

Konzept und Design einer autonom funktionsfähigen Knoten-Plattform für Wireless Mesh Backbones

Alexander Gladisch, Martin Arndt, Robil Daher, Martin Krohn, Djamshid Tavangarian

Lehrstuhl für Rechnerarchitektur
Universität Rostock
Albert-Einstein-Straße 21
D-18059 Rostock
vorname.nachname@uni-rostock.de

Abstract: Wireless Mesh Networks (WMNs) werden aufgrund ihrer Flexibilität als effiziente, drahtlose Alternative zur Versorgung von strukturschwachen Gebieten eingesetzt. Das Design vollständig autonom funktionsfähiger Mesh-Knoten stellt jedoch noch immer eine große Herausforderung dar. Basierend auf einer modularen Netzwerkarchitektur wird in diesem Artikel das Konzept für eine Commercial Off-The-Shelf (COTS) Hardware und Open-Source-Software (OSS) basierte Mesh-Knoten-Plattform entwickelt, welche den unabhängigen Betrieb eines solchen Kommunikationsnetzwerkes ermöglicht. Dabei werden auch Konzepte zur Energieversorgung berücksichtigt. Resultierend aus dieser Entwicklung werden beispielhaft zwei funktionsfähige Prototypen präsentiert.

1 Einführung

Wireless Mesh Networks (WMNs) werden aufgrund ihrer Flexibilität als effiziente, drahtlose Alternative zur Versorgung strukturschwacher Gebiete eingesetzt. Dies trifft nicht nur auf kleinere Städte oder ländliche Gemeinden zu, in denen kein drahtgebundener Breitbandanschluss verfügbar ist [BW08], sondern auch international, z.B. in Entwicklungsländern oder abgelegenen Regionen ohne Infrastruktur. WMNs sind besonders flexibel, da sie in der Regel aus autonomen (d.h. unabhängig von der vorhandenen Infrastruktur und ohne zentrale Netzwerkarchitektur funktionsfähigen) Knoten bestehen, die miteinander kooperieren, um Clients mit den benötigten Netzwerkdiensten zu versorgen [AW05]. Zurzeit existieren verschiedene Netzwerktechnologien, die Mesh-Eigenschaften unterstützen, darunter sind u. a. UMTS-LTE, IEEE 802.16 WiMAX, IEEE 802.11 WLAN und IEEE 802.15 WPAN. Theoretisch können diese Technologien sowohl in Backbone- als auch in Zugriffs-Netzwerken verwendet werden, praktisch stellt IEEE 802.11 in WMNs momentan für beide Netzwerkschichten die meistverwendete Technologie dar [LZ06]. Aufgrund der weiten Verbreitung der Technologie ist WLAN besonders kostengünstig, weshalb sie bei der Entwicklung einer autonomen Knoten-Plattform vordergründig berücksichtigt wird.

Das Aufbauen autonomer Mesh-Knoten stellt eine große Herausforderung dar, da deren Selbstständigkeit auf zwei Ebenen betrachtet werden muss: Kommunikationsnetzwerk und Energieversorgung. Im Kommunikationsnetzwerk sind Techniken wie Selbst-Organisation und -Konfiguration sowie Selbst-Heilung zu unterstützen. Dafür müssen

verschiedene Routing-Protokolle, QoS-Mechanismen, etc. berücksichtigt werden. Für eine autonome Energieversorgung müssen verschiedene alternative Energiequellen in Kombination mit Energiespeichern wie Akkumulatoren analysiert werden. Der Einsatz eines geeigneten Energie-Management-Mechanismus ist dabei obligatorisch.

Der aktuelle Stand der Technik zeigt, dass die Autonomie der Knoten durch aktuell verfügbare Plattformen bisher nicht oder nur unzureichend gewährleistet ist. In dieser Arbeit werden die Anforderungen an eine solche Plattform sowohl für das Kommunikationsnetzwerk als auch für die Energieversorgung beschrieben und ein Konzept für ein entsprechendes Plattformdesign auf Basis von Commercial Off-The-Shelf (COTS) Hardware und Open-Source Software (OSS) erstellt. Anschließend werden zwei Prototypen präsentiert, die basierend auf den Ergebnissen des vorgestellten Konzeptes aufgebaut wurden.

