Faculteit Ingenieurswetenschappen
Vakgroep Informatietechnologie
Voorzitter: Prof. Dr. Ir. P. LAGASSE

# Transparante Bluetooth netwerkoplossingen voor context-aware ziekenzorgtoepassingen

door
Tom PLUYM

Promotoren: Prof. Dr. Ir. I. MOERMAN, Prof. Dr. Ir. R. VAN DE WALLE, Dr. Ir. P. VERHOEVE
Thesisbegeleiders: Dr. F. DE KEUKELAERE, P. DE MIL, B. JOORIS
In samenwerking met Televic NV

Scriptie ingediend tot het behalen van de academische graad van
licentiaat in de informatica, optie: softwareontwikkeling

Academiejaar 2005-2006

Faculteit Ingenieurswetenschappen
Vakgroep Informatietechnologie
Voorzitter: Prof. Dr. Ir. P. LAGASSE

# Transparante Bluetooth netwerkoplossingen voor context-aware ziekenzorgtoepassingen

door
Tom PLUYM

Promotoren: Prof. Dr. Ir. I. MOERMAN, Prof. Dr. Ir. R. VAN DE WALLE, Dr. Ir. P. VERHOEVE
Thesisbegeleiders: Dr. F. DE KEUKELAERE, P. DE MIL, B. JOORIS
In samenwerking met Televic NV

Scriptie ingediend tot het behalen van de academische graad van
licentiaat in de informatica, optie: softwareontwikkeling

Academiejaar 2005-2006

# Voorwoord

Het verwezenlijken van dit eindwerk gedurende het afgelopen academiejaar is een erg leerrijke ervaring geweest; naast de mogelijkheid om mijn technische kennis uit te breiden, kreeg ik ook de kans om mijn vaardigheden op het vlak van communicatie en organisatie aan te scherpen. Ik dank dan ook mijn promotoren voor het boeiende en praktijkgerichte onderwerp, dat me toeliet om op korte tijd kennis te maken met heel wat technologieën. Ik wil uiteraard ook mijn dank uitdrukken aan mijn begeleiders; zij spaarden tijd noch moeite om me te ondersteunen bij het realiseren van de diverse aspecten van het eindwerk. Tot slot dank ik mijn vriendin Ilse en mijn ouders voor hun steun gedurende de voorbije jaren.

Tom Pluym, mei 2006

# Toelating tot bruikleen



Tom Pluym, mei 2006

# Transparante Bluetooth netwerkoplossingen voor context-aware ziekenzorgtoepassingen

door

Tom PLUYM

Scriptie ingediend tot het behalen van de academische graad van

licentiaat in de informatica, optie: softwareontwikkeling

Academiejaar 2005-2006

Promotoren: Prof. Dr. Ir. I. MOERMAN, Prof. Dr. Ir. R. VAN DE WALLE, Dr. Ir. P. VERHOEVE

Thesisbegeleiders: Dr. F. DE KEUKELAERE, P. DE MIL, B. JOORIS

In samenwerking met Televic NV

Faculteit Ingenieurswetenschappen

Universiteit Gent

Vakgroep Informatietechnologie

Voorzitter: Prof. Dr. Ir. P. LAGASSE

## Samenvatting

Televic is een Belgische onderneming gespecialiseerd in de ontwikkeling en de integratie van informaticasystemen voor de zorgsector. De onderneming maakt gebruik van een heterogene infrastructuur voor de implementatie van haar ziekenzorgtoepassingen. De wens om communicatielinks te creëren tussen diverse elementen binnen deze infrastructuur ligt aan de oorsprong van dit eindwerk. Om de goede werking van de toepassing te garanderen, is de te gebruiken communicatielink sterk afhankelijk van de context waarin deze ingezet zal worden.
Het eerste deel van deze scriptie staat in het teken van het realiseren van een softwarelaag voor de contextgevoelige en transparante vorming van een draadloos netwerk met behulp van de Bluetooth communicatietechnologie; binnen dit netwerk kunnen zich verschillende types vaste en mobiele communicatiepartners bevinden met diverse besturingssystemen.
In de tweede helft van de scriptie richten we onze aandacht op de opstellingen voor de thuiszorg en meerbepaald op de Telenet Digibox. We onderzoeken waarom een Digibox toepassing geen toegang heeft tot het internet, ondanks de aansluiting op het Telenet netwerk. Aan de hand van een analyse van het verkeer tussen de Digibox en de kabelmodem, leggen we de structuur van het Telenet netwerk bloot; op die manier bepalen we de precieze oorzaken van de connectiviteitsproblematiek. We realiseren hiervoor een oplossing aan de hand van een compacte Linux computer tussen de Digibox en de kabelmodem, en de Click Modular Router software. Aan de hand van de Linux computer, zullen we eveneens de mogelijkheid bieden om de Digibox te verbinden met randapparaten. We gebruiken hiervoor de kabelvervangende mogelijkheden van Bluetooth voor de communicatie met het randapparaat, en zetten webservices in voor het aanbieden van de functionaliteit van het apparaat aan de Digibox toepassing.

## Trefwoorden

Bluetooth, Click, contextgevoeligheid, iDTV-netwerk, transparante netwerkvorming, webservices

# Transparent Bluetooth network solutions for context-aware healthcare applications

Tom PLUYM

Thesis supervisors: Prof. Dr. Ir. I. MOERMAN, Prof. Dr. Ir. R. VAN DE WALLE, Dr. Ir. P. VERHOEVE
Thesis assistants: Dr. F. DE KEUKELAERE, P. DE MIL, B. JOORIS

***Abstract*** **This article outlines the networking issues one experiences in the context of the heterogeneous infrastructure for accommodating healthcare applications. We introduce Bluetooth[1] as an excellent communications technology for creating context-aware ad-hoc networks between the healthcare application's fixed and mobile devices. Next, we focus on the patient's home environment, and investigate the networking issues involved in using the Telenet Digibox for bringing a healthcare application into the patient's home.**

***Keywords*** **Bluetooth, Click, context-awareness, iDTV-network, transparent networking, web services**

## I. Introduction

Televic[2] is a Belgian information systems developer and integrator that specializes in hardware and software for healthcare applications. In order to fit its costumers' needs, Televic deploys software applications on a number of systems and platforms. Depending on the location and role of the user, a different device type will be used.

The first important group of devices are those for use in an infirmary environment. The major device in this context is the *Interactive Patient Terminal (IPT)*. The IPT is a compact Windows XP terminal with a large touch screen. The IPTs will be integrated in hospital rooms and display relevant patient care information.

The second group comprises the devices intended for use in the patient's home environment. In this context, the device of major importance is the Telenet Digibox. By deploying healthcare applications on a platform for interactive digital television, Televic introduces the application in the patient's home environment without forcing the patient into using a computer.

A number of various mobile devices form the last group; it contains among others the *Interactive Nurse Terminal (INT)* and a glucose meter. The INT is a Pocket PC running the Windows Mobile 5 operating system. Each nurse has his own INT, and is supposed to carry the device when on service. The INT displays confidential patient information and provides the nurse with constant access to the healthcare application.

Using different mobile and fixed device types for enrolling a healthcare application implies the need for efficient and transparent communications links among these devices and other infrastructure elements. In this article we focus on three essential links.

We show how to create a transparent context-aware wireless link between the IPTs and INTs. The existence of such a link allows to transparently authenticate the nurse to the IPT, and enables us to exchange data between the IPTs' and INTs' healthcare application instances.

Next, we investigate the issues involved in providing the Telenet Digibox with access to public internet servers. By providing such a communications link, we enable the Digibox healthcare application to communicate with servers for retrieving and storing medical information.

As the Digibox is a dedicated interactive television solution, it has little or no support for connecting peripheral devices. We show how to provide the Digibox with access to peripheral devices like a glucose meter.

## II. IPT/INT Communications Link

We have chosen Bluetooth[1] for implementing the transparent communications link between the IPT and the INT.

Bluetooth is an emerging wireless communications technology for Personal Area Networking, that features a 10 m range and low power consumption. Because of the lack of transparent network formation capabilities, Bluetooth's use in applications is nowadays mainly limited to cable replacement. In order to overcome these limitations, we have implemented a software layer that controls the Bluetooth protocol stack, and realizes the transparent formation of ad-hoc networks; we call this software layer the *Transparent Communications Layer*.

One of the issues involved in implementing the Transparent Communications Layer, was obtaining a suitable API for programming the Bluetooth protocol stack. Franson BlueTools[3] is an affordable API for Windows XP and Windows Mobile 5, that supports the major vendors' Bluetooth protocol stack implementations. Moreover it offers the facilities required for implementing the Transparent Communications Layer.

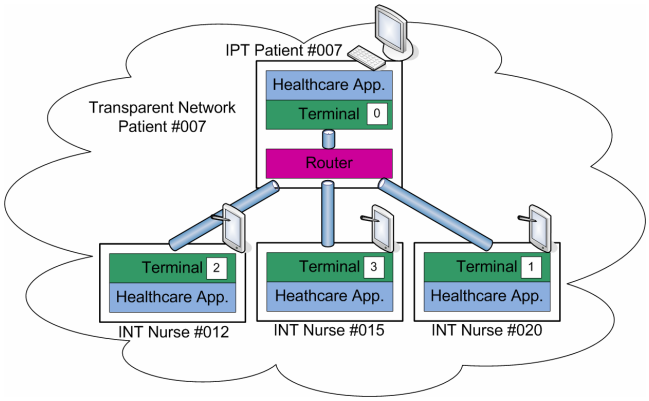



Figure 1: A Transparent Network topology

The major task of the Transparent Communications Layer is the formation of Bluetooth *Transparent Networks*. As shown in Figure 1, each device that intends to join a Transparent Network, is supposed to run the *Terminal* component of the Transparent Communications Layer. The Terminal

component is responsible for searching and selecting devices running the Router component; once a suitable Router has been found, the Terminal initiates a point-to-point link with the Router device. The Router component manages a Transparent Network; this involves assigning network addresses to Terminal components, and forwarding packets among them. In the case of a healthcare application, the INT runs the Terminal component. The IPT is required to run both Router and Terminal components.

By combining Bluetooth with a software layer for transparent network formation, we are able to implement a framework for the creation of transparent communications links between IPTs and INTs. By thoroughly selecting the potential Router devices, and by using a notion of context based on the Bluetooth received signal strength indicator, we are able to significantly reduce the number of user interventions, and provide the healthcare application with a sufficiently transparent wireless networking solution.

## III. Digibox/Internet Communications Link

In spite of the Digibox' connection to the Telenet network, a Digibox software application isn't able to access the public internet. The first step to solving this disability, is gaining insight in the network that causes it; this can be accomplished by conducting a traffic analysis on the Digibox' return channel with the help of the Ethereal Protocol Analyzer[4] software. By monitoring the return channel traffic, we have learned that the Telenet DHCP servers assign network addresses, based on the DHCP request source's hardware address. In case of a DHCP request originating from a Digibox, an address in the private address range 10.0.0.0/8 will be assigned. Note that these ranges are meant for use in private networks; companies often use them for their intranet in order to avoid reserving a public IP address for each host. Because these addresses don't guarantee worldwide uniqueness, they can't be used for public internet access. Usually, this issue is solved by using Network Address Translation (NAT)[5]; this technique enables multiple hosts to access the public internet using a single public IP address. Because of its private IP address, the Digibox is part of Telenet's private iDTV-network, which comprises all Digiboxes and interactive television support servers. Clearly, the iDTV-network has no provisions for NAT.

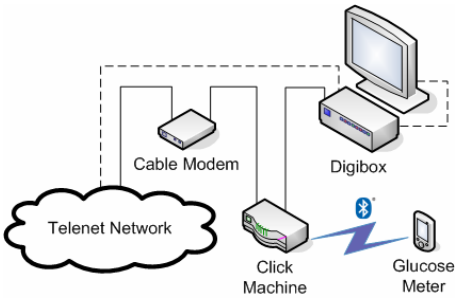



Figure 2: Digibox setup with Click Machine and glucose meter

The above observations have inspired us to solve the connectivity issue by performing selective NAT. As shown in Figure 2, we've installed a small Linux computer – the *Click Machine* – in between the Digibox and the cable modem; the Click Machine performs NAT by running the Click Modular Router[6] software and a dedicated Click script.

The outlined approach enables us to provide the Digibox healthcare application with internet access, without interrupting the device's interactive television functionality.

## IV. Digibox/Glucose Meter Communications Link

We've mentioned before that the Digibox has no provisions for connecting peripheral devices like a glucose meter. Because of the meter's limited memory capacity, and because of the error proneness of manual registration, we would like to provide the Digibox health care application with the means for accessing the measurements accumulated in the meter's memory. We implement this link by using the Click Machine's capabilities for providing web services with Tomcat[7], and connecting peripheral devices.

As shown in Figure 2, we've opted to use Bluetooth as a cable replacement technology for connecting the glucose meter to the Click Machine. We use a servlet for presenting access to the glucose meter to the Digibox application. The application can access the meter's measurements by using HTTP transactions in combination with a simple XML based protocol. This approach allows the application to use the peripheral device without being bothered by device specific aspects. Note that a device specific servlet is required to be installed on the Click Machine.

In a number of scenarios, one will prefer the device specific aspects to be handled by the Digibox application; this will require a more advanced generic web service, and the use of a real time protocol, as a lot of peripheral devices have stringent timing constraints for their communications.

## V. Conclusion

Despite the healthcare application's heterogeneous infrastructure, we were able to realize communications links among diverse elements. The reader will have noticed the importance of Bluetooth technology for implementing these links; we use Bluetooth as a transparent network support technology for the IPT/INT link, while it provides us with a uniform access point for connecting peripheral devices to the Click Machine.

Installing the Click Machine in between the Digibox and the cable modem, enabled us to provide internet connectivity to the Digibox applications. The Click Machine also manifested itself as a flexible framework for presenting web service to Digibox applications, for example for accessing the measurements accumulated in a glucose meter's memory.

## References

[1] Bluetooth.com – The Official Bluetooth Wireless Info Site. *http://www.bluetooth.com/*.
[2] Televic NV. *http://www.televic.com/*.
[3] Franson Technologies AB. *http://www.franson.com/*.
[4] Ethereal – Network Protocol Analyzer. *http://www.ethereal.com/*
[5] Computer Networks – A Top-Down Approach Featuring the Internet, Second Edition. James F. Kurose and Keith W. Ross. Pearson Education Inc. p 352. (2003).
[6] The Click Modular Router Project. *http://www.read.cs.ucla.edu/click*.
[7] Apache Tomcat. *http://tomcat.apache.org/*.

# Inhoudsopgave

**Hoofdstuk 1 Situering ..... 1**

1.1 Infrastructuur ..... 2

1.1.1 Infrastructuur voor de zorginstelling ..... 3

1.1.2 Infrastructuur voor de thuiszorg ..... 4

1.2 Scenario's ..... 6

1.2.1 Communicatie IPT/IVT ..... 6

1.2.2 Communicatie Digibox/internet ..... 7

1.2.3 Communicatie Digibox/glucosemeter ..... 9

1.2.4 Communicatie Digibox/IVT ..... 10

1.3 Conclusie ..... 12

**Hoofdstuk 2 Transparante Bluetooth Communicatielaag ..... 13**

2.1 Algemeen ontwerp ..... 14

2.1.1 Entiteiten en terminologie ..... 14

2.1.2 Vereisten ..... 16

2.1.3 Problematiek standaardoplossingen ..... 19

2.2 Bluetooth ..... 24

2.2.1 Oorsprong en standaardisering ..... 24

2.2.2 Operationele werking ..... 24

2.2.3 De Bluetooth protocol stack ..... 30

2.2.4 Bluetooth API's ..... 37

2.3 Implementatie met Bluetooth ..... 44

2.3.1 Zoeken naar potentiële Routers ..... 44

2.3.2 Selectie van potentiële Routers ..... 56

2.3.3 Contextgevoelige selectie van de Router ..... 60

2.3.4 Lidmaatschap van een Transparant Netwerk ..... 70

2.3.5 Uitbreiding naar Linux ..... 77

2.4 Conclusie ..... 79

**Hoofdstuk 3 De Click Machine ..... 81**

3.1 Communicatie Digibox/internet ..... 82

3.1.1 Problematiek en conceptuele oplossing ..... 82

3.1.2 Implementatie ..... 88
3.2 Communicatie Digibox/Randapparaat ..... 94
3.2.1 Problematiek en conceptuele oplossing ..... 94
3.2.2 Implementatie ..... 95
3.3 Conclusie ..... 100

**Hoofdstuk 4 Conclusie ..... 102**

# Hoofdstuk 1
# Situering

Sinds enkele jaren berichten de media bijna dagelijks over de vergrijzing van de West-Europese bevolking, en de gevolgen hiervan op onze manier van leven en werken. Stilaan groeit bij politici en sociale partners het besef dat ingrijpende maatregelen nodig zijn om de huidige standaard inzake sociale zekerheid in stand te houden. Een eerste maatregel was het in 2005 goedgekeurde Generatiepakt[1].

Eén van de sectoren die een essentiële rol zullen spelen bij het opvangen van de vergrijzing is de zorgsector. Het aantal zorgbehoevende mensen stijgt immers sterk, terwijl er jaarlijks steeds minder geschikt personeel beschikbaar komt op de arbeidsmarkt[2].

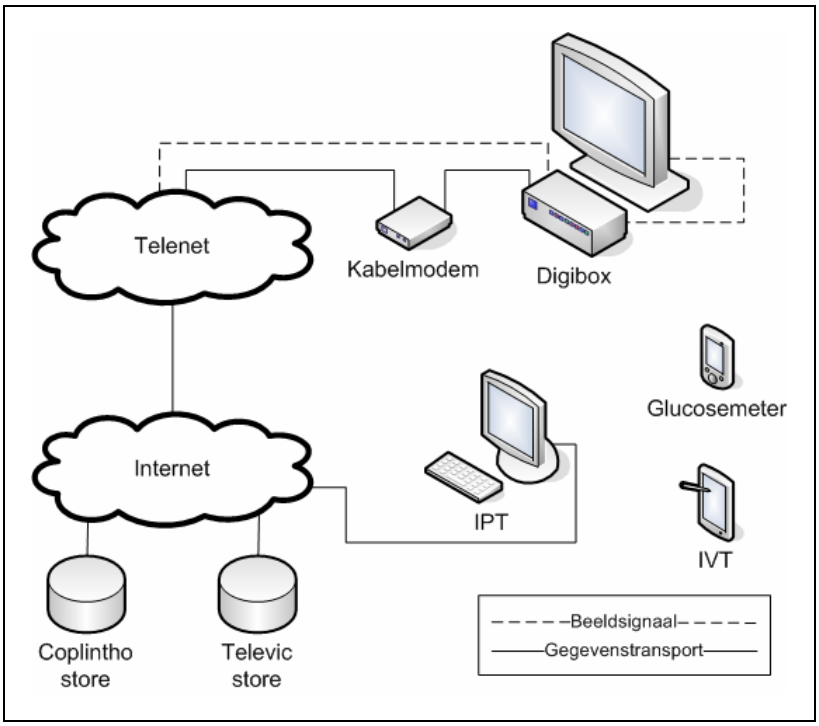
Om de problemen van de zorgsector te verzachten, streeft men een hogere efficiëntie na, en past men informatisering toe waar mogelijk. De kwalitatieve en menselijke aspecten van de geleverde diensten mogen hierbij uiteraard niet uit het oog verloren worden. Heel wat instellingen en overheden hebben hun hoop dan ook op een verantwoord gebruik van informatietechnologie gevestigd om een antwoord te bieden op de problematiek van de zorgsector. In de voorbije jaren werden een aantal onderzoeksprojecten opgestart, zoals Coplintho[3] op lokaal niveau en MobiHealth[4] in Europees samenwerkingsverband.

Televic[5] is een Belgische onderneming gespecialiseerd in de ontwikkeling en de integratie van informaticasystemen voor de zorgsector. Het bedrijf werkt momenteel aan een totaaloplossing voor gebruik in zorginstellingen en de thuiszorg. De doelstellingen van de totaaloplossing omvatten ondermeer een geïnformatiseerde zorgregistratie en een geavanceerd gegevensbeheer. De motivatie voor deze thesis bestaat uit de nood aan oplossingen voor een aantal concrete problemen die het bedrijf ondervindt bij de ontwikkeling van dit systeem. Zoals de titel van de thesis reeds laat vermoeden, betreft het vooral problemen omtrent de transparante communicatie tussen de systeementiteiten binnen een netwerk.

Vooraleer we deze problemen introduceren aan de hand van een aantal scenario's, zullen we eerst de beschikbare infrastructuur in meer detail bekijken. Tot slot zullen we een aantal conceptuele oplossingen voorstellen en verwijzen we naar de technische invulling verderop in deze scriptie.

# 1.1 Infrastructuur

We bekijken in dit onderdeel een overzicht van de infrastructuur, zoals Televic die in gedachten heeft voor de realisatie van een totaalsysteem voor de informatisering van zorginstellingen en de thuiszorg. Bij de ontwikkeling wordt er zoveel als mogelijk gebruik gemaakt van reeds aanwezige elementen en systemen. Indien het vooralsnog nodig is om de bestaande infrastructuur aan te vullen met extra hardware, probeert men standaardtoestellen te gebruiken waar dit mogelijk is. De infrastructuur waar men van uitgaat wordt weergegeven in Figuur 1.




Figuur 1: Overzicht infrastructuur

De weergegeven elementen kunnen ondergebracht worden in drie groepen, afhankelijk van de omgeving waarin ze gebruikt zullen worden. Een eerste groep omvat de infrastructuur voor de realisatie van het scenario in de zorginstellingen. Een tweede groep bestaat uit de elementen die ingezet zullen worden in een oplossing voor de thuiszorg. Er zijn uiteraard eveneens een aantal – eventueel mobiele – gemeenschappelijke elementen die in beide omgevingen een rol zullen vervullen.

