bedeutet, eine gültige Identität zu bekommen. Auf der anderen Seite können ein oder mehrere Knoten versuchen, den Austausch von Vertrauensinformationen zu manipulieren, um eigene Angriffe zu verschleiern oder andere Knoten zu diffamieren.

Unberechtigte gültige Identität

Ein Knoten kann nur dann im Mesh-Netzwerk teilnehmen, wenn mindestens 10% aller sich bereits im Netzwerk befindlichen Knoten seine Identität als *marginal* oder *trusted* bewerten. Da jeder Knoten lokal seine Vertrauensbeziehungen verwaltet und diese nicht durch ein Protokoll des Netzwerkes verändert werden können, kann dies nur durch Überzeugung der Personen im Netzwerk geschehen. Und genau dies ist der Ablauf des zentralen Plenums.

Kontrolliert ein Angreifer bzw. eine Angreiferin 10% aller Knoten, welche sich berechtigt im Mesh-Netzwerk befinden, so kann dieseR beliebig viele neue Knoten im Netzwerk integrieren. Dieses Manko betrifft jedoch alle Schwellwertschemata, da die Sicherheit auf der Annahme beruht, dass die Anzahl der kompromittierten Knoten nie die Schwelle übersteigt.

Eine weitere Möglichkeit ist, sich physisch Zugang zu Knoten zu verschaffen und dort die eigene Identität einzutragen. Dies beinhaltet auch direkte Angriffe auf das Betriebssystem oder die Software der Mesh-Knoten. Diese Angriffe können nicht durch die OLSR-Software samt Plugins abgewehrt werden. Einzig die Geräte, welche z.B. auf Dächern platziert sind und zu denen relativ unbemerkt Zugang erfolgen kann, sollten durch gute Passwörter geschützt sein. Die installierte Software sollte ebenfalls auf das Notwendigste beschränkt bleiben, um unnötige Schwachstellen zu vermeiden.

Manipulation von Trust-Nachrichten

Trust-Nachrichten werden per Broadcast mit dem MPR-Verfahren im Netzwerk verteilt. Will ein Knoten Einfluss auf diese Nachrichten nehmen, kann er:

- den Nachrichteninhalt verändern,
- veraltete Nachrichten erneut senden (Replay-Angriff),
- unter falscher Identität Informationen senden (Spoofing),

- ausgewählte Nachrichten nicht weitersenden oder unbrauchbar machen.

Alle Nachrichten werden durch eine AbsenderInnen-Signatur geschützt. Eine *Manipulation des Dateninhalts* wird somit erkannt, solange der Schlüssel des absendenden Knotens nicht kompromittiert ist. Diese Signatur schützt ebenfalls vor *Spoofing*, da diese nur der bzw. die echte AbsenderIn erzeugen kann.

Replay-Angriffe können durch Sequenznummern, welche bei jedem Versenden von Trust-Nachrichten inkrementiert werden, verhindert werden. Erhalten Knoten eine solche Nachricht, überprüfen sie zunächst, ob die Sequenznummer größer als die zuletzt gesendete ist. Ist dies nicht der Fall, wird die Nachricht verworfen.

Im Header eines OLSR-Paketes existiert bereits eine Sequenznummer, welche jedoch nur eine 16 Bit-Länge hat. Dies ist nicht ausreichend, da es schon nach kurzer Zeit (etwa 18 Stunden bei einer Trust-Nachricht je Sekunde) zu einem Überlauf und Neubeginn kommt. Deshalb werden 32 Bit-Sequenznummern verwendet. Hier beträgt die Zeitspanne eines Überlaufs mehr als 100 Jahre (bei einer Trust-Nachricht je Sekunde).

Durch die Broadcast-Versendung der Nachrichten können diese auf verschiedenen Wegen zu allen Knoten gelangen. Eine effektive *Blockierung* ist somit nur möglich, wenn der Zugang zum gesamten oder einem Teil des Netzwerkes nur durch den egoistischen / angreifenden Knoten stattfinden kann.

Um zu erkennen, wann die Verteilung der Trust-Nachrichten blockiert wird, senden alle Knoten eine signierte Antwort - *Acknowledge*-Nachricht (ACK) - inklusive der Sequenznummer an den absendenden Knoten. Der Knoten wartet eine bestimmte Zeitspanne auf die Antworten. Empfängt er kein ACK, ist entweder der Knoten ausgeschaltet, es existiert ein Verbindungsproblem (Überlastung oder Kollision) oder die Nachricht wurde blockiert.

Der Knoten wartet nun auf eine TC-Nachricht von dem nicht erreichbaren Knoten. Solange, keine solchen Nachrichten eintreffen, geht er davon aus, dass der Knoten sich momentan nicht im Netzwerk befindet. Trifft eine TC-Nachricht ein, so sendet er per unicast seine Trust-Liste an diesen Knoten mit der Bitte um eine Empfangsbestätigung (ACK). Trifft diese nicht ein, so geht er von einer Blockierung seiner Vertrauenslisten aus.

Jeder Knoten verfügt über Topologieinformationen durch die TC-Nachrichten. Auf Grundlage dieser Daten und den erhaltenen ACK kann er ermitteln, auf welchem Link die Nachricht blockiert wurde. Nicht entscheiden kann er, welcher der beiden Knoten des blockierten Links der verursachende Knoten ist. Abbildung 6.5 zeigt dieses Dilemma.

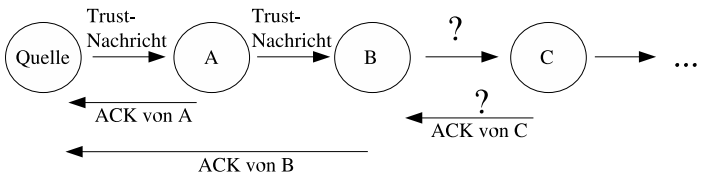

Abbildung 6.5: Entscheidungsdilemma, ob B oder C blockiert

Ein Knoten kann durch die ihm vorliegenden Informationen nicht herausfinden, ob Knoten B zwar ein ACK sendet, aber die Trust-Nachricht nicht an C weiterleitet, ob B die Nachricht an C versandt hat, aber das ACK blockiert oder ob C die Trust-Nachricht erhalten hat, sie nicht weiter versendet und kein ACK sendet.

Im Abschnitt Egoistische Knoten (6.9) wird beschrieben, wie die Blockierung einem Knoten zugeordnet werden kann. Innerhalb des hier beschriebenen Vertrauensaustauschprotokolls wird eine Blockierung bemerkt, die konkrete Identifizierung erfolgt jedoch mit einem weiteren Protokoll.

6.4.4 Trust-Liste und Trust-Nachrichten

Die Trust-Liste eines jeden Knotens besteht aus allen im Netzwerk befindlichen Knoten. Auch Knoten, welche länger in keinen TC-Nachrichten vorhanden waren und sich damit augenscheinlich nicht mehr im Netzwerk befinden, werden für einen längeren Zeitraum (3 Monate) gespeichert. Dies stellt sicher, dass diese Knoten auch nach einer Abwesenheit, wie einem längerem Urlaub, im Anschluss wieder im Netzwerk bekannt sind und teilnehmen können.

Neben der eigenen Sicht der Vertrauensbewertungen werden die Vertrauensbewertungen aller anderen Knoten ebenfalls gespeichert. Hinzu kommen die von den Knoten verwendeten Sequenznummern, um die Aktualität von Trust-Nachrichten zu gewährleisten. Die *Trust-Tabelle* kann folgendermaßen aussehen:

	eigeneID	Ident 01	Ident 02	Ident 03	....	Ident X
Fingerprint	AC..	4F..	1B..	DD..	....	F5..
letzte Sequenz-Nr.	123	565	266262	22526	....	3508223
eigene ID	trusted	trusted	unknown	marginal	....	unknown
Ident 01	trusted	trusted	unknown	unknown	....	unknown
Ident 02	marginal	marginal	trusted	marginal	....	marginal
Ident 03	unknown	unknown	unknown	trusted	....	unknown
....	....	....	....	....	....	....
Ident X	marginal	marginal	unknown	marginal	....	trusted

Um die Größe der zu übertragenden Daten zu minimieren, wird zur Identifizierung einzelner Knoten nicht der gesamte öffentliche Schlüssel (mindestens 1024 Bit) übertragen, sondern deren eindeutiger Fingerprint (160 Bit). Zum schnellen Zugriff werden diese einmalig berechnet und in der ersten Zeile der Trust-Tabelle gespeichert.