Im Weiteren ist diese Arbeit wie folgt gegliedert: In Kapitel 2 wird der aktuelle Stand der Technik beschrieben. Kapitel 3 stellt beispielhaft die Architektur eines WMNs vor, in dem das entwickelte Plattformdesign eingesetzt werden kann. Kapitel 4 beschreibt das Konzept für den Aufbau einer Knoten-Plattform. Dabei wird zwischen Netzwerkkomponenten, Protokollen und Diensten sowie Hard- und Softwareplattform unterschieden. Nachfolgend werden in Kapitel 5 zwei konkrete Prototypen vorgestellt. Schließlich wird der Artikel in Kapitel 6 zusammengefasst.

2 Stand der Technik

In Aufbau und Implementierung unterscheiden sich die verfügbaren Lösungen von 802.11-basierten Knoten-Plattformen in zwei Design-Philosophien [Bu03]: Hardware-Defined Radio (HDR) und Software-Defined Radio (SDR).

In der HDR-Technologie ist die Hardware-Plattform für die Funkschnittstelle komplett verantwortlich, während die Software-Plattform, in Form von Betriebssystem und Treibern, der Anbindung an Nutzerapplikationen sowie Netzwerkservices dient. Unternehmen wie Cisco, Lucent-Alcatel, 3COM und andere stellen ihre eigenen Plattform-Produkte für Hardware und Software her. Für ihren angestammten Anwendungszweck bieten derartige Produkte ausreichend Leistung, stellen jedoch proprietäre Lösungen dar. Die Mehrheit dieser ist nur von den Herstellern selbst mit vielen Beschränkungen in Hard- und Software erweiterbar. Auch wenn einige Produkte mit OSS-Plattformen angeboten werden, wie z.B. WLAN-AP/Router von LinkSys und Asus, bleiben die Beschränkungen von Hardware-Ressourcen ein wesentliches Problem für die Integration neuer Software- und Hardware-Module, was sich z. B. in einer Speicher-Beschränkung für ein neues, speicherintensiveres Routing-Protokoll, oder einer mangelnden Erweiterbarkeit um eine neue WLAN-Karte zeigt. Ganz im Gegensatz zu diesen fertigen Produkten werden skalierbarere Lösungen meist basierend auf COTS Hardware-Komponenten und OSS aufgebaut [RM08]. Die entsprechenden Hardware-Plattformen sind meist mit General-Purpose-CPU's (ARM, XScale, x86, etc.) ausgerüstet. Als Bauform der Mainboards werden oft PC/104, ECX oder nicht standardisierte Größen, ausgestattet mit mehreren mini-PCI-Slots zur Anbindung von WLAN-Karten, eingesetzt. Hardwarekomponenten wie

Netzwerkschnittstellen, Antennen, etc. werden in Zusammenhang mit System-Design und Netzwerkplanung eingebaut. Ein Beispiel für eine Firma, die mit vorgefertigten Mesh-Knoten dieses Design verfolgt ist Saxnet [Sa09]. Die Software-Plattform ist meist Linux basiert, wobei spezielle Linux-Distributionen wie DSL, Puppy oder das stark modifizierbare Buildroot mit kleinem Footprint verwendet werden. Zusätzliche Software-Komponenten werden nach Bedarf integriert, insbesondere Module für Routing-Protokolle, Netzwerk-Services, Nutzer-Applikationen, etc. In der Summe entstehen komplette Knoten-Plattformen.

In der SDR-Technologie hingegen wird die Funkschnittstelle nur partiell, d. h. nur der Signal-Empfang und -Versand, in Hardware implementiert, während die gesamte Signalbearbeitung von der Software-Plattform durchgeführt wird. Nach unserem Kenntnisstand bietet das GNU Radio Open-Source-Projekt, als Software-Plattform, in Zusammenhang mit der ETTUS USRP und einer General-Purpose-Prozessor-basierenden Hardware-Plattform, momentan die einzige SDR-technologie-basierende Open-Source-Lösung für WLAN-Knoten, wobei zusätzliche OSS-Pakete für den Aufbau einer vollständigen Knoten-Plattform benötigt werden. Dickens et. al [DD08] beschreibt den Aufbau eines solchen Knotens für Forschungszwecke, wobei der vorgestellte Prototyp beschränkte Erweiterungsmöglichkeiten, insbesondere der Funkschnittstellen, anbietet.

Beide beschriebenen Möglichkeiten, sowohl die verfügbaren HDR-basierten Lösungen, als auch die verfügbaren Lösungen von SDR-Technologie decken die Anforderung an Design- und System-Flexibilität in Zusammenhang mit der erwarteten Funktionalität und hoher Erweiterbarkeit sowie Skalierbarkeit von Netzwerk-Knoten kaum ab. Alternativ bietet die Kombination von OSS und COTS-Hardware-Komponenten eine vielversprechende Variante zum Aufbau flexibler Knoten-Plattformen, welche die entsprechenden Anforderungen erfüllen sollte.

