

Wireless LAN basierend auf Cellular IP

Projektarbeit SS 2001 - Sna06

Studierende: Marc Steiner / IT3b
Thomas Wendel / IT3a

Dozent: Dr. Andreas Steffen

Partnerfirma: FutureLAB AG
Schwalmenackerstrasse 4
8400 Winterthur
<http://www.futurelab.ch>

Inhaltsverzeichnis

1 Zusammenfassung	4
2 Aufgabenstellung	5
2.1 Beschreibung	5
2.2 Aufgaben	5
3 Einführung Cellular IP	6
3.1 Cellular Internet	6
3.2 Cellular IP Protokollübersicht	6
3.2.1 Merkmale	6
3.2.2 Routing	7
3.2.3 Handoff	7
3.2.4 Paging	8
4 Realisierung	9
4.1 Topologien	9
4.1.1 Dedicated Net	9
4.1.2 Shared Net	10
4.2 Installiertes Netzwerk	11
4.3 Node als Repeater	13
4.3.1 Installation einer Repeaternode	13
4.3.2 Überlegungen zu einer Repeaternode	13
5 Installationsanleitung	14
5.1 Hardware und Treiber	14
5.1.1 Verwendete Hardware	14
5.1.2 PCMCIA-Adapter- und WaveLAN-Karten-Treiber	15
5.1.3 Treibereinstellungen	15
5.2 Software	16
5.2.1 Betriebssystem	16
5.2.2 Wireless Tools	17
5.2.3 Cellular IP v1.1	17
6 Konfigurationsanleitung	18
6.1 Wired-Network	18
6.2 Cellular IP	18
6.2.1 Anpassung des tcl/tk-Scripts	18
6.2.2 Anpassung des Makefiles	18
6.2.3 Anpassung der Konfigurationsfiles	19
7 Bedienungsanleitung	20
7.1 Starten des Gateway	20
7.2 Starten der Node	20
7.3 Starten der Mobile Hosts	20
8 Probleme mit der Software	22
8.1 cipnode.c	22
8.1.1 ARP is no IP	22
8.1.2 Beacon-Signal	23
8.2 cipmobile.c	24
8.3 Sonstige Mängel	25

9 Messungen	26
9.1 Shared Topologie	26
9.1.1 Abdeckung	26
9.2 Dedicated Topologie	27
9.2.1 Abdeckung	27
10. Projektverlauf	31
11 Schlusswort	32
11.1 Fazit / Ausblick	32
11.2 Dank	32
12 Anhang	33
12.1 Konfigurationsfiles	33
12.1.1 Mobile Host	33
12.1.2 Shared Topologie	33
12.1.3 Dedicated Topologie	34
12.2 Messwerte	36
12.3 Glossar	37
12.4 Literaturverzeichnis	38
12.5 CD-Verzeichnisstruktur	38

1 Zusammenfassung

Der Mensch wird immer mobiler. Telefonieren auf offener Strasse oder im Auto war vor ein paar Jahrzehnten noch ein kühner Traum, heute sind die kleinen GSM-Mobiltelefone nicht mehr wegzudenken. Auch das Internet hat sich zu einem wichtigen Bestandteil unserer Gesellschaft entwickelt. In der Geschäftswelt so wie auch im Heimbereich werden immer mehr Dinge übers Internet erledigt. Kommunizieren per E-Mail oder VoIP, einkaufen im Online-Shop, Video-on-demand, Banktransaktionen, um nur einige zu nennen.

Das Internet hat allerdings noch einen wesentlichen Nachteil: Es ist nicht mobil. Es ist zwar durchaus möglich seinen Notebook mittels GSM-Telefon ans Internet anzubinden, was aber in Anbetracht der langsamen Verbindungsgeschwindigkeit kaum angewendet wird. Die grossen Telefongesellschaften prophezeien zwar schnellere Bandbreiten mittels GPRS und UMTS, doch beide Techniken kommen die Anbieter teuer zu stehen und werden eventuell wie auch WAP zum Flop.

Warum also nicht den IEEE Standard 802.11, der sich schon recht gut zu etablieren scheint und auch für den Heimanwender erschwinglich ist, verwenden? In kleineren Räumen wird 802.11 Funk-LAN-Equipment bereits eingesetzt. Doch leider ist es wegen der beschränkten Reichweite noch nicht möglich zum Beispiel ein ganzes Bürogebäude oder gar ein Stadtteil grossflächig abzudecken. Man bräuchte ein flächendeckendes Netz von Basisstationen basierend auf der Idee von GSM. Cellular IP ist ein Protokoll, welches die Realisierung eines auf Zellen basierenden WaveLAN Netzes mit einer grossflächigen Abdeckung und guter Skalierbarkeit ermöglicht. Es wurde an der Columbia University in Zusammenarbeit mit Ericsson entwickelt.

Das Ziel unserer Projektarbeit war, Cellular IP im Auftrag der Firma FutureLAB AG zu untersuchen und einige Erfahrungen damit zu sammeln. Ist ein solches Funknetz einfach realisierbar? Erfüllt es alle die von der Columbia University versprochenen Eigenschaften? Ist es schnell genug? Wie läuft ein Zellenwechsel, genannt Handoff ab?

Um diese Punkte untersuchen zu können, bauten wir zuerst eine Versuchsanordnung im Labor auf. Es standen uns insgesamt sechs PC's im Towergehäuse und zwei Notebooks zur Verfügung. In der zweiten Phase vernetzten wir das E-Gebäude der Zürcher Fachhochschule Winterthur indem wir vier Nodes an geeigneten Orten aufstellten.

Nach dem Studium der Dokumentation hatten wir einen sehr positiven Eindruck von Cellular IP. Wir dachten, dass die Installation und Inbetriebnahme des Cellular IP Netzwerkes keine grossen Probleme mit sich bringen würde, da jeder Schritt sorgfältig dokumentiert ist. Wir einigten uns darauf, in der dritten Woche die Nodes an die Standorte zu verteilen, damit wir noch genügend Zeit hätten um Messungen durchzuführen und einige Erfahrungen mit Cellular IP zu sammeln. Leider rechneten wir nicht damit, dass die Installation der WaveLAN Karten ein Problem sein würde. Diese Tatsache brachte uns eine Woche in den Rückstand. Dann stellte sich heraus, dass die Cellular IP Software sowohl auf den Nodes wie auch auf den Mobile Hosts nicht fehlerslos lief. Die Fehler zu suchen und zu debuggen nahm die meiste Zeit in Anspruch. Wir konnten erst Ende der fünften Woche beginnen, die Nodes an die Standorte zu verteilen und einige Messungen durchzuführen.

Die Messversuche brachten zu Tage, dass eine relativ gute Signalqualität eine Voraussetzung ist, um Dienste wie FTP oder WWW nutzen zu können. Da das E-Gebäude der ZHW an einigen Stellen sehr massiv gebaut ist, genügen vier Nodes für die Vernetzung dieses Gebäudes nicht.

Wir kamen zum Schluss, dass Cellular IP in der jetzigen Form nicht zufriedenstellend ist und auf eine eventuelle nächste Version gewartet oder ein anderes Produkt in Betracht gezogen werden muss.

Winterthur, 18. Mai 2001

Marc Steiner

Thomas Wendel

2 Aufgabenstellung

2.1 Beschreibung

Wireless LANs werden immer mehr zum Allgemeingut. Interessant sind vor allem grossflächige Netze, die einen ganzen Campus, ein Firmenareal oder sogar eine ganze Stadt lückenlos abdecken.

Die Firma Ericsson hat zusammen mit der Columbia University ein spezielles Routing-Protokoll für Wireless LANs entwickelt, das sich "Cellular IP" nennt. Wie der Name sagt, ist es sehr stark an GSM-Netze angelehnt, basiert aber ausschliesslich auf IP. Mobile Benutzer sind in einem Zellenbereich angemeldet, wo sie im Ruhezustand auch erreicht werden können. Der Benutzer meldet sich nur, wenn er in einen neuen Bereich eintritt. Es besteht ein Cache für die letzte Route und den letzten Aufenthaltsort jedes Benutzers. Damit kann der Control-Overhead sehr klein gehalten werden. Weiter kann sehr schnell geroutet werden, so dass es möglich sein sollte, fast unterbruchsfrei aus einem fahrenden Auto heraus kommunizieren zu können.

Im Rahmen der Projektarbeit soll das E-Gebäude der ZHW mit einem WLAN auf der Basis des Cellular IP Verfahrens ausgerüstet werden. Die Verkabelung erfolgt provisorisch auf der Basis von optischen Gigabit-Ethernet Verbindungen. Sowohl die Basisstationen, wie auch die mobilen Hosts sollen unter Linux in Betrieb genommen werden. Es soll die Reichweite des Netzes und die Robustheit des Zellen-Handoffs untersucht werden. Weiter soll das bestehende Authentisierungsverfahren untersucht und eventuelle Verbesserungen vorgeschlagen werden.

Sollte sich das Versuchsnetz in der Erprobung bewähren, so dass es eine Alternative zu UMTS darstellen könnte, soll das Netz in einer weiteren Phase auf das ganze ZHW Areal und einen Teil der Stadt Winterthur ausgedehnt werden.

2.2 Aufgaben

- Einarbeitung in das "Cellular IP" Konzept. Installation der "Cellular IP" SW und der Linux WaveLAN-Treiber auf den Wireless Clients und Access Nodes. Konfiguration des Gateways. Austesten aller Funktion im Labor und anschliessende Installation der Infrastruktur im E-Gebäude.
- Untersuchung der Abdeckung und Reichweite. Einfluss von Störungen.
- Messung der Zeit für einen dynamischen Handoff beim Zellenwechsel.
- Austesten des Authentisierungsverfahrens und der "Mobile IP" Funktionalität.
- Kurze Installations- Bedienungsanleitung, Dokumentation der Resultate.
- Ausblick auf Verbesserungs- und Erweiterungsmöglichkeiten.

3 Einführung Cellular IP

3.1 Cellular Internet

Die meisten Versuche das Internet mobil zu machen basieren auf Adressübersetzung mittels Location Directories. So zum Beispiel auch Mobile IP. In diesem Protokoll werden die Pakete zu einem Mobile Host, welcher temporär eine Adresse zugewiesen bekommen hat, durch ganz normales IP-Routing befördert. Dadurch ist Mobile IP gut skalierbar und ermöglicht globale Mobilität. Der Nachteil ist jedoch, dass sich ein Mobile Host nicht allzu schnell fortbewegen kann. Ein reibungsloser Handoff von einer Zelle zur anderen wäre in diesem Fall nicht mehr gewährleistet, da nach jeder Migration in eine andere Zelle eine neue temporäre IP-Adresse bezogen werden muss.

Zellenartige Mobiltelefoniesysteme basieren auf einem anderen Konzept als Mobile IP. Anstatt in erster Linie globale Mobilität zu gewährleisten, sind solche Systeme, wie zum Beispiel GSM, darauf ausgelegt, einen reibungslosen Handoff innerhalb eines Netzes mit mehreren Zellen durchzuführen. Roaming zwischen solchen Netzen wird dann mittels eines skalierbarem Forwarding-Protokoll ermöglicht.

Allerdings kann die Anzahl der User derart ansteigen, dass ein schnelles Durchsuchen der Lookupdatenbank nicht mehr funktioniert. Dazu kommt noch, dass jedes Mobiltelefon nach einer Migration Informationen ans Netz sendet, was die Performance zusätzlich schwächt. Dies kann jedoch umgangen werden, wenn jedes Telefon jeweils nur hie und da, bei Übertritt in ein neues Netz oder wenn es in ein aktives Gespräch verwickelt ist, diese Informationen sendet. Die Position eines Mobiltelefons ist dem Netz so aber nur ungefähr bekannt. Soll ein Anruf auf ein inaktives Telefon erfolgen, muss das Telefon zuerst in einer limitierten Anzahl Zellen gesucht werden. Dies wird „Passive Connectivity“ genannt und ermöglicht dem Zellennetzwerk eine relativ grosse Anzahl User zu verwalten, ohne dass das Netz mit Managementinformationen überflutet wird.

Auf Zellen basierende Netze offerieren viele Eigenschaften, welche die Performance von zukünftigen Wireless-IP Netzen anhaben kann ohne dabei Einbussen der Flexibilität, Skalierbarkeit und Robustheit zu erleiden, welche für ein IP-Netz charakteristisch sind. Die fundamentalen Unterschiede eines Zellennetzes zu einem auf IP basierendem Netz macht eine einfache Umsetzung der „Zellentechnik“ auf IP allerdings schwierig. Telefonie in einem zellenbasiertem Netz setzt eine bestehende Verbindung voraus um kommunizieren zu können. Ein IP-Netz hingegen führt packetbasiertes Routing durch. Cellular Internet sollte Moblity-Management wie ein Zellennetz durchführen, sollte aber dennoch einfach, robust und flexibel wie ein herkömmliches IP-Netz sein. Das Protokoll sollte Pico-, Campus- oder aber auch Metropolitan-Area Netze handeln können.

3.2 Cellular IP Protokollübersicht

Im folgenden soll eine Übersicht der Merkmale und des Algorithmus des Protokolls gegeben werden.