In der zweiten Zeile werden die zuletzt verwendeten Sequenznummern der verschiedenen Knoten vermerkt. Ab der dritten Zeile werden die Vertrauensbewertungen jedes einzelnen Knotens gespeichert. Die dritte Zeile betrifft dabei den eigenen Knoten.

Die Speichergröße der Tabelle richtet sich nach der Anzahl der sich im Netzwerk befindlichen Knoten. Je Knoten werden etwa 1226 Bit Speicher benötigt - 1024 Bit für den öffentlichen Schlüssel, 160 Bit für den Fingerprint und 32 Bit für die Sequenznummer. Zur Speicherung einer Vertrauensbeziehung reichen 2 Bit aus (00-unknown, 01-marginal, 10-trusted, 11-untrusted). Befinden sich 100 Knoten im Mesh-Netzwerk, werden zur Speicherung 17,4 kByte benötigt[2]. Diese Anforderung kann von den verwendeten Geräten erfüllt werden.

Durch *Trust-Nachrichten* wird die eigene Sicht auf andere Knoten im Netzwerk verteilt. In der Nachricht werden der eigene Fingerprint, eine inkrementierte Sequenznummer, die Vertrauenssicht (2. Zeile in der Trust-Tabelle) und eine Signatur über die gesamte Nachricht versandt. Der Aufbau der Nachricht ist in der Abbildung 6.6 zu sehen.

Nachdem ein Knoten eine Trust-Nachricht empfangen hat, sendet er ein ACK als Bestätigung. Im ACK-Paket befindet sich die eigene Identität, die Sequenznummer der eingegangenen Trust-Nachricht und eine Signatur über die gesamte Nachricht. Abbildung 6.7 beschreibt den Aufbau der Nachricht.

[2] 100x 1226 Bit Grunddaten und 100x100x 2 Bit Vertrauenstabelle = 142.600 Bit (17,4 kByte)

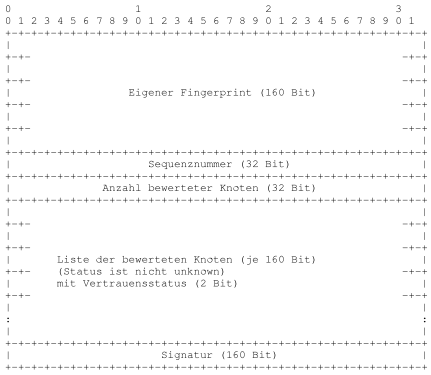

Abbildung 6.6: Trust-Nachricht

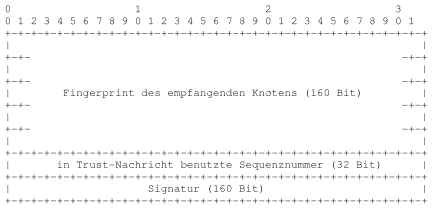

Abbildung 6.7: Trust-Acknowledge

Die Größe der Trust-Nachrichten ist nicht ganz so bedeutend wie die der TC-Nachrichten. Sei werden nur nach einer Änderung und in bestimmten großen Intervallen (einmal pro Stunde) versandt. Das periodische Versenden berücksichtigt, dass Knoten nicht immer online sind, wenn geänderte Trust-Listen verschickt werden.

Die Größe einer Trust-Nachricht ergibt sich aus dem eigenen Fingerprint (160 Bit), der 32 Bit-Sequenznummer und einem 32 Bit-Wert, welcher die Anzahl der folgenden Knoten angibt. In der Liste befinden sich nur die Knoten, zu denen eine Vertrauensmeinung existiert. Die Vertrauensbewertung *unknown* wird nicht versandt.

Befinden sich im Mesh-Netzwerk 100 Knoten, welche eine Vertrauensbewertung zu allen anderen Knoten haben, müssen 16.200 Bit Bewertungen (Fingerprint und 2 Bit-Vertrauensstufe) versandt werden. Die resultierende Trust-Nachricht hat dann eine Größe von 2 kByte. Realistischer ist jedoch die Annahme, dass die Knoten nur zu einem Teil der anderen eine Bewertung vorgenommen haben. Haben Knoten im Durchschnitt nur Bewertungen für ein Drittel aller Identitäten, reduziert sich die Größe der Trust-Nachricht auf 680 Byte. Die Übertragung in dieser Grpößenordung stellt kein Problem in Mesh-Netzwerken dar.

Wie die Knoten die langen öffentlichen Schlüssel bekommen und damit in der Lage sind, Signaturen zu überprüfen, wird im folgenen Abschnitt unter der Integritätssicherung beschrieben.

6.5 Nachrichteninhalt und Integrität

Um den *Nachrichteninhalt* end-to-end sichern zu können und dem weiteren Ziel, dass die Verschlüsselung möglichst wenig Rechenleistung beansprucht, gerecht zu werden, werden symmetrische Sitzungsschlüssel verwendet. Der absendende Knoten generiert einen zufälligen 128 Bit großen Schlüssel. Dieser Schlüssel wird mit dem öffentlichen Schlüssel des empfangenden Knotens codiert und am Anfang jeder Nachricht versandt.

Die *Integrität* von Nachrichten wird in zwei Stufen gesichert. Das Plugin Secure OLSR leistet eine hop-to-hop-Sicherung, wobei nicht ein gemeinsamer Netzwerkschlüssel benutzt wird, sondern der öffentliche Schlüssel jedes Knotens. Die Signatur schützt das gesamte OLSR-Datenpaket, welches aus mehreren einzelnen Nachrichten bestehen kann. Dies bedeutet auch, dass nach Empfang eines Datenpaketes die Nachrichten einzeln auf unterschiedlichen Pfaden weiter versandt werden können.

Die zweite Stufe ist die Integritätssicherung einzelner Nachrichten end-to-end. Hierzu werden die Nachrichten vom absendenden Knoten signiert, auf dem Weg nicht verändert und erst am Ziel auf Gültigkeit überprüft.

Befindet sich das Ziel außerhalb des Mesh-Netzwerkes und fließen die Daten über einen Gateway, so kann die Integritäts- und Inhaltssicherung nur bis zum Gateway-Knoten erreicht werden. Um die Daten auch außerhalb des Mesh-Netzwerkes zu schützen, bedarf es Protokollen, welche nicht durch die OLSR-Software geleistet werden können.

Recht einfach läßt sich die Zusammenarbeit der OLSR-Software mit der allgemein eingesetzten PGP-Software erreichen. Kennt ein Knoten den öffentlichen Schlüssel eines Ziels im Internet, kann er mit den beschriebenen Verfahren eine end-to-end-Verschlüsselung und Signierung realisieren.

6.5.1 Schlüsselverteilung

Eine offene Frage ist noch die Verteilung der öffentlichen Schlüssel (Identitäten). Diese ist dann notwendig, wenn ein neuer Knoten in das Netzwerk kommt oder erstmals eine Kommunikationsbeziehung etabliert wird. Er braucht die Schlüssel seiner Kommunikationspartner (Nachbarknoten, Gateways etc.), als auch diese seinen Schlüssel zum Überprüfen der Signaturen.

Durch TC- und Trust-Nachrichten kennen die Knoten die Fingerprints aller im Netzwerk befindlichen Geräte. Ein passender öffentlicher Schlüssel zu einem Fingerprint wird mit einer Key-Request-Nachricht angefordert. Im Key-Reply-Paket befindet sich

der selbst-signierte öffentliche Schlüssel.

Die Zuordnung eines öffentlichen Schlüssels zu einem Fingerprint ist eindeutig und unveränderlich[3]. Ein Inhalts- oder Integritätsschutz erübrigt sich.

Der/die BesitzerIn des Schlüssels kann immer auf ein Key-Request antworten. Deshalb wird an diesen Knoten zuerst die Key-Request-Nachricht versandt. Bei einer neuen Kommunikationsbeziehung zwischen zwei Knoten wird es meistens passiern, dass der angefragte Knoten noch nicht die Identität des absendenen Knotens kennt. Er antwortet auch ohne Signaturüberprüfung, wenn sein Fingerprint in der Nachricht steht. Danach leitet er ein eigenes Key-Request-Verfahren ein.