### 1.1.1 Infrastructuur voor de zorginstelling

In het geval van de zorgstelling gaat men er van uit dat er nog geen geschikte infrastructuur aanwezig is voor het aanbieden van softwaretoepassingen.

Men heeft de ambitie om een computerterminal te integreren in elke kamer, of zelfs in elk bed. In het vervolg verwijzen we naar deze terminal met de benaming *Interactieve Patiënten Terminal* of *IPT*. De IPT is gebonden aan een bepaalde kamer of ziekenbed, en is zich bewust van de identiteit van de patiënt die zich in het bed bevindt. De IPT is een compacte computer gebaseerd op een Intel architectuur, voorzien van Microsoft Windows XP. De IPT wordt als *vast* beschouwd in de mate dat hij gemonteerd is tegen de muur, of geïntegreerd wordt in het ziekenbed. Hieruit volgt dat we mogen veronderstellen dat de IPT op eenvoudige wijze van stroom kan voorzien worden, en bovendien beschikt over relatief veel rekenkracht. Gezien de keuze van de architectuur en het besturingssysteem, is het relatief goedkoop en eenvoudig om de standaardconfiguratie aan te passen of uit te breiden. Voor de interactie met de gebruiker, is de IPT voorzien van een aanraakscherm. Er is communicatie mogelijk met de buitenwereld via een netwerkinterface, zodat er een verbinding bestaat met het interne netwerk van de zorginstelling en eventueel met het publieke internet. Verder zullen we de IPT uitrusten met de mogelijkheden voor draadloze gegevensuitwisseling met Bluetooth.

De plaatsing in een publieke omgeving van een scherm, waarop mogelijk gevoelige informatie ingevoerd of getoond zal worden, brengt een aantal problemen inzake veiligheid en bescherming van de persoonlijke levenssfeer met zich mee. Een mogelijke oplossing is het gebruik van een kleine draagbare terminal die draadloos gegevens kan uitwisselen met de IPT. In wat volgt zullen we naar een dergelijke mobiele terminal verwijzen met de naam *Interactieve Verpleegkundige Terminal* of *IVT*.

De IVT is een compacte draagbare computer van het type PDA[1], voorzien van een aanraakscherm voor de interface met de gebruiker, en de mogelijkheden voor draadloze gegevensuitwisseling met de IPT. Voor de realisatie van de IVT zal gebruik gemaakt worden van een Pocket PC met Microsoft Windows Mobile 5. Het is relatief moeilijk en duur om de standaardconfiguratie van een Pocket PC te wijzigen; men doet er dan ook goed aan om zich bij de ontwikkeling van een toepassing zoveel mogelijk te houden aan wat standaard beschikbaar is in de meeste producten.

---

[1] *Personal Digital Assistant*

Het is de bedoeling dat elke zorgverstrekker beschikt over een eigen IVT, zodat deze op een eenvoudige en veilige manier gegevens in verband met de dienstverlening kan raadplegen. Door de IVT te introduceren in de zorgtoepassing is het mogelijk om in te staan voor voldoende veiligheid, en wordt de naleving van de privacy van de patiënt gegarandeerd; de IVT is immers gebonden aan een bepaald lid van het verzorgend personeel, terwijl bovendien de omvang van het bijhorende scherm voldoende klein is om *shoulder surfing*[2] te vermijden.

In 1.2.1 bespreken we een scenario waarbij er draadloze gegevensuitwisseling plaatsvindt tussen de IPT en de IVT. De problematiek van een dergelijk scenario situeert zich vooral rond het gebrek aan transparantie bij het vormen van ad-hoc netwerken. De oplossing van het probleem mondt uit in een framework voor de transparante vorming van ad-hoc netwerken met Bluetooth. We bespreken de ontwikkeling van dit framework in Hoofdstuk 2.

Voor informatie omtrent de implementatie van een zorgtoepassing die steunt op de IPT, de IVT en onderlinge gegevensuitwisseling, verwijzen we naar de thesis van O. Christiaens[6].

### 1.1.2 Infrastructuur voor de thuiszorg

De infrastructuur voor de thuiszorg bevindt zich bovenaan in Figuur 1. Het betreft een televisietoestel en een kabelaansluiting, aangevuld met een kabelmodem en een Telenet Digibox. De penetratie van de Vlaamse kabel is uitzonderlijk hoog (97% van de gezinnen[7]), zodat we er van uit kunnen gaan dat het televisietoestel en de kabelaansluiting reeds aanwezig zijn. Tegenover de erg hoge kabelpenetratie staat echter dat momenteel slechts 2,5% van de gezinnen over digitale televisie beschikt[7].

Om digitale televisie mogelijk te maken is er – in het geval van Telenet – een Digibox nodig. Naast de mogelijkheden van digitale televisie is het mogelijk om op het toestel gebruikerssoftware te draaien, bijvoorbeeld een toepassing voor de thuiszorg.

De Digibox maakt gebruik van een zogenaamd *terugkeerkanaal* (Eng. *return channel*) om de invoer van de gebruiker naar de Telenet servers te sturen en op die manier interactieve toepassingen mogelijk te maken. In de praktijk is het terugkeerkanaal niets anders dan een klassieke aansluiting op een IP netwerk. Er is echter jammer genoeg gebleken dat toepassingssoftware op de Digibox geen toegang heeft tot het publieke internet. We tonen eerst het nut van een dergelijke netwerkverbinding aan in scenario 1.2.2 en bespreken vervolgens de technische aspecten van de oplossing in Hoofdstuk 3.

---

[2] Het stiekem verwerven van gevoelige informatie door directe observatie, bijvoorbeeld door over iemands schouder mee te kijken op een computerscherm.

Een tweede tekortkoming van de Digibox is het gebrek aan zowel aansluitmogelijkheden voor randapparaten als voorzieningen voor draadloze communicatie. Het zou erg interessant zijn indien het mogelijk was om bepaalde medische randapparatuur te gebruiken via een toepassing op de Digibox. We vermelden bijvoorbeeld de glucosemeter uit Figuur 1. Door de Digibox en de glucosemeter te voorzien van de mogelijkheid om gegevens uit te wisselen via Bluetooth, willen we het gebruik ervan vergemakkelijken en de automatische uitwisseling van meetgegevens mogelijk maken. We tonen het nut van deze gegevensverbinding aan in scenario 1.2.3 en verwijzen naar Hoofdstuk 3 voor de uitwerking van de technische aspecten van de oplossing.

Voor meer informatie over de realisatie van een zorgtoepassing voor het Digibox platform verwijzen we naar de scriptie van E. Cant[8].

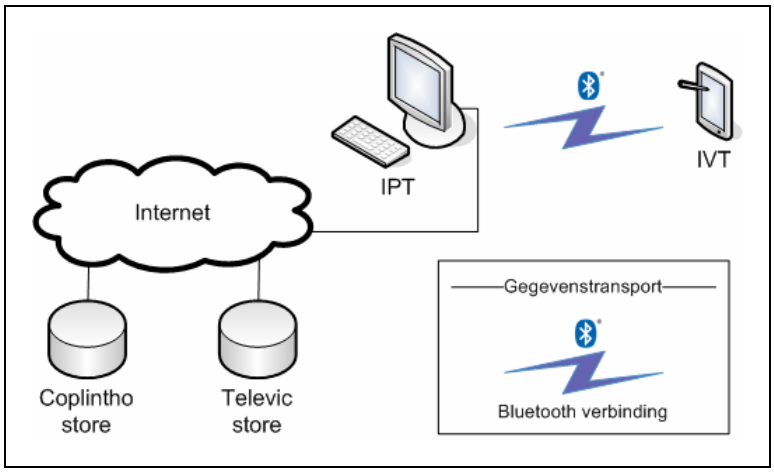
# 1.2 Scenario's

De bedoeling van dit onderdeel is het schetsen van een aantal concrete scenario's die optreden bij het gebruik van de zorgtoepassing. De scenario's maken elk gebruik van een aantal van de infrastructuurelementen die gedefinieerd werden in 1.1.

Met de scenario's willen we de nood aanhalen aan een mogelijkheid om gegevens uit te wisselen tussen een aantal infrastructuurelementen van de zorgtoepassing. We vermelden telkens wat de problemen hierbij zijn en stellen een conceptuele oplossing voor. Tot slot verwijzen we voor elk scenario naar het hoofdstuk waar we de technische aspecten van het probleem en de bijhorende oplossing in detail bespreken.

## 1.2.1 Communicatie IPT/IVT

We beschouwen de opstelling voor de zorginstelling zoals weergegeven in Figuur 2. De belangrijkste infrastructuurelementen die we zullen gebruiken voor dit scenario zijn de IPT en de IVT. Voor meer informatie over de infrastructuurelementen verwijzen we naar 1.1.




Figuur 2: Opstelling zorginstelling – IPT/IVT

Het scenario verloopt als volgt:

1. De patiënt bevindt zich in het ziekenbed en de geïntegreerde IPT is ingeschakeld.
2. De verpleegkundige komt binnen in de kamer.
3. De IVT van de verpleegkundige onderzoekt de aanwezigheid van een IPT.
4. De IVT maakt met zo weinig mogelijk tussenkomst van de verpleegkundige een gegevensverbinding met de IPT van de te verzorgen patiënt.
5. De verpleegkundige gebruikt het grote gebruiksvriendelijke scherm van de IPT om het dossier van de patiënt op te vragen.

6. Aangezien de IVT gebonden is aan een bepaalde verpleegkundige, kan deze aan de IPT bewijzen dat de gebruiker een zorgverstrekker is, en er dus beveiligde gegevens mogen bekeken en gewijzigd worden.

7. Vertrouwelijke gegevens wensen we niet weer te geven of te wijzigen op het grote publieke IPT scherm. Aangezien het echter mogelijk is om data uit te wisselen met de IVT, zullen we de betreffende gegevens enkel tonen en wijzigen met behulp van het kleine private IVT scherm.

8. Wanneer de verpleegkundige de kamer verlaat wordt de communicatie tussen de IPT en de IVT verbroken.

Het is duidelijk dat we de IPT en de IVT moeten voorzien van een communicatietechnologie die de communicatie tussen de toestellen op een zo transparant mogelijke manier verzorgt; een uitgebreide interactie met de verpleegkundige telkens er een verbinding gemaakt wordt, zou de efficiëntie van de zorgtoepassing immers sterk nadelig beïnvloeden. Om de interactie met te gebruiker te minimaliseren zullen we de netwerkoplossing contextgevoelig maken, zodat de vorming van het draadloze netwerk zal afhangen van de context waarin de IVT zich bevindt.

We zullen in Hoofdstuk 2 Bluetooth voorstellen als de aangewezen communicatietechnologie voor onze toepassing. In zijn standaardvorm, vereist deze technologie heel wat manuele tussenkomst van de gebruiker. We zullen dan ook het transparantieprobleem in meer detail onderzoeken en een oplossing voorstellen onder de vorm van een Transparante Bluetooth Communicatielaag in Hoofdstuk 2.

Het bekomen framework werd ingezet bij de implementatie van een concrete zorgtoepassing door O. Christiaens[6].

### 1.2.2 Communicatie Digibox/internet

Voor dit scenario gaan we uit van de infrastructuur in een thuisopstelling zoals weergegeven in Figuur 3. Voor meer informatie over de infrastructuurelementen verwijzen we naar 1.1.

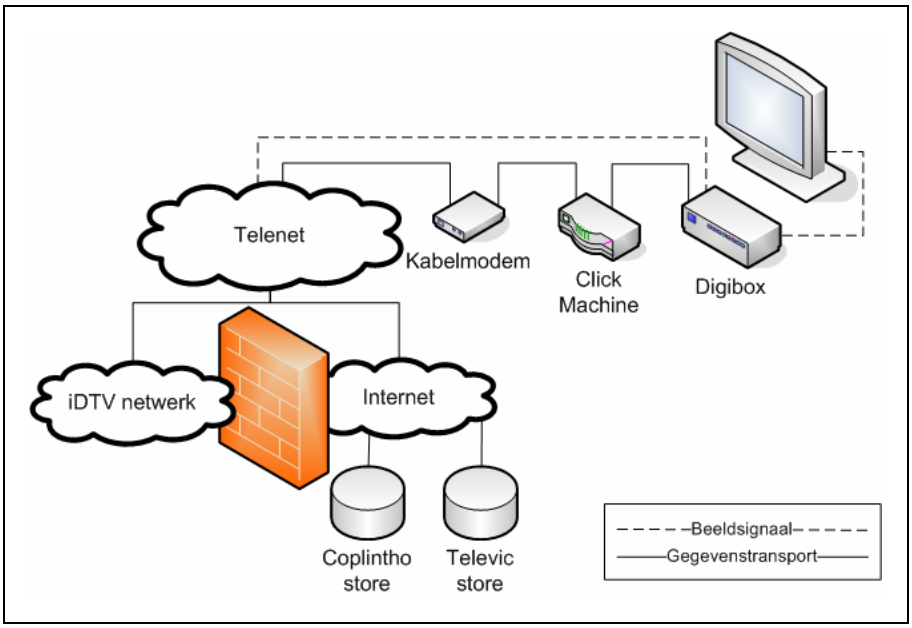
Een mogelijk scenario met deze infrastructuur verloopt als volgt:

1. De thuisverpleegkundige komt langs bij zijn patiënt.

2. De thuisverpleegkundige voert een aantal verzorgingstaken uit.

3. Na de verzorging maakt de verpleegkundige gebruik van de zorgtoepassing op de Digibox om medische informatie en gegevens voor facturatie in te voeren in de databank.

4. Het invoeren van de nieuwe informatie zorgt onmiddellijk voor het bijwerken van de databanken van de betrokken partijen, zoals bijvoorbeeld het RIZIV, de huisarts, enz.




Figuur 3: Opstelling thuiszorg – Digibox/internet

Hoewel de Digibox voorzien is van een terugkeerkanaal, is uit de praktijk gebleken dat het toch niet mogelijk is om een verbinding te maken met servers in het publieke internet. Het is namelijk zo dat Telenet een apart privaat netwerk voorzien heeft voor zijn interactieve televisieactiviteiten, het zogenaamde *iDTV*-netwerk.

Een verbinding met het publieke internet zou het ondermeer mogelijk maken om de Coplintho store en de Televic store te benaderen. Beide servers bieden een aantal specifieke diensten aan voor de opslag en de verwerking van gegevens gebonden aan een zorgtoepassing.

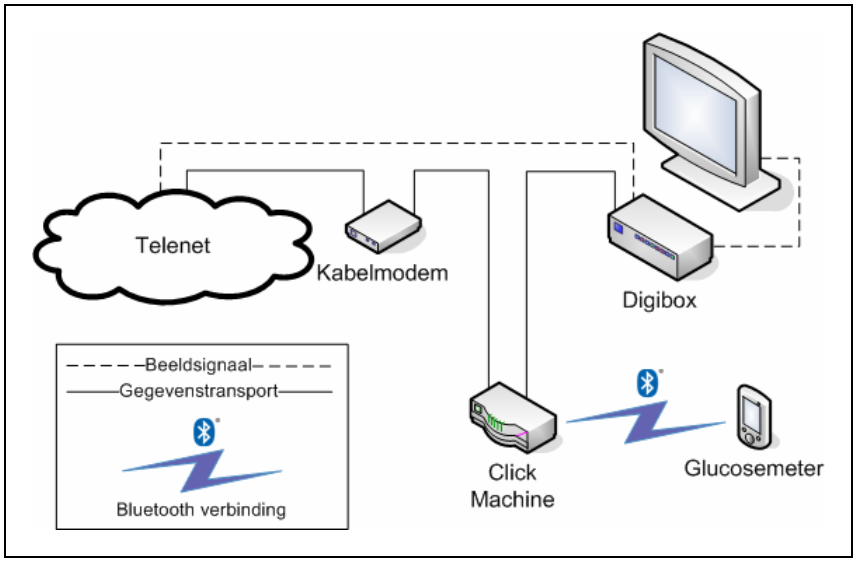
Opdat de zorgtoepassing op de Digibox de servers in het publieke internet zou kunnen bereiken, zullen we de aanvragen naar servers in het publieke internet onderscheppen tussen de Digibox en de kabelmodem. Hiertoe voorzien we een stuk hardware – de *Click Machine* – met bijhorende software die we fysiek tussen de Digibox en de kabelmodem zullen plaatsen zoals weergegeven in Figuur 3. De interactieve televisiefunctionaliteit mag hierbij uiteraard niet verstoord worden.

We behandelen de technische aspecten van het probleem en de bijhorende oplossing in 3.1 in Hoofdstuk 3.

### 1.2.3 Communicatie Digibox/glucosemeter

Het scenario dat we in dit onderdeel zullen bespreken houdt nauw verband met het scenario uit 1.2.2 *Communicatie Digibox/internet*. In dit voorgaande scenario werd namelijk melding gemaakt van de invoer van medische informatie op een Digibox zorgtoepassing door een thuisverpleegkundige. We doelen met deze gegevens bijvoorbeeld op de meetresultaten die verzameld werden in de glucosemeter van een diabeteslijder.

Voor een adequate behandeling van suikerziekte is het belangrijk dat de meetgegevens van een langere termijn goed bijgehouden worden. Het zal echter meteen duidelijk zijn dat de manuele invoer van dergelijke gegevens in een zorgtoepassing vervelend en foutgevoelig is.




Figuur 4: Opstelling thuiszorg – Digibox/glucosemeter

Een meer gebruiksvriendelijk scenario gaat als volg:

1. De patiënt meet enkele malen per dag zijn bloedsuikerspiegel met behulp van een glucosemeter.
2. Wanneer de glucosemeter zich voldoende dicht bij de Digibox bevindt, worden de gegevens automatisch uitgelezen uit de meter en worden ze verder verwerkt door de zorgtoepassing.
3. De meetgegevens van een langere termijn zijn eenvoudig toegankelijk met behulp van de software op de Digibox.

Zoals reeds vermeld in 1.1 *Infrastructuur*, zijn de voorzieningen voor het aansluiten van randapparatuur op de Digibox uiterst beperkt, en zijn er geen draadloze

communicatiemogelijkheden. We zullen echter ook in dit scenario een oplossing trachten te bieden met behulp van de in 1.2.2 geïntroduceerde Click Machine.

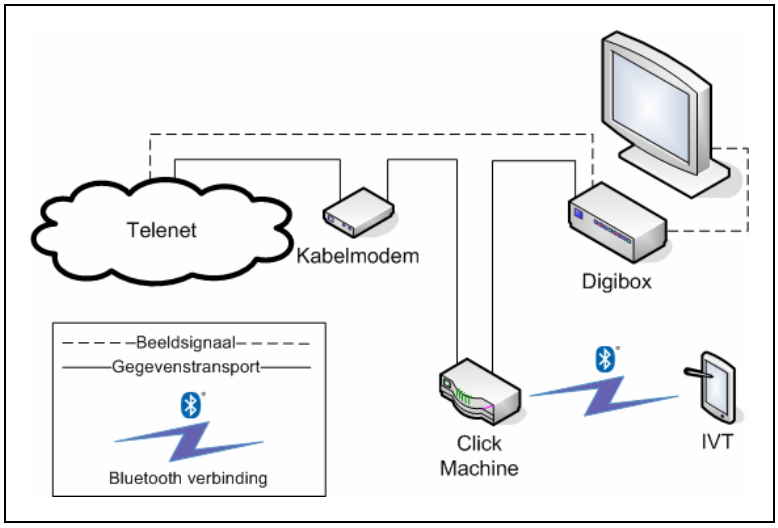
We zullen het via de Click Machine mogelijk maken om aan aantal randapparaten te gebruiken van op de Digibox. Daartoe sluiten we het gewenste randapparaat aan op de Click Machine en voorzien we een framework waarmee dit via een webservice toegankelijk gemaakt wordt voor de Digibox. Deze laatste kan de randapparaten benutten via de netwerkverbinding met de Click Machine.

Heel wat randapparaten voor medisch gebruik zullen in de toekomst uitgerust worden met een Bluetooth interface voor draadloze gegevensuitwisseling[4]. We zullen dan ook de Click Machine en de glucosemeter voorzien van de nodige Bluetooth hardware, en een oplossing voorstellen die gebruikt maakt van Bluetooth voor de gegevensuitwisseling.

We onderzoeken de technische aspecten van dit probleem en de bijhorende oplossing in meer detail in 3.2 in Hoofdstuk 3.

### 1.2.4 Communicatie Digibox/IVT

We stelden in scenario's 1.2.1 en 1.2.3 Bluetooth communicatielinks voor tussen respectievelijk de IPT en de IVT, en tussen de Digibox en een randapparaat. Het zou erg interessant zijn indien we deze scenario's zouden kunnen combineren om op die manier een communicatielink te realiseren tussen de Digibox en de IVT. We motieveren deze communicatielink met een scenario dat zich afspeelt in de opstelling in Figuur 5.




Figuur 5: Opstelling thuiszorg – Digibox/IVT

1. De verpleegkundige komt langs bij de patiënt thuis.
2. Van zodra de verpleegkundige de woonkamer betreedt, ontstaat er een communicatielink tussen de Digibox en de IVT van de verpleegkundige.

3. De IVT geldt als identiteitsbewijs en zorgt voor de transparante authenticatie van de verpleegkundige bij de zorgtoepassing op de Digibox.
4. De zorgtoepassing op de Digibox kan onmiddellijk door de verpleegkundige gebruikt worden, en toont bijvoorbeeld automatisch de uit te voeren taken.

Het is duidelijk dat een communicatielink tussen de Digibox en de IVT zorgt voor een hogere efficiëntie bij het gebruik van de zorgtoepassing op de Digibox.