3.2.1 Merkmale

Cellular IP beinhaltet die wesentlichen Eigenschaften für Mobilitätsmanagement, passive Netzverbindung und Handoff-Kontrolle, das Design baut aber im wesentlichen auf IP auf. Die Basisstation, Node genannt, ist die universelle Komponente des Cellular IP-Netzes. Sie dient als Wireless Access Point, routet IP-Pakete und beinhaltet auch die Zellenkontrollfunktionen, vergleichbar mit den Mobile Switching Centers (MSC) und den Base Station Controllern (BSC) in GSM-Netzen. Die Nodes bauen auf normalen IP-Forwarding Engines auf, jedoch wurde das IP-Routing durch Cellular IP-Routing und ein Location Management ersetzt. Das Cellular IP-Netzwerk ist via Gateway-Router mit dem Internet verbunden. Innerhalb des Cellular IP-Netzes werden die Mobile Hosts durch eine ihnen fix zugewiesene IP-Adresse identifiziert.

3.2.2 Routing

Der genaue Standort eines Mobile Host ist den einzelnen Nodes so wie auch dem Gateway nicht bekannt. Der Mobile Host ist dafür verantwortlich, dass er gefunden wird, indem er mittels Page- bzw. Route-Update Pakete dem Netz mitteilt, über welche Node er erreichbar ist. Uplink-Pakete werden Hop-für-Hop vom Mobile Host zum Gateway geroutet. Der Weg, welche diese Pakete nehmen, ist gegeben, da jede sich auf dem Weg befindliche Node die Pakete an ihr Uplink-Port weiterleitet. Jede Node merkt sich gleichzeitig, an welchem Interface sie ein Paket erhalten hat und von welchem Mobile Host es stammt und speichert diese Information im Route-Cache. Es entsteht sogenanntes Paging- bzw. Route-Cache Mapping. Downlink-Pakete werden anhand des Route-Cache vom Gateway zu den Mobile Hosts geroutet. Der Gateway und die Nodes brauchen dabei nicht zu wissen, wie die Topologie des Netzes aussieht.

Nach dem Prinzip der Passive Connectivity, wechselt ein Mobile Host in den Idle-Mode wenn er nichts mehr senden will und die Einträge in den Route-Caches der einzelnen Nodes laufen nach einer im System definierten Zeit ab. Wenn nun aber der Mobile Host trotzdem Daten erwartet, zum Beispiel UDP-Pakete, so muss er gelegentlich ein Route-Update Packet verschicken, um die Route zum Gateway aufrecht zu erhalten.

3.2.3 Handoff

Die Nodes eines Cellular IP Netzes senden periodisch ein Beacon-Signal, welches Informationen über das Netz enthält. Ein Mobile Hosts ermittelt anhand dieses Beacon, welche Node das qualitativ beste Signal hat und führt gegebenenfalls einen Handoff von der schwächeren zur stärkeren Node durch. Man unterscheidet zwischen drei verschiedenen Arten von Handoffs: Hard Handoff, semisoft Handoff und soft Handoff.

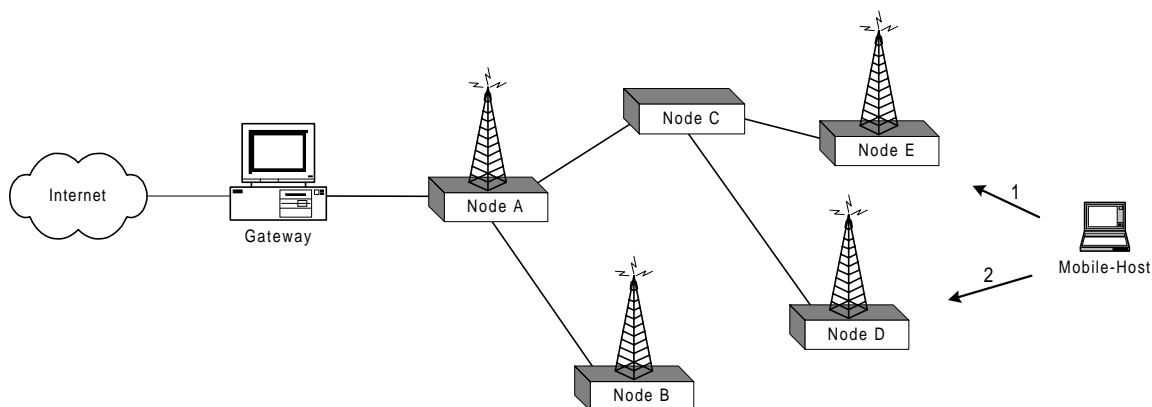


Abbildung 1 Handoff in einem Cellular IP Netz

Harter Handoff

Der Mobile Host empfängt Daten via Node E. Gleichzeitig bewegt er sich in Richtung Node D. Wenn der Mobile Host merkt, dass das Beacon-Signal von Node D stärker ist als das von Node E, sendet er ein Route-Update-Paket an Node D und teilt so dem Netz mit, dass er neu nun über diese Node erreichbar ist. Node D leitet dieses Route-Update-Paket seinerseits an Node C weiter. Der Route-Cache Eintrag in Node C wird ersetzt. Node C leitet nun alle für den Mobile Host bestimmten Pakete nicht mehr zu Node E sondern zu Node D. Node E merkt von all dem nichts und löscht den Route-Cache Eintrag vom Mobile Host nach einer spezifizierten Zeit.

Sollten Datenpakete unmittelbar vor dem Route-Update Paket bei Node C ankommen, werden diese noch zu Node E weitergeleitet. Da der Mobile Host mittlerweile nicht mehr auf Node E hört, sind diese Pakete verloren, was sich schlecht auf die Performance auswirkt.

Semisoft Handoff

Um diesen Verlust zu verhindern, sendet der Mobile Host bevor er die Node wechselt ein Route-Update Paket mit gesetztem Semisoft-Handoff Flag an Node D. Node C erkennt, sobald sie dieses Update-Paket erhält, dass der Mobile Host zu Node D wechseln will und sendet alle Datenpakete sowohl an Node E wie auch an Node D. Der Mobile Host kann nun den Wechsel vollziehen und sendet ein normales Route-Update Paket an Node D. Sobald Node C dieses erhält, weiss sie, dass der Mobile Host den Wechsel vollzogen hat und schickt die Daten nur noch zu Node D.

Soft Handoff

Ein Soft Handoff ist nur mit einem Mobile Host möglich, der auf zwei Nodes gleichzeitig „hören“ kann. Gleich wie beim harten Handoff sendet der Mobile Host ein Route-Update Paket an Node D. Mit dem einen Kanal wechselt der Mobile Host auch sofort auf diese neue Node, empfängt aber weiterhin so lange Pakete von Node E bis die ersten Datenpakete über Node D eintreffen.

3.2.4 Paging

Wie bereits erwähnt, wechselt ein Mobile Host vom aktiv in den idle Zustand, wenn er weder Daten empfangen noch senden will, um den Verkehr auf dem Netz möglichst klein zu halten und um Batterie zu sparen. Sollte der Gateway nun plötzlich Daten aus dem Internet erhalten, die für den Mobile Host bestimmt sind, muss dieser dennoch auffindbar sein. Zu diesem Zweck sendet er periodisch ein Page-Update Paket an die Node mit dem besten Signal. Ein Page-Update Paket wird auf die selbe Weise wie ein Route-Update Paket bis zum Gateway geschickt. Sämtliche auf dem Weg liegende Paging-Cache Einträge werden erneuert. Auch die Aufgabe des Paging-Cache ist ähnlich wie jene des Route-Cache: Das Auffinden eines Mobile Hosts im Netz. Der Unterschied besteht in der Gültigkeitsdauer der Einträge und den verschiedenen Arten, wie sie upgedated werden. Eine Übersicht bietet die folgende Tabelle:

	Paging-Cache	Routing-Cache
update durch	alle Uplink-Pakete (Daten-, Route- und Paging-Pakete)	Route-, und vom Mobile Host kommende Datenpakete
update bei	betreten einer neuen Paging-Area oder nach der Paging-update-time	Zellenwechsel oder nach der Route-update-time
Zweck	Routen von Downlink-Paketen falls kein Route-Cache Eintrag vorhanden ist	Routen von Downlink-Paketen

Tabelle 1 Paging- und Route-Cache

4 Realisierung

4.1 Topologien

4.1.1 Dedicated Net

Aufgrund der Idee von FututeLAB, die Basisstationen an Strassenlampen zu platzieren, war es naheliegend eine Topologie aufzubauen, in der mehrere Basisstationen hintereinander geschaltet sind. Da das Netz aber im Gebäude getestet werden sollte, haben wir uns für zwei „Stränge“ und somit für die folgende Topologie entschieden.

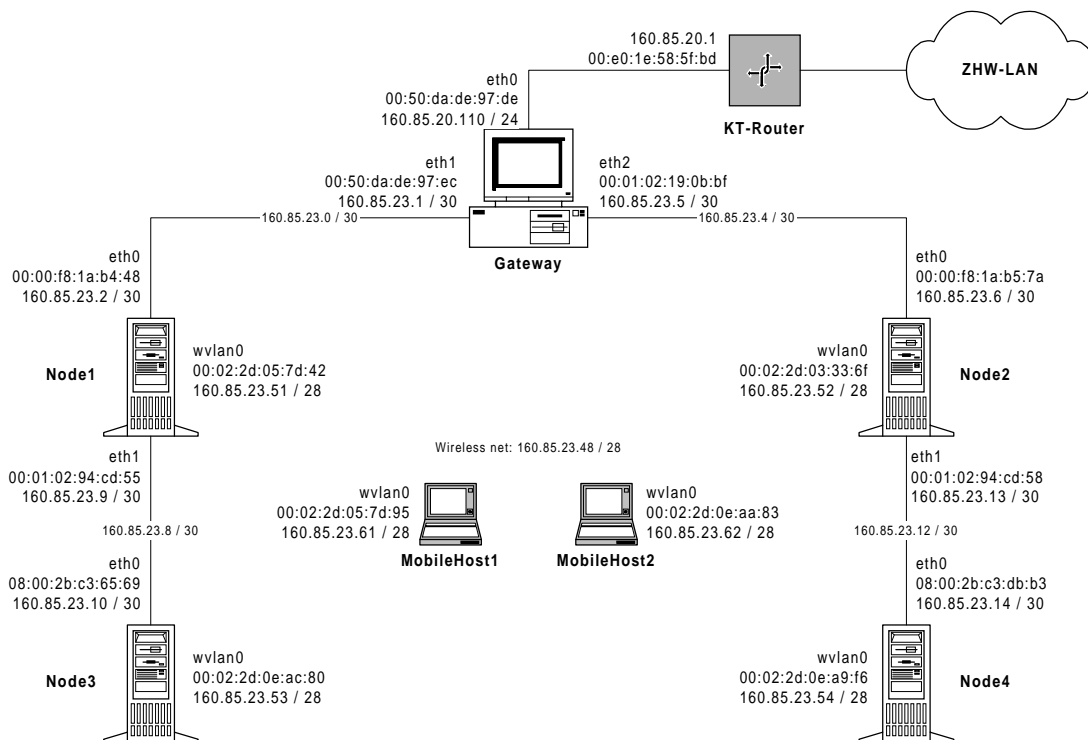


Abbildung 2 Dedicated Net

Damit eine saubere und einfache Konfiguration der Routetabelle möglich war, haben wir für jede Verbindung zwischen den Nodes ein eigenes Subnetz gewählt. Mit der Subnetzmaske 255.255.255.252 hatten wir genau vier Adressen (Netz-, zwei Host-, und Broadcastadresse) zur Verfügung. Nach aussen war unser Gateway über den KT-Router am ZHW-LAN angeschlossen. Im KT-Router haben wir einen Eintrag einfügen lassen, damit von aussen das ganze 23er-Netz (160.85.23.0/24) an unseren Gateway weitergeleitet wurde. So war es möglich, mit den Mobile Hosts ins Internet zu gelangen oder umgekehrt. Das heisst, Anfragen an die Mobile Hosts wurden über den KT-Router an unseren Gateway gesendet. Dieser wählte dann per Cellular IP die richtige Route zum gewünschten Mobile Host.

Wir testeten diese Topologie zuerst im Labor um sicher zu gehen, dass das Netz auf diese Weise funktionierte. Als es darum ging, die Nodes im E-Gebäude zu verteilen, gab es für uns drei Möglichkeiten:

1. Die Nodes weiterhin direkt zu verkabeln (evt. mit Glasfasern),
2. über das ZHW-LAN die Hops zu erzwingen (Simulation des dedicated Net in einer shared Topologie) oder
3. ein wirkliches shared Net zu realisieren.

Die erste Möglichkeit war diejenige, welche uns am sympathischsten war. Wir entschieden uns dennoch für die Simulation eines dedicated Net über ein shared Net. Dies wäre allenfalls eine einfache Lösung gewesen, mit einfachen, vorhandenen Mitteln das Verhalten einer dedicated Topologie zu testen.

4.1.2 Shared Net

Wir überlegten uns, alle benötigten Interfaces der Rechner (Nodes und Gateway) am ZHW-LAN anzuhängen und so die Simulation der Hops zu bewerkstelligen. Dies bedingte natürlich, dass wir auf diesen Interfaces IP-Adressen aus dem ZHW-LAN verwendeten. Wir bekamen also vom ZID für jede von uns benutzten Netzwerkkarten IP-Adressen die zum ZHW-LAN gehören und so per DHCP zugewiesen werden konnten. Da aber alle die Netzmaske des flachen ZHW-LANs besitzen, hätte sich das Routing mühsam gestaltet. Die einzige Möglichkeit wäre gewesen, die automatisch gesetzten Routen zu löschen und manuell durch andere zu ersetzen, welche direkt auf die Adressen der Nodes routen. Dies erschien uns nicht sehr komfortabel und auch wenig sinnvoll, da uns für die Situation im Gebäude ein shared Net wesentlich einfacher und geeigneter erschien. Wir beschlossen deshalb, für einen ersten Test ein shared Net zu installieren, dessen Topologie in Abbildung 3 ersichtlich ist. Abbildung 4 zeigt den Grundriss des E-Gebäudes mit den Standorten der Nodes im shared Net. Resultate der Messungen, so wie Erkenntnisse zu dieser Lösung sind in Kapitel 9 beschrieben.