Auch andere Knoten im Netzwerk haben den öffentlichen Schlüssel gespeichert. Dies sind mindestens die direkten Nachbarknoten. Erhält der anfragende Knoten keine Antwort auf seinen Key-Request, sendet er per Broadcast eine Anfrage an alle Knoten mittels dem MPR-Verfahren aus. Vielleicht ist der angefragte Knoten gerade offline, aber andere Knoten kennen seinen öffentlichen Schlüssel.

Um die Schlüsselverteilung im gesamten Netzwerk zu optimieren, speichern alle Zwischenknoten den öffentlichen Schlüssel im Key-Reply-Paket in ihre Trust-Tabellen ab.

Nach erfogreichem Schlüsselaustausch kann vom selben Knoten für eine bestimmte Dauer kein neuer Austausch initiiert werden. Entsprechende Anfragen werden ignoriert. Dies dient der Vermeidung von DoS-Angriffen.

6.5.2 Angriffe auf Integrität und Inhalt

Für die Verschlüsselung der Inhaltsdaten werden "wohl untersuchte" Algorithmen wie AES oder Tripple-DES eingesetzt. Ein Brechen dieser Verschlüsselung kann somit nur über die Protokollimplementierung erfolgen. Die Verfahren an sich werden momentan als sicher angesehen.

Der absendende Knoten erzeugt den symmetrischen Sitzungsschlüssel, welcher bei jeder Nachricht ein anderer ist[4]. Mit dem öffentlichen Schlüssel des empfangenden Knotens codiert, wird er mit der Nachricht selbst versandt. Die gesamte Nachricht ist durch die Signatur des absendenden Knoten gesichert.

[3] Unter der Annahme, dass das asymmetrische Verfahren sicher ist und keine zwei öffentlichen Schlüssel mit demselben Fingerprint gefunden werden können.

[4] Aus Optimierungsgründen kann über einen begrenzten Zeitraum der selbe Sitzungsschlüssel benutzt werden.

EinE AngreiferIn hat somit keine Chance, unbemerkt Daten innerhalb eines Paketes zu verändern. Sich als eine andere Identität auszugeben, scheitert am Fehlen des geheimen Schlüssels zur Erzeugung der Signatur. Die Integrität als auch der Inhaltsschutz der Daten kann damit als sicher eingestuft werden.

Bleiben noch Angriffsmöglichkeiten auf die Schlüsselverteilung. Manipulationen an der Zuordnung zwischen Fingerprint und öffentlichem Schlüssel sind nicht möglich, da sofort erkennbar. Ein bösartiger Knoten kann versuchen, die Kommunikation eines Schlüsselaustausches zu blockieren. Dies kann durch Nicht-Weiterleitung oder durch Veränderung und damit Unbrauchbarmachen der Daten geschehen. Solche Daten werden vom empfangenden Knoten ignoriert. Ein Knoten, welcher die öffentlichen Schlüssel seiner KommunikationspartnerInnen nicht bekommt, kann mit ihnen keine Daten austauschen. Die Integrität kann nicht überprüft werden. Daten können nicht verschlüsselt versandt werden.

In den meisten Fällen ist davon auszugehen, dass es einem oder mehreren Knoten nicht gelingt, die Kommunikation zu allen Knoten, welche den öffentlichen Schlüssel kennen, zu blockieren (Broadcast-Key-Request). Dies tritt nur dann ein, wenn wenige oder gar keine Knoten die Identität kennen oder die Topologie des Netzwerkes so ist, dass der gesamte Datenverkehr durch die angreifenden Knoten fließen muss.

Zur Identifizierung eines blockierenden Knotens wird wieder das ACK-Verfahren angewendet. Jeder Knoten, welcher ein Key-Request erhält, sendet ein ACK oder den gesuchten Schlüssel, falls er ihn kennt. In einem Punkt unterscheidet sich dieses Verfahren vom TC- und Trust-Versand. Wie in Abbildung 6.8 zu erkennen, kann ein böswilliger Knoten das Key-Request- und die ACK-Pakete passieren lassen, aber die Nachricht mit dem öffentlichen Schlüssel blockieren. Eine Identifizierung des defekten Links ist nicht mehr möglich.

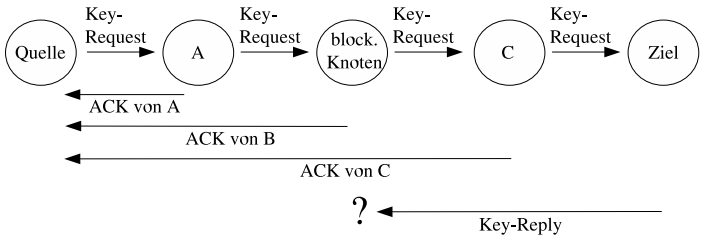

Abbildung 6.8: Blockierung des Key-Reply-Paketes

Um den defekten Link bzw. den böswilligen Knoten trotzdem aufspüren zu können,

initiiert der Knoten mit dem gesuchten öffentlichen Schlüssel ein ACK-Verfahren für das Antwortpaket. Jetzt kann der böswillige Knoten von "beiden Seiten" gefunden werden.

Um Replay-Angriffe der ACK-Nachrichten zu erkennen, befindet sich im Key-Request- als auch im Key-Reply-Paket eine Zufallszahl, welche im ACK-Paket signiert zurückgesendet wird. Im TC- und Trust-Verfahren übernahm die Sequenznummer diese Aufgabe.

6.5.3 Key-Request- und verschlüsselte Nachrichten

Der Aufbau der einzelnen Nachrichten ist in den Abbildungen 6.9 und 6.10 zu erkennen. Das Standard-Datenformat für verschlüsselte Nachrichten und einer end-to-end-Signatur befindet sich in Abbildung 6.11.

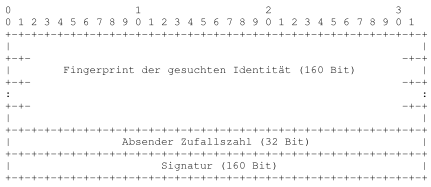

Abbildung 6.9: Key-Request-Nachricht

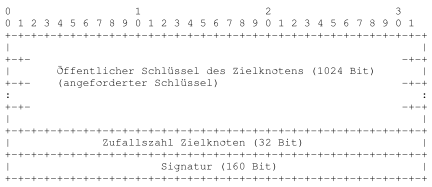

Abbildung 6.10: Key-Reply-Nachricht

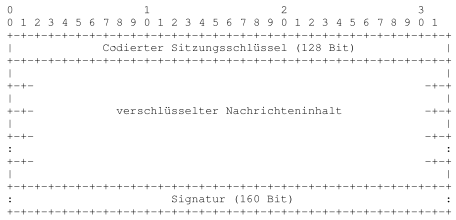

Abbildung 6.11: Verschlüsselte Nachricht

6.6 Sicheres Routing

Durch die Signaturen im Secure OLSR Plugin werden Manipulationen an TC-Nachrichten zuverlässig erkannt. Da die Daten unverschlüsselt übertragen werden müssen, erüb-

rigt sich der Schutz des Inhaltes. Eine Verschlüsselung wäre nur möglich, wenn die Nachrichten per unicast an jeden einzelnen Knoten verschickt werden könnten. Da die Knoten durch die TC-Nachrichten aber erst Kenntnis von neuen Knoten bekommen, kann dies für neue Knoten nicht gewährleistet werden.

Im Secure Plugin werden zur Sicherung der Aktualität und zum Verhindern von Replay-Angriffen Zeitstempel (timestamps) verwendet. Sie bieten den benötigten Schutz, auch wenn beim Anspruch der Optimierung eventuell Sequenznummern verwendet werden können.

Um ein Spoofing von Hello-Nachrichten und der darin enthaltenen Daten (Nachbarknoten) zu vermeiden, müssen diese ebenfalls vom absendenden Knoten signiert werden. Da die Nachrichten nicht weitergeleitet werden (Reichweite one-hop), erfüllt das Secure OLSR Plugin ebenfalls diese Aufgabe.

6.6.1 Angriffe aufs Routingprotokoll

Durch hop-to-hop- und end-to-end-Signaturen kann jede Veränderung am Inhalt zuverlässig erkannt werden. Die Identifizierung des/der AbsenderIn ist eindeutig möglich. Spoofing kann ausgeschlossen werden.