Zoals weergegeven in Figuur 5, maken we ook in dit scenario gebruik van de Click Machine voor het realiseren van een oplossing. De Click Machine dient enerzijds via Bluetooth te communiceren met de IVT, terwijl deze anderzijds de mogelijkheid om te communiceren met de IVT aan de Digibox aanbiedt als een webservice. Merk op dat we, in tegenstelling tot het vorige scenario, Bluetooth hier gebruiken als volwaardige netwerktechnologie, eerder dan louter als kabelvervanger. Hoewel we dit scenario niet volledig zullen uitwerken in het kader van dit eindwerk, leggen we in 2.3.5 en 3.2 wel de basis voor het realiseren van een oplossing.

## 1.3 Conclusie

In dit hoofdstuk hebben we de context geschetst waarin het onderwerp van dit eindwerk zich situeert.

In de inleiding hebben we een aantal problemen van de zorgsector aangegrepen om de nood aan informatisering aan te tonen. Televic is een onderneming die met behulp van informatietechnologie een totaaloplossing voor de zorgsector wil realiseren, om het hoofd te bieden aan de gestelde problemen.

We gaven een overzicht van de beschikbare infrastructuur voor de implementatie van de zorgtoepassing. We toonden bovendien met behulp van een aantal concrete scenario's aan waarom het transparant uitwisselen van gegevens tussen de infrastructuurentiteiten belangrijk is voor het slagen van de zorgtoepassing. De nood aan een transparante netwerkoplossing tussen de infrastructuurelementen vormt de motivatie voor dit eindwerk.

Tot slot merken we nog op dat het streven naar een transparante oplossing met zo weinig mogelijk nutteloze interactie met de gebruiker, volledig past binnen de huidige trend naar *Ubiquitous Computing*. In deze denkrichting streeft men naar alomtegenwoordige maar tegelijk verborgen technologie, die op een intuïtieve manier geïntegreerd is in ons dagelijks leven en werken.

Om het belang van transparantie in onze netwerkoplossingen aan te halen, besluiten we dit hoofdstuk met een citaat van de vader van Ubiquitous Computing uit [9]:

*"[…] There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods."*

*– M. Weiser*

# Hoofdstuk 2
# Transparante Bluetooth Communicatielaag

In dit hoofdstuk gaan we op zoek naar een oplossing voor de problemen die werden aangebracht met behulp van het scenario in 1.2.1 *Communicatie IPT/IVT*. Met het genoemde scenario hebben we aangetoond dat een efficiënte zorgtoepassing nood heeft aan een efficiënte en transparante netwerkoplossing om de IPT's van de patiënten en de IVT's van de verpleegkundigen onderling te laten communiceren.

De communicatie in de zorgtoepassing blijft beperkt tot gegevensuitwisseling tussen een IPT en een IVT. Het onderzoek in dit hoofdstuk concentreert zich dan ook vooral op het transparante karakter van de oplossing, en minder op de netwerktopologie op zich. Het zal blijken dat een groot deel van de transparantie ingevuld kan worden aan de hand van contextgevoeligheid (Eng. *context-awareness*).

Opdat de entiteiten binnen de zorgtoepassing op een eenvoudige manier zouden kunnen communiceren, zullen we in dit hoofdstuk tot een totaaloplossing proberen te komen die op een eenvoudige manier gebruikt kan worden door de toepassing. Deze totaaloplossing bestaat enerzijds uit een communicatietechnologie, en anderzijds uit een softwarelaag, waarmee we transparante communicatie – steunend op die technologie – mogelijk zullen maken. In het vervolg zullen we naar deze softwarelaag verwijzen met de term *Transparante Communicatielaag*.

We beginnen dit hoofdstuk met de formulering van de algemene vereisten en eigenschappen van de netwerkoplossing. We bekijken eveneens of er standaardoplossingen beschikbaar zijn, en welke communicatietechnologie het meeste potentieel heeft om onze netwerkoplossing te ondersteunen.

Bluetooth zal de beste keuze blijken om een netwerkoplossing te bouwen die aan onze vereisten voldoet; we onderzoeken deze technologie dan ook in meer detail, en gaan na hoe we kunnen programmeren met Bluetooth.

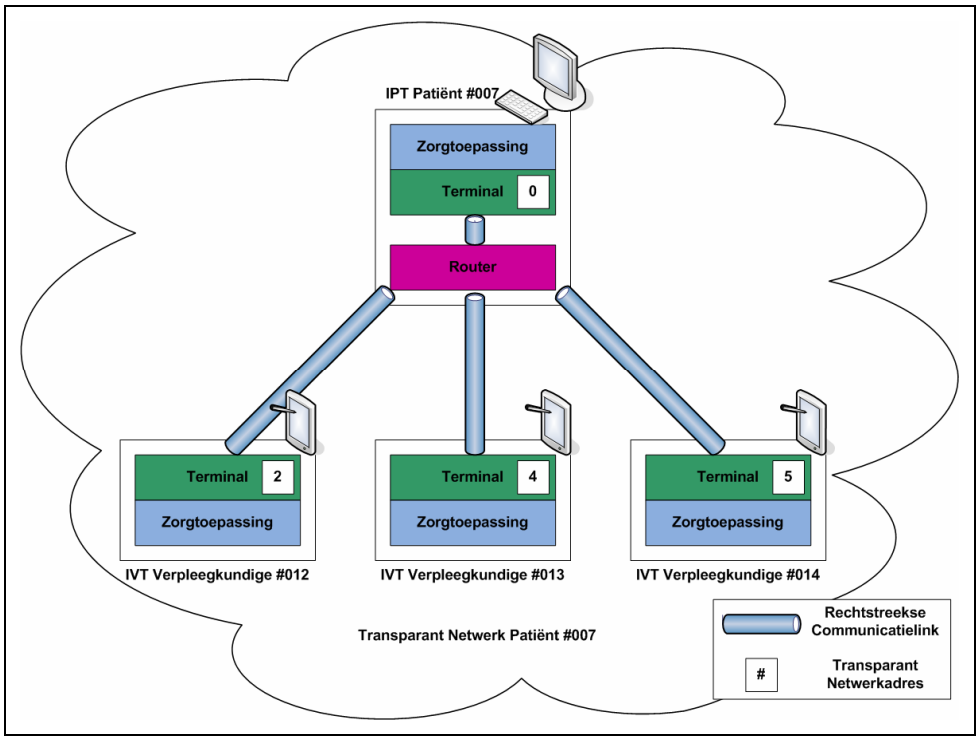
Tot slot tonen we hoe de Transparante Communicatielaag met behulp van Bluetooth geïmplementeerd werd.

## 2.1 Algemeen ontwerp

De eerste stap bij het oplossen van een probleem, is het formuleren van de vereisten waaraan onze oplossing moet voldoen. Eenmaal onze vereisten vastliggen, kunnen we deze aanwenden om de beschikbare communicatietechnologieën en standaardoplossingen te evalueren.

### 2.1.1 Entiteiten en terminologie

We introduceren in deze sectie de belangrijkste entiteiten en elementen van de Transparante Communicatielaag; we definiëren daarbij de band tussen de entiteiten van deze softwarelaag, en de elementen van de zorgtoepassing. We geven alvast een overzicht van een *Transparant Netwerk* in Figuur 6.




Figuur 6: Overzicht Transparant Netwerk

## Transparante Communicatielaag

De *Transparante Communicatielaag* is het geheel van de software ter realisatie van een framework voor de transparante vorming van ad-hoc netwerken. Alle entiteiten die binnen dit framework wensen te communiceren, dienen voorzien te zijn van een welbepaald deel van de Transparante Communicatielaag.

De IPT's en de IVT's van de zorgtoepassing zullen worden voorzien van de juiste componenten van de Transparante Communicatielaag. In Figuur 6 worden deze componenten weergeven met de *Terminal*- en *Router*-blokken in de IPT en de IVT's.

# Transparant Netwerk

Een *Transparant Netwerk* is een ad-hoc netwerk dat op een transparante manier ontstaat wanneer een aantal entiteiten, voorzien van de Transparante Communicatielaag, zich in een correcte configuratie en binnen elkaars zendbereik bevinden.

In het geval van de zorgtoepassing wensen we een Transparant Netwerk te creëren wanneer een IPT en één of meerdere IVT's zich binnen elkaars bereik bevinden. Het Transparant Netwerk dat aldus ontstaat, hoort bij een welbepaalde patiënt of ziekenkamer. Een dergelijk Transparant Netwerk wordt weergegeven in Figuur 6.

# Terminal

Een *Terminal* is een entiteit die wenst te communiceren met andere Terminals binnen de context van een Transparant Netwerk; een Terminal is hiertoe voorzien van het Terminal-component van de Transparante Communicatielaag. Een Terminal kan mobiel of vast zijn; in het geval hij mobiel is, zullen we dit een *Mobiele Terminal* noemen.

Binnen de zorgtoepassing beschouwen we de IPT als een Terminal, terwijl de IVT – als gevolg van zijn mobiel karakter – een Mobiele Terminal is. Zoals weergegeven Figuur 6 bevatten de IPT en de IVT's bijgevolg het Terminal-component van de Transparante Communicatielaag.

# Transparant Netwerkadres

Het *Transparant Netwerkadres* geldt als een unieke identiteit van een Terminal binnen de context van een Transparant Netwerk. Een Terminal verwerft een Transparant Netwerkadres voor de duur van zijn lidmaatschap van het Transparant Netwerk. De netwerkadressen worden toegekend door de Router-entiteit.

# Router

Een *Router* is verantwoordelijk voor het beheer van een Transparant Netwerk, en staat bijgevolg in voor de logische communicatiekanalen tussen de Terminals van dat Transparant Netwerk. De Router onderhoudt daartoe een rechtstreeks communicatiekanaal met elke Terminal van het Transparant Netwerk. Elk Transparant Netwerk bevat precies één Router. Merk op dat deze definitie een stervormige topologie impliceert.

Een entiteit die een rol als Router van een Transparant Netwerk vervult, kan indien nodig eveneens optreden als Terminal binnen dat Transparant Netwerk. Er is evenwel geen rechtstreekse communicatie tussen de Router-componenten van twee Transparante Netwerken.

Zoals weergegeven in Figuur 6, vervult de IPT in de context van de zorgtoepassing zowel de rol van Router als van Terminal. Bijgevolg bevat de IPT zowel het Router-component als het Terminal-component van de Transparante Communicatielaag.

## 2.1.2 Vereisten

Om onze doelstellingen te verduidelijken, formuleren we in deze sectie de vereisten die we stellen aan de te realiseren netwerkoplossing. We onderscheiden daarbij de functionele vereisten, de transparantievereisten, en de vereisten van algemene aard.

Eenmaal de vereisten vastliggen, kunnen we de waarde van bestaande oplossingen in de context van onze probleemstelling beter evalueren. Indien we vaststellen dat er geen standaardoplossingen beschikbaar zijn, zullen goedgedefinieerde vereisten ons helpen bij het ontwerp en de implementatie van een oplossing.

### Functionele vereisten

Het doel van de Transparante Communicatielaag is het aansturen van de vorming van een ad-hoc netwerk tussen de Terminals die onderling gegevens wensen uit te wisselen; het resultaat is het ontstaan van een Transparant Netwerk.

Eén van de aanwezige entiteiten zal optreden als Router, en instaan voor het beheer van het Transparant Netwerk. Dit houdt ondermeer het toekennen van Transparante Netwerkadressen, en het routeren van gegevenspakketten tussen de Terminals in. De entiteit die zal optreden als Router is vast gedefinieerd, en reeds vóór de vorming van het Transparant Netwerk gekend.

De Transparante Communicatielaag stuurt op een zo autonoom mogelijke manier de vorming van een Transparant Netwerk. Het Router-component is verantwoordelijk voor het bekendmaken van het bestaan van een Transparant Netwerk. Wanneer het Terminal-component ingeschakeld is, zoekt dit naar mogelijkheden om aan te sluiten bij een Transparant Netwerk. Indien er meerdere keuzes zijn, probeert het Terminal-component een zo goed mogelijke netwerkkeuze te maken. Hiervoor wordt gesteund op een notie van context, en op de precieze configuratie van het component.

Wanneer een Terminal lid wordt van een Transparant Netwerk, wordt deze op de hoogte gebracht van de reeds aanwezige leden van het netwerk. Bovendien worden de reeds aanwezige leden op de hoogte gebracht van het nieuwe lidmaatschap. Een Terminal, die lid

is van een Transparant Netwerk, kan op elk moment nagaan welke andere Terminals er zich op dat moment in dat Transparant Netwerk bevinden.

Het is mogelijk om met elke Terminal van hetzelfde Transparant Netwerk te communiceren via broadcast-, multicast- en unicastberichten.

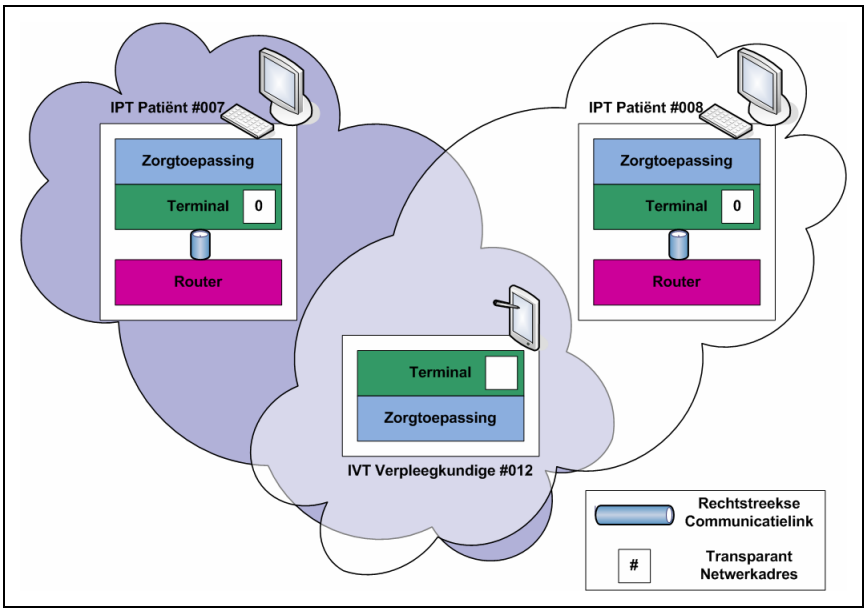
Tot slot probeert een Terminal zo goed mogelijk in te schatten wanneer een verbinding met het netwerk niet langer nodig of gewenst is. Elke Terminal van een Transparant Netwerk wordt op de hoogte gebracht wanneer een lid dat netwerk verlaat.

# Transparantievereisten

We streven naar een zo hoog mogelijke mate van transparantie bij de vorming en het beheer van een Transparant Netwerk. Bovendien zijn ook alle details, verbonden met de communicatie tussen de Terminals in het netwerk, transparant voor de gebruiker.

De gebruiker is zich niet bewust van de ondersteunende technologie voor draadloze communicatie.

Een Terminal hoeft geen voorkennis te bezitten over de mogelijke Transparante Netwerken, en kan nagaan welke Transparante Netwerken beschikbaar zijn binnen bereik.




Figuur 7: Overlappende Transparant Netwerken

Zoals weergegeven in Figuur 7, is het mogelijk dat Transparante Netwerken overlappen; in dit geval zijn er meerdere keuzes mogelijk bij de selectie van het netwerk. Indien dit het geval is, selecteert de Terminal zo autonoom mogelijk een netwerk. Voor deze selectie wordt gesteund op contextgevoeligheid, en de precieze configuratie van de Terminal, afhankelijk van de toepassing.

Eenmaal het gewenste Transparant Netwerk geselecteerd is, is het verwerven van een lidmaatschap voor dit netwerk volledig transparant voor de gebruiker. Hetzelfde geldt voor de communicatie binnen de context van het netwerk.

Een Terminal probeert zo goed mogelijk in te schatten wanneer het lidmaatschap van het Transparant Netwerk niet langer nodig is.

# Algemene vereisten

## Prestaties

We zullen de prestaties van de netwerkoplossing evalueren aan de hand van de tijd nodig om een Transparant Netwerk te vormen, en het aantal interacties met de gebruiker dat hiervoor nodig is. Het is duidelijk dat we beide prestatie-indicatoren wensen te minimaliseren.

## Robuustheid

Het uiteindelijke doel van de netwerkoplossing is het inzetten ervan in een productieomgeving; een zorginstelling is een veeleisende omgeving met een lage tolerantie voor onderbrekingen en programmafouten. De Transparante Communicatielaag en de ondersteunende communicatietechnologie zullen bijgevolg voldoende robuust moeten zijn om in een dergelijke omgeving te functioneren.

## Compatibiliteit

De doelplatformen voor de zorgtoepassing zijn enerzijds een Intel architectuur voorzien van het Microsoft Windows XP besturingssysteem, en anderzijds een Pocket PC met Microsoft Windows Mobile 5. In een latere fase wensen we de oplossing eveneens in te zetten onder Linux.

Om bruikbaar te zijn, moet de Transparante Communicatielaag compatibel zijn met de genoemde doelplatformen.

## Energieconsumptie

Eén van de doelplatformen betreft een Pocket PC met beperkte batterijcapaciteit. We houden bijgevolg rekening met deze beperking bij het realiseren van de netwerkoplossing.

## Commerciële haalbaarheid

Het commerciële aspect van de zorgtoepassing is van ondergeschikt belang in de context van dit eindwerk. Het is echter duidelijk dat in de praktijk de commerciële haalbaarheid van de oplossing afhangt van de kwaliteit en de prijs van al haar individuele onderdelen. We zullen bijgevolg enkel oplossingen overwegen waarvan de kost in verhouding staat met de waarde die ze toevoegt aan de zorgtoepassing.

# 2.1.3 Problematiek standaardoplossingen

Het heeft uiteraard geen zin om een framework te bouwen voor het transparant vormen van ad-hoc netwerken, indien er reeds adequate standaardoplossingen beschikbaar zijn.

Een eerste belangrijke vereiste waaraan een standaardoplossing moet voldoen, is de compatibiliteit ervan met de doelplatformen van de ziekenzorgtoepassing. Het gebruik van een Pocket PC voor de realisatie van de IVT blijkt de meest beperkende factor te zijn bij de keuze van een communicatietechnologie; we wensen de enkelvoudige uitbreidingssleuf van het toestel immers niet kwijt te spelen aan een uitbreidingskaart voor draadloze communicatie. Bijgevolg moeten we ons op dit vlak aanpassen aan wat de toestellen standaard te bieden hebben.

Het uitvoeren van een kort marktonderzoek toont aan dat er drie communicatietechnologieën courant ondersteund worden; het betreft IrDA[10], 802.11 WLAN[11] en Bluetooth[12]. IrDA is gebaseerd op gericht infrarood licht, terwijl de twee laatste technologieën steunen op radiotransmissie in de 2,4 GHz ISM[3] band.

In Tabel 1 wordt een overzicht gegeven van de Pocket PC's van twee toonaangevende fabrikanten, en de communicatietechnologieën die ze ondersteunen.

Tabel 1: Overzicht Pocket PC's en ondersteunde communicatietechnologieën

| Model | IrDA | 802.11 WLAN | Bluetooth | Prijs[4] (Incl. BTW) | Bron |
|---|---|---|---|---|---|
| Dell X51 | v1.2 | 802.11b | v1.2 | € 400,- | *www.Dell.be* |
| Dell X51v | v1.2 | 802.11b | v1.2 | € 484,- | *www.Dell.be* |
| HP iPAQ rx1950 | v1.2 | 802.11b | / | € 274,- | *www.PDAShop.be* |
| HP iPAQ hx2190 | v1.2 | / | v1.2 | € 321,- | *www.PDAShop.be* |
| HP iPAQ hx2490 | v1.2 | 802.11b | v1.2 | € 388,- | *www.PDAShop.be* |
| HP iPAQ hx2790 | v1.2 | 802.11b | v1.2 | € 480,- | *www.PDAShop.be* |

In wat volgt bekijken we in welke mate de beschikbare technologieën standaardoplossingen voorzien, en of ze eventueel in aanmerking komen om de Transparante Communicatielaag te ondersteunen.

[3] *Industrial, Scientific and Medical*

[4] Prijzen april 2006

# IrDA

In 1994 werd de Infrared Data Association (IrDA)[10] opgericht; de IrDA is een samenwerkingsverband met als doel de standaardisering en ontwikkeling van infrarood als draadloze communicatietechnologie.

In 2004 waren 98% van de verkochte PDA's uitgerust met IrDA[13]; geen enkele andere draadloze communicatietechnologie wordt in die mate ondersteund.

IrDA maakt gebruik van gericht infrarood licht, en dringt niet doorheen obstakels. Er is bijgevolg een zichtlijn nodig tussen de communicerende toestellen. De maximale onderlinge afstand is meestal beperkt tot 1,5 m, en de werkingshoek bedraagt ongeveer 30°.

Hoewel de nieuwste IrDA standaard transfersnelheden tot 4 Mbps ondersteunt, zijn de prestaties van de IrDA poorten van de beschouwde Pocket PC's beperkt tot 115 Kbps.

De afhankelijkheid van IrDA van een zichtlijn is in strijd met de transparantievereisten, aangezien de gebruiker de communicatiepartners dicht bij elkaar moet brengen, en deze vrij precies moet richten. De gebruiker moet zich bijgevolg bewust zijn van de eigenschappen van de communicatietechnologie.

Aangezien we de genoemde tekortkoming niet met intelligente software kunnen verhelpen, beschouwen we IrDA als ongeschikt voor onze doeleinden.

# IEEE 802.11 Wireless LAN

802.11 is een populaire netwerktechnologie voor draadloze LAN's. De standaardisering van de technologie is in handen van het IEEE[11].