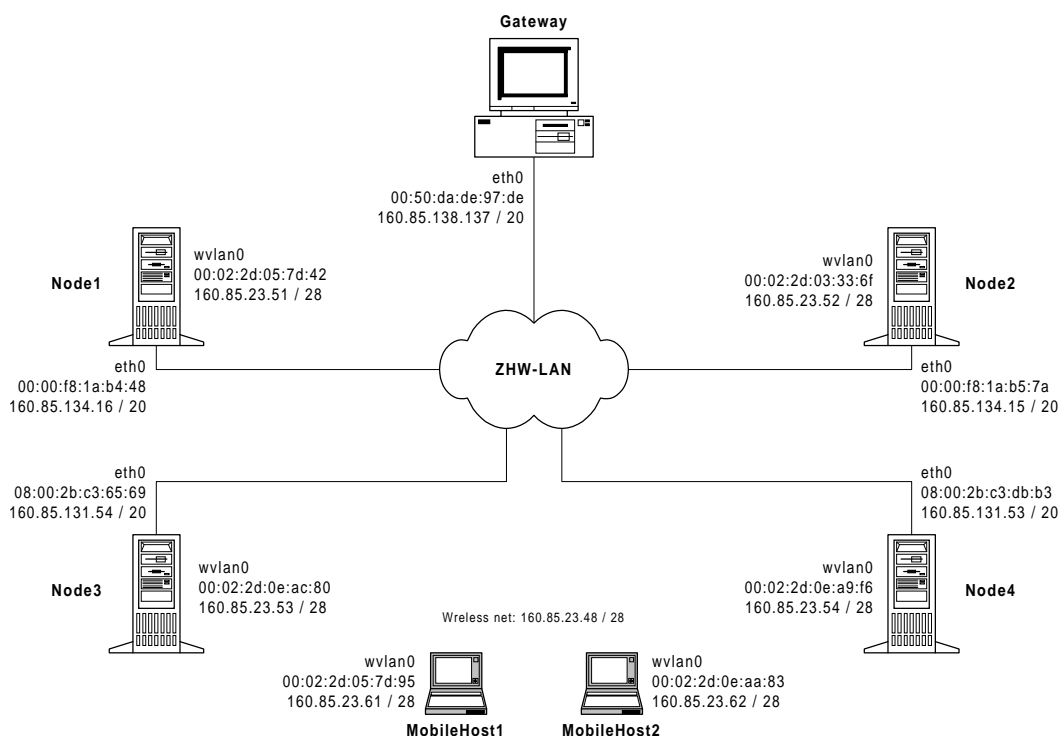


Abbildung 3 Shared Net

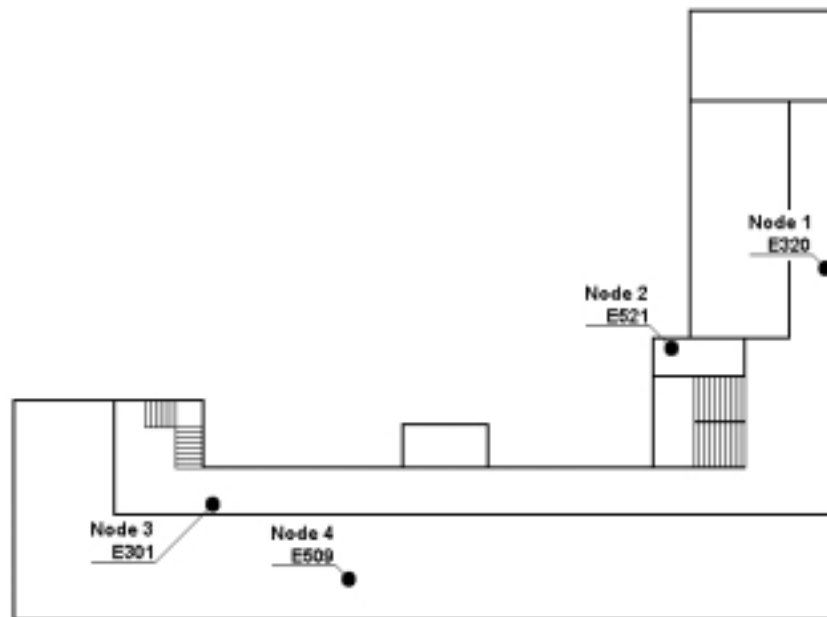


Abbildung 4 Verteilung der Nodes im shared Net

4.2 Installiertes Netzwerk

Da das Resultat im shared Net leider nicht sehr zufriedenstellend war, beschlossen wir möglichst schnell auf die ursprünglich angestrebte Topologie umzustellen. Wir suchten also neue geeignete Standorte, an denen wir die Möglichkeit hatten, die Verbindungen zwischen den Nodes direkt zu patchen. Die Standorte konnten wir bis auf einen mehr oder weniger belassen. Die neue Verteilung der Nodes ist in Abbildung 5 ersichtlich.

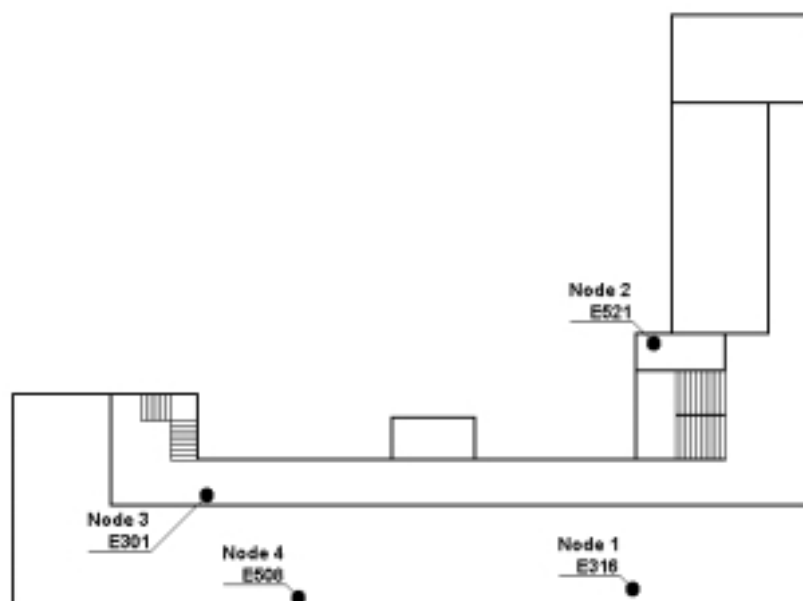


Abbildung 5 Verteilung der Nodes im dedicated Net

Es gab aber zwei Probleme zu beheben. Erstens gab es keine direkte Leitung zwischen dem zweiten (E300) und dem vierten (E500) Stockwerk. Aus diesem Grund zogen wir eine eigene Leitung vom E221 ins E521. Von diesen Räumen war es möglich alle Standorte zu erreichen. Das zweite Problem war demzufolge nicht schwierig zu eruieren. Die einzelnen Verbindungen zwischen den Nodes waren teilweise über 100m lang. Deshalb mussten wir noch vereinzelt Hub's zwischenschalten. Die folgende Grafik zeigt die gepatchten Verbindungen.

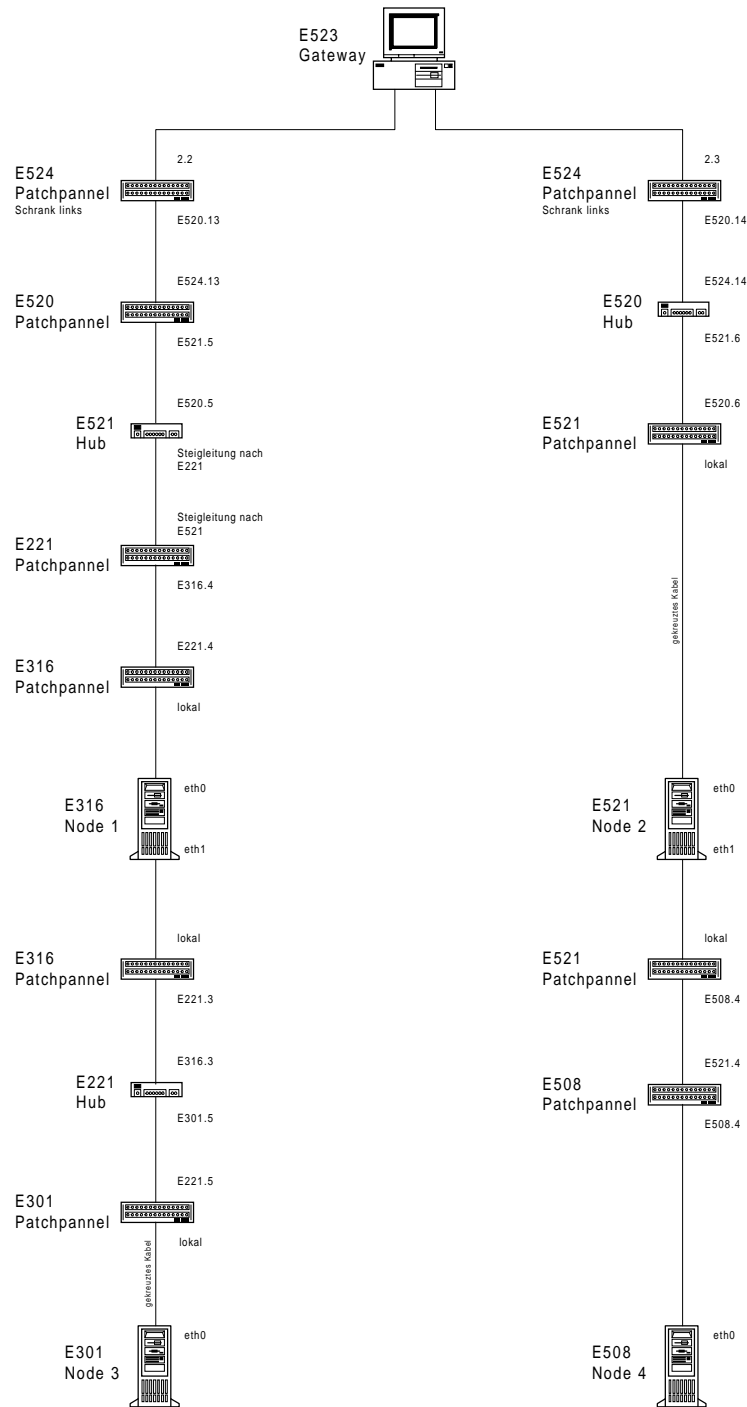


Abbildung 6 Plan der gepatchten Verbindungen

Es ist klar, dass diese Lösung für eine dauerhafte Installation unsinnig ist. Es war aber in unserer Situation die einzige realisierbare Möglichkeit, um die angestrebte Topologie mit den gewählten Standorten im E-Gebäude zu bewerkstelligen. Zudem ist dies auch nur eine temporäre Lösung für Testzwecke.

4.3 Node als Repeater

Die Firma FutureLAB brachte die Idee auf, einen PC, welcher nur eine WaveLAN Funkkarte als einziges Interface besitzt als Repeater einzusetzen. Dies kann Sinn machen, wenn zum Beispiel in einem Teil eines Gebäudes der Empfang sehr schlecht ist und keine Möglichkeit besteht, eine weitere Node per Kabel ins Cellular IP Netz einzuhängen.

4.3.1 Installation einer Repeaternode

Der PC wird genau gleich wie die anderen Nodes aufgesetzt (siehe Kapitel 5). Lediglich das Konfigurationsfile ist etwas anders. Die Cellular IP Software muss ein wenig überlistet werden, da sie für diesen Einsatz nicht vorgesehen ist.

Das Programm `cipnode` parst die Zeilen des Konfigurationsfile, in welchen die Neighbours angegeben werden, nach den Schlüsselwörtern `wire` und `wireless`. Findet es das Wort `wire`, liest es, nachdem es den Interfacenamen eingelesen hat, auch noch die IP-Adresse dieses Nachbarn. Beim Schlüsselwort `wireless` wird jedoch die Stelle, an der die IP-Adresse steht übersprungen. Dies macht nur einen Sinn, wenn man von der Überlegung ausgeht, dass man nicht genau weiss, welche IP's eventuelle Mobile Hosts haben werden. Sauberer wäre gewesen, wenn diese Stelle auch geprüft wird und man erst dann von einem Mobile Host als Nachbar ausgeht wenn sie leer ist.

Will man nun einen Uplink über eine WaveLAN Karte realisieren, muss man das Programm überlisten, indem man das Schlüsselwort `wire` an den Anfang dieses Tuples stellt. Nur so wird die IP-Adresse ausgewertet. Der Interfacename muss weiterhin `wvlanX` lauten.

4.3.2 Überlegungen zu einer Repeaternode

Die Möglichkeit eine Repeaternode ins Netz mit einzubinden haben wir ganz am Schluss der Projektarbeit noch ein wenig erprobt. Doch die Zeit um Messungen mit einem solchen Repeater durchzuführen hatten wir leider nicht mehr, wir machten jedoch ein paar Überlegungen dazu.