Replay-Angriffe zur Verbreitung von veralteten Topologieinformationen werden durch den Zeitstempelaustausch verhindert.

Nur Knoten, welche sich bereits im Netzwerk befinden, können TC-Nachrichten versenden und werden von den anderen Knoten beachtet. Daten von unbekannten Knoten werden einfach verworfen (Bekanntheit kleiner 10%). Ein Knoten, welcher im Netzwerk akzeptiert ist, kann auf beliebige Weise seine Hello- als auch TC-Daten verändern.

Zwei Möglichkeiten zur Manipulation der Nachbarschaftslisten eines Knotens sind vorstellbar. Dies betrifft sowohl die Hello- als auch die TC-Nachrichten. Einerseits kann ein Nachbarknoten "vergessen" werden und damit findet kein Versand von Daten über den manipulierenden Knoten zu diesem Knoten statt. Eine zweite Möglichkeit ist das "Erfinden" von neuen Nachbarknoten. Dies können vollkommen neue und damit im Netzwerk unbekannte Knoten sein, als auch existierende, zu denen dann augenscheinlich ein Funkkontakt besteht.

Das Erfinden von neuen Knoten hat keinen Effekt auf das Netzwerk, weil kein anderer Knoten eine Vertrauensbeziehung zu diesem hat und dieser damit vom Rest des Netzwerkes ignoriert wird.

Wenn ein Knoten einen anderen, im Netzwerk etablierten Knoten fälschlicherweise als einen direkt erreichbaren Nachbarn ausgibt, wird dieser andere Knoten ihn nicht in seiner Nachbarschaftsliste führen. Der Link ist somit asymmetrisch, die Knoten können sich gegenseitig nicht als MPR einsetzen und die Verbindung wird für das Routing nicht benutzt.

Anders sieht die Sache aus, wenn zwei Knoten zusammenarbeiten und sich als gegenseitige Nachbarknoten ausgeben. Dies kann von außen nicht erkannt werden. Der Effekt, den dieses Szenario hat, ist ggf. eine unnötige Umleitung von Daten über diese beiden Knoten. Dies erhöht den Datenverkehr im Netzwerk. Die Sicherheitsmechanismen werden davon nicht beeinflusst. Die beiden Knoten bekommen jedoch einen größeren Einfluss, um z.B. Datenverkehr von bestimmten Knoten zu blockieren.

Wird ein Knoten von einer/einem AngreiferIn ignoriert, obwohl eine direkte Funkverbindung zu ihm besteht, kann dies nicht erkannt werden. Der betroffene Knoten muss sich nun einen anderen Nachbarschaftsknoten suchen, über den er seine Daten weiterleiten kann. Umgekehrt wird dieser Knoten aber auch keine Daten vom Angreifer / von der Angreiferin weiterleiten.

Gibt es für einen Knoten keinen anderen als den einen, welcher ihn ignoriert, so hat er keine Chance, am Mesh-Netzwerk teilzunehmen. Dieser Effekt kann jedoch auch durch eine Funkstörung durch andere elektronische Geräte verursacht werden. Diese Person wird sich also mit den technischen AdministratorInnen des Mesh-Netzwerkes auf Fehlersuche begeben und das Verhalten des Angreifers / der Angreiferin entdeckt werden.

Ähnlich dem Schlüsselaustausch-Protokoll kann ein böswilliger Knoten die Weiterleitung von TC-Nachrichten blockieren. Die sendenden Knoten müssen eine Möglichkeit bekommen, zu bemerken, wenn ihre Daten nicht zu allen Knoten gelangen. Dies wird durch ein ACK-Paket erreicht. Das Verfahren entspricht dem, welches für die Trust-Nachrichten angewendet wird.

Knoten warten auf ACK aller in der Trust-Liste vermerkten Knoten. Erhalten sie keine Antwort, warten sie auf Nachrichten (TC oder Trust) vom anderen Knoten, um sicher zu gehen, dass dieser online ist. Erhalten sie ein solches Paket, wird per unicast-Versand das TC-Paket erneut verschickt. Gibt es wieder kein ACK, wird von einer Blockierung ausgegangen. Die Nachbarknoten des defekten Links werden zur Überwachung aufgefordert und das TC-Paket erneut versandt. Für den Ausschluss eines erkannten Knotens gilt das im Abschnitt Trust-Nachrichten Beschriebene.

6.7 Kommunikationsbeziehung und Anonymität

Wie bereits im Kapitel 4 beschrieben, gibt es nur zwei grundlegende Verfahren zur Verschleierung der Kommunikationsbeziehung: Mixe und Dummy-Traffic. Beide sind in Mesh-Netzwerken nur bedingt einsetzbar.

Dummy-Traffic erzeugt eine enorme Netzlast, die durch die Bandbreitenbeschränkung in Funknetzwerken die eigentliche Nutzung unmöglich machen würde. Jeder Knoten müsste permanent Daten zu allen anderen Knoten senden.

Der Schutz, den *Mixe* bieten, hängt maßgeblich von der Anzahl der NutzerInnen ab, welche in einem kleinen Zeitintervall die Daten durch einen Mix leiten. In Funknetzwerken ist die Anzahl der NutzerInnen eher klein (20-100). Durch die Eigenschaft der Funkübertragung ist entweder Senden oder Empfangen möglich. Ein Mix müßte über einen größeren Zeitraum eingehende Nachrichten sammeln und diese gemixt weiterversenden. Dies erhöht die Latenzzeit bedeutend mehr als Mix-Verfahren in kabelbasierten Netzen.

Dritter Punkt, der gegen einen Einsatz von Mixen spricht, ist die Strecke, die ein Datenpaket von dem/der SenderIn zur/zum EmpfängerIn zurücklegt. Im ungünstigen Fall befinden sich beide Knoten in Funkreichweite und das Datenpaket "wandert" mehrmals durch das gesamte Mesh-Netzwerk, da sich die Mixe an deren Enden befinden. Die Bandbreite für die eigentlichen Daten wird enorm verkleinert.

Sehr zu empfehlen ist der Ansatz von Mixen im nicht funkbasierten Netzwerk. Stellen die Gateways Mixe für externe Netze dar, kann bereits im OLSR-Protokoll definiert werden, dass externer Verkehr nur als Mix-Paket versandt wird.

Möchte ich im Mesh-Netzwerk anonym sein, so kann dies nur mit wechselnden bzw. nicht korrelierbaren Identitäten passieren. Die Identität (öffentlicher Schlüssel) bildet jedoch die Grundlage für den Topologieaufbau und die bisher genannten Schutzmechanismen. Anonymität kann somit innerhalb des Mesh-Netzwerkes nicht erreicht werden.

Einen marginalen Schutz bietet das periodische Neuerstellen der öffentlichen Schlüssel. Geschieht dies in kleinen Gruppen, welche sich gegenseitig die benötigten 10% Vertrauen verschaffen, könnte somit tagesweise die Identität gewechselt werden. Die Anonymität gilt aber nicht für diese Gruppe.

Da die Knoten fast immer stationär sind, ist die Korrelation von Daten mit einer bestimmten Sendestärke an einem bestimmten topologischen Punkt immer möglich. Knoten und Personen können so identifiziert werden, ohne dass die Identität beachtet werden muss.

Das Fazit ist, dass die Anonymität im Mesh-Netzwerk nicht geschützt werden kann und dieses Schutzinteresse damit nicht erfüllt ist.

6.8 Egoistische oder böswillige Knoten

Das Verfahren *eindeutiges iteratives Probing*, welches in [50] beschrieben wurde, leistet eine zuverlässige Erkennung von egoistischen und böswilligen Knoten. Es beruht jedoch auf dem reaktiven DSR-Protokoll, das mit Source-Routen arbeitet. Die Route, welche ein Datenpaket nimmt, ist vorgegeben und ein gezieltes Testen der einzelnen Knoten auf einer Route ist möglich.

OLSR ist im Gegensatz dazu ein proaktives Protokoll, bei dem der absendende Knoten das Datenpaket an denjenigen Knoten leitet, der aus seiner Sicht auf der kürzesten Verbindung zum Ziel liegt. Jeder Knoten auf dem Weg entscheidet selbst, welches der nächste hop ist.