802.11 wil een draadloos alternatief zijn voor bekabelde netwerken, zoals bijvoorbeeld netwerken gebaseerd op Ethernet of Token Ring. De IEEE 802.11-standaard kan gebruik maken van verschillende frequentiebanden; de meest populaire variant 802.11b maakt gebruik van de licentievrije 2,4 GHz ISM band.

802.11b is in staat tot transfersnelheden van maximum 11 Mbps. Gezien haar bedoelde toepassing bezit de technologie een vrij hoog bereik; binnenshuis varieert deze van ongeveer 30 m tot zelfs 90 m. Een 802.11b netwerk wordt gekarakteriseerd door een kanaalnummer; er zijn 13 dergelijke nummers beschikbaar.

We hebben de ambitie om een netwerk te definiëren voor elke kamer, of zelfs voor elk ziekenbed; het is duidelijk dat het aantal kanaalnummers in dit geval ontoereikend is voor de gemiddelde zorginstelling. Gezien het vrij grote bereik zal er namelijk overlap optreden, en is het in de praktijk onmogelijk om een netwerk te selecteren, zelfs met een beperkte interactie met de gebruiker. Deze laatste zal immers uit een hele lange lijst netwerken zijn

keuze moeten maken. Dit zorgt ervoor dat we de toepassing onmogelijk voldoende transparant kunnen maken.

In vergelijking met de andere beschouwde technologieën, consumeert 802.11b significant meer energie; in de praktijk is het zelfs onmogelijk dat een verpleegkundige zijn dienstronde doorkomt zonder extra batterijcapaciteit of herlaadmogelijkheden.

De genoemde tekortkomingen zijn gekende eigenschappen van 802.11b, en vormen geen bezwaar voor de meeste courante toepassingen van de technologie. We hebben echter aangetoond dat 802.11b niet aan onze vereisten voldoet, en bijgevolg ongeschikt is voor de ondersteuning van de Transparante Communicatielaag.

# Bluetooth

Bluetooth is een draadloze communicatietechnologie die aanvankelijk ontwikkeld is binnen de telecommunicatie-industrie. Tegenwoordig gebeurt de standaardisering in de context van de Bluetooth Special Intrest Group (SIG)[12].

Bluetooth maakt net als 802.11b gebruik van de licentievrije 2,4 GHz band, maar is bedoeld voor *Personal Area Networks* of *PAN's*. Een PAN is een netwerk met een bereik beperkt tot ongeveer 10 m rond een persoon. In de praktijk wordt Bluetooth voornamelijk als kabelvervanger gebruikt, bijvoorbeeld in handenvrije GSM kits.

Als gevolg van haar dienstgeoriënteerde architectuur, is Bluetooth meer dan louter een communicatietechnologie. Bluetooth definieert *profielen* ter ondersteuning van deze architectuur; een profiel bepaalt hoe de transportfunctionaliteit van Bluetooth ingezet zal worden om transportdiensten te verschaffen aan een welbepaalde dienst. Er bestaat met andere woorden nooit zomaar een communicatielink tussen twee Bluetooth apparaten; een dergelijke verbinding bestaat steeds in de context van een welbepaalde dienst, en wordt geconfigureerd volgens de regels van het passende profiel.

Verder bezit Bluetooth uitgebreide voorzieningen ter ondersteuning van ad-hoc netwerken; deze omvatten ondermeer de mogelijkheid om apparaten binnen bereik op te sporen, en om de diensten aangeboden door een bepaald toestel op te vragen.

Het aantal Bluetooth netwerken dat op een bepaalde locatie kan coëxisteren, bezit geen harde bovengrens. Het is echter zo dat met het aantal netwerken ook de onderlinge interferentie toeneemt, en bijgevolg de individuele prestaties van elk van de aanwezige netwerken dalen.

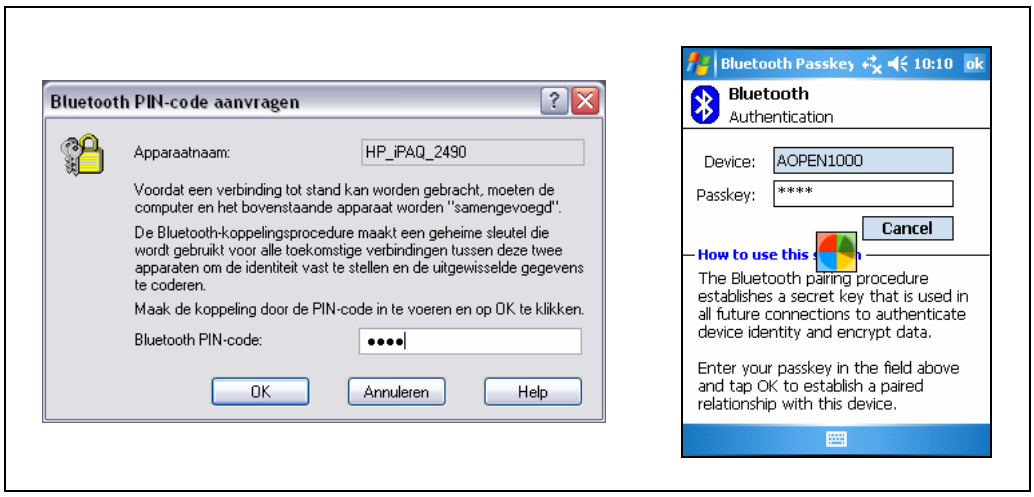
We beschikken over de volgende hardware en software om na te gaan of één van de standaard Bluetooth diensten in aanmerking komt als transparante netwerkoplossing:

- Een Windows XP computer, voorzien van een Belkin Bluetooth USB dongle. De dongle wordt geleverd met de laatste versie (v. 4.0) van de Broadcom Bluetooth software, die een uitgebreid gamma Bluetooth diensten aanbiedt.
- Een HP iPAQ hx2490 Pocket PC, eveneens voorzien van de nieuwste Broadcom Bluetooth software.

De standaarddiensten die eventueel in aanmerking zouden komen als standaardoplossing zijn de *Netwerktoegang* dienst en de *Seriële Bluetooth-poort* dienst. We bespreken deze hieronder kort en geven aan wat de problemen zijn.

# Netwerktoegang

Kort samengevat is de Netwerktoegang dienst (Eng. *PAN service*) een implementatie van IP over Bluetooth. De dienst emuleert in de beide deelnemende communicatiepartners een netwerkadapter, en een cross-over kabel tussen deze adapters. Het besturingssysteem merkt geen verschil met een standaard netwerkadapter.




Figuur 8: Bluetooth authenticatieprocedure onder Windows XP en Windows Mobile 5

Jammer genoeg gooien de beveiligingsvoorzieningen van Bluetooth roet in het eten; telkens wanneer twee apparaten die nog niet eerder communiceerden dit wensen te doen, moeten ze een authenticatieprocedure doorlopen. Zoals getoond in Figuur 8, houdt deze procedure het invoeren van een PIN code op beide schermen in. Het is bij deze dienst niet mogelijk om de authenticatie uit te schakelen; Bluetooth emuleert immers eveneens de fysieke veiligheid van de cross-over netwerkkabel.

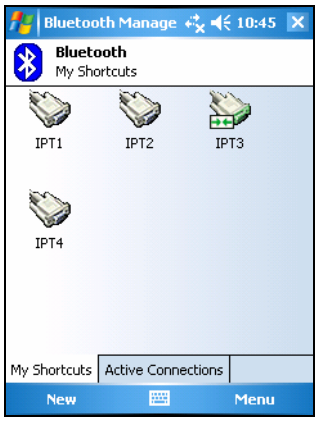
Het is duidelijk dat deze oplossing onmogelijk voldoende transparant gemaakt kan worden.

# Seriële Bluetooth-poort

De Seriële Bluetooth-poort dienst emuleert standaard seriële poorten op de beide deelnemende communicatiepartners, en het bestaan van een seriële kabel daartussen. Deze dienst is zo'n beetje het manusje-van-alles onder de Bluetooth diensten.

In tegenstelling met de Netwerktoegang dienst, kan men bij deze dienst de beveiligingsvoorzieningen uit Figuur 8 uitschakelen. Jammer genoeg is het evenmin mogelijk om de Seriële Bluetooth-poort voldoende transparant te maken. De problematiek wordt weergegeven in Figuur 9; men moet op elke communicatiepartner manueel een uitgaande geëmuleerde poort creëren, voor elk toestel waarmee men wil communiceren. Voor de IVT zou dit betekenen dat er een uitgaande poort moet gecreëerd worden voor elke IPT, zoals weergegeven in Figuur 9.

We besluiten dat de Seriële Bluetooth-poort dienst niet geschikt is als transparante netwerkoplossing.




Figuur 9: Problematiek Seriële Bluetooth-poort dienst

# Conclusie

We hebben in deze sectie aangetoond dat de hedendaagse Pocket PC's een drietal populaire communicatietechnologieën ondersteunen; het betreft IrDA, 802.11b WLAN en Bluetooth.

Als gevolg van de nood aan een zichtlijn tussen de communicatiepartners, en het erg beperkte bereik van de technologie, concludeerden we dat IrDA niet in aanmerking komt voor onze doeleinden. 802.11b WLAN bleek om praktische redenen, zoals een te groot bereik en een tekort aan kanaalnummers, evenmin geschikt als transparante netwerkoplossing.

Uit ons korte onderzoek is gebleken dat Bluetooth het grootste potentieel bezit voor de ondersteuning van onze netwerkoplossing. De technische karakteristieken van de technologie voldoen volledig aan onze vereisten. Jammer genoeg zijn de standaarddiensten van deze technologie niet voldoende transparant. In tegenstelling tot de overige beschouwde diensten, vermoeden we dat we dit transparantieprobleem kunnen oplossen met intelligente software. We zullen dan ook op Bluetooth steunen om, met de tussenkomst van de Transparante Communicatielaag, in een transparante netwerkoplossing te voorzien.

# 2.2 Bluetooth

In het voorgaande onderdeel hebben we aangetoond dat Bluetooth het grootste potentieel bezit voor de ondersteuning van een transparante netwerkoplossing. We geven in dit hoofdstuk dan ook een inleiding tot de razend populaire Bluetooth technologie.

We bekijken kort hoe de Bluetooth standaardspecificatie tot stand kwam, en wat de drijvende kracht achter de ontwikkeling van de technologie geweest is. Vervolgens geven we een overzicht van de algemene operationele werking van Bluetooth. Tot slot bestuderen we de standaardspecificatie; we focussen daarbij op een aantal specifieke elementen die in het bijzonder van belang zullen zijn voor de ontwikkeling van de Transparante Communicatielaag.

## 2.2.1 Oorsprong en standaardisering

Bluetooth vindt haar oorsprong in de onderzoekslabo's van het Zweedse telecommunicatiebedrijf Ericsson. In 1994 deed een onderzoeksgroep er een studie naar de haalbaarheid van de ontwikkeling van een kabelvervangende, draadloze netwerktechnologie met een klein bereik van ongeveer 10 m; de nadruk lag hierbij op een beperkte productiekost en een lage energieconsumptie. Het initiatief kon op heel wat interesse rekenen van de bedrijfswereld en, wat begon als een onderzoeksproject met beperkte middelen, monde in 1998 uit in de *Bluetooth Special Interest Group (SIG)*[12].

De Bluetooth SIG werd aanvankelijk opgericht door Ericsson, Intel, IBM, Nokia en Toshiba, en is ondertussen uitgegroeid tot een consortium met meer dan 2000 leden, waaronder een aantal belangrijke spelers uit diverse sectoren. De doelstelling van de SIG is de ontwikkeling van Bluetooth binnen een internationaal samenwerkingsverband en het promoten van het gebruik van de technologie in producten. Bij de ontwikkeling en commercialisering van de technologie, stelt het consortium de volgende vereisten voorop: integratie van zender en ontvanger op één chip, lage kost, robuuste gegevenstransmissie en laag energieverbruik.

Het samenwerkingsverband resulteert periodiek in de publicatie van een publiek toegankelijke standaardspecificatie. We zullen in wat volgt de meest recente publicatie gebruiken, namelijk versie 2.0 + EDR[14].

## 2.2.2 Operationele werking

Bluetooth is van de meet af aan bedoeld geweest voor gebruik in ad-hoc netwerken met een klein bereik van ongeveer 10 m. Een netwerk met een straal van 10 m rond de gebruiker

worden vaak een *Personal Area Network (PAN)* genoemd. Het is duidelijk dat het ad-hoc karakter van Bluetooth een aantal bijzondere voorzieningen vereist. De ontwerpers voorzagen Bluetooth dan ook van een aangepaste topologie, evenals een aantal procedures en modi, om het ad-hoc karakter te ondersteunen.

# Topologie

Het centrale begrip in de Bluetooth netwerktopologie is het *piconet*; dit is een verzameling Bluetooth apparaten waarvan de fysieke lagen onderling gesynchroniseerd zijn. Een piconet ontstaat wanneer twee of meerdere apparaten eenzelfde fysiek kanaal gebruiken. De communicatie tussen twee Bluetooth stations gebeurt steeds binnen de context van een piconet.

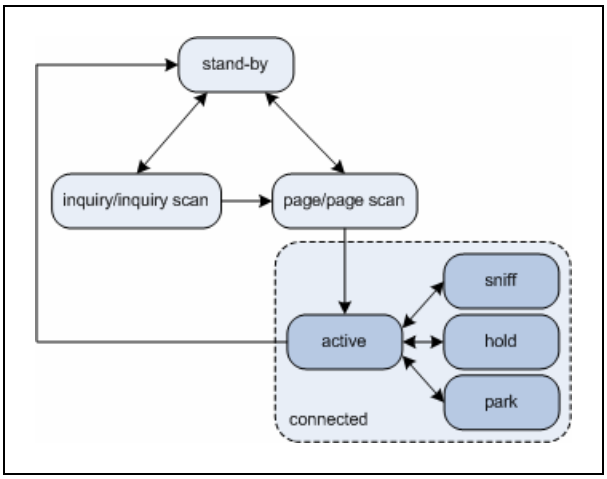
Binnen een bepaald piconet is er steeds precies één toestel dat optreedt als *master*; de overige apparaten werken in *slave* modus. Het master apparaat geeft ondermeer aan welke de gevolgde hoppingvolgorde is, en beheert de klok van het piconet. De slave stations synchroniseren hun hoppingvolgorde en klok met die van de master. Er kunnen ten hoogste 256 apparaten lid zijn van eenzelfde piconet. Het aantal toestellen dat simultaan actief kan deelnemen aan de communicatie is beperkt tot 8, waaronder de master. Rechtstreekse communicatie tussen twee apparaten is enkel mogelijk tussen een master en een slave. Opdat twee slaves gegevens zouden kunnen uitwisselen, moet de master van het piconet optreden als doorgeefluik; de master vervult dus de rol van *switch* in het piconet.

Meerdere piconetten kunnen gelijktijdig en onafhankelijk van elkaar bestaan op eenzelfde locatie. Elk toestel kan in ten hoogste één piconet de rol van master aannemen. Het aantal piconetten waarin een apparaat voorkomt in de slave rol is in theorie onbeperkt. Er zijn in de standaard procedures voorzien die het mogelijk maken dat de master en een slave in een piconet van rol wisselen.

Indien twee of meer piconetten één Bluetooth apparaat gemeenschappelijk hebben, ontstaat er een *scatternet*. In deze situatie is er communicatie mogelijk tussen apparaten van de verschillende piconetten binnen het scatternet. Alle communicatie tussen de piconetten gebeurt dan uiteraard via het gemeenschappelijke apparaat, dat nu herhaaldelijk van het ene naar het andere piconet moet springen.

# Procedures en operationele modi

In de context van een piconet doorloopt een Bluetooth apparaat gedurende zijn werking een aantal toestanden. De mogelijke toestanden worden weergegeven in Figuur 10. Merk op dat een gegeven apparaat zich tezelfdertijd in meerdere modi kan bevinden als het deelneemt aan verschillende piconetten. In de context van één bepaald piconet bevindt een gegeven toestel zich echter in precies één van de modi uit de figuur.




Figuur 10: De operationele modi van Bluetooth

Gezien het belang van de operationele modi en procedures, zullen we deze hieronder beknopt bespreken. Een uitvoerige bespreking van elk van de modi valt buiten het bestek van deze scriptie. We verwijzen dan ook naar [15] en [16] voor een uitvoerige bespreking van de toestanden. De precieze technische details worden beschreven in [14].

## Inquiry Mode/Inquiry Scan Mode

De *inquiry* mode en de bijhorende procedure zijn van groot belang voor de ondersteuning van het ad-hoc karakter van Bluetooth.

Een Bluetooth apparaat maakt gebruik van de *inquiry* of *device discovery procedure* om apparaten binnen bereik op te sporen. Apparaten die wensen gevonden te worden door andere toestellen dienen zich in de *inquiry scan mode* te bevinden. Het zoekende apparaat verstuurt actief aanvragen, waarop geantwoord wordt door de apparaten in inquiry scan mode.

Een Bluetooth apparaat bezit een wereldwijd uniek apparaatadres; het resultaat van de inquiry procedure is het verwerven van de apparaatadressen van de toestellen binnen bereik.

De device discovery procedure zal een essentieel onderdeel blijken bij de realisatie van de Transparante Communicatielaag. Concreet zullen we deze procedure gebruiken voor het inschatten van de context van de Mobiele Terminal, en voor het opsporen van toestellen die kunnen optreden als Router. We bespreken dit in 2.3.1 *Zoeken naar potentiële Routers*.

## Page Mode/Page Scan Mode

Wanneer een apparaat een verbinding wenst te maken met een ander toestel binnen zijn bereik, neemt dit apparaat hiertoe de *page mode* aan. In deze modus verstuurt het apparaat verbindingsaanvragen naar de gewenste communicatiepartner. Indien de communicatiepartner zich in de *page scan* toestand bevindt, ontvangt deze de aanvragen en wordt er aldus een gegevensverbinding opgebouwd tussen de toestellen.

Om de verbindingsprocedure te initialiseren, heeft men enkel het apparaatadres van de gewenste communicatiepartner nodig. Dit adres kan bekomen worden met de device discovery procedure, maar men kan het bijvoorbeeld ook vragen aan de gebruiker of ophalen uit een databank. Indien het Bluetooth apparaatadres van het toestel gekend is, hoeft er dus geen tijd gespendeerd te worden aan de device discovery procedure.

## Connected Mode

Het voorgaande verbindingsproces zorgt voor de verbinding van de toestellen in een piconet. Men zegt dat de toestellen zich in de *connected* modus bevinden.

Zoals weergegeven in Figuur 10, is de connected toestand in feite een verzamelnaam voor een aantal meer specifieke subtoestanden; we onderscheiden de *active*, *hold*, *sniff* en *park* modi. Al deze subtoestanden bezitten de gemeenschappelijke eigenschap dat ze de synchronisatie met het piconet in stand houden.

De active modus levert de beste prestaties voor wat betreft de doorvoer, de latentietijd en de betrouwbaarheid bij het verzenden en ontvangen van gegevens. In deze toestand is het energieverbruik dan ook het hoogst. De hold, sniff en park modi zijn energiebesparende toestanden waarin de prestaties van de interface afnemen.

Het lidmaatschap van een piconet impliceert de toekenning van een adres voor de adressering van het toestel binnen het piconet; in de hold, sniff en park modi is dit een *Active Member Address (AMA)*, terwijl de park modus een *Parked Member Address (PMA)* impliceert. We bespreken zo meteen kort beide adresseringsmethodes.

# Identificatie en adressering

Bluetooth voorziet verschillende manieren om een apparaat te identificeren en te adresseren. Welke manier precies van toepassing is, hangt af van de toestand waarin het toestel zich bevindt, en het doel van de adressering.

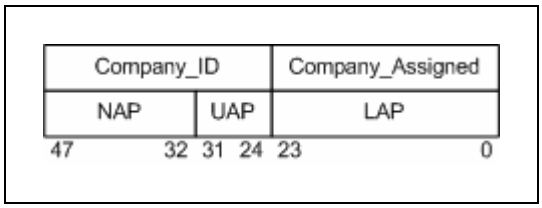
## Bluetooth Device Address

Elk Bluetooth apparaat bezit een wereldwijd unieke identiteit, het *Bluetooth Device Address*. Dit is een apparaatadres van 6 bytes dat, wat de vorm betreft, identiek is aan het fysieke adres van klassieke netwerkhardware. De apparaatadressen worden beheerd en toegekend door de IEEE Registration Authority[17].

De toekenning van apparaatadressen gebeurt niet lukraak; het adres bestaat namelijk uit een aantal velden die een bepaalde betekenis bezitten.

Zoals blijkt uit Figuur 11, is het apparaatadres onderverdeeld in *Company_ID* en *Company_Assigned* velden. Zoals de benaming reeds laat vermoeden, is het Company_ID veld verbonden met een bepaald bedrijf. Het betreft meestal de producent van de Bluetooth

chip, en dus niet noodzakelijk de fabrikant van het eindproduct, zoals bijvoorbeeld van een Pocket PC.




Figuur 11: Formaat Bluetooth apparaatadres

Een producent wordt geacht elk exemplaar van zijn producten te voorzien van een apparaatadres dat begint met zijn Company_ID. De inhoud van het tweede deel van het adres kan vrij gekozen worden door de producent. Elk product van de producent moet echter een uniek Company_Assigned veld bezitten; op die manier garandeert men de wereldwijde uniciteit van het apparaatadres.