Wir erachten den Einsatz einer Repeaternode eher als eine temporäre Lösung, die schnellstmöglich durch eine mit fest installierter Node abgelöst werden sollte. Gründe die zu diesem Schluss führten sind folgende:

- Was geschieht, wenn sich in einem Teil des Gebäudes, der über einen Repeater abgedeckt ist, viele mobile Benutzer aufhalten? Unter Umständen kann die Bandbreite von 11 MBit/s nicht mehr ausreichen um eine genügend schnelle Verbindung zur nächsten kabelgebundenen Node zu gewährleisten.
- Der gewonnene Radius kann nicht besonders gross sein, da sich die Zellen der nächsten Node und dem Repeater zu einem gewissen Stück überschneiden müssen, um dem Repeater Anschluss ans Netz zu ermöglichen.
- Der Einsatz als Repeater ist in der Cellular IP Software eigentlich nicht vorgesehen. Dies zeigt sich auch in dem kleinen Überlistungstrick, welcher im Konfigurationsfile vorgenommen werden muss.

5 Installationsanleitung

5.1 Hardware und Treiber

5.1.1 Verwendete Hardware

Die ZHW stellte uns für die Projektarbeit folgende Hardware zur Verfügung:

- 2 Notebooks: Marke: MAXDATA
 Modell: NB Vision 340C Celeron / 500 MHz
- 5 PC's: Die PC's sind wie folgt ausgestattet. Die genaue Bezeichnung der eingebauten Netzwerkkarten ist weiter unten aufgeführt

Gateway: Pentium III / 550 MHz
 Netzwerkkarten: 3com

Node 1: Pentium II / 350 MHz
 Netzwerkkarten: 3com und Digital Equipment

Node 2: Pentium II / 350 MHz
 Netzwerkkarten: 3com und Digital Equipment

Node 3: Pentium III / 550 MHz
 Netzwerkkarte: Digital Equipment

Node 4: Pentium III / 550 MHz
 Netzwerkkarte: Digital Equipment

- Netzwerkkarten: Marke: 3com
 Modell: EtherLink 10/100 PCI Network Interface Card 3C905B-TX-NM

 Marke: Digital Equipment
 Modell: DE500-XA FAST ETHERWORKS PCI 10/100MB

Von der Firma FutureLAB wurden die PCMCIA-Adapter Cards und die WaveLAN-Karten zur Verfügung gestellt.

- PCMCIA-Adapter: 11Mbps Wireless LAN PCI/PC Adapter
- WaveLAN-Karten: Marke: Orinoco
 Modell: HS 11Mb PC Card EXT
 Lucent, 128RC4, Gold, World Card

Wir haben auf den WaveLAN Karten ein Firmwareupdate von Version 6.06 auf Version 6.16 durchgeführt. Dies muss allerdings unter Windows gemacht werden, da keine Software für Linux existiert.

5.1.2 PCMCIA-Adapter- und WaveLAN-Karten-Treiber

Bei SuSE Linux 7.1 Kernel 2.2.18 ist Version 3.1.22 des PCMCIA-Treibers bereits vorhanden. Ein Update auf Version 3.1.25 ist nicht empfehlenswert, da ansonsten die benötigten Wireless Tools nicht mehr richtig funktionieren.

Es gibt abgesehen von den unzähligen Versionen grundsätzlich drei mögliche Treiber für die WaveLAN Karten.

wvlan_cs	Lucent WaveLAN / IEEE 802.11 device driver Dies ist der benötigte Treiber für unsere Karten. Er ist bereits im PCMCIA-Treiber 3.1.22 enthalten.
wavelan_cs	AT&T GIS WaveLAN PCMCIA device driver Auch dieser ist im PCMCIA-Treiber 3.1.22 enthalten.
wavelan2_cs	Diese Version wird von ORiNOCO zur Verfügung gestellt. Sie unterstützt aber keine Wireless Extensions und ist somit uninteressant für uns, da die Wireless Tools nicht verwendet werden können.

5.1.3 Treibereinstellungen

In dem File `/etc/pcmcia/wireless.opts` können die gewünschten Einstellungen für die WaveLAN Karte gesetzt werden. Es ist möglich, verschiedene Profile zu erstellen, damit je nach eingesteckter Karte die richtigen Einstellungen übernommen werden. Genauere Beschreibungen für die Benutzung von solchen Profilen kann in diversen Linux-Büchern nachgelesen werden.

In der Datei sind bereits viele Einträge für diverse Karten enthalten. Es ist also wichtig, dass die MAC-Adresse der jeweils benutzten Karte im File nur einmal konfiguriert wird.

Der bestehende Eintrag für die Lucent WaveLAN Karten sieht folgendermassen aus:

```
# Lucent Wavelan IEEE
# Note : wvlan_cs driver only, and version 1.0.4+ for encryption
support
*,*,*,00:60:1D:*|*,*,*,00:02:2D:*)
  INFO="Wavelan IEEE example (Lucent default settings)"
  ESSID="Wavelan Network"
  MODE="Managed"
  RATE="auto"
  KEY="s:secu1"
# To set all four keys, use :
# KEY="s:secu1 [1] key s:secu2 [2] key s:secu3 [3] key s:secu4 [4]
key [1]"
# ;;
```

Da unsere WaveLAN Karten von Lucent sind und somit die MAC-Adresse 00:02:2d:*:*:* besitzen, wird dieser Eintrag für die Karten übernommen. Deshalb muss entweder dieser Eintrag mit den gewünschten Werten modifiziert, oder falls ein eigener Eintrag gemacht werden will die entsprechende MAC-Adresse in der ersten Zeile gelöscht werden. Es gibt in diesem File auch ein Generic example Eintrag, der als Beispiel für die eigene Konfiguration verwendet werden kann.

Unsere Konfiguration sieht wie folgt aus:

```
* , * , * , * )
INFO="Cellular IP"
ESSID="TEST"
MODE="Ad-Hoc"
CHANNEL="5"
RATE="11M"
```

Es ist wichtig, dass der MODE auf Ad-Hoc eingestellt wird. ESSID und CHANNEL müssen bei allen Rechner, die im Netz sind, gleich sein. Die restlichen Einträge können frei gewählt werden.

Falls das System beim Starten der WaveLAN Karten blockiert, ist dies wahrscheinlich ein Problem mit der Vergabe der IRQs. In `/var/log/messages` ist dies ersichtlich.

Um dieses Problem zu beheben müssen die IRQs fest gesetzt werden die vom Adapter und der WaveLAN Karte benutzt werden sollen.

- In `/etc/rc.config` oder mit YaST die Variable `PCMCIA_PCIC_OPTS="irq_list=10"` setzen.
- In `/etc/pcmcia/config.opts` folgende Zeile eintragen:
`module "wvlan_cs" opts "irq_list=3"`

Die ORiNOCO WaveLAN-Karten signalisieren beim starten mit zwei Tönen, ob sie richtig konfiguriert sind. Bei einem hohen gefolgt von einem tiefen Ton ist die Karte richtig erkannt, es ist jedoch noch keine IP-Adresse zugeteilt worden. Bei zwei hohen Tönen ist die Karte richtig konfiguriert.

5.2 Software

5.2.1 Betriebssystem

Die Columbia University installierte ihr Netzwerk auf Red Hat Linux Kernel 2.2.14. Wir entschieden uns für die Installation von SuSE Linux 7.1 Kernel 2.2.18, da wir auch schon frühere Praktika mit SuSE Linux 7.1 durchgeführt haben und mit YaST und anderen SuSE Konfigurations Tools gut zurecht kommen. Für die Mobile Hosts ist es notwendig, eine grafische Oberfläche zu installieren. Wir benötigten den KDE2-Windowmanager.

Um Cellular IP verwenden zu können, muss zusätzlich das Paket `libpcap` installiert werden. `libpcap` ist eine Bibliothek die von Netzwerk-Snifferprogrammen benutzt wird. Sie stellt diesen eine Schnittstelle zur Verfügung, um Pakete vom Netzwerk zu holen und zu analysieren. Wir benutzten jedoch nicht das von SuSE mitgelieferte `libpcap`-Paket, sondern die von der Columbia University empfohlene Red Hat Version `libpcap-0_4-19_i386.rpm`. Der Grund dafür war, dass der Verzeichnisbaum der erstellt wird, bei SuSE und Red Hat nicht genau gleich aufgebaut ist. Da die Cellular IP Software aber unter Red Hat geschrieben wurde, war es für uns einfacher gerade die Red Hat Version zu verwenden anstatt den ganzen Verzeichnisbaum oder alle nötigen Source-Files anzupassen.

5.2.2 Wireless Tools

Damit die Wireless Tools richtig benutzt werden können, muss folgendes gemacht werden:

- Die Datei `wireless_tools_20.tar.gz` in ein gewünschtes Verzeichnis kopieren und entpacken.
- Kompilieren (`Makefile`)
- Damit die drei Tools `iwconfig`, `iwspy` und `iwpriv` Systemweit aufrufbar sind, müssen die ausführbaren Files nach `/sbin` kopiert werden.
- Um die man-Pages verwenden zu können müssen diese nach `/usr/share/man/man8` kopiert werden.

Die Handhabung der Wireless Tools kann in den entsprechenden man-Pages nachgelesen werden.

5.2.3 Cellular IP v1.1

Die Datei `linux_cellularip.tar.gz` muss lediglich in ein gewünschtes Verzeichnis kopiert und dort entpackt werden. Auf der CD befinden sich im Verzeichnis `/Software/Cellular IP/modifizierte Files` die Source-Code Dateien, in denen wir Bugs behoben oder sonstige Änderungen am Sourcecode vorgenommen haben. Dies sind `cipnode.c` und `cipmobile.c`. Die Originalversion kann mit diesen überschrieben werden.

6 Konfigurationsanleitung

Bevor Cellular IP gestartet werden kann, muss das Wired-Network und Cellular IP richtig konfiguriert werden.

6.1 Wired-Network

Als erstes ist es wichtig, dass die IP-Adressen aller im Netz beteiligten Interfaces richtig zugewiesen werden. Weiter sind die Routingtabellen im Wired-Network so zu setzen, dass die Rechner ihre Nachbarn erreichen können. Dies ist bei einer sauberen Subnetierung automatisch der Fall. Die Defaultrouten müssen nicht unbedingt eingetragen werden, da später Cellular IP den Uplink-Nachbar, der als Defaultrouter angesehen werden kann, aus dem Konfigurationsfile liest. Bei den Mobile Hosts wird die im Moment aktive Basisstation von der Cellular IP Software automatisch als Defaultrouter in die Routingtabelle eingetragen. Deshalb wird dort kein Defaultrouter-Eintrag gemacht.

6.2 Cellular IP

Um Cellular IP starten zu können, müssen zuerst noch diverse Files den gewünschten Anforderungen und Gegebenheiten angepasst werden.

6.2.1 Anpassung des tcl/tk-Scripts

Die Cellular IP Software wird auf den Mobile Hosts mit dem tcl/tk-Script `cip` gestartet. Die Version und der Pfad von `wish` (windowing shell von tcl/tk) muss im `cip`-Script überprüft und allenfalls angepasst werden.

In unserem Fall benutzen wir `wish8.3`. Die erste Zeile des `cip`-Scripts sieht also folgendermassen aus:

```
#!/usr/bin/wish8.3 -f
```

6.2.2 Anpassung des Makefiles

Bevor die Cellular IP Software kompiliert wird, muss das Makefile so modifiziert werden, damit die gewünschten Features übersetzt und evt. Debuginformationen später angezeigt werden. Ein Feature ist eingeschaltet, wenn der Unterstrich am Wortende entfernt wird.

Wenn anstatt WaveLAN Karten Aironet Karten verwendet werden, muss also in der Zeile

```
$(CC) $(CFLAG) -c cipmobile.c -DWICACHE -DANCACHE_ -DDEBUG_ -DD_DEBUG_
```

der Unterstrich bei `-DANCACHE` (Aironet) entfernt und bei `-DWICACHE` (WaveLAN) gesetzt werden. Das selbe kann mit den Debuginformationen gemacht werden. Falls diese also erwünscht sind, muss der Unterstrich entfernt werden.

In der Zeile

```
$(CC) $(CFLAG) -c cipnode.c -DDEBUG_ -DMOBILE_IP_
```

können ebenfalls die Debuginformationen für die Nodes ein- oder ausgeschaltet werden. `-DMOBILE_IP` muss nur aktiviert werden, wenn Mobile IP verwendet werden möchte.

Zusätzlich kann die Funktion `-DFILTER` eingesetzt werden. Gemäss Source Code (`cipnode.c`) gibt es auf dem WirelessLAN manchmal sogenannte Strange-Packets. Diese werden mit dem gesetzten Filter unterdrückt. Der Filter prüft alle Pakete, ob die IP-Version gleich 4 ist. Falls dies nicht der Fall ist, wird das Paket nicht weiter verarbeitet.

Ein von uns selber geschriebener Filter kann durch einfügen von `-ARPFILTER` in derselben Zeile gesetzt werden. Dieser prüft die Pakete, ob es sich um eine ARP-Message oder nicht. Falls ja wird das Paket von Cellular IP nicht weiter berücksichtigt (siehe dazu Kapitel 8).

6.2.3 Anpassung der Konfigurationsfiles

Die von uns verwendeten Konfigurationsfiles sind im Anhang und auf der beiliegenden CD enthalten.

Gateway und Node

Erklärungen zu den Einträgen in den Konfigurationsfiles für Gateway und Node sind in der folgenden Tabelle ersichtlich.