Die bisherigen Protokolle leisten, dass ein fehlerhafter Link entdeckt wird. Nicht entschieden werden konnte, welcher der beiden Knoten dieses Links sich egoistisch oder böswillig verhält.

Im folgenden Verfahren wird versucht, den schuldigen Knoten zu finden. Der Knoten, welcher den fehlerhaften Link bemerkt hat, sendet an alle Nachbarknoten der beiden verdächtigten Knoten die Aufforderung, diese durch *overhearing* zu überwachen. Die Nachbarknoten sind aus den TC-Nachrichten bekannt. Diese Aufforderung wird verschlüsselt.

Die Nachbarknoten kontrollieren, ob der verdächtigte Knoten Daten weiterleitet. Dies geschieht in zufälligen Intervallen über einen längeren Zeitraum (mehrere Stunden). Wird festgestellt, dass der Knoten viele Daten fremder Knoten nicht weiter leitet, eigene Daten jedoch versendet, wird der Vertrauensstatus dieses Knotens auf *untrusted* gesetzt. Die Kontrolle muss über einen längeren Zeitraum erfolgen, um kurzzeitige Frequenzstörungen oder unbeabsichtigte Überlastungen eines Knotens nicht falsch zu bewerten.

Zu beachten ist, dass ein Knoten keine Daten zu einem Knoten leitet, dessen Vertrauensverhälnis *untrusted* ist. Die Vertrauensverhältnisse sind allen Knoten im Netzwerk bekannt und können somit berücksichtigt werden.

Durch Abhören der Daten in seiner Umgebung kann ein egoistischer Knoten erkennen, dass alle seine Nachbarknoten ein Datenpaket von einem anderen Knoten bekommen haben. Er muss nun entscheiden, ob es sich um normale Broadcast-Nachrichten handelt oder es eine Aufforderung ist, ihn zu überwachen. Entscheidet er sich für Letzteres, wird er ein protokollkonformes Verhalten zeigen. Da die Überwachung mehrere Stunden dauern kann, muss er die ganze Zeit über kooperatives Verhalten zeigen.

Mit dem Verfahren wird somit keine vollständige Kooperation erreicht, jedoch ein Knoten nach seiner Entdeckung für einen längeren Zeitraum zu kooperativem Verhalten gezwungen.

6.8.1 Manipulation der Erkennung

Falsche Verdächtigungen führen zu einer höheren Ressourcennutzung der Knoten, welche die Überwachung durchführen. Führt ein Knoten eine Überwachung durch, ist es aber egal, ob er einen oder alle seine Nachbarknoten überwacht. Denunziation führt damit zu keinen Einschränkungen der eigentlichen Protokollfunktion.

Replay-Angriffe sind möglich, führen aber nur zu der eben beschriebenen falschen Verdächtigung.

Ein Knoten kann nur soviel unkooperatives Verhalten zeigen, dass die ihn überwachenden Knoten nicht entscheiden können, ob das Verhalten bewußt oder nicht selbst verschuldet auftritt. Diesen Grenzbereich wird es leider geben, da Funkprobleme immer auftreten können und falsche Verdächtigungen schwerer wiegen, als Knoten die ein wenig unkooperativ sind und nicht entdeckt werden.

EinE AngreiferIn kann noch zwei weitere Verfahren anwenden, um zu vermeiden, Daten von anderen Knoten weiterleiten "zu müssen".

Ein Knoten kann, obwohl er in der Reichweite mehrerer Nachbarknoten liegt, einen Knoten auswählen und nur diesen in seinen Hello- als auch TC-Nachrichten angeben. Augenscheinlich verfügt er nur über einen symmetrischen Link, über den er Daten versenden kann bzw. über den Daten an ihn geschickt werden können.

Eine andere Methode ist das Setzen der Vertrauensverhältnisse auf *untrusted* aller Knoten, ausgenommen den einen ausgewählten. Dies führt ebenfalls zum beschriebenen Effekt, da Links zu nicht vertrauenswürdigen Knoten als nicht existent angesehen werden.

Beide Varianten werden von den Nachbarknoten bemerkt. Sie haben jedoch keine Möglichkeit dieses egoistische Verhalten nachzuweisen, da - zwar selten - die Situation unverschuldet entstehen kann.

Einziger Ausweg bleibt die längere Beobachtung der Situation und die Klärung auf dem Plenum. Personen können eine Begründung einfordern, warum sie als nicht vertrauenswürdig eingestuft werden. In ihren Trust-Listen kann jede Person erkennen, wie andere sie einstufen.

Egoistisches Verhalten kann in allen Fällen erkannt werden und damit theoretisch zum Ausschluss des Knotens führen. Diese Gefahr reicht in meinen Augen aus, um Knoten zum kooperativen Verhalten als bessere Strategie für eine längere Teilnahme am Mesh-Netzwerk zu bewegen.

6.8.2 Scan-Nachrichten

In einer Scan-Nachricht wird einem Knoten mitgeteilt, einen anderen Knoten in seiner direkten Umgebung zu überwachen. Die Nachricht enthält damit nur die Identität des Absenders und des zu überwachenden Knotens. Die Integrität wird durch die Signatur und die Vertraulichkeit durch Verschlüsselung mit dem gemeinsamen DH-Schlüssel gesichert. In Abbildung 6.12 ist der Aufbau zu sehen.

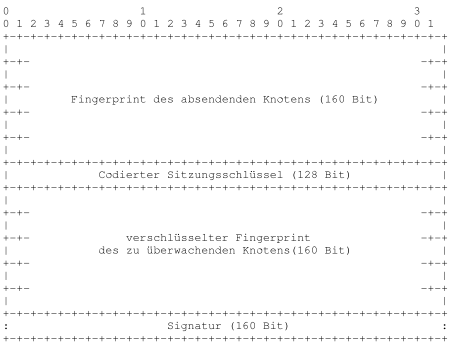

Abbildung 6.12: Aufforderung zur Überwachung

6.9 Fairness

Wenn die Bandbreite in einem Mesh-Netzwerk stark ausgelastet ist, kann es zu unfairem Verhalten kommen. Grund ist die Weiterleitung nach dem Prinzip *first-come first-serve*. In diesem Verfahren wird nicht berücksichtigt, welcher Knoten wie viele Daten versenden will.

Durch *Load-Balancing* kann eine gerechte Verteilung versucht werden. Knoten melden ihre Sendewünsche an und der durchleitende Knoten vergibt eine Art *Senderecht*. Die Senderechte werden nach dem Prinzip vergeben, dass die maximale Bandbreite für alle sendewilligen Knoten gleichmäßig aufgeteilt wird.

In der Literatur gibt es verschiedene Verfahren, welche Quality-of-Service-Mechanismen in Mesh-Netzwerken implementieren wollen (siehe Kapitel 2.3.4). Wenn nicht nur wenige Knoten, sonder alle solche QoS-Garantien bekommen, ist dies ein weiteres Verfahren des Load-Balancings.

In dieser Arbeit wird der Punkt Fairness nicht weiter betrachtet, da die Entwicklung eines solchen Systems für dynamische Mesh-Netzwerke zu umfangreich wäre und erst nach einer Analyse des Verkehrsaufkommen in Community-Netzwerken überhaupt eine Entscheidung getroffen werden kann, inwieweit es zu unfairem Verhalten kommt.

6.10 Denial-of-Service-Angriffe

Neben dem *jamming*, welches die Funkübertragung auf physikalischer Ebene stört, kann es auch zu DoS-Angriffen kommen, welche die Spezifika der Protokollimplementierung nutzen. Folgende Teile des Protokolls können davon betroffen sein:

- *Timestamp-Austauschverfahren im Secure-OLSR-Plugin* - ständige Initialisierung des Zeitstempelaustauschs
- *Trust-Nachrichten* - ständiges Fluten veränderter Trust-Nachrichten im Netzwerk und Zwang der Knoten zu einer Aktualisierung ihrer Trust-Listen
- *Key-Request-Nachrichten* - ständige Anfragen nach einem öffentlichen Schlüssel durch einen oder mehrere Knoten (Identitäten)
- *Scan-Nachrichten* - Verteilen der Überwachungsaufforderung an alle Knoten
- *Verschlüsselte Nachrichten* - Zwang eines oder mehrerer Knoten zum ständigen Entschlüsseln und Überprüfen von Signaturen

Alle Angriffe dieser Art erzeugen eine erhöhte Netzlast. EinE AngreiferIn kann jedoch nur die Bandbreite für die Übertragung nutzen, welche ihm/ihr auch als reguläreR NutzerIn zur Verfügung stehen würde. Der Einsatz von Fairnessprotokollen könnte die maximal nutzbare Bandbreite beschränken, dass eigentliche OLSR-Protokoll kann dies nicht. Ohne Analyse der versandten Daten ist es nicht möglich zu entscheiden, ob große Datenmengen einer Videokommunikation oder Daten, welche von einem böswilligen Verhalten resultieren, über das Mesh-Netzwerk verschickt werden. Ziel ist somit die Erkennung der Angriffe und nicht das Erkennen von hohem Netzwerkverkehr.