Bluetooth maakt voor zijn werking gebruik van bepaalde delen van het apparaatadres; een verdeling van het adres in *NAP*, *UAP* en *LAP* velden steunt op deze observatie. Het onderdeel van het apparaatadres dat Bluetooth gebruikt voor zijn interne keuken, wordt *significant* genoemd. Het significante gedeelte wordt vervolgens nogmaals opgedeeld in een *Upper Address Part (UAP)* van 8 bits, en een *Lower Address Part (LAP)* van 24 bits. De overige 16 bits zijn niet van belang voor de interne werking, en kregen bijgevolg de benaming *Non-significant Address Part (NAP).*

Bij de bespreking van de Bluetooth protocol stack in 2.2.3, gaan we dieper in op het belang van de significante delen van het apparaatadres voor de werking van de technologie.

## Friendly Name

De *Friendly Name* is een identiteit die door de gebruiker van een toestel ingesteld kan worden, en is dan ook niet uniek. Vanuit het standpunt van de eindgebruiker is dit de meest gebruikte adresseringsmethode, bijvoorbeeld bij de manuele selectie van een communicatiepartner na het uitvoeren van een device discovery procedure.

## Active Member Address

Indien een apparaat gesynchroniseerd is met een piconet en bovendien actief deelneemt aan de communicatie daarbinnen, bezit het apparaat een 3 bits *Active Member Address (AMA).* Er kunnen ten hoogste acht apparaten gelijktijdig actief zijn in de context van eenzelfde piconet; één van deze apparaten is de master.

## Parked Member Address

Het *Parked Member Address (PMA)* is een 8 bits adres dat toegekend wordt aan een apparaat dat zich synchroniseert met een piconet maar niet actief deelneemt aan de

communicatie. Aan de hand van dit secundaire adres, kunnen tot 256 apparaten lid zijn van eenzelfde piconet.

# Dienstgebaseerde architectuur en beveiliging

Door zijn dienstgebaseerde architectuur, is Bluetooth veel meer dan louter een communicatietechnologie. Als gevolg van het dienstgebaseerde karakter, wordt een gegevensverbinding, tussen twee apparaten in een piconet, uitsluitend opgebouwd binnen de context van een welbepaalde toepassing of dienst. Ook de beveiligingsvoorzieningen van Bluetooth zijn dienstafhankelijk.

Ter ondersteuning van zijn dienstgeoriënteerd karakter, voorziet Bluetooth applicatieprofielen en een protocol voor het adverteren en zoeken van diensten. We bespreken hieronder kort beide voorzieningen.

## Applicatieprofielen

Applicatieprofielen zijn modulaire gestandaardiseerde elementen van de Bluetooth protocol stack. Elk profiel biedt transportkanalen aan voor de implementatie van diensten van een bepaalde klasse. Verder staat het profiel eveneens in voor de beveiligingsaspecten geassocieerd met de dienstklasse. Voor zijn werking steunt een profiel op de onderliggende lagen van de protocol stack.

Merk op dat een profiel zelf geen toepassing is, maar slechts een abstractie van de rest van de protocol stack. Via een interface worden de diensten van deze abstractie aangeboden aan hogere lagen, om de implementatie van een bepaalde klasse van diensten te ondersteunen. Een profiel beschrijft met andere woorden compatibele contactpunten, die een applicatie bij de implementatie van een dienst kan gebruiken om te communiceren met gelijkaardige toepassingen en diensten. Door de profielen te standaardiseren, is het mogelijk om de interoperabiliteit te garanderen tussen apparaten en gelijkaardige toepassingen van verschillende producenten.

## Beveiliging

Bluetooth biedt ondersteuning voor authenticatie en encryptie. In het kader van de dienstgerichte architectuur kan elk applicatieprofiel een eigen beveiligingspolitiek definiëren.

De voorzieningen van Bluetooth voor authenticatie werken op het niveau van apparaten, en worden ondergebracht in een procedure voor paarvorming (Eng. *pairing*). De Bluetooth paarvormingsprocedure houdt in dat twee communicatiepartners onderling hun identiteit bewijzen; daartoe wordt aan de gebruiker gevraagd om op beide apparaten eenzelfde PIN code in te voeren. Wanneer de codes identiek zijn, bewijst dit de identiteiten van de betrokken apparaten, en bovendien dat de communicatie weldegelijk gewenst is. De

paarvormingsprocedure dient ten hoogste één keer uitgevoerd te worden voor elk paar toestellen. We maakten reeds kennis met de paarvormingsprocedure bij de bespreking van de Netwerktoegang dienst in 2.1.3. Het is evenwel zo dat niet alle diensten het uitvoeren van de paringsprocedure vereisen.

De voorzieningen voor encryptie werken op het niveau van een dienst. Indien het karakter van de dienst dit vereist, kan de geassocieerde communicatielink versleuteld worden. Aangezien hiervoor de PIN code van de Bluetooth authenticatieprocedure gebruikt wordt, moeten de apparaten een paar vormen, alvorens men encryptie kan gebruiken.

We merken tot slot op dat de beveiligingsvoorzieningen van Bluetooth niet bruikbaar zijn in een transparant scenario; dit zou immers vereisen dat de programmacode de PIN codes instelt, hetgeen inherent onveilig is. De beveiligingaspecten van de transparante netwerkoplossing worden daarom toevertrouwd aan de applicatielaag; voor meer informatie hieromtrent verwijzen we naar [6].

### Service Discovery Protocol

Het *Service Discovery Protocol (SDP)* is een protocol waarmee een apparaat zijn diensten kan adverteren, en de diensten van andere toestellen kan opvragen.

We behandelen het SDP in meer detail in 2.3.2 *Selectie van potentiële Routers*.

## 2.2.3 De Bluetooth protocol stack

De elementen die samen de functionaliteit van een Bluetooth toepassing realiseren, kunnen ondergebracht worden in een gelaagde en modulaire structuur, de Bluetooth protocol stack. De elementen van elke laag realiseren diensten die gebruikt kunnen worden door hogere elementen van de stack, en steunen hiervoor op de functionaliteit van de onderliggende lagen. In Figuur 12 wordt een weergave getoond van een Bluetooth protocol stack.

Zowat elke producent heeft zijn een eigen interpretatie van de protocol stack. Om de productdiversiteit te stimuleren, laat men de fabrikanten bovendien heel wat vrijheid bij de keuze of een bepaald stackelement al dan niet geïmplementeerd wordt in het eindproduct. Een fabrikant selecteert met andere woorden zelf welke elementen van de protocol stack hij wenst te implementeren in zijn product. De stack uit Figuur 12 wordt evenwel beschouwd als een minimale implementatie.

In Figuur 12 tonen we het gedeelte van de protocol stack dat gedefinieerd wordt in de *Bluetooth Core Specification*[14]. Zoals weergegeven in de figuur, kan men de protocol stack opsplitsen in een *controller* gedeelte en een *host* gedeelte.

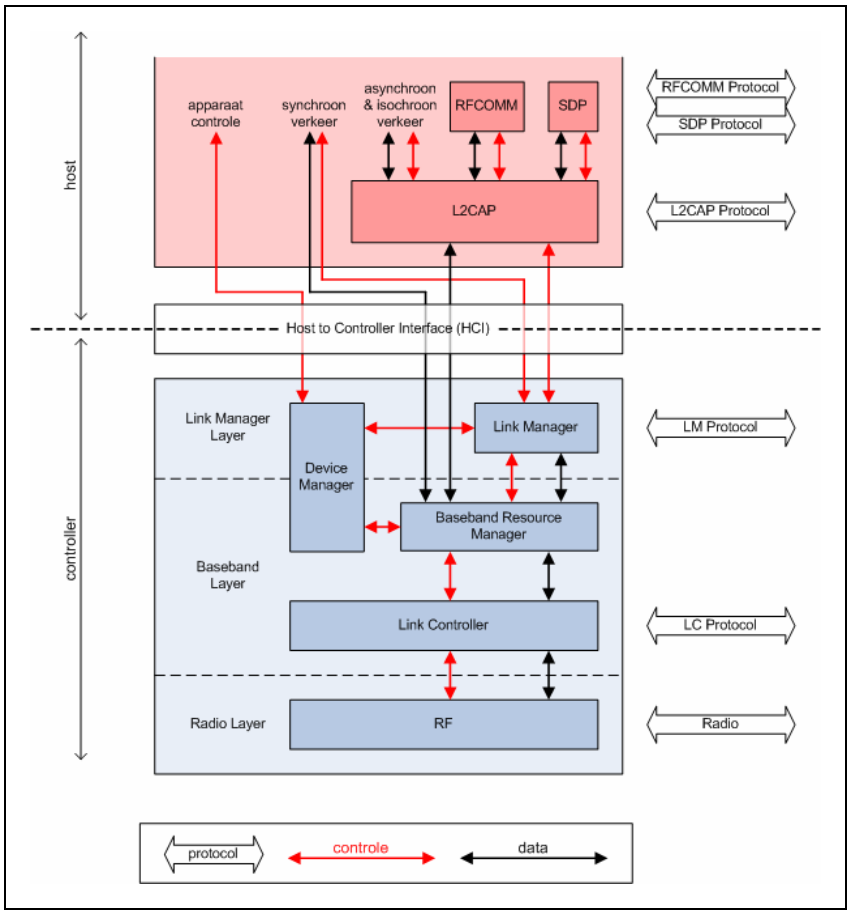
De Bluetooth controller wordt praktisch altijd gerealiseerd met behulp van hardware en omvat de *Radio Layer*, de *Baseband Layer*, de *Link Manager Layer* en het controller

gedeelte van de *Host to Controller Interface (HCI)*. Een Bluetooth USB dongle is een voorbeeld van een implementatie van de controller.

De HCI is de interfacelaag die zorgt voor een wederzijdse abstractie van host en controller. In het voorbeeld van een Bluetooth USB dongle, zorgt de HCI voor de communicatie tussen de twee gedeelten van de protocol stack via USB.

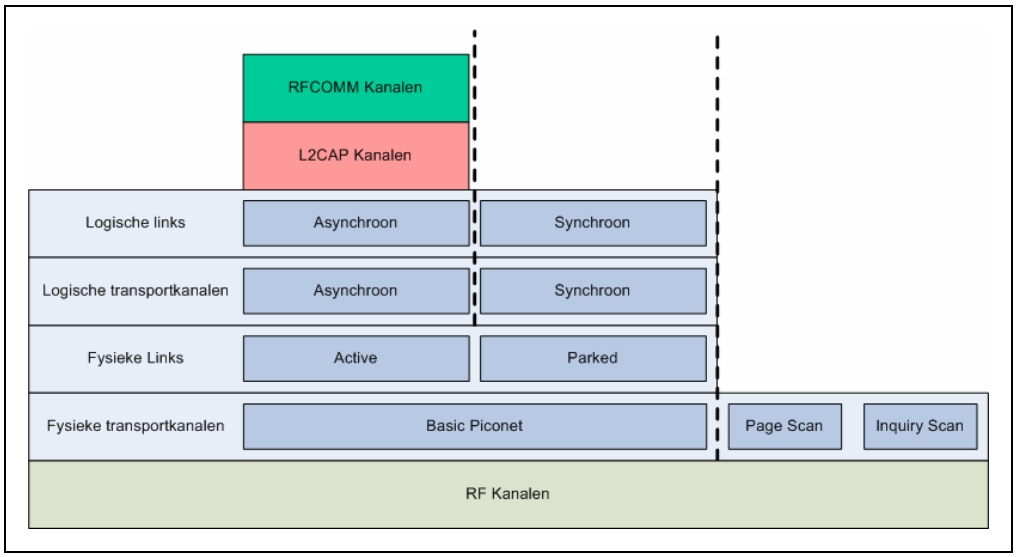
Het host gedeelte bestaat uit het corresponderende deel van de HCI, het *Logical Link Control and Adaptation Protocol (L2CAP),* het *Service Discovery Protocol (SDP)* en het *RFCOMM* protocol. Daarnaast kan het host gedeelte een arbitrair aantal applicatieprofielen bevatten, afhankelijk van de precieze implementatie. In het geval van de Bluetooth USB dongle, wordt de host implementatie meegeleverd als systeemsoftware.

Boven de host komen eventuele toepassingen.




Figuur 12: Een weergave van de Bluetooth protocol stack

In wat volgt bespreken we de elementen van de protocol stack; we gaan hierbij op zoek naar elementen die nuttig zullen zijn bij de implementatie van de Transparante Communicatielaag, zoals het Device Discovery Protocol, het Service Discovery Protocol en een gegevenstransport dat geschikt is voor onze doeleinden. We geven alvast een overzicht van de Bluetooth gegevenstransportarchitectuur in Figuur 13.




Figuur 13: De Bluetooth gegevenstransportarchitectuur

# Controller

We onderzoeken hieronder de elementen van de controller. We willen met dit onderzoek nagaan welke de diensten zijn die de controller aanbiedt, en wat de precieze kwaliteit van deze diensten is. Eenmaal we deze kennis bezitten, kunnen we de controller als een *black box* beschouwen tijdens de ontwikkeling van onze toepassing.

## Radio layer

De Radio layer is de fysieke laag van Bluetooth en zorgt voor de communicatie over het draadloze medium. De Radio layer biedt hiertoe 79 RF kanalen aan op de licentievrije 2,4 GHz band. De gebruikte frequenties zijn 2,4000 GHz tot en met 2,4835 GHz.

Afhankelijk van de karakteristieken van de Radio Layer, zijn Bluetooth apparaten in drie vermogenklassen beschikbaar. De eigenschappen van de vermogensklassen worden getoond in Tabel 2. De meeste apparaten behoren tot vermogensklasse 2 of 3.

Tabel 2: Bluetooth vermogensklassen

| **Vermogensklasse** | **Maximaal Uitgangsvermogen** | **Bereik** | **Vermogenscontrole** |
|---|---|---|---|
| 1 | 100 mW (20 dBm) | ~100 m | Verplicht |
| 2 | 2,5 mW (4 dBm) | ~10 m | Optioneel |
| 3 | 1 mW (0 dBm) | ~1 m | Optioneel |

Merk op dat een Bluetooth apparaat uitgerust kan zijn met vermogensregeling; het doel hiervan is het minimaliseren van de energieconsumptie, het maximaliseren van de kanaalkwaliteit en het optimaliseren van het algemene interferentieniveau. De eventuele vermogenscontrole wordt aangestuurd vanuit de Link Manager.

# Baseband layer

De Baseband layer is in staat om fysieke kanalen tot stand te brengen, steunend op de RF kanalen van de Radio layer.

## Fysieke kanalen

Bluetooth steunt op *Frequency Hopping Spread Spectrum (FHSS)* voor het creëren van fysieke kanalen; bij deze techniek bestaat een fysiek kanaal uit een sequentie van RF kanalen, die achtereenvolgens worden benaderd. De sequentie wordt herhaald in de tijd. Voor een overzicht van FHSS verwijzen we naar [16].

Een fysiek kanaal komt tot stand door het volgen van een bepaalde FHSS hopping volgorde; de Baseband Layer zorgt in deze context voor de synchronisatie van de apparaatklok, en het aanhouden van de juiste hopping sequentie. Afhankelijk van de precieze manier waarop de laag dit doet, kan men fysieke kanalen bekomen met verschillende eigenschappen. De precieze hopping sequentie wordt – afhankelijk van het kanaaltype – afgeleid van een bepaald Bluetooth apparaatadres. Een fysiek kanaal is onbetrouwbaar.

We bespreken hieronder kort de belangrijkste types.

### Basic Piconet Physical Channel

Binnen een piconet bestaat er precies één *Basic Piconet Physical Channel*, waarmee apparaten in de context van dat piconet onderling kunnen communiceren. De doorvoercapaciteit van dat kanaal wordt bijgevolg gedeeld door alle leden van het piconet. De hopping sequentie van het kanaal wordt ondermeer bepaald door het apparaatadres van de master van het piconet.

### Page Scan Physical Channel

Het *Page Scan Physical Channel* wordt enkel gebruikt door de paging procedure voor de signalisatie tijdens de verbindingsopbouw, en bezit bijgevolg een vrij korte levensduur. Voor het realiseren van dit kanaaltype wordt een korte hopping sequentie relatief traag doorlopen; de sequentie wordt ondermeer bepaalde door het apparaatadres van de communicatiepartner waarmee men een verbinding wenst te creëren.

### Inquiry Scan Physical Channel

Het *Inquiry Scan Physical Channel* wordt gebruikt door de inquiry procedure voor het zoeken van apparaten in de nabijheid. Het kanaal ontstaat door het relatief traag doorlopen van een korte hopping sequentie; de hopping volgorde wordt afgeleid uit een daartoe gereserveerd apparaatadres. Een belangrijke eigenschap van dit kanaal is het ongecoördineerde karakter; apparaten kunnen het kanaal namelijk naar wens benaderen en op die manier botsingen veroorzaken.

## Fysieke links

Een fysieke link is een abstractie van een Baseband connectie tussen Bluetooth apparaten. Fysieke links laten toe om gegevens en controle-informatie van hogere elementen te transporteren met behulp van een fysiek kanaal. Elke fysieke link is daartoe geassocieerd met precies één fysiek kanaal; zoals weergegeven in Figuur 13, steunt een fysieke link in de praktijk altijd op het basic piconet physical channel. Een fysieke link is onbetrouwbaar.

Een fysieke link voegt een notie van *toestand* toe aan een fysiek kanaal, in overeenstemming met de stroombesparingsmodi en de bijhorende adresseringsmogelijkheden. Er zijn bijgevolg twee types fysieke links, namelijk *Active* en *Parked*.

## Logische transportkanalen

Een logisch transportkanaal is een communicatielink tussen de master en de slaves van het piconet. De logische transportkanalen van een piconet steunen op eenzelfde fysieke link.

### Synchroon logisch transportkanaal

Een synchroon logisch transportkanaal is verbindingsgeoriënteerd en circuitgeschakeld. Het transportkanaal biedt een constante bandbreedte van 64 Kbps in beide richtingen, gebaseerd op de reservering van tijdslots van het onderliggende fysieke kanaal.

Dit type transportkanaal wordt in de praktijk bijna uitsluitend gebruikt voor het transport van geluidsgegevens, en is bijgevolg niet van belang voor onze toepassing.

### Asynchroon logisch transportkanaal

De tijdslots die niet werden gereserveerd door een synchroon logisch transportkanaal, zijn beschikbaar voor dit type transportkanaal. Een asynchroon logisch transportkanaal is pakketgeschakeld, niet verbindingsgeoriënteerd, en biedt geen garanties over de beschikbare bandbreedte.

De standaard definieert één type asynchroon logisch transportkanaal, namelijk *Asynchronous Connectionless (ACL).* Een ACL logisch transportkanaal zorgt voor een punt-naar-meerpuntsverbinding tussen de master en de slaves binnen het piconet. De communicatie vindt altijd plaats tussen een master en een slave, behalve in het geval van een broadcast, waarbij de master alle slaves in het piconet gelijktijdig benadert.

Om de betrouwbaarheid van het kanaal te verhogen, maken ACL pakketten gebruik van een controlesom, en past men een eenvoudig 1-bit herverzendingsschema toe. Er wordt in functie van het precieze pakkettype aan voorwaartse foutcorrectie (Eng. *Forward Error Correction*, *FEC*) gedaan. Het ACL logisch transport biedt geen garanties over de volgorde waarin de ontvangen gegevens afgeleverd worden aan hogere lagen.

Ondanks de voorzieningen voor het opsporen van pakketfouten, onderkent de standaardspecificatie dat de betrouwbaarheid van dit kanaaltype niet voor alle

toepassingen volstaat; we besluiten dan ook dat het gebruik van ACL transportkanalen in een kritieke omgeving een hogere mate van betrouwbaarheid zal vereisen.

Een ACL logisch transportkanaal beschikt over een maximale bandbreedte van 780,8 Kbps. De laatste versie van de standaardspecificatie definieert een *Enhanced Data Rate (EDR)* uitbreiding die transfersnelheden tot 3 Mbps toelaat.

### Logische links

Logische links laten toe om logische transportkanalen in te zetten voor het transport van data en signalisatie van hogere lagen. Een logische link maakt hiertoe gebruik van een logisch transportkanaal. Eén logisch transportkanaal is in staat om meerdere logische links te dragen; dit kan met behulp van multiplexering of door de links die gebruik maken van het transportkanaal af te wisselen.

## Link Manager layer

De *Link Manager layer* definieert geen nieuwe kanaaltypes, maar is in de plaats verantwoordelijk voor het creëren van Baseband logische links tussen apparaten. Daarbij wordt uiteraard gesteund op de diensten van de Baseband layer.

De taak van de Link Manager wordt bemoeilijkt doordat niet elk Bluetooth apparaat dezelfde mogelijkheden bezit, bijvoorbeeld als gevolg van een verschillende stackversie. De Link Manager moet bijgevolg een afweging maken tussen de gevraagde dienstkwaliteit, en de mogelijkheden van de controllers van de communicatiepartners, waartussen men een verbinding wenst te realiseren. De Link Manager blijft verantwoordelijk voor het verdere beheer van een logische link, eenmaal deze tot stand gebracht is.

De Link Manager zorgt voor het optimaliseren van de kwaliteit van de logische links door het regelen van het zendvermogen, indien het apparaat uitgerust is met de mogelijkheden voor vermogenscontrole.

## Device Manager

De *Device Manager* behoort tot de Baseband layer en de Link Manager layer. Het is een belangrijk onderdeel van de stack dat zorgt voor het beheer van de ingebouwde beveiligingsfunctionaliteiten van Bluetooth die we besproken hebben in 2.2.2.

Naast de verantwoordelijkheden inzake beveiliging, staat de Device Manager in voor de device discovery procedure. Het is opmerkelijk dat deze functionaliteit volledig door de controller gerealiseerd wordt. We bespreken de device discovery procedure uitgebreid in 2.3.2 *Selectie van potentiële Routers*.