	Gateway	Node
GW	YES	NO
default router's IP address	IP-Adresse des Default-Routers	-
neighbor, uplink direction	-	(Medium, Interfacename, Dest.IP)
leaf neighbours(s)	(Medium, Interfacename, Dest.IP falls Medium = wire)	(Medium, Interfacename, Dest.IP falls Medium = wire)
paging cache	YES/NO	YES/NO
semisoft	YES/NO	YES/NO
route-timeout	Gültigkeitsdauer der Routing-Cache-Einträge in Millisekunden	Gültigkeitsdauer der Routing-Cache-Einträge in Millisekunden
paging-timeout	Gültigkeitsdauer der Paging-Cache-Einträge in Millisekunden	Gültigkeitsdauer der Paging-Cache-Einträge in Millisekunden
GW IP address	IP-Adresse des Gateways (Interface Richtung Defaultrouter)	IP-Adresse des Gateways

Tabelle 2 Einträge im Konfigurationsfile für Gateway und Nodes

Mobile Host

Für die Mobile Hosts sind folgende Einstellungen nötig.

	Beschreibung
wireless interface	Interfacename der WaveLAN-Karte
mobile's IP address	IP-Adresse des Mobile Host
air interface name	-
route-update-time	Maximale Zeit zwischen Route-Update-Packets in Millisekunden
paging-update-time	Maximale Zeit zwischen Page-Update-Packets in Millisekunden
active-state-timeout	Zeit die der Mobile Host im aktiven Zustand verweilt, wenn keine Daten empfangen werden
handoff	Art des Handoffs (0=forced, 1=SNR based)

Tabelle 3 Einträge im Konfigurationsfile für Mobile Hosts

7 Bedienungsanleitung

Wenn die Cellular IP Software auf allen beteiligten Geräten installiert ist, wie es im vorherigen Kapitel beschrieben ist, kann sie gestartet werden. Allerdings muss man dazu als Super User beim System angemeldet sein, da das Programm den pcap-Filter verwenden kann. Es spielt keine Rolle, ob man zuerst den Gateway, die Nodes oder ein Mobile Host startet. Die Gateway- bzw. Node Software kann aus der normalen Linux Konsole gestartet werden. Die Mobile Host Software hingegen wird aus einem tcl/tk-Script gestartet und benötigt deshalb eine grafische Bedienungsfläche.

7.1 Starten des Gateway

1. In das Verzeichnis wechseln, in dem sich das Programm `cipnode` befindet. Das Programm vom Gateway ist das selbe wie das der Nodes. Die Unterscheidung, ob es sich um einen Gateway oder eine Node handelt geschieht mittels Konfigurationsfile.
2. Den Gateway mit folgendem Befehl starten:

```
./cipnode <Gateway-Konfigurationsfile>
```

Falls kein Konfigurationsfile angegeben wird, verwendet die Software defaultmässig `cipnode.conf`. Dies gilt auch für das Starten der Nodes. Bei einem Mobile Host wird `cipmobile.conf` verwendet.

7.2 Starten der Node

1. In das Verzeichnis wechseln, in dem sich das Programm `cipnode` befindet.
2. Die Node mit folgendem Befehl starten:

```
./cipnode <Node-Konfigurationsfile>
```

Wird die Node von einem anderen PC mittels `telnet` oder `ssh` gestartet, sollte man das Programm mit dem Befehl `nohup` im Hintergrund laufen lassen. Der Grund dafür ist, dass diese zusätzliche Verbindung ebenfalls Verkehr auf dem Netz verursacht, welcher vom Gateway natürlich angezeigt wird. Dies kann zu reichlich Verwirrung führen.

```
nohup ./cipnode <Node-Konfigurationsfile> &
```

7.3 Starten der Mobile Hosts

1. Ein Konsolenfenster öffnen und in das Verzeichnis wechseln, in dem sich das `cip`-Script befindet.
2. Den Mobile-Host mit folgendem Befehl starten:

```
./cip <Mobile Host-Konfigurationsfile>
```

Folgendes Fenster sollte nach ca. 10 Sekunden erscheinen:

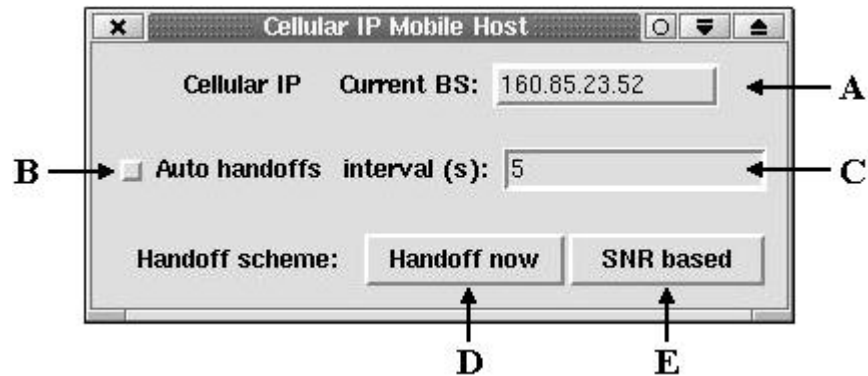


Abbildung 7 Cellular IP Mobile Host GUI

- A: In diesem Feld wird die IP-Adresse der Basisstation angezeigt, an welche sich der Mobile Host angemeldet hat.
- B: Ist diese Checkbox angewählt, führt der Mobile Host periodisch einen Handoff durch, ungeachtet ob die aktuelle Node zur Zeit das stärkste Signal hat.
- C: In diesem Feld kann das Zeitintervall angegeben werden, in welchem der automatische Handoff stattfindet.
- D: Ein Handoff kann auch vom Benutzer mit dieser Taste erzwungen werden.
- E: Nach einem automatischen oder vom Benutzer erzwungenen Handoff kann mit diesem Button wieder auf SNR-basierter Handoff geschaltet werden.

Ein automatischer bzw. ein vom Benutzer ausgelöster Handoff kann nur erfolgen, wenn der Mobile Host bereits mehrere Nodes in seiner Liste hat. Sollte nur ein einziger Node vorhanden sein, wird dies dem Benutzer mitgeteilt.

8 Probleme mit der Software

Wir hatten verschiedene Probleme mit der Cellular IP Software. Diese werden inklusive unseren Lösungen, auf den folgenden Seiten beschrieben.

8.1 cipnode.c

8.1.1 ARP is no IP

Als wir den Gateway und die Nodes starteten, hatten wir teilweise Pakete, in denen die Source und Destination IP-Adresse nicht richtig ausgegeben wurde. Hier ein Beispiel:

```
+++++++ PACKET RECEIVED!!
*****
packet from wvlan0, size:60
Src MAC: 00 E0 1E 58 5F BD
Dst MAC: 00 50 DA DE 97 DE
src IP: 95.189.160.85      dst IP: 20.1.0.80      type: Data
```

Auffällig war auf den ersten Blick, dass die IP-Adresse 160.85.20.1 die zur MAC 00 E0 1E 58 5F BD gehört und demzufolge Source-Adresse ist, um zwei Byte verschoben war. Wir wussten bis dahin jedoch noch nicht, wo dieser Fehler passiert ist. Mit der Tatsache, dass einige Pakete richtig dargestellt wurden und wir nach langem durchsuchen des Source Codes keinen Fehler bezüglich ein- oder auslesen der Pakete gefunden haben, machten wir uns dran, die Pakete vollständig auszugeben und zu analysieren. So konnten wir mit Hilfe von Herrn Steffen die Ursache lokalisieren.

Diese Pakete waren keine IP-, sondern ARP-Pakete. Da nur IP-Pakete erwartet werden, wurde bei der Programmierung ein fester Offset zum Auslesen der IP-Adressen verwendet. Bei einer ARP-Message zeigt dieser Offset mitten in die src-MAC, was natürlich zu einem „falschen“ Output führt. In der folgenden Grafik ist dies dargestellt.

```
src:MAC 00:E0:1E:58:5F:BD      IP 160.85.20.1
dst:MAC 00:50:DA:DE:97:DE      IP 160.85.20.110
```

Ethernet-Header (14 Bytes)

00	50	DA	DE	97	DE	00	E0	1E	58	5F	BD	08	06
dst MAC						src MAC						ARP	

ARP-Message (28 Bytes)

00	01	08	00	06	04	00	02	00	E0	1E	58	5F	BD	A0	55	14	01	00	50	DA	DE	97	DE	A0	55	14	6E
Ether		IP		LM		LP		Reply		src MAC						src IP		dst MAC						dst IP			

LM: Lenght of MAC-address

LP: Lenght of IP-address

IP-Header

V	TS	L	FI	FC	TL	P	C	src IP		dst IP		options/padding			
---	----	---	----	----	----	---	---	--------	--	--------	--	-----------------	--	--	--

V: IP Version + Header Lenght

TS: Type of Service

L: Total Lenght

FI: Fragment Identification

FC: Fragment Control

TL: Time to Live

P: Protocol

C: Checksum

Abbildung 8 Vergleich von ARP-Message und IP-Header

Der Fehler entpuppte sich also als nicht berücksichtigter Fall in der Software. Wahrscheinlich wurden diese Pakete von der Columbia University als sogenannte „strange packets“ angesehen und somit auch per FILTER eliminiert. (siehe Kapitel 6.2.2).

Diese Pakete hätten theoretisch nicht gestört. Die Cellular IP Software behandelt diese Pakete folgendermassen:

Kommt ein solches ARP-Paket von oben, schickt zum Beispiel der Gateway das Paket an alle seine Leaf Neighbours weiter, da die IP-Adresse in seinem Cache nicht vorhanden ist.

Kommt ein solches Paket von unten, wird das Paket vom Gateway aus dem gleichen Grund über eth0 an den Router weitergeleitet.

Da Cellular IP diese Pakete aber nicht behandeln muss, schrieben wir unseren eigenen Filter, der die Pakete überprüft, ob es sich um ein ARP-Paket handelt, um somit diese Unschönheit zu bereinigen. Folgenden Programmcode haben wir eingefügt:

```
#ifndef ARPFILTER
/* filter ARP-Messages out */
  if (mynode->pkt[13] == 06) {      /* Byte 13 of ethernet header
defines                                     type of packet) */
#ifdef DEBUG
  printf("It is ARP-Message! Filtered out!\n");
#endif
  return;
}
#endif
```

8.1.2 Beacon-Signal

Nachdem das ARP-Problem behoben war, versuchten wir die Mobile Hosts zu starten. Es wurde aber leider keine Basisstation im GUI eingetragen. Eine erste Vermutung, dass wir nicht überall auf dem gleichen Kanal arbeiteten, erwies sich als Fehlschlag. Weiter vermuteten wir, dass das Beacon-Signal nicht richtig gesendet oder empfangen wurde. Also begannen wir dieses zu analysieren. Auch dies war nicht die Ursache des Problems. Allerdings fanden wir einen weiteren Bug, der aber nicht weiter tragisch war.

Im Beacon-Signal befindet sich ab Byte 41 der Name der Station. Beim Einfügen von anschliessenden Leerzeichen wurde jedoch das letzte Byte des Stationsnamens überschrieben. Der Stationsname wird aber auf diesem Stand der Software nie aus dem Beacon-Signal ausgelesen und ist deshalb auch (noch) nicht relevant.

8.2 cipmobile.c

Der Grund warum die Software noch nicht funktionierte, lag beim Mobile Host.

Durch Überwachung der Prozesse bzw. Threads beobachteten wir, dass nach kurzer Zeit die Kindprozesse abstürzten respektive zu Zombies wurden. Mit Hilfe des GNU Debugger `gdb` fanden wir die Ursache.

Im folgenden werden die für den Fehler verantwortlichen Code-Zeilen aus der Funktion `thread_handle_beacon()` aufgezeigt und beschrieben.

```
char *arg, temp[20];

strcpy(arg, "ether host ");

sprintf(temp, "%02x:% 02x:% 02x:% 02x:% 02x:% 02x", \
        beacon_dst[0], beacon_dst[1], beacon_dst[2], \
        beacon_dst[3], beacon_dst[4], beacon_dst[5]);

strcat(arg, temp);
```

Für den Zeiger `*arg` wird kein Speicher reserviert. Dies führt zu einem Segmentation Fault und somit zum Absturz.

Unserer Meinung nach ist diese Lösung ohnehin sehr umständlich. Wir haben deshalb den String `"ether host "` direkt in den `sprintf`-Befehl hineingenommen und deshalb die Grösse der Variable `temp` auf 29 geändert. Anschliessend wird `arg` gleich `temp` gesetzt. Die Zeilen sehen dann wie folgt aus:

```
char *arg, temp[29];

sprintf(temp, "ether host %02x:% 02x:% 02x:% 02x:% 02x:% 02x", \
        beacon_dst[0], beacon_dst[1], beacon_dst[2], \
        beacon_dst[3], beacon_dst[4], beacon_dst[5]);

arg = temp;
```

Uns ist klar, dass auch dies nicht die beste Lösung ist. Da es aber nicht unsere Aufgabe war, den Source-Code zu reviewen und wir möglichst schnell damit beginnen wollten, Cellular IP in Betrieb zu nehmen, einigten wir uns darauf, diesen „Bugfix“ so zu belassen.

8.3 Sonstige Mängel

Es sind uns noch weitere Mängel aufgefallen, die für die Weiterentwicklung des Programms berücksichtigt werden sollten. Wir sind diesen Mängeln jedoch nicht weiter auf den Grund gegangen, haben uns aber eine Mögliche Lösung des Problems überlegt.