3 Beispiel einer modularen Netzwerkarchitektur

Als Beispiel für den Einsatz der Knoten-Plattform soll eine modulare Multi-Layer Architektur dienen, die für die Versorgung von infrastrukturschwachen Gebieten vorgesehen ist. In ihr sind drei grundlegende Netzwerk-Ebenen definiert: Zugriffsnetz (Access), Backbone und Zubringer (Supply/Internet), wie in Abbildung 1 dargestellt. Im Netzwerk können drei Haupttypen von Knoten nach dem IEEE 802.11s-Draft unterschieden werden: (1) Der Mesh Point (MP) als Relay-Station (RS) dient zur Paket-Weiterleitung in der Backbone-Ebene; (2) Der Mesh Access Point (MAP) ermöglicht zum einen den Zugriff der Clients (Mobile Station – MS) auf das Netzwerk und zum anderen die Anbindung an die Backbone-Ebene; (3) Das Mesh Point Portal (MPP) dient als Gateway-Station (GS) zur Anbindung der Backbone-Ebene an die Zubringer-Ebene. Zur Überbrückung von großen Strecken und zur Erhöhung der Datenübertragungsrate können an allen Knoten der Ebenen spezialisierte Antennen (z.B. Yagi- oder omnidirektionale Antennen) verwendet werden. Die Anwendungsbereiche der beschriebenen Architektur reichen von der Versorgung von stationären Teilnehmern, wie Häusern oder Installationen, bis hin zur Unterstützung von mobilen Clients (Mobile Stations, MS), die sich innerhalb der abgedeckten Fläche bewegen.

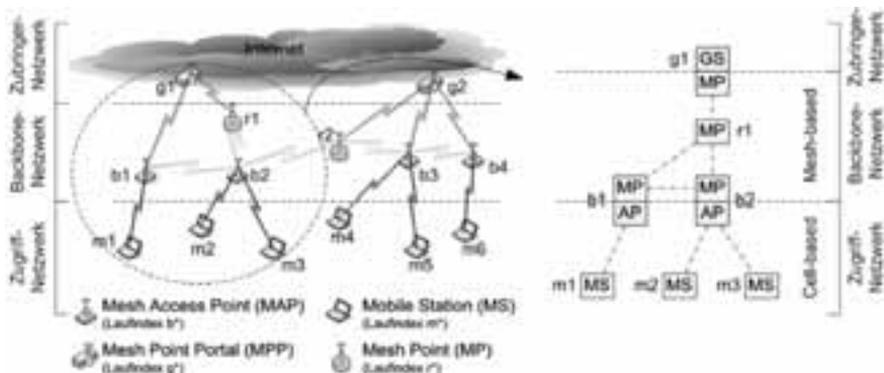


Abbildung 1: Darstellung der Beispielnetzwerkarchitektur

4 Konzept und Design der Knoten-Plattform

4.1 Netzwerkkomponenten

WLAN Technologien

Aufgrund der weiten Verbreitung des 802.11b/g Standards im Endkundenbereich, bietet sich dieser besonders zur Realisierung der Zugriffsebene an. Durch die hohe Bandbreite, die 11 unabhängigen Kanäle im 5 GHz Band und die deutlich niedrigeren Interferenzen im 5 GHz Bereich, bietet der 802.11a/h Standard die beste Grundlage für den Einsatz in der Backbone-, bzw. Zubringer-Ebene [JP03]. Ebenfalls interessant für die Verwendung im Zubringer- und Backbone-Netzwerk ist der neue Draft 802.11n, sobald er endgültig als Standard verabschiedet wurde [DA03].