Ziel der DoS-Angriffe ist es, einen oder mehrere Knoten so mit Protokollfunktionen zu belasten, dass sie nicht mehr in der Lage sind, eigene Datenkommunikation durchzuführen. Da nur Daten von gültigen Identitäten im Netzwerk weitergeleitet werden, haben nur AngreiferInnen, welche sich als normaleR NutzerIn im Netzwerk befinden, eine Möglichkeit diese durchzuführen.

Im Timestamp- und Key-Request-Verfahren wurden zeitliche Schranken eingebaut. Nach einem erfolgten Austausch oder mehrerer nicht erfolgreicher Versuche, kann derselbe Knoten für eine gewisse Dauer keinen neuen Austausch initiieren.

Bei Erhalt von Trust-Nachrichten muss ein Knoten seine Trust-Liste aktualisieren, wenn die verwendete Sequenznummer höher als die gespeicherte ist. Falls nicht, wird das Paket sofort verworfen. Der Aufwand für das Speichern ist gering, da nur eine Zeile der Tabelle überschrieben werden muss und keine weiteren kryptographischen Rechnungen notwendig sind. Der Angriff führt somit zu keinem Erfolg.

Theoretisch ist es möglich, allen Knoten im Netzwerk eine Aufforderung zum Überwachen aller Nachbarknoten zu schicken. Der Aufwand eines Knotens ist gleich, egal ob ein oder mehrere NachbarInnen überwacht werden. Die Überwachung erfolgt in einem längeren Intervall und der Knoten entscheidet selbst, ob er die Überwachung zum Senden eines eigenen Datenpaketes unterbricht. Die Wirkung des Angriffs über Scan-Nachrichten ist ebenfalls gering und vernachlässigbar.

Die Signaturen ankommender Nachrichten zu überprüfen und ggf. die Inhalte zu entschlüsseln, gehört zu den Standardaufgaben jedes Knotens. Die aufwändigeren Rechenoperationen asymmetrischer Verschlüsselung wurden durch Einsatz von Hashwerten, Fingerprints etc. versucht zu minimieren. Die Inhaltsverschlüsselung erfolgt mit einem symmetrischen Sitzungsschlüssel. Dies bedeutet, ein Angriff kann einen Knoten zwingen, ständig Rechenoperationen durchzuführen. Dies muss der Knoten aber auch im Normalfall leisten. Der Angriff verursacht somit nur eine ständige Belastung eines Knotens, aber keinen Ausfall oder Blockierung eigener Sendewünsche.

Das Fazit der Analyse von Protokoll-spezifischen Angriffen ist somit, dass diese eine Wirkung auf die Auslastung der Knoten haben, jedoch nicht effektiv einen Knoten beeinträchtigen können.

Alle Angriffe können einfach erkannt und einem (oder mehreren) Knoten zugeordnet werden, da diese unter einer gültigen Identität auftreten müssen. Auswertungs-Logs können zum Beispiel für das Plenum ein Grund sein, einen Knoten vom Netzwerk auszuschließen.

6.11 Zusammenfassung

Folgende Protokolle und Nachrichtentypen werden verwendet:

- *Hello-Nachrichten* zum Auffinden von Nachbarknoten mit Signatur des Secure-OLSR-Plugins
- *TC-Nachrichten* und ACK zur sicheren Verteilung von Topologieinformationen
- *Trust-Nachrichten* und ACK für sichere Verteilung der Trust-Listen
- *Key-Request*, Reply and ACK zum Versand von öffentlichen Schlüsseln
- *Scan-Nachrichten* als Aufforderung zur Kontrolle einzelner Knoten
- *Verschlüsselte und signierte Nachrichten* für den gesamten Datenverkehr

Durch die benannten Protokolle und Datenstrukturen können die Ziele 1-4 und 6 aus Kapitel 6.1 erreicht werden.

Ziel 5 - der Schutz der Kommunikationsbeziehung und Anonymität, bleibt eine offene Frage. Ausführlich im Unterpunkt 6.7 erklärt, ist nur das Verfahren der Dummy-Traffics zum Schutz einer Kommunikationsverbindung anwendbar. In welcher Form dies, bei der beschränkten Bandbreite von Funknetzwerken und einer größeren Anzahl von Knoten, einsetzbar ist, stellt eine offene Grundsatzfrage dar, welche in dieser Arbeit leider nicht beantwortet werden kann.

Der Wunsch nach Anonymität kann nur durch wechselnde Identitäten erreicht werden. Die Aufnahme von neuen Identitäten wird nicht durch das Protokoll, sondern durch nicht-technische Vereinbarungen geregelt. Anonymität kann somit nicht durch das Protokoll geleistet werden. Durch Lokalisierungs- und Zuordnungsmöglichkeiten von Funkwellen würde ich diesen Punkt generell als äußerst schwer erreichbar einschätzen.

Nicht näher beachtet wurde das Ziel der fairen Verteilung der Bandbreite, da für eine erfolgreiche Entwicklung umfangreiche Labortest notwendig gewesen wären, die innerhalb dieser Arbeit nicht geleistet werden konnten. Im entsprechenden Kapitel wurden jedoch theoretische Möglichkeiten aufgezeigt.

Die faire Verteilung der Bandbreite ist ein Unterziel des Sicherheitspunktes "Verfügbarkeit der Kommunikation". Hier spielen noch viele weitere Punkte, wie DoS-Angriffe, eine wichtige Rolle, welche im Rahmen dieser Arbeit Beachtung fanden.

Trotz der benannten Einschränkungen wurde in dieser Arbeit erstmals eine Sicherheitsarchitektur für Community-Mesh-Netzwerke entworfen, welche zwar nicht umfassend, jedoch alle Interessen der Beteiligten berücksichtigt und abwägt.

In der Entwicklung der Architektur wurde der Fokus auf die Sicherheitsmechanismen gelegt. Performancesteigerungen und Optimierungsmöglichkeiten sind an vielen Stellen möglich und notwendig. Das Protokoll des Schwellwertschemas wurde so ausgelegt, dass es dynamisch an die Bedürfnisse und Vereinbarungen in einem Gemeinschaftsnetzwerkes angepasst werden kann.

7 Zusammenfassung und Ausblick

Ziel dieser Diplomarbeit war die Entwicklung einer geschützten Wireless-Mesh-Architektur. In einer kurzen Einleitung wurden die verwendeten Begriffe erläutert und die Grundstruktur eines Mesh-Netzwerkes beschrieben.

Um eine sichere Mesh-Architektur möglichst konkret zu beschreiben, wurden danach verschiedene Einsatzszenarien skizziert und die von den Beteiligten gewünschten Sicherheitsbedürfnisse ermittelt. Die Hauptpunkte dieser Sicherheitsanforderungen waren eine eindeutige Identifizierung der Mesh-Knoten, ständige Verfügbarkeit der Kommunikation, Integrität der Daten, teilweise die Geheimhaltung des Inhaltes der zu übertragenden Daten sowie die der gesamten Kommunikationsbeziehung.

Bemerkenswert waren die sehr unterschiedliche Komplexität der Sicherheitswünsche und der Mesh-Struktur (Anzahl Knoten, Entfernung, Mobilität usw.). Ein erstes Ergebnis ist somit die Festellung, dass eine geschützte Wireless-Mesh-Architektur auf ein konkretes Szenario abgestimmt werden sollte. Eine allumfassende Architektur für alle Einsatzfälle ist nicht notwendig, erhöht die Netzwerklast bzw. vermindert die Performance und führt zu Widersprüchen, welche nur sehr aufwendig aufzulösen sind. Ein Beispiel für den letzten Punkt ist der Sicherheitswunsch nach eindeutiger Identifizierung aller Geräte und dem augenscheinlich entgegengesetzten Wunsch nach Anonymität einzelner Knoten.