- Basisstationen, die bereits in der Liste eingefügt und sogar auch schon aktiv sind, werden zum Teil nochmals in die Liste aufgenommen.
Dies könnte behoben werden, indem vor dem Einfügen zuerst geprüft wird, ob die Basisstation schon in der Liste ist oder nicht.
- Falls der Mobile Host den Empfangsbereich verlässt, wird der Benutzer auf keine Weise gewarnt.
Hier könnte man bei ausbleiben des Beacon-Signals nach einer gewissen Zeit, eine Warnmeldung (zum Beispiel in Form einer Message-Box) ausgeben.
- Die Cellular IP Software kann nur gestartet werden, wenn man als Super User angemeldet ist. Das kommt daher, dass pcap nur als Super User geöffnet werden kann. Für eine spätere Lösung wäre dies sicherlich nicht akzeptabel.
Um dieses Problem zu beheben müsste man wahrscheinlich einen anderen Ansatz für die ganze Software in Betracht ziehen.
- Das cip-Script stürzt manchmal beim Starten ab. Ursachen für dieses Verhalten haben wir keine gefunden.

9 Messungen

9.1 Shared Topologie

Im shared Net lief Cellular IP sehr unstabil. Folgende Beobachtungen haben wir gemacht:

- Daten-Pakete die von einem Mobile Host an den Gateway gesendet wurden kamen nur sehr selten an.
- Der Gateway erkannte nicht, wann ein Paket für ihn selber bestimmt war. Das heisst, er leitete alle Pakete weiter an seinen im Konfigurationsfile eingetragenen Defaultrouter.

Den Gründen für diese Fehlverhalten sind wir nicht weiter nachgegangen, da wir die verbleibende Zeit dafür aufwenden wollten, die dedicated Topologie im E-Gebäude zu realisieren und zu testen.

Zudem war es in der shared Topologie nicht möglich, ins Internet zu gelangen, um so das Verhalten beim surfen oder bei einem ftp download zu beobachten.

9.1.1 Abdeckung

Um trotzdem etwas über die Abdeckung zu erfahren, gingen wir durch das Gebäude und notierten uns, welche Nodes an welchen Orten empfangen werden konnten und welche als aktive Basisstation eingetragen wurde.

Beim Testen der Abdeckung haben wir festgestellt, dass an bestimmten Orten nicht immer dieselben Nodes als momentane Basisstation verwendet wurden. Je nach dem, von welcher Seite her der Ort angegangen wurde, war eine andere Station aktiv. Sogar wenn der Mobile Host direkt vor eine im Moment nicht aktive Node gestellt wurde, behielt der Mobile Host die weiter entfernte Node als aktive Basisstation eingetragen.

In der folgenden Tabelle ist ersichtlich auf welchen Stockwerken, welche Nodes

- ✓ ✓ aktiv als Basisstation verwendet wurden.
- ✓ gefunden, aber nicht (oder nur teilweise z.B. wenn Host am Fenster) aktiv wurden.
- ✗ nicht erreichbar waren.

Stockwerk	Node 1	Node 2	Node 3	Node 4
E200	✓ ✓	✗	✓	✗
E300	✓ ✓	✓	✓ ✓	✗
E400	✓	✓ ✓	✓ ✓	✗
E500	✗	✓ ✓	✓	✓ ✓
E600	✗	✓ ✓	✓	✓ ✓

Tabelle 4 Abdeckung im E-Gebäude mit shared Net

Bei Betrachtung der Tabelle könnte man meinen, das Gebäude sei genügend abgedeckt (teilweise sogar redundant). Es wird jedoch bei dieser Messung keine Aussage darüber gemacht, an welchem Ort auf dem Stockwerk welche Nodes erreichbar sind. Es ist also Möglich, dass im hinteren Teil eines Geschosses zwei Nodes aktiv sein können, während im vorderen keine Basisstation ansprechbar ist. Wir können aber daraus schliessen, dass die mit ✗ bezeichneten Nodes in den entsprechenden Stockwerken mit Sicherheit nicht erreichbar sind.

9.2 Dedicated Topologie

9.2.1 Abdeckung

Unser Test sah in diesem Fall folgendermassen aus:

Wir teilten alle Stockwerke in die aus der Abbildung 8 ersichtlichen Sektoren ein. Da wir nicht in alle Zimmer Zutritt hatten, haben wir uns für die Messungen nur auf die effektiv eingezeichneten Sektoren konzentriert.

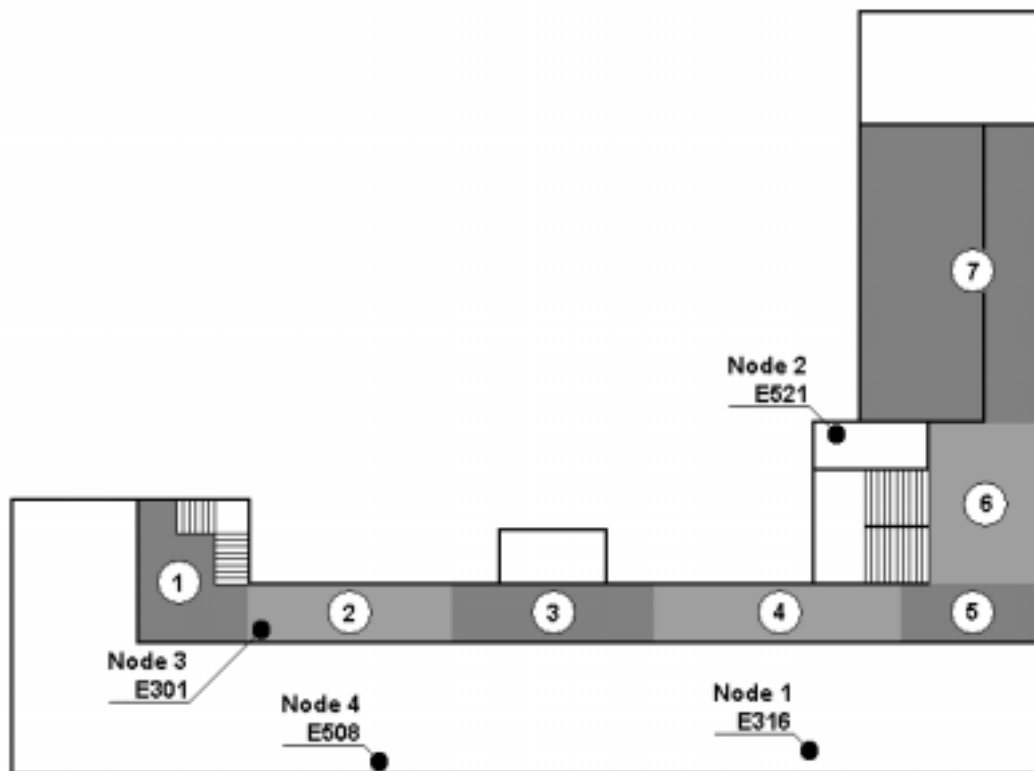


Abbildung 9 Aufteilung der Stockwerke in Sektoren

Wir messen mit dem Tool *iwspy* innerhalb jedes Sektors die Qualität der Luftverbindung zu den Nodes, die im momentanen Sektor als aktive Basisstation gewählt werden konnten. Da die Qualität sehr empfindlich reagiert, haben wir innerhalb eines Sektors zehn Werte gemessen und anschliessend die durchschnittliche Qualität berechnet. Die Tabelle mit den Messwerten ist im Anhang ersichtlich.

Die Qualität ist ein Integerwert. In der Praxis liegt dieser zwischen 0 und ca. 85. Ab einer Qualität von etwa 10 konnten wir auf das Internet zugreifen. Diese Qualität war aber an sehr wenigen Orten im Gebäude erreicht. Wir müssen daraus schliessen, dass vier Nodes für eine komplette Abdeckung des E-Gebäudes nicht ausreichend sind.

Im Folgenden werden die Messresultate in Diagrammen dargestellt. Pro Stockwerk existiert ein Diagramm. In diesen sind jeweils auf der x-Achse die Sektoren eingetragen. In jedem Sektor ist die Qualität für alle Nodes eingetragen.

Es kann davon ausgegangen werden, dass die Verbindung im jeweiligen Sektor in Ordnung ist, falls zu einer Basisstation eine Qualität von mindestens 10 gemessen wird.

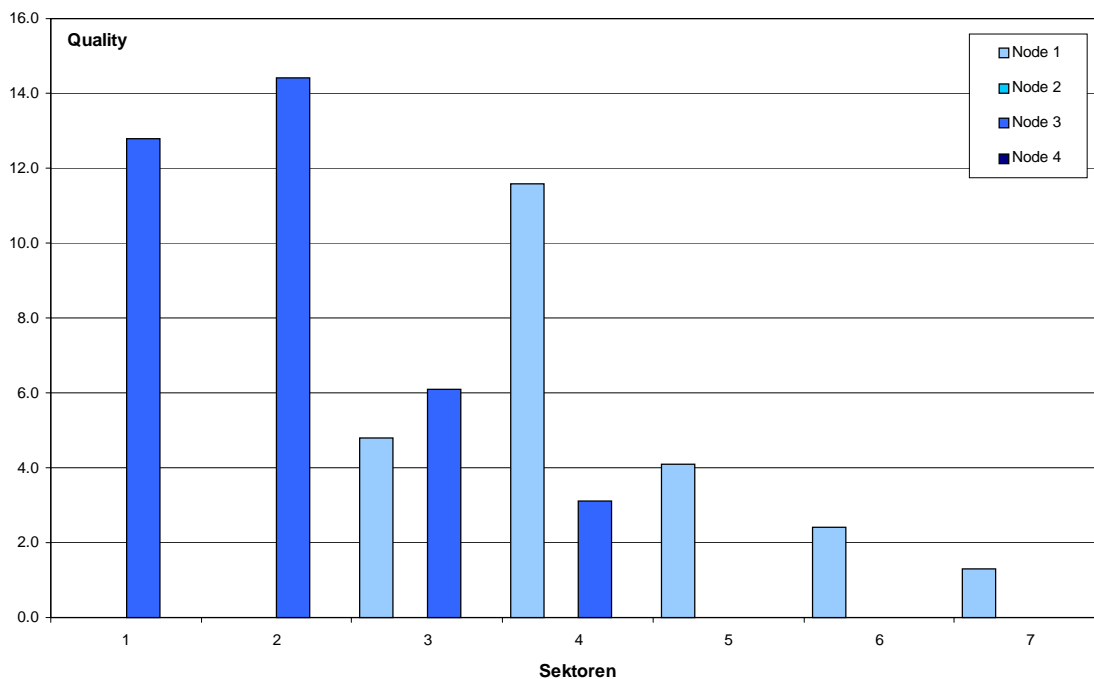


Diagramm 1 Erreichbare Nodes im E200

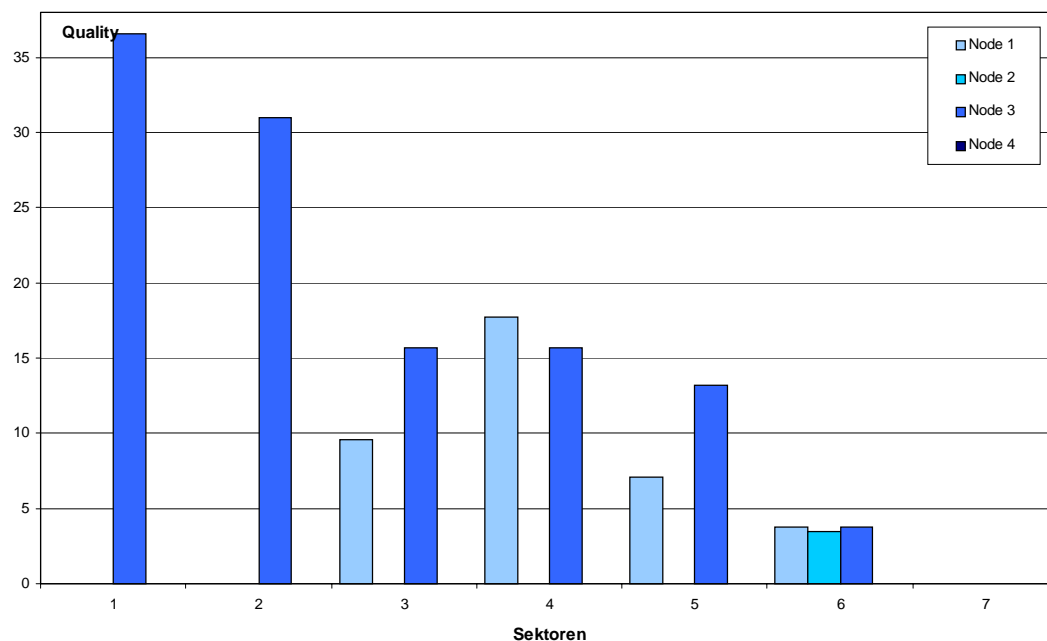


Diagramm 2 Erreichbare Nodes im E300

Stockwerk E300 ist das am besten abgedeckte Geschoss. In Sektor 7 (ZID) ist keine Basisstation mehr erreichbar. Das Signal ist wie aus dem Diagramm ersichtlich schon in Sektor 6 sehr schwach. Die Glastür zum ZID ist für dieses Signal ein zu grosses Hindernis.