Schnittstellen und Antennen

Die Anzahl der drahtlosen Schnittstellen leitet sich aus dem Einsatzzweck und damit dem Knotentyp ab, der Sachverhalt ist in Tabelle 1 dargestellt. Die Einschränkung auf vier Schnittstellen ergibt sich aus der Anzahl zur Verfügung stehender Kanäle sowie der großen Anzahl der damit entstehenden Möglichkeiten zur Gestaltung der Netzwerktopologie. An dieser Stelle sollen zwei Fälle näher betrachtet werden: (1) Es ist nur ein Netzwerkinterface vorhanden, wobei für das gesamte Netzwerk nur ein gemeinsamer Kanal zur Kommunikation verwendet wird. Dabei werden omnidirektionale Antennen eingesetzt. In diesem Szenario sind der Datendurchsatz und die Resistenz des Systems auf Störeinflüsse relativ gering. Dagegen ist die Installation des Systems leicht durchzuführen, die Wartung wird vereinfacht und die Kosten des Gesamtsystems sind vergleichsweise niedrig. (2) Die Knoten besitzen jeweils vier Schnittstellen, wovon drei für den Backbone-Bereich eingesetzt und jede mit einem nicht überlappenden Kanal betrieben wird. Für die Verbindung der Knoten kommen gerichtete Antennen zum Einsatz. In diesem Szenario verfügt das gesamte Netzwerk über eine hohe Performance und ist resistent gegen Störeinflüsse. Es werden Algorithmen zur Kanalverteilung (z.B. Hyacinth [RC05]) benötigt, um Performance-Einbußen durch sich überlappende Kanäle zu vermeiden. Wartung und Installation des Systems sind komplex und erfordern einen erhöhten Aufwand.

Tabelle 1: Konfiguration der Knotentypen

Knotentyp	Anzahl (A) der drahtlosen Netzwerkschnittstellen	Eingesetzte Technologien	Eingesetzte Antennentypen
MP	$1 \leq A \leq 4$	802.11a, 802.11n (5 GHz)	gerichtet
MAP	$2 \leq A \leq 4$	802.11a, 802.11n (2,4 bzw. 5 GHz), 802.11g	gerichtet, omnidirektional
MPP	$1 \leq A \leq 4$	802.11a, 802.11n (5 GHz)	gerichtet

4.2 Protokolle und Dienste

Autonomie und Fernwartung

Um die Selbstorganisation und –konfiguration der Knoten zu gewährleisten, muss eine möglichst modulare Software-Plattform entwickelt werden (eine entsprechende Eigenentwicklung wird in Kapitel 4.3 beschrieben), in die alle für den optimalen Betrieb eines Netzwerks notwendigen Funktionen integriert werden können. So sind ein Mechanismus zur strukturierten Vergabe oder zur dynamischen Konfiguration von IP-Adressen (z.B. über DHCP oder eigens entwickelte Adressvergabemechanismen), ein Algorithmus zur Verteilung von Funk-Kanälen der drahtlosen Schnittstellen (z.B. Hyacinth [RC05]) sowie eine geeignete Power-Management-Software (die aktuell von den Autoren entwickelt wird) für den dauerhaften und autonomen Betrieb der Plattform unverzichtbar. Um die Plattform effizient wartbar zu gestalten, sind Geräte- bzw. Funktionsparameter fernadministrierbar (z.B. über SNMP oder SSH) zu entwerfen.

Routing

Für die Wahl geeigneter Kommunikationspfade innerhalb des Netzwerks muss ein geeignetes Protokoll im Software-Framework bereitstehen. Insbesondere für drahtlose Netzwerke aller Art (Ad-hoc Netzwerke, MANETs und WMNs) steht eine große Anzahl verschiedener Routing-Protokolle zur Verfügung. In [HX02] werden die verschiedenen Design-Philosophien von Routing-Protokollen (proaktiv, reaktiv, hierarchisch, geographisch) sowie typische Vertreter (OLSR, AODV, uvm.) beschrieben und klassifiziert. Die Leistungsfähigkeit des eingesetzten Routing-Protokolls hängt stark von der im Protokoll genutzten Metrik ab. Neben den in den Protokollen genutzten Standard-Metriken können auch Metriken wie Expected-Transmission-Count (ETX) [CA03] bzw. -Time (ETT) und deren Erweiterung [DP04] in drahtlose Routing-Protokolle integriert werden. Als besonders effizient für Mesh-Netzwerke hat sich die Metrik EMO (Extended Medium Observation, [Ko08] erwiesen. Welches Protokoll sich für den Einsatz auf der Knoten-Plattform eignet, hängt also stark vom Einsatzszenario ab und muss nach Bedarf entschieden werden. Gegebenenfalls ist für spezielle Anwendungsfälle die Entwicklung eigener Lösungen und deren Integration in das Framework sinnvoll.

Quality of Service und Lastbalancierung

Um die vorhandenen Netzwerk-Ressourcen optimal nutzen zu können, werden geeignete Mechanismen zur Verteilung der im Netzwerk entstehenden Last benötigt, die in das Software-Framework eingebunden werden können. Durch Lastbalancierung kann es dem Gesamtsystem ermöglicht werden, Ressourcen- sowie Mobilitätsmanagement zu unterstützen und dadurch QoS-Eigenschaften für die Anbindung der Teilnehmer zu

verbessern. Dies kann beispielsweise mittels QoS-orientierter Lastbalancierungs-Mechanismen wie in [DT06] erreicht werden. Weitere Möglichkeiten stellen IntServ (RFC2210) oder DiffServ (RFC2475) sowie die Unterstützung des Standards 802.11e dar. Auch Eigenentwicklungen können bei geeigneter Integration in das Software-Framework verwendet werden.