# Host to Controller Interface

De *Host to Controller Interface* of HCI splitst de protocol stack op in een host en een controller gedeelte. In veel gevallen betekent dit ook de scheiding tussen respectievelijk software en hardware.

Het belangrijkste onderdeel van de HCI is een implementatie van een host en een controller gedeelte voor een bepaalde interfacetechnologie. Veelgebruikte technologieën zijn USB, PCMCIA en UART. De laatste interface wordt in het bijzonder veel gebruikt in Pocket PC's.

Het controller gedeelte van de HCI zorgt voor de adaptatie tussen de gebruikte interfacetechnologie en de rest van de controller. Het host gedeelte van de HCI wordt meestal geïmplementeerd in de vorm van een driver en bijhorende applicatiesoftware, en zorgt voor de interface met het besturingssysteem.

De HCI laag is het laagste element van de protocol stack dat rechtstreeks kan benaderd worden bij de programmering van de protocol stack.

# Host

Bij de programmering van de protocol stack zullen we vooral te maken krijgen met het host gedeelte van de stack. Het is dan ook belangrijk om een goed algemeen inzicht te verwerven in de protocol elementen van dit deel van de stack.

## Logical Link Control and Adaptation Protocol

Het *Logical Link Control and Adaptation Protocol* of L2CAP is verantwoordelijk voor de multiplexering van de gegevensstromen en protocollen van de hogere lagen van de stack via een ACL logische link. De kanalen die hierdoor ontstaan worden L2CAP kanalen genoemd.

Een L2CAP kanaal wordt gekarakteriseerd door een *Protocol Service Multiplexer* (PSM) waarde; hiermee wordt het hogere protocol aangegeven waarvoor het L2CAP kanaal een transportlink verzorgt. Er kunnen meerdere L2CAP kanalen bestaan met eenzelfde PSM waarde. De PSM waarden voor het SDP en het RFCOMM protocol zijn bijvoorbeeld respectievelijk 0x0001 en 0x0003.

Het L2CAP voorziet een aantal extra maatregelen voor de detectie van beschadigde of ongeldige pakketten; indien men L2CAP kanalen echter wenst te gebruiken in een kritieke omgeving, doet men er goed aan om extra aangepaste betrouwbaarheidsvoorzieningen in te bouwen.

### RFCOMM

Het RFCOMM protocol is een transportprotocol dat L2CAP kanalen aanbiedt onder de vorm van een ETSI TS 07.10 geëmuleerde seriële verbinding. Dit houdt ondermeer voorzieningen voor flow control in, gebaseerd op het RTS/CTS schema.

Het RFCOMM protocol laat toe om verschillende RFCOMM kanalen te creëren, en deze te identificeren aan de hand van een RFCOMM kanaalnummer.

Gezien de gunstige eigenschappen van het RFCOMM protocol, zullen we gebruik maken van RFCOMM kanalen bij de implementatie van de Transparante Communicatielaag.

### Service Discovery Protocol

We zullen het Service Discovery Protocol (SDP) uitgebreid bespreken in 2.3.2 *Selectie van potentiële Routers*. Voorlopig is het voldoende op te merken dat het SDP toelaat om de diensten van een Bluetooth apparaat te adverteren; dit zorgt voor een losse koppeling tussen gebruikers en aanbieders van een bepaalde dienst.

## 2.2.4 Bluetooth API's

Bij de bespreking van Bluetooth in 2.2.3, hebben we aangehaald dat de Bluetooth protocol stack uit twee grote belangrijke delen bestaat, namelijk de controller en de host; de controller wordt meestal geïntegreerd in een chip, terwijl de host bestaat uit drivers en eventuele toepassingsprogramma's.

Indien men de protocol stack wenst te programmeren, dient men dit te doen door het manipuleren van het host gedeelte van de stack met behulp van een API. Er zijn momenteel twee grote spelers die de host software leveren, namelijk Broadcom en Microsoft. Indien we de protocol stack wensen te programmeren zullen we rechtstreeks of onrechtstreeks gebruik moeten maken van de API's van deze bedrijven; we bekijken in deze sectie kort wat de mogelijkheden hieromtrent zijn.

### Broadcom

Tot een jaar geleden werd zowat alle Bluetooth computerhardware voor het Windows platform afgeleverd met de Widcomm Bluetooth software van Broadcom. Gezien de ervaring van Broadcom, is de kwaliteit van de software erg hoog, en zijn de ondersteunde functionaliteiten en applicatieprofielen zeer uitgebreid.

Jammer genoeg is de API behorende bij de Broadcom protocol stack niet vrij beschikbaar, en zijn de producten van Broadcom vooral bedoeld voor bedrijven die een erg groot productvolume realiseren, zoals bijvoorbeeld HP. De prijs van deze software is dan ook te

hoog in verhouding met de toegevoegde waarde die we ermee kunnen bereiken in onze toepassing.

# Microsoft

De tweede grote speler op het vlak van Bluetooth software is Microsoft. Ondanks de grote populariteit van de technologie, biedt Microsoft pas sinds Windows XP SP 1 (2002) en Windows CE[5] 4.0 (2001) ondersteuning voor Bluetooth. Het gevolg is dat Microsoft een achterstand opgelopen heeft op het vlak van Bluetooth, en dat de meeste Bluetooth computerhardware nog steeds met de software van Broadcom geleverd wordt. De software van Microsoft beidt bovendien slechts beperkte mogelijkheden aan de eindgebruiker, en wordt vandaag nog steeds als prematuur beschouwd door de fabrikanten van Pocket PC's. Dell is één van de weinige fabrikanten die het aangedurfd heeft om in 2005 een Pocket PC af te leveren voorzien van de Microsoft Bluetooth software. Het voordeel voor onze toepassing is dat de Microsoft API's vrij beschikbaar zijn.

Men wordt verondersteld gebruik te maken van C/C++ voor het programmeren met de Microsoft API's. Bovendien hebben we meermaals ervaren dat de documentatie van de API's in veel gevallen onvoldoende correct en duidelijk is; in [18] bevindt zich echter een goed overzicht van deze API's en een uitgebreid voorbeeld van de mogelijkheden voor het ontwikkelen van een Bluetooth toepassing.

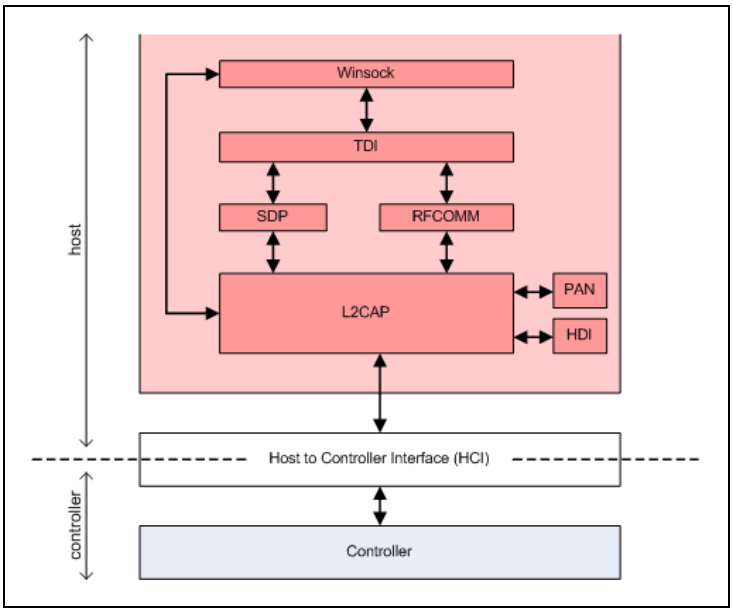
Een overzicht van de Microsoft Bluetooth protocol stack wordt weergegeven in Figuur 14. Merk op dat er twee lagen getoond worden die we nog niet eerder besproken hebben, namelijk TDI en Winsock.

De TDI-laag heet voluit de *Transport Driver Interface*. De taak van deze laag is de abstractie van het asynchrone karakter van de onderliggende lagen van de stack. Deze laag biedt de functionaliteit van de Bluetooth stack aan via de Windows Sockets (Winsock) interface.

Een alternatieve manier om de stack aan te sturen is de zogenaamde Bluetooth API. Deze API is meer gericht op het beheer van de aanwezige Bluetooth hardware van het apparaat. In tegenstelling tot de Winsock API, zijn de mogelijkheden van de Bluetooth API verschillend voor Windows CE en Windows XP.

---

[5] Windows CE is een familie van besturingssystemen voor embedded toestellen met beperkte prestaties. Het besturingssysteem kan naar wens gewijzigd worden volgens de noden van het toestel. Voorbeelden van leden van de familie zijn *Windows Mobile* en *SmartPhone*, voor respectievelijk Pocket PC's en GSM's.




Figuur 14: De Microsoft Bluetooth protocol stack

# Windows Sockets

De Windows Sockets interface is een softwarelaag die het aan toepassingssoftware toelaat om op een eenvormige manier gebruik te maken van verschillende netwerktechnologieën, waaronder Bluetooth. Indien men gebruik maakt van de Microsoft protocol stack is dit de aangewezen manier om voor de stack te implementeren.

De Winsock interface laat toe om de volgende Bluetooth operaties uit te voeren:

- Zoeken naar apparaten (device discovery).
- Zoeken naar services (service discovery).
- Een punt-naar-puntverbinding realiseren tussen twee apparaten; het resultaat is de creatie van sockets die de communicatie tussen de communicatiepartner verzorgen.
- Een Bluetooth dienst adverteren; connecties met deze dienst worden ontvangen door een server socket.

Voor de communicatie tussen de sockets van twee verbonden apparaten kan men gebruik maken van het L2CAP of het RFCOMM protocol. In ons geval zullen we gebruik maken van het RFCOMM protocol omwille van de voordelen die we aanhaalden bij de bespreking van de Bluetooth protocol stack. Merk op dat deze benadering toelaat om een verbinding te onderhouden met een apparaat dat zelf geen gebruik maakt van Winsock, op voorwaarde dat het RFCOMM protocol ondersteund wordt.

Het grote voordeel van Winsock is de hoge mate van abstractie en de mogelijkheid om de C++ code op eenvoudige wijze te porteren tussen Windows CE en Windows XP.

## Windows XP Bluetooth API

De Bluetooth API voor Windows XP dient uitsluitend voor het beheer van de aanwezige Bluetooth hardware in het apparaat. Voor communicatie met andere apparaten moet er gebruik gemaakt worden van de Winsock interface.

De belangrijkste functionaliteiten van de Windows XP Bluetooth API zijn:

- Opvragen van de aanwezige Bluetooth controllers en de toestand waarin ze zich bevinden.
- Instellen of het al dan niet mogelijk is voor andere toestellen om de controller te ontdekken via een device discovery, of er verbinding mee te maken.
- Oproepen van de dialogen van de authenticatieprocedure. De authenticatieprocedure op zich is slechts in beperkte mate controleerbaar.

## Windows CE Bluetooth API

In tegenstelling tot de Bluetooth API voor Windows XP, kan de API voor Windows CE heel wat meer dan uitsluitend administratieve functionaliteiten:

- Volledig aansturen van device discovery en service discovery.
- Volledig aansturen van de authenticatieprocedure.
- Het instellen van de operationele toestand van het Bluetooth apparaat; dit omvat eveneens de mogelijkheid om een apparaat in één van de stroombesparingsmodi te brengen.
- Het aanmaken van fysieke links.
- Het aanmaken van synchrone en asynchrone logische links.
- Configuratie van de Bluetooth controller.

De Windows CE Bluetooth API is vrij complex om mee te werken, en is eerder bedoeld voor de fabrikanten van Windows CE apparaten, uitgerust met Bluetooth. Voor gegevensverbindingen tussen applicaties gebruikt men bij voorkeur de Winsock interface.

## Wrapper API's

De API's van zowel Broadcom als Microsoft zijn vrij complex om mee te programmeren en zijn bovendien enkel rechtstreeks beschikbaar voor toepassingen geschreven in C++. Een aantal softwarehuizen heeft dan ook vrij recent een aantal initiatieven genomen om de complexiteit van de genoemde API's te abstraheren in een vorm die beter beantwoordt aan de meeste courante toepassing. Deze zogenaamde *wrappers* kunnen bovendien vaak gebruikt worden in combinatie met talen met kortere ontwikkelingscycli zoals C#. We bespreken hieronder kort de drie wrappers die naar onze mening het meest in aanmerking komen om een reële toepassing te ontwikkelen.

## 32Feet.NET

De 32Feet.NET[19] wrapper is een compacte bibliotheek voor de Microsoft Bluetooth protocol stack. De bibliotheek bevat een aantal klassen die de Winsock functionaliteiten abstraheren in een objectgeoriënteerd model. De klassen kunnen gebuikt worden met de talen van het .NET framework (C#, VB, …). Bij het gebruik van deze klassen blijft de Winsock structuur evenwel erg goed zichtbaar.

De belangrijkste klassen van de bibliotheek zijn BluetoothListener en BluetoothClient.

Een instantie van BluetoothListener wordt via het aanroepen van een operatie gebonden met een te kiezen identiteit (*Globally Unique Identifier, GUID*). Vervolgens wordt volledig volgens de regels van Winsock geluisterd naar binnenkomende verbindingen.

Het doel van de BluetoothClient is een verbinding te creëren met een BluetoothListener. Hiertoe kan de BluetoothClient een device discovery uitvoeren en op die manier de apparaatadressen van apparaten binnen bereik zoeken. Het is eveneens mogelijk dat het toepassingsprogramma het apparaatadres aangeeft. Het is echter niet mogelijk om de diensten van een apparaat op te vragen met behulp van een service discovery. Om een verbinding te maken met een BluetoothListener, gebruikt de BluetoothClient het apparaatadres van het toestel en de GUID-identiteit. Merk op dat de GUID-identiteit bij voorbaat gekend moet zijn. Het resultaat van de verbindingsprocedure is het tot stand komen van een stream tussen de betrokken apparaten.

32Feet.NET is vrij te verkrijgen en bovendien eenvoudig in gebruik. De uniforme interface zorgt dat het mogelijk is om op een eenvoudige manier software te schrijven voor Windows XP en Windows Mobile 5. Jammer genoeg is er geen ondersteuning voor het Service Discovery Protocol, en is er enkel communicatie mogelijk tussen apparaten die beide voorzien zijn van de Microsoft protocol stack en de 32Feet.NET software.

## High Point BTAccess

De BTAccess[20] wrapper van High Point Software richt zich specifiek op Pocket PC's met de Broadcom Bluetooth protocol stack, zoals bijvoorbeeld de iPAQ-reeks van HP. Er zijn klassenbibliotheken voorzien voor gebruik met C++ en voor de talen van het .NET framework (C#, VB, …). Het is niet mogelijk om de wapper te gebruiken voor de ontwikkeling van een Windows XP toepassing.

De belangrijkste klassen van de bibliotheek zijn BtStack en BtDevice.

De BtStack klasse is een abstractie van de Broadcom protocol stack en dient vooral voor administratieve doeleinden, zoals het in- en uitschakelen van de Bluetooth radio en het wijzigen van een aantal algemene beveiligingsinstellingen. Verder bezit deze klasse operaties voor het starten en stoppen van de device discovery procedure; de zoekresultaten

worden tijdens het zoeken via events doorgegeven aan de toepassing. Het resultaat van de Device Discovery procedure is het verwerven van een aantal BtDevice instanties.

Een instantie van de BtDevice klasse laat het toe om te communiceren met één bepaald Bluetooth apparaat. Met de operaties van BtDevice kan men ondermeer de Bluetooth authenticatieprocedure aansturen. Verder kan er verbinding gemaakt worden via de volgende profielen: PIM, OBEX, Dial-up, PAN en het seriële poort profiel.

Voor de PIM en OBEX profielen zijn er speciale operaties voorzien om visitekaartjes uit te wisselen en om een bestand te versturen of te ontvangen. Een connectie met de Dial-up, PAN en seriële poort profielen levert de naam van een geëmuleerde seriële poort op. Er moet gebruik gemaakt worden van de voorzieningen voor seriële communicatie van de gebruikte programmeertaal om deze poort verder te benaderen.

De aankoopprijs van BTAccess bedraagt $ 750,- voor een onbeperkt aantal gebruikers en toepassingen. Jammer genoeg kan er met BTAccess uitsluitend geprogrammeerd worden op Pocket PC's met de Broadcom Bluetooth protocol stack. Deze wrapper is bijgevolg niet geschikt voor onze doeleinden.

## Franson BlueTools

De laatste wrapper die we in dit overzicht zullen beschouwen is Franson BlueTools[21]. Deze wrapper is uniek omwille van de ondersteuning van de beide belangrijke Bluetooth protocol stacks. Bovendien kan de bibliotheek zowel voor ontwikkeling onder Windows XP als voor ontwikkeling onder Windows CE gebruikt worden. De ondersteunde programmeertalen zijn de talen van het .NET framework.

De belangrijkste klassen van BlueTools zijn Network, Device en Service.

De Network klasse is verantwoordelijk voor het beheer van de Bluetooth controllers van de computer, en bezit ondermeer een operatie voor het uitvoeren van een device discovery. Het resultaat van het uitvoeren van deze operatie is het verkrijgen van een lijst met beschikbare Bluetooth apparaten, elk geabstraheerd onder de vorm van een instantie van de Device klasse.

De Device klasse biedt een operatie aan waarmee men de diensten van het Bluetooth apparaat kan opvragen via een service discovery. Het resultaat van deze operatie wordt teruggegeven onder de vorm van een lijst met de beschikbare diensten van het apparaat. Eenmaal men over een referentie naar een dienst van een apparaat beschikt, kan men verbinding maken met deze dienst. Als resultaat bekomt men een stream die enkel steunt op RFCOMM voor het gegevenstransport. Het is dus mogelijk om te communiceren met apparaten die geen exemplaar van BlueTools bevatten, op voorwaarde dat het RFCOMM protocol ondersteund wordt.

De Service klasse is eveneens van belang wanneer men zelf een dienst wil aanbieden en adverteren. Hiertoe ontwerpt men een klasse die overerft van de Service klasse. De bekomen klasse moet ten minste een handler bevatten om de binnenkomende verbindingen af te handelen. Eenmaal men de Service klasse geregistreerd heeft bij de locale Network instantie, kan de dienst ontdekt worden met het Service Discovery Protocol en kan de klasse binnenkomende connecties afhandelen.

Een licentie voor Franson BlueTools kost $ 149,- per ondersteund framework. Er zijn bibliotheken beschikbaar voor het .NET framework en het .NET compact framework. De licentie mag gebruikt worden voor een onbeperkt aantal exemplaren van eenzelfde toepassing.

# Conclusie

We hebben in dit onderdeel aangetoond dat er heel wat API's en wrappers beschikbaar zijn voor de programmering van de Bluetooth protocol stack. Voor de ontwikkeling van de Transparante Netwerklaag geven we de voorkeur aan twee van de genoemde mogelijkheden, namelijk de Bluetooth Winsock interface en Franson BlueTools.

De Winsock API is interessant omwille van de vrij uitgebreide mogelijkheden en zijn vrije beschikbaarheid. Bovendien verbindt men zich niet aan een ander bedrijf voor het afleveren van een belangrijk component van de software. Een nadeel van de API is de vereiste om de toepassing gedeeltelijk in C++ te schrijven. Het is bovendien zo dat heel wat producenten de Microsoft protocol stack als prematuur beschouwen, en er bijgevolg momenteel nog weinig Pocket PC's voorzien zijn van deze stack.

De BlueTools wrapper steekt met kop en schouders boven de andere wrappers uit omwille van de ondersteuning voor beide grote protocol stacks via een uniforme interface, zowel onder het .NET framework als het .NET compact framework. Bovendien laat de opbouw van de klassenbibliotheek toe om de protocol stack op een elegante wijze te programmeren. Het nadeel is uiteraard de afhankelijkheid van een derde partij voor de Bluetooth communicatie.

Gezien de voordelen van BlueTools, hebben we voor dit eindwerk voor deze wrapper gekozen voor de implementatie van de Transparante Communicatielaag.

## 2.3 Implementatie met Bluetooth

We bespreken in dit onderdeel de implementatie van de Transparante Communicatielaag met behulp van Bluetooth. Gezien de doelstellingen van dit eindwerk, concentreren we ons vooral op uitdagingen inzake transparantie en contextgevoeligheid. Concreet zullen we vier belangrijke fases van naderbij bekijken.

We beginnen in 2.3.1 met een overzicht van de procedures waarmee een Terminal kan zoeken naar toestellen die optreden als Router binnen een gewenst Transparant Netwerk. We bekijken in de genoemde sectie eveneens hoe we aan de hand van apparaatkenmerken reeds bij voorbaat zoveel mogelijk kandidaten buiten beschouwing kunnen laten. Eenmaal we over een lijst met nabijgelegen toestellen beschikken, bekijken we in 2.3.2 hoe we kunnen nagaan of deze daadwerkelijk in aanmerking komen als potentiële Router. Het is uiteraard mogelijk dat er meerdere Transparante Netwerken beschikbaar zijn; we onderzoeken dan ook in 2.3.3 of het mogelijk is om de huidige context van een Terminal in te schatten, en of we aan de hand van die informatie het aantal mogelijke netwerken kunnen reduceren. Tot slot bekijken we in 2.3.4 wat het lidmaatschap van een Transparant Netwerk inhoudt, en hoe dit lidmaatschap op een correcte manier beëindigd kan worden.

### 2.3.1 Zoeken naar potentiële Routers

Een Terminal initieert de verbindingsprocedure voor het verbinden met een Transparant Netwerk met het zoeken van toestellen die een rol kunnen vervullen als Router.