Im Folgenden wurden kryptographische Algorithmen und Protokolle erläutert, mit deren Hilfe die geforderten Schutzbedürfnisse erfüllt werden können. Neben Kryptosystemen mit symmetrischen Schlüsseln und asymmetrischen Schlüsselpaaren wurde auf Hashfunktionen (Fingerprint) sowie auf Protokolle zur Authentifizierung und zum Schlüsselaustausch eingegangen. Da das Routing in einem Mesh-Netzwerk im Gegensatz zu traditionellen Netzwerken in besonderem Maße gefährdet ist, wurden ebenfalls einige speziell auf Ad-hoc-Netze und damit auch auf Mesh-Netzwerke zugeschnittene Routingprotokolle vorgestellt. Im Abschluss des Kapitels wurde noch einmal auf die Notwendigkeit weiterer Sicherheitsmechanismen und die damit verbundenen Probleme hingewiesen.

Im nächsten Schritt wurden bestehende Arbeiten und Implementierungen zur Verbesserung der Sicherheit in Mesh-Netzwerken besprochen und bewertet. Keine der vorgestellten Arbeiten konnte die Sicherheitsanforderungen restlos erfüllen. Auf der einen

Seite wird von unrealistischen Annahmen ausgegangen - so funktioniert das Routing auf Grundlage vorher ausgetauschter Schlüssel. Wie diese jedoch ohne eine Routingfunktion ausgetauscht werden, bleibt offen. Auf der anderen Seite werden sehr oft nur einzelne Aspekte einer sicheren Architektur betrachtet.

Durch die am Anfang analysierten Sicherheitsbedürfnisse zeigte sich jedoch auch, dass für einzelne Einsatzszenarien bereits einige Arbeiten existieren, welche die Bedürfnisse fast erfüllen. Zum Beispiel hat in einer Gebäudevernetzung der Betreiber oder die Betreiberin physikalischen Zugriff auf alle Geräte und kann alle Geräte mit einer eindeutigen Identifizierung und einem gemeinsamen Netzwerkschlüssel programmieren. Da sich alle Geräte - sofern funktionsfähig - kooperativ verhalten und der Dateninhalt nicht verschlüsselt werden muss, wird mit mehren Protokollen eine geschützte Architektur erreicht. Offen sind Anpassungen für ein energiesparsames Protokoll, welche für die Verfügbarkeit des Netzwerkes notwendig sind. In anderen Einsatzszenarien zeigen sich jedoch noch viele offene Sicherheitsfragen, welche keines der genannten Protokolle erfüllen kann.

Im folgenden Schritt wurde eine sichere Mesh-Architektur für Community-Netzwerke entwickelt. Die Schutzinteressen in diesem Szenario konnten durch keines der vorher analysierten Protokolle erfüllt werden. Aufbauend auf einer OLSR-Implementierung, welche aufgrund ihrer offenen Plugin-Funktionalität und dem proaktivem Ansatz ausgewählt wurde, wurde eine sichere Mesh-Architektur beschrieben.

Grundlage für den Zugang und Ausschluss von Mesh-Geräten ist ein Schwellwertschema, welches auf die Vertrauensverhältnisse der NutzerInnen zueinander aufbaut. Ausführlich wurden die einzelnen Schutzmethoden beschrieben und Angriffsmöglichkeiten untersucht. Es war jedoch nicht möglich, alle Schutzinteressen in vollem Umfang umzusetzen. Die Anonymität der NutzerInnen konnte nur teilweise erreicht werden. Der Schutz der Kommunikationsbeziehung und das faire Verteilen der Bandbreit sind zwei Forschungspunkte in Mesh-Netzwerken, für die es noch keine zufriedenstellenden Lösungen gibt. Innerhalb dieser Arbeit konnten nur Ideen entwickelt und Vorschläge gegeben werden.

Eine Implementierung exemplarischer Maßnahmen schien nicht sinnvoll, da erst durch das Gesamtkonzept ein wirklicher Schutz ereicht wird und die Schutzmethoden aufeinander aufbauen. Eine Gesamtimplementierung war in dieser Diplomarbeit nicht leistbar.

Mesh-Netzwerke sind ein spannendes Forschungsthema, welches momentan durch viele kommerzielle und Community-Versuche immer mehr an Bedeutung gewinnt. Etliche technische Fragestellungen bleiben noch offen. Sicherheitsmechanismen müssen im Blick auf Skalierbarkeit und Performance optimiert werden. Noch gibt es Bedenken

der NetzanbieterInnen bezüglich der korrekten Abrechnung genutzter Dienstleistungen, als auch von AnwenderInnen, welche in einer immer stärker digitalisierten Welt um ihre Privatsphäre fürchten. Die Einführung von Mesh-Produkten mit einer Sicherheitsarchitektur, welche auf die Bedürfnisse der Beteiligten zugeschnitten ist, kann hier das benötigte Vertrauen schaffen.

Die Integration des Mesh-Ansatzes in Anwendungsszenarien wie dem öffentlichen Internetzugang könnte dem Konzept zu einem Durchbruch verhelfen. Hier zeigt sich durchaus ein gesellschaftlicher Ansatz - die einfache und kostengünstige Bereitstellung von Internetzugängen für die gesamte Bevölkerung.

Ich hoffe, mit dieser Arbeit einen Beitrag zur Etablierung von Mesh-Netzwerken geleistet zu haben. Die Fokussierung auf die öffentliche Unik-OLSR-Software ermöglicht eine einfache Umsetzung des Schutzkonzeptes. Kontakt zu den BetreuerInnen der Software wurde bereits aufgenommen. Finden sich weitere UnterstützerInnen kann in einem kurzem Zeitrahmen diese Arbeit implementiert und in weiteren Laborversuchen untersucht werden.

Literaturverzeichnis

[1] Research-Unternehmen IDC. 25,000 Hotspots in Europa. *http://www.newsbyte.ch/suche2.cfm?id=65841*, 25 January 2005.

[2] Marktforscher VDC. Maschendraht aus null und eins. *http://www.ecin.de/news/2005/07/25/08518/*, 25 July 2005.

[3] Thomas Thaler. berlin backbone - ein drahtloses freies netz. *http://www.freifunk.net/magazin/praxisberichte_projekte/bbb_thomas_thaler*, 19 October 2003.

[4] diverse AutorInnen. Mobiles Ad-hoc-Netzwerk. *http://de.wikipedia.org/wiki/Mesh_Network*, 17 June 2005.

[5] diverse AutorInnen. Wireless mesh network. *http://de.wikipedia.org/wiki/Wireless_mesh_network*, 4 July 2005.

[6] Pradip Lamsal. Understanding Trust and Security. *http://wiki.uni.lu/secan-lab/Lamsal2001.html*, 20 October 2001.

[7] Yih-Chun Hu. Self-Organizing Neighborhood Wireless Mesh Networks. *http://research.microsoft.com/mesh/*, 2004.

[8] Yu-Kwong Kwok Liza Lai-Yee Shek. An integrated approach to scatternet traffic management in Bluetooth ad hoc networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking, VOL 45*, 45:99 – 118, 2004.

[9] diverse AutorInnen. IEEE 802.11s. *http://en.wikipedia.org/wiki/IEEE_802.11s*, 1 February 2006.

[10] ZigBee Alliance. ZigBee Alliance. *http://www.zigbee.org/*, 2005.

[11] TG 5. IEEE 802.15 WPAN Task Group 5. *http://www.ieee802.org/15/pub/TG5.html*, 9 August 2005.

[12] diverse AutorInnen. Software defined radios. *http://en.wikipedia.org/wiki/Software-defined_radio*, 26 October 2005.

[13] Brian Zill Richard Draves, Jitendra Padhye. Comparison of routing metrics for static multi-hop wireless networks. *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 30 September 2004.

[14] diverse AutorInnen. Ad hoc protocol list. *http://en.wikipedia.org/wiki/Ad_hoc_protocol_list*, 13 July 2005.