Sicherheit

Grade in frei zugänglichen drahtlosen Netzwerken spielen Sicherheitsmechanismen eine wichtige Rolle. Insbesondere die Mechanismen der Protokollerweiterung 802.11i, welche die Authentifizierungs- und Autorisierungsmechanismen des 802.1X Standards enthält und die AES-Verschlüsselung unterstützt, müssen in der Knoten-Plattform implementiert werden. Als Sicherheitsprotokolle können EAP-TLS oder EAP-TTLS in Zusammenhang mit RADIUS zum Einsatz kommen. Ebenso sollte eine passende Software zur Systemüberwachung und zum Systemmanagement integriert werden.

4.3 Software-Plattform

Modularität

Um für die genutzten Protokolle und Dienste eine einheitliche Schnittstelle auf die benötigten Funktionen des Betriebssystems wie z.B. den Netzwerk-Stack oder die Treiber zu schaffen und eine einheitliche Datenbasis für alle Protokolle und Dienste vorzuhalten, bietet sich eine Agenten-basierte Softwarearchitektur an. Das Grundgerüst des Agenten stellt die genannte Funktionalität über definierte Schnittstellen bereit. Die z.B. als Plugin realisierten Protokolle und Dienste können über diese Schnittstellen die von ihnen benötigten Daten abrufen und entsprechende Steuerbefehle auslösen. Somit wird weiter vom Betriebssystem und den verwendeten Treibern abstrahiert und eine modular erweiterbare Firmware geschaffen.

Betriebssystem

Um die erforderliche Grundfunktionalität für die benötigten Dienste bereitzustellen, wird ein geeignetes Betriebssystem (BS) benötigt. Dieses sollte möglichst diverse Hardwarekomponenten und Prozessor-Architekturen unterstützen, damit die Software-Plattform flexibel einsetzbar bzw. portabel ist. Besonders gut für diesen Einsatzzweck geeignet ist Linux. Durch den frei verfügbaren Quellcode ist es flexibel anpassbar, auch der Großteil der für den Betrieb der Plattform benötigten Funktionen ist in Linux bereits enthalten. Die Entwicklung eigener Software für Linux-Systeme ist problemlos möglich, da eine Vielzahl von Programmiersprachen mit zugehörigen Compilern für die unterschiedlichsten Hardware-Plattformen zur Verfügung stehen. Um ein möglichst Ressourcen sparendes System zu erstellen, sollte ein speziell auf die Hardware-Plattform angepasstes, auf Linux basierendes Buildroot-System [Br09] verwendet werden. Es beinhaltet nur die genau für diesen Einsatzzweck benötigten Softwarekomponenten, was die optimale Nutzung der vorhandenen Ressourcen sicherstellt.

Treiber

Um die Funktionalität der eingesetzten Protokolle und Dienste bereitzustellen, müssen die Treiber der drahtlosen Netzwerkschnittstellen besondere Anforderungen erfüllen. Der Treiber muss in der Lage sein, detaillierte Informationen über das drahtlose Medium zu liefern und Befehle für die Steuerung der Hardware (außerhalb der normalen

Funktionen im End-Kunden-Bereich) entgegenzunehmen. Für Linux bieten sich daher insbesondere die Treiber Madwifi [Ma09] bzw. die Nachfolger Ath5k und Ath9k [Wk09] für WLAN-Karten mit Atheros-Chipsätzen an, da sie diese Anforderungen erfüllen. Um den Funktionsumfang der Treiber noch zu erweitern, können sie mit dem Linux-Programm HostAP kombiniert werden.

4.4. Hardware-Plattform

Jede Hardware-Plattform besitzt einige grundlegende Komponenten. Sie besteht aus einem Mikrocomputer mit einem zentralen Prozessor, Arbeitsspeicher, Festspeicher und verschiedenen Ein- und Ausgabeschnittstellen. Dazu gehören Netzwerkschnittstellen, serielle Schnittstellen zur Behebung von Störungen sowie USB- und Mini-PCI-Schnittstellen zur einfachen Erweiterung sowie zur Erhöhung der Flexibilität des Geräts, z. B. mit zusätzlichem externen Speicher. Um die Plattform so wartungsarm wie möglich zu realisieren, muss auf bewegliche Teile wie Festplatten und Lüfter verzichtet werden, da diese Komponenten besonders fehleranfällig sind [GH99]. Um die Komponenten der Plattform vor Umwelteinflüssen zu schützen, wird weiterhin ein wetterfestes Gehäuse benötigt. Im Folgenden wird auf die einzelnen Kriterien detaillierter eingegangen.