Het doel van deze sectie is tot een procedure te komen, waarmee een Terminal een minimale lijst kan maken van potentiële Routers binnen zijn bereik. De eerste stap van deze fase bestaat dan ook uit het detecteren van de nabijgelegen Bluetooth apparaten. Vervolgens proberen we deze lijst zo veel mogelijk te reduceren, louter steunend op de apparatenkenmerken van de gevonden apparaten. Het is met andere woorden niet bedoeling om in deze fase reeds met de apparaten te communiceren of te onderhandelen.

Bluetooth laat toe om aan de hand van de device discovery procedure na te gaan welke toestellen zich binnen het bereik bevinden van het zoekende apparaat. Een Terminal kan dan ook gebruik maken van deze procedure om de mogelijke communicatiepartners in zijn nabijheid te zoeken.

We beginnen dit onderdeel met een onderzoek van de device discovery procedure. Vervolgens gaan we na hoe we de lijst met potentiële Routers kunnen reduceren aan de hand van informatie die we verwerven uit de device discovery. Aangezien de device discovery procedure een aantal nadelige eigenschappen bezit, stellen we in deze sectie eveneens een alternatieve zoekprocedure voor; deze alternatieve zoekprocedure zal in

bepaalde specifieke situaties een antwoord bieden op de tekortkomingen van de device discovery procedure. We besluiten dit onderdeel met een korte bespreking van de implementatie van de genoemde zoekprocedures in de Transparante Communicatielaag.

# Zoeken met de device discovery procedure

In deze sectie zullen we de device discovery procedure van naderbij onderzoeken. We bestuderen daarbij eerst wat de Bluetooth standaardspecificatie over dit onderwerp te vertellen heeft. Vervolgens bekijken we in welke mate de Winsock API en BlueTools ondersteuning bieden voor deze procedure. Gezien het karakter van onze toepassing, wensen we zo weinig mogelijk tijd te spenderen aan het uitvoeren van een device discovery. We voeren dan ook een experiment uit om te bepalen hoeveel de minimale duur van de procedure bedraagt. Tot slot bekijken we een aantal reductiemethodes om, louter op basis van apparaatkenmerken, tot een minimale lijst met potentiële Routers te komen.

## De device discovery procedure in de SIG standaard

Een apparaat neemt tijdens de device discovery procedure één van twee rollen op zich. Enerzijds is er een apparaat dat actief zoekt naar communicatiepartners; dit is het *inquiring device*. Anderzijds bepaalt elk apparaat of het al dan niet gevonden wil worden; de toestellen die dit wensen worden *discoverable* genoemd, en luisteren actief naar aanvragen van het zoekende toestel. De apparaten die noch zoeken, noch gevonden willen worden zijn niet van belang in de context van deze procedure.

Het zoekende apparaat maakt gebruik van een *Inquiry Scan Physical Channel* voor het versturen van de aanvragen; we kwamen dit fysieke kanaal reeds eerder tegen bij de bespreking van de Bluetooth protocol stack in 2.2.3.

In normale omstandigheden wordt een fysiek kanaal gedefinieerd door een aantal karakteristieken van het piconet, zoals een gedeelte van het apparaatadres van de master. In het geval van een device discovery is er uiteraard nog geen sprake van een piconet en wordt het fysieke kanaal volledig bepaald door één van de daartoe gereserveerde toegangscodes. Een daartoe opgerichte werkgroep heeft 64 toegangscodes[22] gereserveerd, waarvan er momenteel twee gebruikt worden: de *General/Unlimited Inquiry Access Code (GIAC)* en de *Limited Dedicated Inquiry Access Code (LIAC)*. De LIAC toegangscode mag enkel gebruikt worden in specifieke gevallen en gedurende een korte periode. Bovendien moeten zowel het zoekende als de gezochte apparaten zich tijdens deze periode bewust zijn van het bijzondere karakter van de discovery sessie. Aangezien onze toepassing continu naar partners zoekt, zullen we de GIAC gebruiken. De apparaten die wensen gevonden te worden beluisteren hiertoe continu het fysieke kanaal gedefinieerd door de GIAC toegangscode.

Wanneer een luisterend apparaat een ontdekkingsaanvraag ontvangt van een zoekend apparaat, dient het zich bekend te maken door te antwoorden op de aanvraag. Gezien het ongecoördineerde karakter van de procedure en de onbetrouwbaarheid van het onderliggende fysieke kanaal, bestaat er een reële kans dat er een botsing zal optreden wanneer twee of meerdere apparaten tegelijk op een inquiry aanvraag antwoorden. Men moet dan ook naar een evenwicht zoeken tussen de tijd gespendeerd aan de discovery procedure en het aantal apparaten dat effectief gevonden wordt. Gezien de gevoeligheid voor eventuele botsingen, ligt het voor de hand dat een omgeving met veel interferentie een langer zoekinterval zal vereisen om eenzelfde zoekkwaliteit in stand te houden.

De lengte van de inquiry procedure bedraagt ten minste 1,28 s, en wordt gedefinieerd in veelvouden van eenheden van eveneens 1,28 s. Om te garanderen dat het zoekende apparaat alle toestellen in inquiry scan mode vindt, beveelt de Bluetooth specificatie een interval van ten minste 10 eenheden (12,8 s) aan. Sommige auteurs bevelen echter, afhankelijk van de precieze toepassing, inquiry tijden tot een minuut aan[15].

## De device discovery procedure in Franson BlueTools

Hoewel de Bluetooth standaardspecificatie de mogelijkheid vermeldt om de lengte van het zoekinterval in te stellen, gebruikt men in Franson BlueTools steeds een vaste lengte van 12,8 s. Men volgt hieromtrent dus de aanbevelingen van de SIG.

We zullen echter zo meteen aantonen dat dit interval vrij ruim gekozen is, maar anderzijds voor bepaalde andere toepassing veel te kort zal blijken. Een instelbare intervallengte lijkt ons dus de beste keuze.

Het resultaat van de device discovery procedure is het verwerven van het apparaatadres, de naam (Eng. *Friendly Name*) en de apparaatklasse (Eng. *Class Of Device*, COD) van elk toestel binnen bereik. We zullen echter zien dat men nooit kan garanderen dat *alle* apparaten binnen bereik gevonden worden.

## De device discovery procedure in Microsoft Winsock

In tegenstelling tot BlueTools, kan men in de Winsock API de lengte van het interval vrij kiezen in eenheden van 1,28 s. De instelbaarheid is goed nieuws voor het uitvoeren van ons volgende experiment; we zullen in dit experiment een onderzoek doen naar de minimale duur van de procedure, en proberen te achterhalen wat de gevolgen zijn van een te kort zoekinterval.

Het resultaat van de discovery procedure is in dit geval eveneens de verwerving van het apparaatadres, de naam en de apparaatklasse van elk toestel binnen bereik.

## Minimale duur van de device discovery procedure

Gezien het belang van een snelle werking van de Transparante Communicatielaag, willen we proberen om de tijd die doorgebracht wordt in de discovery procedure zo kort mogelijk te

maken. Vanuit het standpunt van een toepassing is dit immers pure wachttijd gedurende dewelke er niets nuttig verricht wordt.

Om de minimale lengte van het zoekinterval te bepalen voeren we een experiment uit met zes toestellen. We kiezen telkens een bepaalde duur voor de discovery procedure en bekijken hoeveel van de aanwezige toestellen er daadwerkelijk gevonden worden.

Om dit experiment uit te voeren hebben we een kort C++ programma geschreven met behulp van de Microsoft Winsock API, aangezien deze toelaat de gewenste lengte van het zoekinterval in te stellen; de lengte kan in overeenstemming met de standaard worden ingesteld in veelvouden van 1,28 s. Aangezien de device discovery procedure volledig geïmplementeerd werd in het controller gedeelte, moet onze toepassing de procedure enkel activeren en nadien de resultaten opvragen. Om dezelfde reden is het inschakelen van de gezochte toestellen voldoende om deze te laten antwoorden op de verzoeken van het zoekende apparaat.

Het testprogramma voert een device discovery uit gedurende het gewenste zoekinterval en rapporteert nadien de resultaten zoals weergegeven in Tabel 3 en Tabel 4. De device discovery wordt voor elk zoekinterval vijf keer herhaald. We bekijken vervolgens in hoeveel van de gevallen elk van de toestellen gevonden werd.

## Toestellen in eenzelfde kamer

In Tabel 3 tonen we de resultaten voor het experiment waarbij alle toestellen gelijkmatig in eenzelfde kamer werden opgesteld binnen een straal van 5 m rond het zoekende apparaat.

Tabel 3: Toestellen in eenzelfde kamer – aantal succesvolle zoekacties naar een toestel per 5 pogingen

| **Zoektijd (x 1,28 s)** | **Toestel 1** | **Toestel 2** | **Toestel 3** | **Toestel 4** | **Toestel 5** | **Toestel 6** | **Succes** |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 0 | 5 | 0 | 0 | 0 | 33% |
| 2 | 5 | 0 | 5 | 0 | 0 | 0 | 33% |
| 3 | 5 | 5 | 5 | 5 | 5 | 3 | 93% |
| **4** | **5** | **5** | **5** | **5** | **5** | **5** | **100%** |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 100% |
| 60 | 5 | 5 | 5 | 5 | 5 | 5 | 100% |

Een eerste vluchtige analyse van de resultaten bevestigt ons vermoeden dat een langer zoekinterval aanleiding geeft tot een hoger succespercentage. Van zodra het interval een lengte aanneemt van vier eenheden, bekomen we het gewenste succespercentage. Het gebruik van langere intervallen is mogelijk, maar dit levert uiteraard geen betere prestaties. Merk op dat de lengte van het interval minder dan de helft bedraagt van de in de standaardspecificatie aanbevolen waarde.

Wanneer we de resultaten voor intervallen met lengte 1,28 s en 2,56 s bekijken, valt het op dat de zoekprocedure steeds dezelfde toestellen oplevert: de vijf uitgevoerde discoveries vonden telkens enkel Toestel 1 en Toestel 3. Dit is een erg belangrijke observatie die het belang van een voldoende lang zoekinterval benadrukt. We kunnen namelijk uit de bekomen resultaten afleiden dat we, ongeacht het aantal keer dat we de procedure herhalen, sommige apparaten nooit zullen vinden indien het zoekinterval te kort gekozen werd. We doen er dus goed aan om, bij de keuze van het interval, voldoende rekening te houden met eventuele interferentie en een mogelijk groot aantal aanwezige apparaten.

## Toestellen in verschillende kamers

Om na te gaan of de afstand van de toestellen een invloed heeft op het succespercentage, verdelen we de apparaten over meerdere kamers en herhaalden we het experiment. Alle apparaten bevonden zich binnen een straal van 10 m van het zoekende toestel. Toestellen 4 en 5 hebben we echter in dezelfde kamer gelaten als het zoekende apparaat. Deze bevinden zich bijgevolg ruimschoots het dichtst bij het zoekende toestel. De bekomen resultaten bevinden zich in Tabel 4.

Tabel 4: Toestellen in verschillende kames – aantal succesvolle zoekacties naar een toestel per 5 pogingen

| **Zoektijd (x 1,28 s)** | **Toestel 1** | **Toestel 2** | **Toestel 3** | **Toestel 4** | **Toestel 5** | **Toestel 6** | **Succes** |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 0 | 5 | 0 | 0 | 0 | 33% |
| 2 | 5 | 0 | 5 | 0 | 0 | 0 | 33% |
| **3** | **5** | **5** | **5** | **5** | **5** | **0** | **83%** |
| 4 | 5 | 5 | 5 | 5 | 5 | 0 | 83% |
| 5 | 5 | 5 | 5 | 5 | 5 | 0 | 83% |
| 60 | 5 | 5 | 5 | 5 | 5 | 0 | 83% |

De resultaten bekomen met toestellen verspreid over meerdere kamers vertonen een aantal opmerkelijke karakteristieken. Het valt onmiddellijk op dat Toestel 6 nooit gevonden wordt, zelfs niet bij extreem lange zoekintervallen. Dit is het gevolg van het beperkte bereik van dat toestel. Verder is er een opmerkelijke gelijkenis met de resultaten uit Tabel 3. Bij korte intervallen van 1,28 s en 2,56 s, zijn het opnieuw toestellen 1 en 3 die telkens gevonden worden, onafhankelijk van het aantal keer dat we het experiment herhalen. In tegenstelling tot wat men zou verwachten, levert het zoeken dus niet toestellen 4 en 5 op, hoewel deze zich het dichtst bij het zoekende apparaat bevinden. Dit bewijst dat men zich niet mag laten verleiden om een te kort interval te kiezen, met de ambitie om communicatiepartners uit te sluiten die zich relatief ver van het zoekende apparaat bevinden.

### Conclusie

We hebben geprobeerd om met dit experiment een minimale waarde te bekomen voor de lengte van de device discovery. In overeenstemming met wat we verwachtten, heeft de lengte van het zoekinterval een belangrijke invloed op het percentage van de aanwezige toestellen dat effectief gevonden wordt.

Hoewel een kort interval reeds aanleiding gaf tot uitstekende zoekresultaten, willen we zeker niet ingaan tegen de SIG aanbevelingen inzake een langere discovery tijd. We toonden immers aan dat een te korte zoekactie steeds dezelfde apparaten oplevert. Als gevolg zorgt een te korte zoektijd ervoor dat sommige toestellen nooit gevonden zullen worden, ongeacht het aantal keer dat men de device discovery uitvoert. We vermelden hier dan ook geen exacte minimale waarde voor de lengte van het zoekinterval. In de plaats wijzen we erop dat men de zoektijd voorzichtig en voldoende groot moet kiezen in functie van de toepassing en de omgeving waarin deze effectief gebruikt zal worden.

Tot slot merkten we op dat een te korte zoektijd geen aanleiding gaf tot het vinden van enkel de dichtste toestellen; indien een apparaat zich binnen bereik bevindt, bestaat er met andere woorden geen duidelijk verband tussen de afstand tot het zoekende toestel, en de kans dat dit toestel gevonden wordt bij een device discovery.

## Reductie van het aantal potentiële Routers

We hebben hierboven besproken hoe het met de Bluetooth device discovery procedure mogelijk is om aanwezige communicatiepartners op te sporen. In de context van de Transparante Communicatielaag zullen we met de device discovery procedure echter een groot aantal toestellen overwegen, dat eigenlijk bij voorbaat niet in aanmerking kwam voor het vervullen van een rol als Router in het gewenste Transparant Netwerk. Denk maar aan de talloze GSM's die uitgerust zijn met Bluetooth, en zich bijgevolg allemaal in de lijst met potentiële Routers zullen bevinden. We bekijken in dit onderdeel dan ook een aantal mogelijkheden om het aantal te overwegen communicatiepartners zo veel als mogelijk te reduceren. We zullen hierbij steunen op informatie die we verkrijgen tijdens de device discovery procedure. Zoals reeds eerder vermeld, is het resultaat van de device discovery procedure het verwerven van het apparaatadres, de naam (Eng. *Friendly Name*) en de apparaatklasse (Eng. *Class Of Device*, COD) van elk toestel binnen bereik. We bekijken op welke manier we deze informatie kunnen benutten om het aantal mogelijke Routers te reduceren.

### Apparaatadres

Na een device discovery beschikken we over het apparaatadres van de aanwezige toestellen. We hebben het apparaatadres reeds besproken in een overzicht van de mogelijk manieren om een Bluetooth apparaat te adresseren in 2.2.2.

Zoals reeds vermeld in het genoemde overzicht, kan een apparaatadres opgedeeld worden in twee semantische velden, namelijk een *Company_ID* en een *Company_Assigned* veld. Een eerste manier om het aantal te overwegen toestellen te reduceren, is het gebruik van het Company_ID veld; we gaan er in dit geval van uit dat de toestellen die deelnamen aan het Transparant Netwerk geleverd worden door een beperkt aantal producenten.

In het geval van de zorgtoepassing zouden we kunnen overwegen om uitsluitend USB Bluetooth dongles van de fabrikant Belkin te gebruiken in de IPT's. Het Company_ID veld van de apparaatadressen van de dongles van deze producent kan een beperkte aantal waarden aannemen, namelijk 00:00:3A of 00:0A:3A. We dienen in dit geval uitsluitend apparaten met één van deze waarden voor het Company_ID veld te overwegen als Router. Op deze manier sluiten we alvast alle GSM toestellen uit, die voorzien zijn van het Company_ID veld van bijvoorbeeld Nokia of Ericsson. De Pocket PC's van HP en Dell gebruiken het Company_ID van Bluetooth chipproducent Texas Instruments, en worden bijgevolg eveneens verwijderd uit de lijst. In een ideaal geval zouden we kunnen beschikken over Bluetooth hardware die ons eigen Company_ID gebruikt.

Een alternatief is het bijhouden van een lijst van de apparaatadressen van de mogelijke communicatiepartners. In dit geval moet er echter vooraf een volledige lijst bestaan, hetgeen heel wat administratie vereist en de transparantie van de toepassing mogelijk niet ten goede komt.

Indien we deze reductietechnieken willen toepassen in de zorgtoepassing, moet de IVT uit het apparaatadres kunnen afleiden of het apparaat al dan niet in aanmerking komt als Router.

## Friendly Name

De Friendly Name is een benaming die de gebruiker van een Bluetooth apparaat vrij kan kiezen. We zouden ervoor kunnen opteren om alle mogelijke Routers een benaming te geven die aangeeft dat ze deze rol wensen te vervullen. Bovendien kan deze benaming eveneens een indicatie geven over het aangeboden Transparant Netwerk.

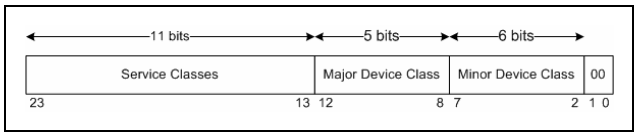
In het geval van de zorgtoepassing zouden we elke IPT een benaming kunnen geven van de vorm *IPT_ PatiëntNummer*. Op deze manier weet de IVT meteen dat het apparaat een IPT is behorend bij de Patiënt met het gegeven nummer.

Merk op dat deze methode eenvoudiger is, maar tegelijk ook een stuk minder veilig dan de methode die steunt op apparaatadressen; het wijzigen van het apparaatadres van een Bluetooth toestel is immers een stuk moeilijker dan het wijzigen van de benaming.

## Class Of Device

Een laatste apparaateigenschap die we kunnen verwerven met behulp van een device discovery, is de apparaatklasse (Eng. *Class Of Device*, COD) van de nabijgelegen toestellen.

De COD is een veld van 24 bits dat de klasse van het apparaat en de geleverde diensten aangeeft. Het formaat van het veld wordt weergegeven in Figuur 15. Het *Service Classes* veld geeft een algemene beschrijving van de diensten die het apparaat aanbeidt; het is uiteraard mogelijk dat er meer dan één dienst aanwezig is op het apparaat. Voorbeelden zijn *Object Transfer*, *Capturing* en *Networking*. Sommige Bluetooth apparaten wijzigen dit veld afhankelijk van de diensten die ze op dat moment echt aanbieden. Er is evenwel geen direct verband tussen het adverteren van diensten op een apparaat en het wijzigen van de COD; alle wijzigingen van het COD vereisen een manueel ingrijpen van de toepassingssoftware.




Figuur 15: Formaat Bluetooth Class Of Device

De twee overige velden geven een indicatie van de klasse waartoe het beschouwde apparaat behoort. Het *Major Device Class* veld beschrijft de apparaatklasse met de grootste granulariteit en bepaalt tegelijk de mogelijke betekenissen van het *Minor Device Class* veld. Mogelijke waarden voor het Major Device Class veld zijn bijvoorbeeld *Wearable*, *Peripheral*, *Phone* en *Computer*. In ons geval zijn we uitsluitend geïnteresseerd in de laatste genoemde waarde; in dit geval bevat de Minor Device Class een meer gedetailleerde beschrijving van het computertype; mogelijke waarden zijn bijvoorbeeld *Desktop*, *Laptop*, *Server*, *PDA*, …

De overige twee bits zijn gereserveerd voor toekomstig gebruik.

Het is duidelijk dat we met dit veld vrij goede reductieresultaten kunnen bekomen, zonder de transparantie van de toepassing te schaden. Voor een precieze beschrijving van de mogelijke waarden van de COD velden verwijzen we naar [22].

In het geval van de zorgtoepassing zou de IVT bijvoorbeeld enkel toestellen kunnen beschouwen met een waarde *Computer* voor het Major Device Class veld, gecombineerd met de *Desktop* waarde voor het Minor Device Class veld.

# Een alternatieve zoekprocedure

In de voorgaande paragrafen hebben we een mogelijke zoekprocedure bekeken die steunt op de Bluetooth device discovery procedure. Tijdens het experiment met deze procedure, toonden we aan dat de zoektijd lang genoeg moet zijn, opdat we alle aanwezige apparaten zouden vinden. Het bleek namelijk dat er een reële kans bestaat dat het gewenste apparaat anders nooit gevonden zal worden, ongeacht het aantal keer dat we de procedure herhalen. Het is duidelijk dat dit in een omgeving met een hoge apparaatdichtheid aanleiding kan geven tot de nood aan extreem lange zoektijden. Het is zelfs helemaal niet ondenkbaar dat

de toepassing onbruikbaar wordt indien er te veel tijd aan de device discovery gespendeerd wordt.

Bij de bespreking van de *paging procedure* in 2.2.2, vermeldden we reeds dat het in principe niet vereist is om een device discovery uit te voeren, alvorens men een verbinding maakt. Het is natuurlijk wel zo dat we dan op voorhand moeten kunnen beschikken over een lijst met de apparaatadressen van de potentiële communicatiepartners. We zullen voor de rest van deze sectie veronderstellen dat een dergelijke lijst voor handen is op het toestel dat de zoekactie uitvoert. Merk op dat we met een dergelijke lijst bij voorbaat de garantie hebben dat de bekomen lijst met potentiële Routers minimaal is; in tegenstelling tot de device discovery procedure, heeft deze alternatieve procedure geen nood aan reductiemethoden.