[15] Jennifer J.-N. Liu Imrich Chlamtac, Marco Conti. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks 1/2003*, 1 January 2003.

[16] Phd Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall, December 2001.

[17] diverse AutorInnen. OLSR daemon. *http://www.olsr.org/*, 30 October 2005.

[18] Kevin H. Grace. Mobile Mesh: Providing Solutions For Mobile Adhoc Networking. *http://www.mitre.org/work/tech_transfer/mobilemesh/*, 4 April 2005.

[19] Samir Das Charles Perkins, Elizabeth Belding-Royer. Ad hoc On Demand Distance Vector. *http://moment.cs.ucsb.edu/AODV/aodv.html*, 17 February 2003.

[20] diverse AutorInnen. Dynamic Source Routing. *http://de.wikipedia.org/wiki/Dynamic_Source_Routing*, 27 October 2005.

[21] Alex Song. Piconet II - A Wireless Ad Hoc Network for Mobile Handheld Devices. *http://piconet.sourceforge.net/thesis/index.html*, 10 November 2001.

[22] Prince Samar Zygmunt J. Haas, Marc R. Pearlman. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. *http://tools.ietf.org/tools/rfcmarkup/rfcmarkup.cgi?draft=draft-ietf-manet-zone-zrp-04.txt*, January 2003.

[23] Tsu-Wei Chen Guangyu Pei, Mario Gerla. Fisheye State Routing in Mobile Ad Hoc Networks. *ICDCS Workshop on Wireless Networks and Mobile Computing*, September 1999.

[24] Robin Kravets Rong Zheng. On-demand Power Management for Ad Hoc Networks. *INFOCOM 2003*, 2003.

[25] Ten-Yueng Hsieh Yu-Chee Tseng, Chih-Shun Hsu. Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks. *IEEE Annual Conference (INFOCOM 2002)*, 2002.

[26] Khaldoun Al Agha Hakim Badis, Thomas Claveirole. QoS with the OLSR protocol. *http://qolsr.lri.fr/*, 2005.

[27] Aleksandr Huhtonen. Comparing AODV and OLSR Routing Protocols. *http://www.tml.tkk.fi/Studies/T-110.551/2004/papers/Huhtonen.pdf*, 2004.

[28] Weilin Wang Ian F. Akyildiz, Xudong Wang. Wireless mesh networks: a survey. *Computer Networks 47 (2005) 445-487*, 1 January 2005.

[29] Corinna Aichele. Selbst organisiert - Mesh Netzwerke. *LINUX Magazin - Sonderheft Network Edition*, 1 September 2004.

[30] Wiana. Wireless Internet Assigned Numbers Authority. *http://www.wiana.org/*, 2005.

[31] Matthew Broersma. Die Zukunft der Funknetze: Mesh soll die Löcher stopfen. *http://www.zdnet.de/mobile/wireless/0,39023428,39121738,00.htm*, 21 April 2004.

[32] Wirtschaftswoche. Vernetzte Feuerwehrleute - Fraunhofer-Institut stattet Rettungskräfte mit Sensoren und Computern aus. *http://www.zdnet.de/news/hardware/0,39023109,39130272,00.htm*, 14 February 2005.

[33] Adfero Ltd. Portsmouth buses use battlefield technology. *http://www.serco.com/media/industrynews/ItemPage.asp?ItemID=3712660*, 26 October 2004.

[34] Donald Wilkins. Mesh Networking Smooths Traffic Flow. *http://www.wsdmag.com/Articles/ArticleID/9665/9665.html*, 1 January 2005.

[35] Markus Honsig. Plug and Drive. *Technology Review Nr.9 (2005)*, 9 September 2005.

[36] Übersetzung: Ben Schwan Eric Hellweg. Hightech-Hilfe für New Orleans. *Technology Review 9/2005*, 5 September 2005.

[37] Tropos Networks. New Orleans selects Tropos Networks to provide metro-scale WiFi network for surveillance camera project. *http://www.tropos.com/company/2004_04_20.html*, 20 April 2004.

[38] Greg Lehane. Ricochet. *http://ntrg.cs.tcd.ie/undergrad/4ba2.01/group1/ricochet.htm*, 2000.

[39] WLAN_weekly Online-Magazin. Peking wird zur Olympiade drahtlos. *http://mobileaccess.de/wlan/?go=newsitem&msgid=1225&sid=*, 14 November 2005.

[40] 4G-Systems. 4G ACCESS CUBE. *http://www.4g-systems.de/*, 2005.

[41] Sanjit Biswas Dan Aguayo, John Bicket. MIT Roofnet. *http://pdos.csail.mit.edu/roofnet/*, 14 September 2005.

[42] diverse AutorInnen. Freie Funknetze (WLAN) im deutschsprachigen Raum. *http://www.freifunk.net/*, 1 January 2005.

[43] LocustWorld Ltd. LocustWorld: The Information Revolution. *http://www.locustworld.com/*, 2005.

[44] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, November 1976.

[45] H. Luo and P. Zerfos and J. Kong and S. Lu and L. Zhang. Self-securing ad hoc wireless networks. *http://citeseer.ist.psu.edu/luo02selfsecuring.html*, 2002.

[46] S. Capkun and L. Buttyan and J. Hubaux. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *citeseer.ist.psu.edu/capkun02selforganized.html*, 1 January 2002.

[47] A. Pfitzmann. Sicherheit in Rechnernetzen: Mehrseitige Sicherheit in verteilten und durch verteilte Systeme. *Vorlesungsskript TU Dresden*, 15 October 2000.

[48] JAP Team an der TU-Dresden. Anonymity and Privacy. *http://anon.inf.tu-dresden.de/*, 2005.

[49] Roger Dingledine und Nick Mathewson. Tor: Ein anonymes Kommunikationssystem für das Internet. *http://tor.eff.org/*, 2005.

[50] Frank Kargl. Sicherheit in Mobilen Ad hoc Netzwerken. *Dissertation Universität Ulm*, 2003.

[51] Edward W. Knightly Violeta Gambiroza, Bahareh Sadeghi. End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks. *http://100x100network.org/papers/gambiroza-mobicom2004.pdf*, September 2004.

[52] Jean-Pierre Hubaux Naouel Ben Salem. A Fair Scheduling for Wireless Mesh Networks. *http://www.cs.ucdavis.edu/ prasant/WIMESH/p10.pdf*, September 2005.

[53] Bruce Schneier. *Angewandte Kryptographie*. Addison-Wesley, 1996.

[54] Manel Guerrero Zapata. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. *http://www.ietf.org/internet-drafts/draft-guerrero-manet-saodv-04.txt*, July 2002.

[55] David B. Johnson Yih-Chun Hu, Adrian Perrig. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. *MobiCom '02*, September 2002.

[56] Brian Neil Levine Kimaya Sanzgiri, Bridget Dahill. A Secure Routing Protocol for Ad Hoc Networks. *http://citeseer.ist.psu.edu/678897.html*, 2002.

[57] Panagiotis Papadimitratos and Zygmunt J. Secure Routing for Mobile Ad hoc Networks. *http://wnl.ece.cornell.edu/Publications/cnds02.pdf*, January 2002.

[58] J. Hubaux L. Buttyán. Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks. *citeseer.ist.psu.edu/article/buttyan01nuglets.html*, 2001.

[59] Jacquet Clausen. Optimized Link State Routing Protocol (OLSR). *http://ietf.org/rfc/rfc3626.txt*, October 2003.

[60] Initiative freifunk.net. Präsentation: Freie funkbasierte Netzwerke. *http://www.freifunk.net/downloads/050406_ffn_present_v10_jpn.sxi*, 10 April 2005.

[61] Bob Dickerson. 802.11 Local Area Wireless Networks. *http://homepages.feis.herts.ac.uk/ mcom0055//osn-04-05-node8.html*, 16 December 2004.

[62] B. Boehm. Sicherheit und Probleme: CSMA-Protokoll. *http://referate.mezdata.de/sj2003/wlan_bianca-boehm/praesentation/sicher1.html*, 2004.

[63] Richard Sietmann. Luftbrücken über Berlin. *c't 2005, Heft 26*, December 2005.

[64] Andreas Tonnesen. Implementing and extending the Optimized Link State Routing Protocol. *UniK University Graduate Center Oslo*, 1 August 2004.

Abbildungsverzeichnis