Mikroprozessor

Als Basis des Systems stehen verschiedene Mikroprozessor-Architekturen zur Auswahl. Neben der weit verbreiteten x86-Architektur sind für eingebettete Systeme insbesondere die ARM-Architektur [Ar09] und entsprechende Derivate davon (z.B. XSCALE von Intel) interessant. Aktuelle ARM-Prozessoren haben im Gegensatz zu x86-Prozessoren einen besonders geringen Stromverbrauch, was die autonome Stromversorgung der Plattform enorm erleichtert. Zusätzlich wird wenig Abwärme erzeugt, was eine lüfterlose und somit wartungsärmere Plattform ermöglicht. Da für diese Architektur entsprechende Compiler für Linux existieren, ist ihr Einsatz in der Plattform empfehlenswert.

Netzwerkschnittstellen

Damit die Hardware-Plattform universell eingesetzt werden kann, müssen die über Mini-PCI oder USB angeschlossenen drahtlosen Netzwerkschnittstellen alle erforderlichen WLAN-Standards unterstützen. Dazu gehören die Standards 802.11a/b/g sowie der Draft 802.11n zum Betrieb des Physical- und des MAC-Layers, aber auch weitere Funktionalität wie Sicherheit, die mit der Standarderweiterung 802.11i bereitgestellt wird. Zusätzlich muss die Netzwerkschnittstelle mit den in der Software-Plattform beschriebenen Treibern betrieben werden können. Dies trifft insbesondere auf WLAN-Karten mit aktuellen Atheros-Chipsätzen zu, weshalb deren Einsatz in der Hardware-Plattform empfohlen wird. Zur Integration der Netzwerkkarten sollte Mini-PCI bevorzugt werden, da es höhere Datenraten als USB zur Verfügung stellt. USB 2.0 (mit Datenraten bis zu 480 MBit/s) stößt bei Anschluss von Draft 802.11n Hardware (mit Datenraten bis zu 600 MBit/s) bereits an seine Leistungsgrenzen.

Stromversorgung

Für den autonomen Betrieb der Plattform wird eine adäquate Stromversorgung benötigt. Hier bietet sich der Einsatz (verschiedener) alternativer Energiequellen wie z.B. Solarzellen oder Windkraft in Kombination mit Energiespeichern wie Akkumulatoren an. Dies ermöglicht die eigenständige Erzeugung der von der Plattform benötigten

Energie und befähigt den Knoten auch Zeiträume, in denen die Energieerzeugung nicht möglich ist, zu überbrücken. Die Steuerung der Energieversorgung übernimmt eine spezielle Power-Management-Software. Je nach geographischer Lage der einzelnen Netzwerkknoten können zudem mehrere Knoten des WMNs zu einem Verbund zusammengefasst und über eine einzelne, besonders leistungsfähige Energiequelle betrieben werden. Der Anschluss der Energiequelle an die Plattform erfolgt über Power-over-Ethernet (PoE) oder einen Niederspannungs-Gleichstrom-Anschluss.

Realisierungsformen

Eine Plattform mit den geforderten Eigenschaften kann, wie in [RM08] beschrieben als Personal Computer (PC) oder Eingebettetes System realisiert werden. Dabei erscheint die Realisierung als Eingebettetes System besonders vorteilhaft. Es basiert auf einer großen Auswahl gut verfügbarer und somit kostengünstiger Hardwarekomponenten und ist bei ausreichender Leistung zum Betrieb der Plattform besonders energiesparend.

5 Prototypische Umsetzung



Abbildung 2: Prototyp 1 (indoor)



Abbildung 3: Prototyp 2 (outdoor)

Anhand der gegebenen Empfehlungen und des erstellten Konzepts der Knotenplattform, wurden zwei Prototypen konstruiert. Um die Plattform im Access- und im Backbone-Bereich von Netzwerken einsetzen zu können, wurde vorrangig auf eine möglichst flexible Zusammenstellung der einzelnen Komponenten geachtet. Es wurde bewusst das genannte Maximum von vier drahtlosen Netzwerkschnittstellen integriert, um eine Vielzahl von Anwendungsszenarien abzudecken. So können die beiden entstandenen prototypischen Knoten-Plattformen zur Implementierung von MP, AP, MAP oder auch MPP mit vier drahtlosen und einer zusätzlichen drahtgebundenen Netzwerkschnittstelle dienen. Beide Prototypen haben gemein, dass sie momentan ohne Power-Management über das Stromnetz bzw. Akku betrieben werden. Die weiterentwickelte Version des zweiten Prototyps wird auf Basis von Solar-Energie in Kombination mit Akkus betrieben, wobei eine Power-Management-Software eingesetzt wird.