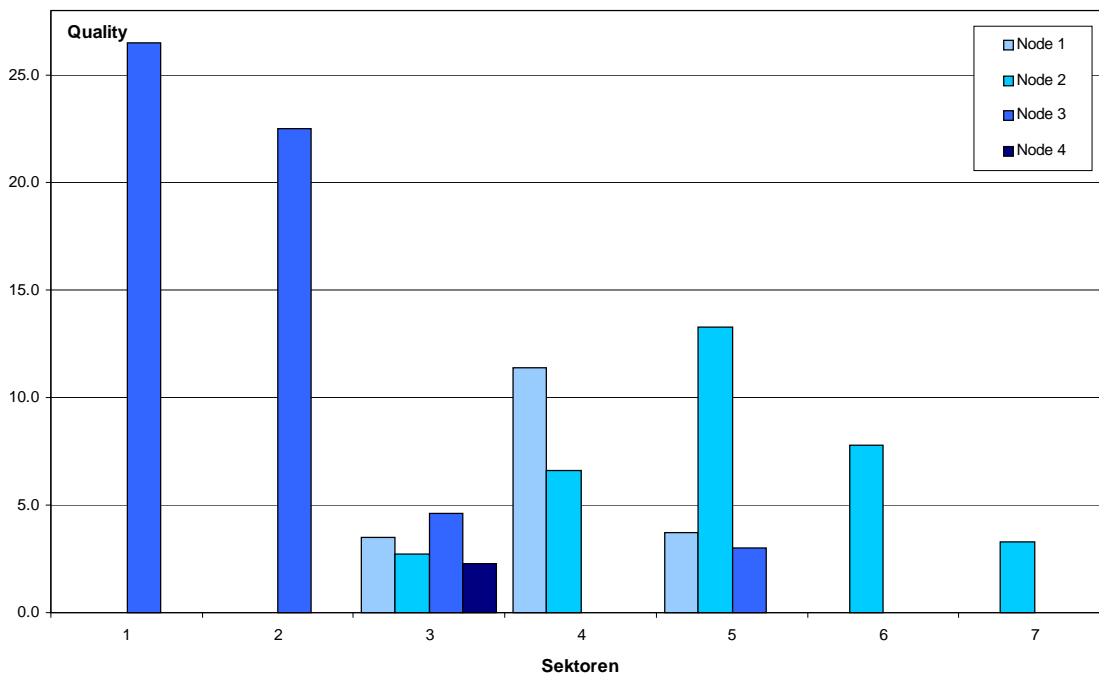


Diagramm 3 Erreichbare Nodes im E400

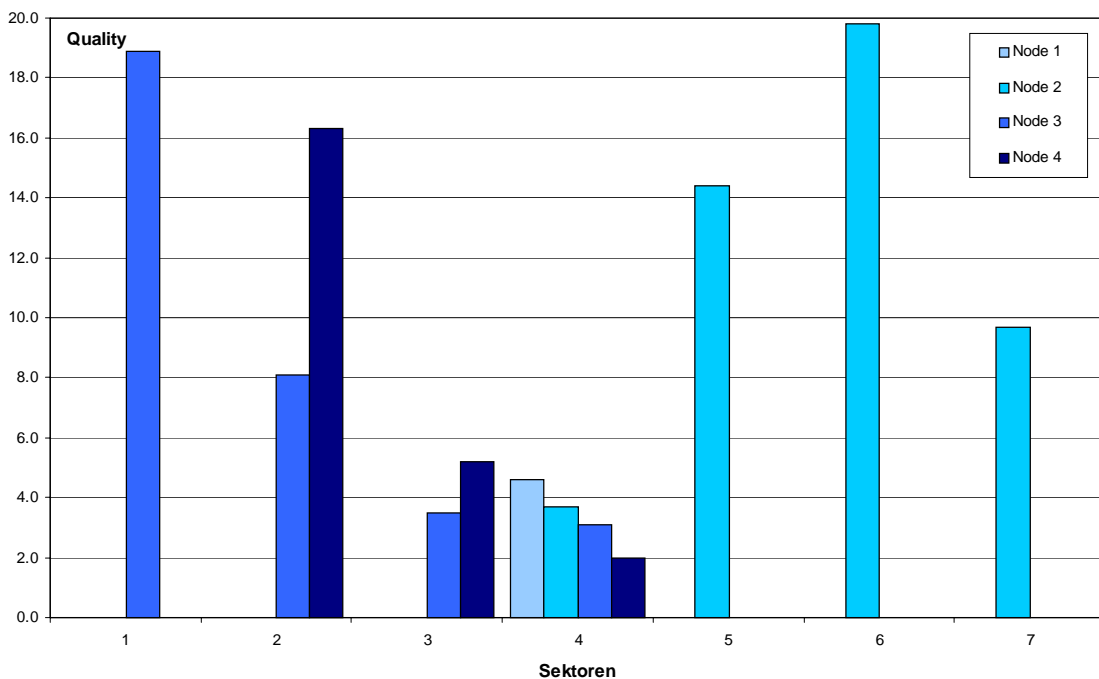


Diagramm 4 Erreichbare Nodes im E500

Im Sektor 4 auf Stockwerk E500 wäre es eventuell möglich, den Empfang zu verbessern, indem Node 4 näher beim Gang platziert würde. Sektor 7 ist ein Grenzfall. Es kommt sehr darauf an, wo genau der Mobile Host steht. Teilweise ist eine Verbindung zu Node 2 möglich.

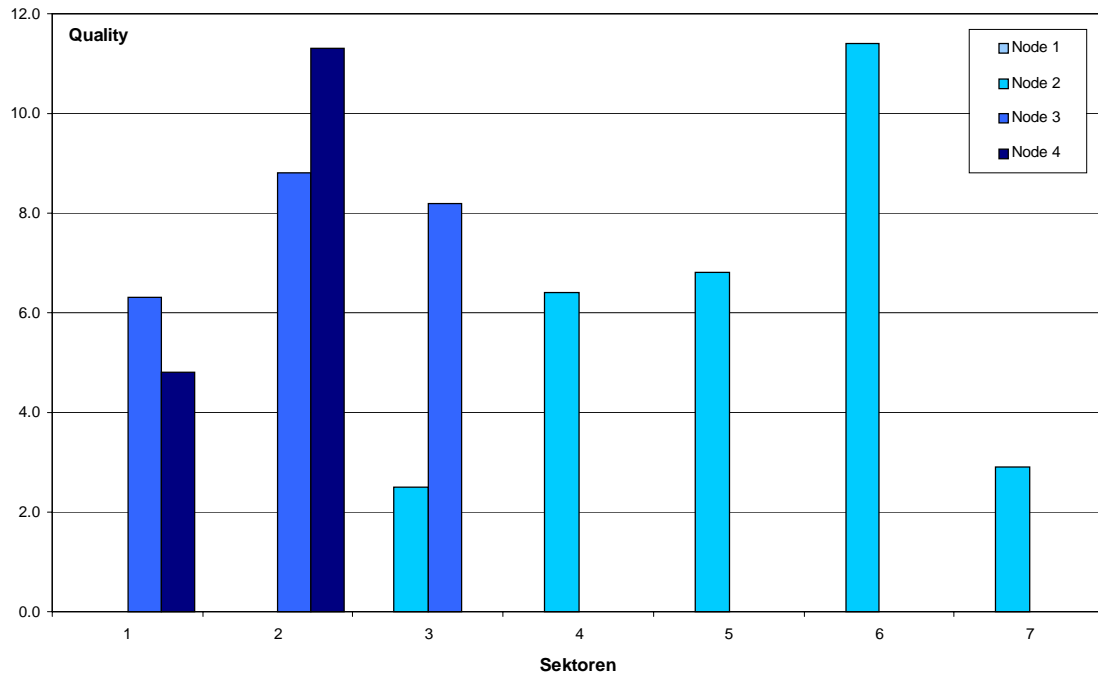


Diagramm 5 Erreichbare Nodes im E600

10. Projektverlauf

Der endgültige Ist-Zeitplan weicht schon nach den ersten paar Tagen drastisch von dem zu Beginn der Projektarbeit aufgestellten Soll-Zeitplan ab. Wir hatten zum Beispiel nicht damit gerechnet, dass das Installieren der PCMCIA Adapter oder der Cellular IP Software grössere Probleme verursachen würde. Die Cellular IP Software wies einige Fehler auf und das Debuggen des relativ unsauberen und nicht gut kommentierten Source-Code nahm viel Zeit in Anspruch. Auch haben sich einige Aufgabenpunkte im Laufe der Zeit geändert, neue wie das Fehlersuchen kamen hinzu. Wir machten Tests mit zwei verschiedenen Topologien anstatt nur mit einer. Durch diese Verzögerungen kamen die Messungen ein wenig zu kurz.

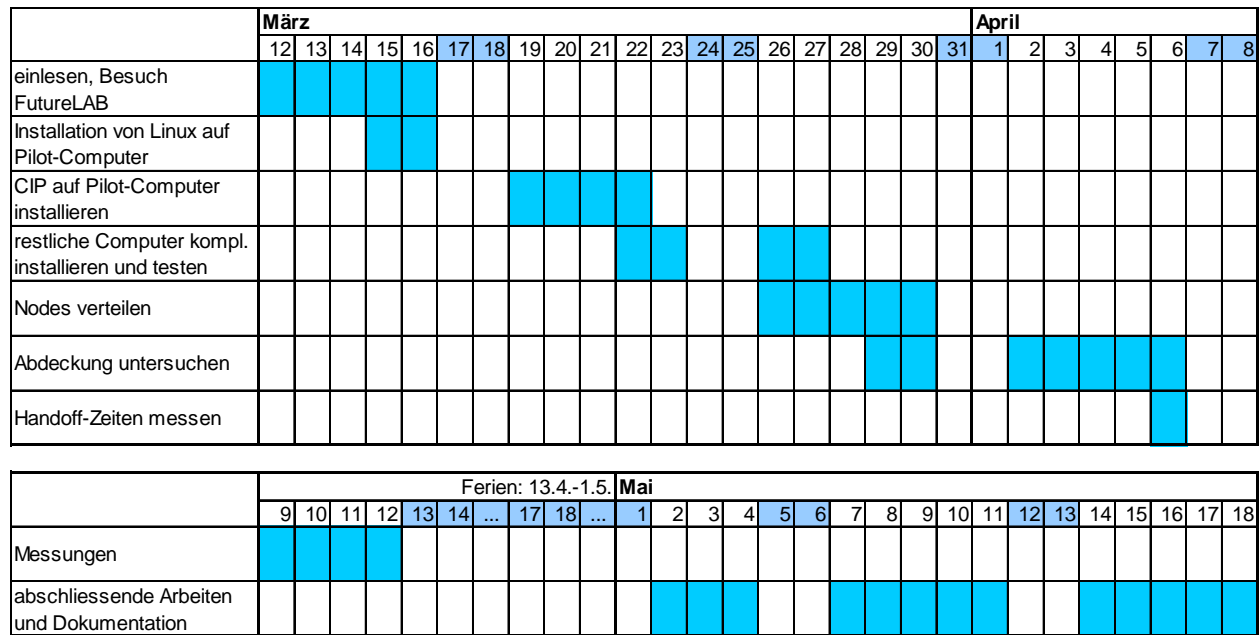


Abbildung 10 Soll-Zeitplan

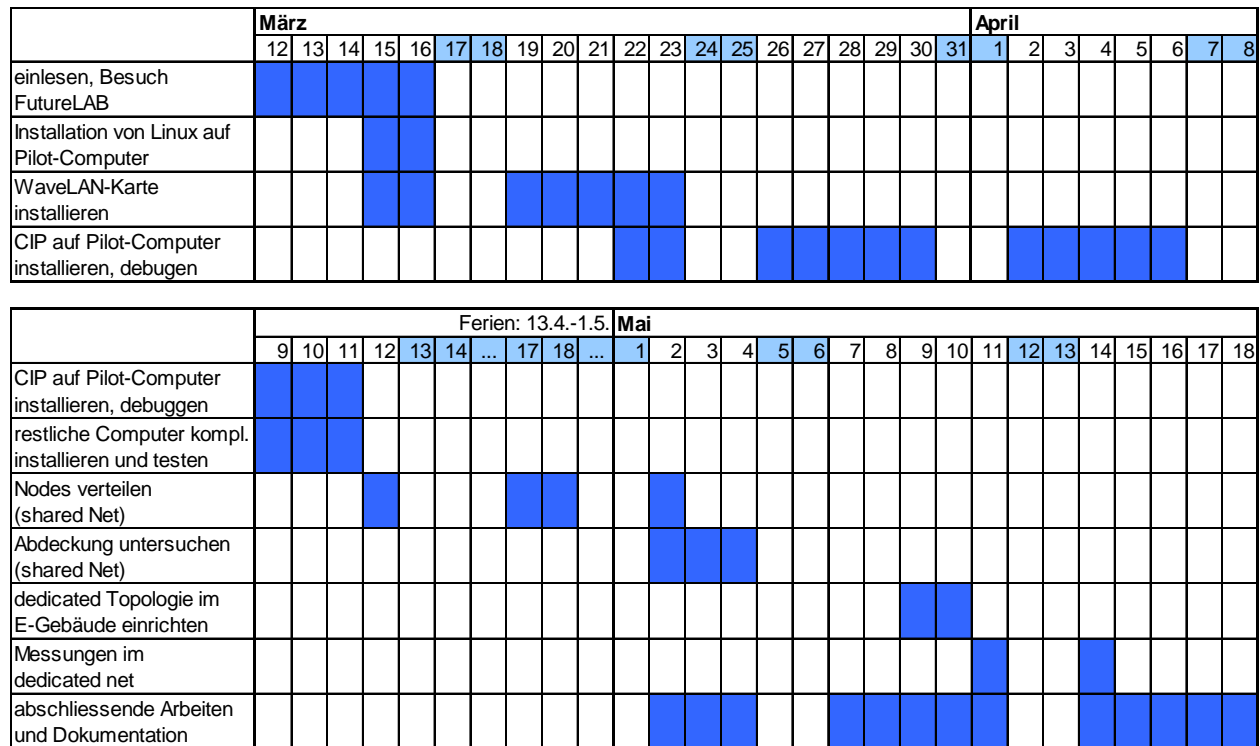


Abbildung 11 Ist-Zeitplan

11 Schlusswort

11.1 Fazit / Ausblick

Cellular IP ist sicherlich eine mögliche Lösung für ein mobiles Internet. Allerdings wird es noch einige Zeit benötigen, um dieses Konzept als kommerzielle Lösung auf den Markt zu bringen.