5.1 Prototyp für den Indoor-Bereich

Um die generelle Umsetzbarkeit zu verifizieren, wurde diese Knoten-Plattform zunächst als Indoor-Variante realisiert. Die Plattform besteht aus den in Tabelle 2 genannten Hardware-Komponenten und wird mit einer speziell zusammengestellten Software-Plattform betrieben, die alle in Kapitel 4 definierten Bestandteile enthält. Eine Darstellung des ersten Prototyps ist in Abbildung 2 zu sehen.

Tabelle 2: Komponenten der Prototypen

Komponente		Prototyp 1 (Indoor-Bereich)	Prototyp 2 (Outdoor-Bereich)
Protokolle u. Dienste	Adressvergabe	DHCP	DHCP
	Routing-Protokoll	OLSR	AODV basiert auf EMO
	Quality of Service	Nicht integriert	Nur 802.11e
	Sicherheitsmechanismus	WPA	WPA2 + RADIUS
Softwareplatt- form	Betriebssystem	Buildroot/Linux: Eigene Zusammenstellung	Buildroot/Linux: Eigene Zusammenstellung
	WLAN-Treiber	Madwifi	Atheros 5k
	Root-Dateisystem	Modifiziertes Buildroot Remote-Update möglich	Modifiziertes Buildroot Remote-Update möglich
Hardwareplattform	Prozessor	Intel Celeron 400 MHz	Intel XScale IXP 425 533 MHz
	Hauptplatine	PC104-Board mit 4 Mini-PCI-Steckplätzen, 2 Ethernet-Schnittstellen 2 RS232-Schnittstellen	Gateworks Avila GW2348 mit 4 Mini-PCI-Steckplätzen, 2 Ethernet-Schnittstellen, 2 RS232 Schnittstellen
	Festwertspeicher	1 GB CF-Karte	512 MB CF-Karte
	Funknetzwerk- Schnittstellen	4 WLAN 802.11b/g Mini-PCI Steckkarten (Atheros-Chipsatz)	4 WLAN 802.11a/b/g Mini-PCI Steckkarten (Atheros-Chipsatz)
	Hauptspeicher	128 MB SDRAM	64 MB SDRAM
	Gehäuse	Eigenentwicklung	Gateworks GW3020, witterungsfest

5.2 Prototyp für den Outdoor-Bereich

Dieser Prototyp (Abbildung 3) wurde im Vergleich zum ersten hinsichtlich der Erfüllung der Anforderungen deutlich verbessert. So ist sein Gehäuse nun auch für den Einsatz im Outdoor-Bereich geeignet, die Energieversorgung wurde auf PoE umgestellt. Durch die Erweiterung der Funk-Netzwerkschnittstellen von 802.11b/g auf 802.11a/b/g Technologie, wird es dem Backbone-Netzwerk ermöglicht, im 5 GHz Frequenzband zu arbeiten und so Interferenzen zu minimieren bzw. die Leistungsfähigkeit zu optimieren. Ebenfalls wurde die Gerätegröße reduziert, unnötige Bestandteile wurden entfernt. Der Prototyp besteht aus den in Tabelle 2 genannten Komponenten und verwendet eine weiterentwickelte Version der Software-Plattform der ersten Version.

6 Zusammenfassung und Ausblick

Im vorliegenden Artikel wurden das Konzept eines autonom funktionsfähigen Mesh-Knotens vorgestellt. Auf Basis der beschriebenen Anforderungen an einen solchen Mesh-Knoten wurde eine entsprechende Knoten-Plattform entwickelt, wobei das Netzwerkdesign sowie die Hard- und Software-Architektur differenziert betrachtet wurden. Während beim Netzwerkdesign insbesondere auf die Konfiguration der Netzwerkkomponenten und verwendete Technologien eingegangen wurde, wurde beim Design der Softwareplattform die zugrunde liegende Open-Source Systemsoftware sowie die benötigten Protokolle und Dienste beschrieben. Beim Design der Hardwareplattform wurden verschiedene Architekturen der Basis-Plattform betrachtet

und ihr autonomer Betrieb sowie mögliche Realisierungsformen mit COTS-Hardware fokussiert. Weiterhin wurden als Beispielimplementierung zwei funktionsfähige Prototypen vorgestellt. Um die Leistungsfähigkeit der Prototypen weiter zu steigern, findet aktuell eine weitere Optimierung der Hard- und Software-Plattform statt.

Literaturverzeichnis

- [Ar09] Webseite von ARM, <http://www.arm.com>
- [AW05] Akyildiz, I. F.; Wang, X.; Wang, W.: Wireless Mesh Networks: A survey. In: Computer Networks Journal (Elsevier), 2005.
- [Ba06] Bundesnetzagentur: Allgemeinzuteilung von Frequenzen in den Bereichen 5150 MHz - 5350 MHz und 5470 MHz - 5725 MHz für Funkanwendungen zur breitbandigen Datenübertragung, WAS/WLAN (Wireless Access Systems including Wireless Local Area Networks), 2006
- [Br09] Buildroot Webseite, <http://buildroot.uclibc.org/>
- [Bu03] Burns, P.G.: Software Defined Radio for 3G. Published by Artech House, 2003
- [Bw08] Bundesministerium für Wirtschaft und Technologie: „Breitband-Verfügbarkeit in Deutschland“, <http://www.zukunft-breitband.de/BBA/Navigation/Breitbandatlas/laenderkarten.html?>, 2008
- [CA03] De Couto, D., Aguayo, D., Bicket, J., Morris, R.: A High-Throughput Path Metric for Multi-Hop Wireless Routing, ACM Mobicom 2003
- [CB03] Chapin, J.M.; Bose, V.G.: Vanu Software Radio System, Vanu Inc., www.vanu.com, 2003
- [DA03] Doefexi, A.; Armour, S.; Beng-Sin Lee; Nix, A.; Bull, D.: An evaluation of the performance of IEEE 802.11a and 802.11g wireless local area networks in a corporate office environment. IEEE International Conference on Communications, 2003
- [DD08] Dickens, M.L.; Dunn, B.P.; Laneman, J.N.: Design and Implementation of a Portable Software Radio, IEEE Communication Magazine, August 2008
- [DP04] Draves, R., Padhye, J., Zill, B.: Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, ACM Mobicom 2004
- [DT06] Daher, Robil; Tavangarian, Djamshid: “QoS-oriented Load Balancing for WLANs”, In Proc. of The First International Workshop on Operator-assisted (Wireless Mesh) Community Networks 2006 (OpComm'06), Berlin, Germany, September 18-19, 2006
- [GH99] Gebauer, J.; Hartmann, H.; Seguin, M.: Clustering mit Windows NT. 1. Auflage, Addison-Wesley, Bonn 1999
- [HX02] Hong, X., Xu, K., Gerla, M.: Scalable routing protocols for mobile ad hoc networks. Network, IEEE , vol.16, no.4, pp.11-21, Jul/Aug 2002
- [JP03] Jain, K., Padhye, J., Padmanabhan, V. N., Qiu, L.: Impact of interference on multi-hop wireless network performance. In: ACM Annual International Conference on Mobile Computing and Networking (MOBICOM), 2003
- [Ko08] Kopp, Heiko; Krohn, Martin; Tavangarian, Djamshid.; Extended Medium Observation (EMO) — A load-aware metric for routing in wireless mesh backbone networks, Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference on , 2008
- [LZ06] Lee, M. J.; Zheng, M.; Ko, J. B.; Shrestha, D.M.: Emerging standards for wireless mesh technolog, IEEE Wireless Communications, vol. 13, no. 2, pp. 56 - 63, April 2006
- [Ma09] Webseite des Projekts Madwifi, <http://madwifi-project.org/>
- [RC05] Raniwala, A.; Chiueh, T.: Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In Proc. of IEEE INFOCOM, 2005.
- [RM08] Riggio, R.; Miorandi, D.; Chlamtac, I.; Scalabrino, N.; Gregori, E.; Granelli, F.; Yuguang F.: Hardware and software solutions for wireless mesh network testbeds, IEEE Communications Magazine, vol.46, no.6, pp.156-162, June 2008
- [Sa09] Webseite der Firma Saxnet, <http://saxnet.de/>
- [Sj03] Steinheider, J.: Software-Defined Radio Comes of Age, Vanu Inc., www.vanu.com, 2003
- [Wk09] Webseite des Linux Wireless Wikis, <http://wireless.kernel.org/>