Zum Einen sind Treiber für die WaveLAN-Technologie speziell unter Linux noch im Anfangsstadium. Das heisst es gibt viele verschiedene Treiber, die teilweise noch im Entwicklungsstadium sind, HOWTOs oder man-Pages die demzufolge nicht dem neuesten Stand entsprechen etc.

Auch die von uns benutzte Cellular IP Software von Columbia University und Ericsson ist für eine kommerzielle Nutzung noch lange nicht reif genug. Es sind noch viele Hürden zum Beispiel bezüglich Authentisierung, Verschlüsselung oder Dinge wie Multiple Gateway Networks zu überwinden. Weiter wäre auch abzuklären, ob und welche Alternativen zu dieser Software eventuell vorhanden sind.

Schlussendlich, steht noch die Frage offen, wie ein solches Produkt überhaupt verkauft, respektive dem User angeboten wird. Wie wird abgerechnet, wer bietet einen solchen Dienst an und wer unterhält ein solches Netz?

Es stehen also noch sehr viele Fragen und Probleme vor uns, bevor wir per Cellular IP im Stadtpark mit dem Kollegen chatten oder im Internet surfen können...

11.2 Dank

Wir bedanken uns bei der Firma FutureLAB AG, insbesondere bei Herrn M. Aebi und Herrn T. Boesch, und unserem Dozenten Herrn Dr. A. Steffen für die gute Zusammenarbeit.

12 Anhang

12.1 Konfigurationsfiles

12.1.1 Mobile Host

Das Konfigurationsfile für den Mobile Host ist für alle Topologien gleich.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
wireless interface:          wlan0
mobile's IP address:        160.85.23.61
air interface:              mobile-air
route-update-time:          200      %in milliseconds
paging-update-time:         10000    %in milliseconds
active-state-timeout:       2000     %in milliseconds
handoff:                     1      %forced (=0) or SNR based (=1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

12.1.2 Shared Topologie

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               gateway_shared.conf                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GW:                               YES % this will be used at run time not compile time

IF YES, default router's IP address: (wire, eth0, 160.85.131.55)

IF NO, neighbour, uplink direction:

leaf neighbours(s):              (wire, eth0, 160.85.134.16), (wire, eth0, 160.85.134.15),
(wire, eth0, 160.85.131.54), (wire, eth0, 160.85.13.53)

paging cache:                     YES

semisoft:                          YES

If yes then size:                  10      %in packets

route-timeout:                     3000    %in milliseconds

paging-timeout:                    30000   %in milliseconds

max number of mobiles in cache:    100

max number of node interfaces:     10

GW IP address:                     160.85.138.137

% A neighbour is defined by medium(wire/wireless), the interface name
% on which the neighbour can be reached and the neighbour IP address.
% In the case of a wireless interface the frequency should be defined.

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               node1_shared.conf                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GW:                               NO % this will be used at run time not compile time

IF YES, default router's IP address:

IF NO, neighbour, uplink direction: (wire, eth0, 160.85.138.137)

leaf neighbours(s):               (wireless, wvlan0)

paging cache:                      YES

semisoft:                           YES

If yes then size:                  10      %in packets

route-timeout:                     3000    %in milliseconds

paging-timeout:                    30000   %in milliseconds

max number of mobiles in cache:    100

max number of node interfaces:     10

GW IP address:                      160.85.138.137

% A neighbour is defined by medium(wire/wireless), the interface name
% on which the neighbour can be reached and the neighbour IP address.
% In the case of a wireless interface the frequency should be defined.
    
```

Für die restlichen Nodes sind die Konfigurationsfiles analog zu Node 1.

12.1.3 Dedicated Topologie

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               gateway_dedicated.conf                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GW:                               YES % this will be used at run time not compile time

IF YES, default router's IP address: (wire, eth0, 160.85.20.1)

IF NO, neighbour, uplink direction:

leaf neighbours(s):               (wire, eth1, 160.85.23.2), (wire, eth2, 160.85.23.6)

paging cache:                      YES

semisoft:                           YES

If yes then size:                  10      %in packets

route-timeout:                     3000    %in milliseconds

paging-timeout:                    30000   %in milliseconds

max number of mobiles in cache:    100

max number of node interfaces:     10

GW IP address:                      160.85.20.110

% A neighbour is defined by medium(wire/wireless), the interface name
% on which the neighbour can be reached and the neighbour IP address.
% In the case of a wireless interface the frequency should be defined.
    
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               nodel_dedicated.conf                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GW:                               YES % this will be used at run time not compile time

IF YES, default router's IP address:

IF NO, neighbour, uplink direction: (wire, eth0, 160.85.23.51)

leaf neighbours(s):                (wireless, wvlan0), (wire, eth1, 160.85.23.10)

paging cache:                       YES

semisoft:                            YES

If yes then size:                   10      %in packets

route-timeout:                      3000    %in milliseconds

paging-timeout:                     30000   %in milliseconds

max number of mobiles in cache:     100

max number of node interfaces:      10

GW IP address:                      160.85.20.110

% A neighbour is defined by medium(wire/wireless), the interface name
% on which the neighbour can be reached and the neighbour IP address.
% In the case of a wireless interface the frequency should be defined.
```

Für die restlichen Nodes sind die Konfigurationsfiles analog zu Node 1.

12.2 Messwerte

Stockwerk	Sektor	aktive Nodes	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	Q ₈	Q ₉	Q ₁₀	Q _m	
E600	1	3	8	6	7	4	4	6	10	3	8	7	6.3	
		4	7	8	6	1	2	7	2	3	3	9	4.8	
	2	3	9	12	14	9	11	11	10	0	4	8	8.8	
		4	13	16	5	9	15	11	11	5	16	12	11.3	
	3	2	0	3	9	0	3	1	1	3	4	1	2.5	
		3	10	9	2	14	10	4	7	5	3	18	8.2	
	4	2	9	2	3	7	6	10	6	6	6	9	6.4	
	5	2	1	11	4	7	11	6	7	8	7	6	6.8	
	6	2	18	12	11	16	9	10	1	9	17	11	11.4	
	7	2	4	3	2	1	4	3	1	1	5	5	2.9	
	E500	1	3	13	19	24	18	16	19	18	16	20	26	18.9
		2	3	5	6	2	3	8	10	9	11	9	18	8.1
			4	3	15	22	22	17	23	16	18	17	10	16.3
		3	3	1	0	5	1	8	4	5	4	4	3	3.5
4			2	3	2	5	17	12	4	3	2	2	5.2	
4		1	7	6	1	7	4	7	2	3	4	5	4.6	
		2	4	1	7	3	5	4	2	3	4	4	3.7	
		3	3	6	6	5	1	4	1	0	4	1	3.1	
		4	1	0	3	3	0	4	1	4	2	2	2.0	
5		2	14	18	6	18	15	14	16	20	10	13	14.4	
6		2	33	14	22	21	33	16	14	14	16	15	19.8	
7		2	16	6	4	6	2	10	13	9	16	15	9.7	
E400		1	3	26	27	26	29	26	35	26	22	24	24	26.5
		2	3	10	22	18	19	17	33	22	31	26	27	22.5
	3	1	5	0	8	3	11	1	2	1	4	0	3.5	
		2	3	3	2	3	0	0	3	4	7	2	2.7	
		3	15	1	1	6	6	10	0	0	2	5	4.6	
		4	2	2	3	5	3	2	0	0	6	0	2.3	
	4	1	12	19	9	7	2	9	17	9	11	19	11.4	
		2	12	6	7	4	9	17	3	3	5	0	6.6	
	5	1	3	10	0	0	11	2	2	3	2	4	3.7	
		2	20	17	19	5	0	18	14	10	20	10	13.3	
		3	5	4	1	2	2	1	5	4	2	4	3.0	
	6	2	3	5	7	12	10	1	8	8	11	13	7.8	
	7	2	5	2	3	6	6	1	2	3	2	3	3.3	
	8	2	3	1	2	3	3	2	1	3	4	0	2.2	
E300	1	3	40	33	38	41	30	35	32	61	28	28	36.6	
	2	3	44	3	32	4	41	35	44	40	30	37	31.0	
	3	1	13	11	6	5	6	7	6	10	12	20	9.6	
		3	7	27	30	1	22	30	3	1	28	8	15.7	
	4	1	9	14	16	10	20	25	2	24	30	27	17.7	
		3	11	8	13	17	17	27	18	30	6	10	15.7	
	5	1	6	8	13	7	14	1	5	5	4	8	7.1	
		3	12	14	24	23	19	2	9	3	14	12	13.2	
	6	1	2	0	10	1	2	3	5	0	12	3	3.8	
		2	4	6	2	3	2	3	4	4	4	3	3.5	
		3	4	3	2	3	4	3	5	1	9	4	3.8	
	7	0											0.0	
	8	1												
		2												
E200	1	3	19	15	8	5	18	19	11	13	11	9	12.8	
	2	3	17	3	1	20	19	20	19	17	10	18	14.4	
	3	1	8	0	3	13	6	7	5	3	3	0	4.8	
		3	3	9	14	2	7	3	7	1	9	6	6.1	
	4	1	9	10	12	17	5	14	19	13	11	6	11.6	
		3	2	5	0	3	6	4	2	3	1	5	3.1	
	5	1	2	3	2	6	4	4	5	3	6	6	4.1	
	6	1	6	2	1	3	0	2	0	2	5	3	2.4	
	7	1	2	3	1	1	1	0	1	2	1	1	1.3	

12.3 Glossar

Active Mobile Host	Ein Mobile Host ist Active, falls er IP-Pakete versendet oder empfängt.
Cellular IP Base Station	siehe Cellular IP Node
Cellular IP Gateway	Cellular IP Node, der mit mindestens einem Interface an ein reguläres IP-Netzwerk angeschlossen ist.
Cellular IP Node	Ein Cellular IP Netzwerk besteht aus verbundenen Cellular IP Nodes. Diese routen IP-Pakete innerhalb des Cellular IP Netzwerkes und kommunizieren mit den Mobile Hosts über das wireless Interface (falls ein solcher vorhanden ist). Ein Cellular IP Node mit einem wireless Interface wird auch Basis Station genannt.
Control packet	Paging-update, route-update packet
Data packet	IP-Paket, das nicht control packet ist
Downlink	in Richtung Mobile Host
Downlink neighbor	alle Nachbarn, ausser Uplink
Handoff	Anmeldevorgang beim Übertritt in eine neue Zelle.
Idle Mobile Host	Ein Mobile Host ist Idle, falls er nicht kürzlich IP-Pakete versendet oder empfangen hat.
Paging Cache	Cache, der von Nodes zusätzlich zum Route Cache verwendet werden kann, um Pakete zu den Mobile Hosts zu routen
Paging-timeout	Verfallzeit für die Einträge im Paging Cache
Paging-update packet	Von Mobile Hosts übertragenes Kontroll Paket für Route Cache updates
Route Cache	Cache, der von allen Nodes benutzt wird, um Pakete zu den Mobile Hosts zu routen.
Route-timeout	Verfallzeit für die Einträge im Route Cache
Route-update packet	Von Mobile Hosts übertragenes Kontroll Paket für Paging Cache updates
Route-update-time	Zeit zwischen zwei route-update packets
Update packet	Paging-update und Routing-update packet
Uplink	in Richtung Gateway
Uplink neighbor	nächster Nachbar eines Nodes in Richtung Gateway
ZID	Zentrale Informatik Dienste der ZHW

12.4 Literaturverzeichnis

„Building the Wireless Internet“

Article by Chip Elliott in the January 2001 Issue of „IEEE Spectrum“ Magazine

IEEE 802.11 Standard

Cellular IP Home Page

<http://www.comet.columbia.edu/cellularip/>

A. G. Valko, “Cellular IP - A New Approach to Internet Host Mobility”

<http://www.comet.columbia.edu/cellularip/pub/ccr99.pdf>

IETF Internet Draft “Cellular IP”

<http://www.comet.columbia.edu/cellularip/pub/draft-ietf-mobileip-cellularip-00.txt>

Linux WaveLAN IEEE 802.11 Treiber

<http://www.fasta.fh-dortmund.de/users/andy/wvlan/>

Agere’s ORiNOCO Home Page (was Lucent’s WaveLAN Home Page)

<http://www.wavelan.com>

Wireless LAN resources for Linux

http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/

Linux-WLAN Project

<http://www.linux-wlan.com/linux-wlan/>

M. Gersbach & M. Kunz, Projektarbeit “VoIP und IPsec über ein Wireless LAN”

http://www.strongsec.com/zhw/PA/Sna01_2000.pdf

OSDN – open source development network

<http://sourceforge.net>

Linux Magazin

<http://www.linux-magazin.de>

12.5 CD-Verzeichnisstruktur