We zullen in dit onderdeel een aantal experimenten uitvoeren om de waarde van de voorgestelde zoekprocedure in te schatten. Voor de proefnemingen voorzien we een kleine C# toepassing steunend op BlueTools voor de programmering van de Bluetooth stack. Het programma start met een lijst van apparaatadressen en probeert aan de hand van elk adres een gegevensverbinding op te zetten met het bijhorende apparaat. In tegenstelling tot de testtoepassing voor de device discovery, moet er ook op het luisterende apparaat een toepassing draaien die de binnenkomende aanvraag beantwoordt en de connectie opzet. Merk op dat dit verschilt van de vorige toepassing in de zin dat we nu meteen ook een gegevensverbinding maken met het luisterende toestel, en niet enkel nagaan of het toestel beschikbaar is binnen bereik.

We beschikken over vier toestellen voor het uitvoeren van dit experiment:

- *Acer*. Een laptop met een Acer USB dongle;
- *Belkin 3*. Een PC met een Belkin USB dongle;
- *HP iPAQ*. Een HP iPAQ hx2490 Pocket PC;
- *Dell Axim*. Een Dell Axim X51 Pocket PC.

Verder gebruiken we, voor de voorstelling van apparaten die zich niet binnen het bereik van het zoekende toestel bevinden, de apparaatadressen van twee bestaande Belkin Bluetooth USB dongles (*Belkin 1* en *Belkin 2*).

## Toestellen in eenzelfde kamer

In een eerste testopstelling verdelen we de beschikbare toestellen over eenzelfde kamer, zodat de maximale onderlinge afstand tussen elke twee apparaten ten hoogste vijf meter bedraagt.

De testtoepassing is zodanig geconfigureerd dat de apparaatadressen van de apparaten in de eerste rij van Tabel 5 achtereenvolgens gebruikt worden voor het oproepen van het bijhorende toestel; de eerste kolom bevat telkens de naam. We herhalen het experiment

zodat elk aanwezig apparaat vijf keer gezocht heeft naar de aanwezige apparaten. We leggen daarbij telkens vast hoelang het oproepen van een bepaald apparaat in beslag neemt. Vervolgens berekenen we uit de bekomen resultaten de gemiddelde lengte van het oproepinterval voor elk paar, bestaande uit een zoekend apparaat en een gezocht apparaat. De aldus bekomen gemiddelden worden weergegeven in Tabel 5. Het zoekende toestel bevindt zich telkens in de eerste cel van de rij, gevolgd door de gemiddelden die voor de zoekactie naar het toestel in de overeenkomstige kolomtitel bekomen werden.

Tabel 5: Toestellen binnen eenzelfde kamer – gemiddelde oproeptijd

| | **Acer (s)** | **Belkin 1 (s)** | **Belkin 2 (s)** | **Belkin 3 (s)** | **HP iPAQ (s)** | **Dell Axim (s)** |
|---|---|---|---|---|---|---|
| **Acer** | - | 5,23 | 5,18 | 1,97 | 0,95 | - |
| **Belkin 1** | - | - | - | - | - | - |
| **Belkin 2** | - | - | - | - | - | - |
| **Belkin 3** | 0,74 | 5,15 | 5,15 | - | 1,01 | - |
| **HP iPAQ** | 1 | 5 | 6 | 0 | - | - |
| **Dell Axim** | 1 | 5 | 5 | 1 | 2 | - |

Bij een vluchtige analyse van de gegevens in Tabel 5, valt het onmiddellijk op dat niet alle cellen een meetwaarde bevatten. Ten eerste vermelden we geen meetwaarde voor een toestel dat zichzelf oproept, aangezien een dergelijke oproep nooit beantwoord wordt. Ten tweede zijn de toestellen *Belkin 1* en *Belkin 2* niet aanwezig, en kunnen we ze bijgevolg niet gebruiken om een zoekactie mee uit te voeren. Het gebrek aan meetwaarden in de laatste kolom, is het gevolg van een fout in BlueTools waardoor de *Dell Axim* in dit specifieke geval nooit antwoordt op verbindingsaanvragen. Het toestel kan echter wel zelf andere apparaten oproepen, zodat er wel meetgegevens beschikbaar zijn voor dit toestel als zoekend apparaat. Tot slot merken we nog op dat de we met de Pocket PC's meetwaarden bekomen met een lagere precisie dan de meetwaarden van de PC's. Dit is het gevolg van de beperkte precisie van de variabelen voor de voorstelling van tijd in het .NET compact framework.

Wanneer we de meetwaarden van naderbij bekijken, valt het onmiddellijk op dat het oproepen opmerkelijk minder tijd in beslag neemt wanneer het toestel fysiek aanwezig is. Er wordt in dat geval namelijk vrij snel een gegevensverbinding gemaakt; dit gebeurt in ongeveer één seconde. Indien het toestel niet aanwezig is, wordt de aanvraag een aantal keer herhaald, om uiteindelijk toch zonder antwoord te blijven. In dit geval loopt de gespendeerde tijd per toestel op tot ongeveer vijf seconden.

## Toestellen in verschillende kamers

Analoog met het voorgaande onderzoek naar de minimale device discovery tijd, herhalen we ook dit experiment met de toestellen verspreid over een aantal naburige kamers. We

herhalen het experiment zoals hierboven beschreven en brengen de resultaten onder in Tabel 6.

Tabel 6: Toestellen in verschillende kamers – gemiddelde oproeptijd

| | **Acer (s)** | **Belkin 1 (s)** | **Belkin 2 (s)** | **Belkin 3 (s)** | **HP iPAQ (s)** | **Dell Axim (s)** |
|---|---|---|---|---|---|---|
| **Acer** | - | 5,30 | 5,18 | 1,97 | 1,11 | - |
| **Belkin 1** | - | - | - | - | - | - |
| **Belkin 2** | - | - | - | - | - | - |
| **Belkin 3** | 0,74 | 5,14 | 5,14 | - | 0,87 | - |
| **HP iPAQ** | 1 | 5 | 5 | 2 | - | - |
| **Dell Axim** | 1 | 5 | 5 | 1 | 1 | - |

Een vergelijking tussen de meetwaarden in Tabel 5 en Tabel 6 toont aan dat het verspreiden van de toestellen over meerdere kamers quasi geen invloed heeft op de bekomen gemiddelden. We kunnen hieruit besluiten dat er geen verband bestaat tussen de onderlinge afstand en de tijd die een toestel nodig heeft om een andere communicatiepartner op te roepen. Indien er echter botsingen optreden als gevolg van interferentie, is het uiteraard wel mogelijk dat de paging procedure meerdere pogingen moet ondernemen; bijgevolg zal het realiseren van een verbinding mogelijk meer tijd in beslag nemen in een omgeving met veel interferentie.

# Vergelijking van de procedures

We hebben in de experimenten met de alternatieve zoekmethode aangetoond dat we in ongeveer één seconde verbinding kunnen maken met een toestel, indien dit zich binnen bereik bevindt. In tegenstelling tot de device discovery procedure, vereist de alternatieve procedure bovendien geen reductie van de lijst met potentiële Routers, gebaseerd op bijvoorbeeld hun apparaatadres of apparaatklasse; de op alternatieve wijze bekomen lijst is immers bij voorbaat gegarandeerd minimaal.

De problematiek van de alternatieve zoekprocedure situeert zich echter in de tijd die we verliezen met elk apparaatadres van een toestel dat zich niet binnen bereik bevindt; per dergelijk apparaatadres verliezen we maar liefst vijf seconden. We toonden in een eerder experiment aan dat de device discovery procedure in die tijd meer dan zes apparaten kan ontdekken. Het is dan ook meteen duidelijk dat de device discovery procedure de beste keuze is in het algemene geval. In een aantal specifieke situaties zal de alternatieve procedure echter een antwoord bieden op de tekortkomingen van een device discovery. We zullen hieronder drie dergelijke situaties bespreken.

## Hoge apparaatdichtheid/beperkt aantal apparaatadressen

Een eerste geval waarin de alternatieve procedure nuttig is, treedt op wanneer de apparaatdichtheid erg hoog is, terwijl we vooraf de lijst met communicatiepartners kennen. Bovendien veronderstellen we dat deze lijst kort is, in verhouding met het aantal apparaten binnen bereik.

Stel bijvoorbeeld dat de apparaatdichtheid in die mate hoog is, dat een device discovery procedure van één minuut nauwelijks volstaat. We kunnen dan, voor een vooraf gedefinieerde lijst van een vijftiental apparaatadressen, eenzelfde tijdprestatie afleveren maar geven hierbij wel de garantie dat, indien het apparaat binnen bereik is, het zeker gevonden wordt. Indien men harde garanties wil dat elk apparaat gevonden wordt, en de lijst beperkt is van omvang, is de voorgestelde zoekprocedure zeker een valabel alternatief voor een device discovery.

## Workflow impliceert apparaatadressen

In bepaalde toepassingen is het mogelijk om een korte lijst met potentiële communicatiepartners af te leiden uit het gedrag van de gebruiker binnen de toepassing.

Indien we bijvoorbeeld veronderstellen dat een verpleegkundige zijn rondes steeds min of meer in eenzelfde volgorde afwerkt, dan kunnen we een korte lijst met toekomstige communicatiepartners afleiden uit de kennis van de huidige partner. Indien de verpleegkundige bijvoorbeeld de kamers afwerkt volgens oplopend kamernummer, beschouwen we in kamer 301 enkel de terminals van kamers 302, 303 en 304 als potentiële communicatiepartners. De mobiele terminal van de verpleegkundige kan ter voorbereiding reeds verbindingen klaarmaken met de terminals op die kamers, zodat de verpleegkundige eenvoudig een keuze kan maken en de gegevensverbinding onmiddellijk beschikbaar is voor gebruik.

## Veiligheidoverwegingen

Een laatste niet onbelangrijk aspect van de device discovery is dat het gebruik ervan een toepassing inherent onveilig maakt.

Een eerste veiligheidsprobleem is het gebrek aan een manier waarop een apparaat kan beslissen of het al dan niet gevonden wenst te worden door een welbepaald toestel; indien een apparaat zich in inquiry scan mode bevindt, antwoordt het immers op elke aanvraag om zich bekend te maken, ongeacht de aanvrager. Dit maakt toestellen in de inquiry scan mode gevoelig voor kwaadwillige gebruikers. De beveiligingspolitiek van heel wat bedrijven en instellingen beschouwt een dergelijke eigenschap dan ook als ontoelaatbaar.

Het tweede aspect betreft de veiligheidsrisico's voor het zoekende apparaat. Een kwaadwillige gebruiker die weet heeft van de werking van het systeem, kan praktisch altijd een toestel zodanig configureren dat het zeker zal gecontacteerd worden door het zoekende

apparaat. Het toestel in inquiry mode is zich van geen kwaad bewust en voegt het apparaat toe aan de lijst met mogelijke communicatiepartners. In sommige toepassingen zal het zoekende toestel na de inquiry procedure onderhandelen met de potentiële communicatiepartners om één partner over te houden. Het is duidelijk dat het zoekende apparaat zich in dit geval blootgeeft en uiterst gevoelig is voor een eventuele aanval.

De alternatieve procedure is minder gevoelig voor dergelijke veiligheidsdreigingen aangezien de betrokken apparaten zich niet in inquiry of inquiry scan mode moeten bevinden, en de lijst met veilige partners reeds bij voorbaat vastligt.

## Implementatie Transparante Communicatielaag

We hebben ervoor geopteerd om beide zoekprocedures te voorzien in de implementatie van het Terminal-component van de Transparante Communicatielaag.

Voor de meeste algemene gevallen zal men voor het zoeken opteren voor de device discovery procedure; indien er geen verbinding is, wordt deze na het verlopen van een instelbaar interval opgestart. Het is met BlueTools niet mogelijk om de lengte van het zoekinterval in te stellen; bijgevolg bedraagt de duur van het zoekinterval steeds 12,8 s. Het is mogelijk om reductie van het aantal mogelijke partners in te schakelen; voor het definiëren van de toegelaten benamingen en apparaatadressen, wordt er gesteund op reguliere expressies. Het is momenteel niet mogelijk om met BlueTools de apparaatklasse van een apparaat op te vragen; bijgevolg biedt de Transparante Communicatielaag geen mogelijkheid om beperkingen op te leggen voor wat betreft de apparaatklasse van de potentiële Routers. Het opvragen van de apparaatklasse zal wel geïmplementeerd worden in een volgende versie van BlueTools.

Voor gebruik in een aantal specifieke gevallen, werd eveneens de alternatieve zoekprocedure geïmplementeerd. Opdat deze gebruikt zou worden, dient men een lijst met te zoeken apparaatadressen aan te geven.

Men kan beide zoekprocedures samen gebruiken en ze onafhankelijk van elkaar in- of uitschakelen.

# 2.3.2 Selectie van potentiële Routers

Eenmaal we een minimale lijst met potentiële communicatiepartners bekomen hebben zoals besproken in 2.3.1, wensen we uit deze lijst de mogelijk Routers te selecteren. Een apparaat is een potentiële Router voor een Terminal, indien deze een Transparant Netwerk beheert waarmee de Terminal wenst te verbinden.

In tegenstelling tot de vorige procedure, zullen we in deze fase selecteren op het niveau van diensten. We maken dan ook gebruik van de dienstgeoriënteerde architectuur van

Bluetooth voor het realiseren en het zoeken van een Transparant Netwerk. De Router zal gebruik maken van het Service Discovery Protocol (SDP) voor het adverteren van zijn Transparant Netwerk. Analoog zal ook de Terminal het SDP gebruiken voor het uitvoeren de tweede selectiefase; we starten hierbij van de minimale lijst met potentiële Routers.

We bespreken hieronder het Service Discovery Protocol, zoals gedefinieerd in de Bluetooth standaardspecificatie. Vervolgens gaan we na op welke manier BlueTools ondersteuning biedt voor diensten en het SDP. Tot slot bespreken we de concrete implementatie van deze fase in de Transparante Communicatielaag

# Het Service Discovery Protocol in de SIG Standaard

In tegenstelling tot de device discovery procedure, bevindt het Service Discovery Protocol zich vrij hoog in de protocol stack. Bovendien maakt het protocol op gecoördineerde wijze gebruik van L2CAP kanalen voor draadloze communicatie tussen de twee betrokken apparaten.

Via het SDP is het mogelijk om een toepassing te transformeren in een dienst, op voorwaarde dat gebruik maakt wordt van Bluetooth voor de draadloze communicatie. Dit biedt als voordeel dat een toepassing die van een dienst gebruik wenst te maken, niet bij voorbaat op de hoogte moet zijn van de apparaten die de gewenste dienst aanbieden. In de plaats worden met behulp van het SDP de beschikbare diensten van een apparaat opgevraagd, en wordt vervolgens voor een bepaalde instantie van een dienst gekozen. Dit zorgt voor een erg losse koppeling tussen aanbieders en gebruikers van diensten.

Een apparaat zou bijvoorbeeld kunnen optreden als krantenkiosk. Er kunnen dan meerdere instanties van een dienst van het type *Krant* aangeboden worden, bijvoorbeeld *De Morgen*, *De Standaard*, … Meerdere apparaten kunnen optreden als krantenkiosk, en eventueel verschillende kranten aanbieden via instanties van een dienst van het type *Krant*. De lezer hoeft niet op de hoogte te zijn van de locatie van de gewenste *Krant*-dienst; hij kan in de plaats aan elk apparaat binnen bereik vragen wat de aangeboden diensten zijn of – meer specifiek – welke instanties van de *Krant*-dienst er beschikbaar zijn.

De twee belangrijkste taken van het SDP zijn het adverteren van een dienst, en het doorzoeken van een apparaat naar beschikbare diensten.

## Het adverteren van een dienst

Het adverteren van een dienst bestaat uit het aanmaken van een Service Record, en het registreren van dit record bij de SDP Server.

Het Service Record bevat de informatie die gebruikt wordt bij het adverteren van één dienst, en verwijst naar één goedgedefinieerde instantie van die dienst op een welbepaald apparaat. Het record omvat ondermeer de volgende elementen:

- *Service Class ID List*. Een beschrijving van het type van de dienst, gebaseerd op een lijst met dienstklassen.
- *Service ID*. Een identiteit die de instantie van de dienst uniek definieert binnen het apparaat.
- *Service Name*. De naam van de dienst in klare tekst.
- *Service Description*. Een beschrijving van de dienst in klare tekst.

Bovendien kan men zelf extra elementen toevoegen aan het record.

Om het latere zoekproces te vergemakkelijken, definieert de standaard een hiërarchie waarbinnen diensten kunnen ondergebracht worden. De hiërarchie bestaat uit namen van elementen uit de protocol stack waarop de dienst steunt. Bovendien definieert [23] een aantal gestandaardiseerde dienstklassen, die zich eveneens binnen deze hiërarchie bevinden. Dit maakt het mogelijk om in grote mate de precisie te kiezen waarmee men naar een bepaalde dienst zoekt.

Eenmaal het record aangemaakt is, wordt het geregistreerd bij de SDP Server, en aldus geadverteerd naar toestellen binnen bereik. Elk apparaat dat zelf diensten kan aanbieden, bevat een SDP Server voor het beheren en adverteren van zijn Service Records.

Merk op dat het adverteren van een dienst een passief proces is; het bestaat enkel uit het registreren van een Service Record bij de SDP Server, en initieert zelf geen communicatie tussen apparaten.

## Het opsporen van een dienst

Het actieve onderdeel van het protocol bestaat uit het zoeken naar diensten van andere apparaten, en het antwoorden op aanvragen in het kader van het SDP.

Alvorens het zoekproces kan beginnen, heeft men het apparaatadres van het te doorzoeken toestel nodig. Een dergelijk apparaatadres kan bekomen worden aan de hand van de zoekprocedures besproken in 2.3.1. Eenmaal het te doorzoeken apparaat gekend is, wordt er een L2CAP logisch kanaal gebruikt voor de communicatie van de SDP-elementen van de betrokken apparaten.

Bij het doorzoeken van een apparaat, kan men een aantal attributen opgeven om de bekomen zoekresultaten te beperken. Deze attributen beperken het deel van de hiërarchie waartoe de gevonden diensten behoren.

# Diensten in Franson BlueTools

BlueTools biedt een uitgebreide ondersteuning voor de dienstgerichte architectuur van Bluetooth. We bespreken hieronder hoe men diensten kan zoeken en adverteren.

## Het adverteren van een dienst

BlueTools laat toe om zelf diensten te adverteren. Het is niet nodig om zelf een volledig Service Record op te bouwen; het volstaat om bij de registratie van een dienst bij de SDP Server een aantal van de record attributen mee te geven als parameter. De verplichte attributen zijn de naam van de dienst, commentaar en de klasse waartoe de dienst behoort.

Indien gewenst, kan men bij het adverteren het te gebruiken *Service Channel Number (SCN)* opgeven. Aan de hand van het SCN, kan men op een dynamische manier het protocolelement adresseren dat de binnenkomende verbindingen met de dienst zal ontvangen. Het is bijvoorbeeld mogelijk om een RFCOMM kanaal te koppelen aan een SCN, om op die manier verbindingen met de dienst te ontvangen via RFCOMM. Men kan een dienst adverteren op een vast SCN, en op die manier zorgen dat elke instantie van de dienst zich op hetzelfde SCN bevindt op verschillende apparaten. Dit biedt het voordeel dat de service discovery procedure niet strikt noodzakelijk is, en er in principe rechtstreeks verbinding kan gemaakt worden met de dienst, indien men weet dat deze aanwezig is. Het is evenwel zo dat het SCN reeds bezet kan zijn door een andere dienst, en het bijgevolg niet mogelijk is om de dienst te adverteren op het opgegeven SCN. Bovendien is het goed mogelijk dat er apparaten bestaan die een compleet andere dienst aanbieden op een bepaald SCN; men is bijgevolg niet zeker dat er een connectie gemaakt wordt met de gewenste dienst. In de handleiding van BlueTools wordt aanbevolen om het SCN automatisch te laten kiezen en gebruik te maken van het SDP om het SCN te verkrijgen.

## Het opsporen van een dienst

BlueTools laat toe om een service discovery uit te voeren aan de hand van een apparaatadres; men kan dit apparaatadres rechtstreeks opgeven in tekstvorm of een adres gebruiken dat eerder verworven werd via de device discovery procedure.

De functie voor het initiëren van de service discovery neemt als parameter een dienstklasse; op deze manier kan men naar wens instellen tot welk deel van de hiërarchie de gevonden diensten mogen behoren. De opgegeven dienstklasse geldt als wortel van de toegelaten deelboom in de hiërarchie. Indien men bijvoorbeeld *L2CAP* opgeeft, bekomt men zo goed als alle diensten die een apparaat aanbiedt; het opgeven van *RFCOMM* levert een iets kleinere deelverzameling van deze resultaten. Het gebruik van de *OBEX* dienstklasse levert anderzijds meestal slechts één dienst per apparaat op. Het komt er dus op aan om een goede dienstklasse op te geven om zo een minimale deelboom van de hiërarchie te bekomen.

Het opzoeken van de diensten van één apparaat neemt gemiddelde ongeveer één seconde in beslag. De gespendeerde tijd vertoont relatief grote schommelingen (tussen 0,30 s en 1,60 s) en blijkt in de praktijk niet merkbaar afhankelijk van het aantal geadverteerde diensten.

Het resultaat van een service discovery is het verwerven van een abstractie van het service record van elke gevonden dienst. Dit bevat ondermeer de benaming van de dienst en het SCN. Indien men beschikt over een apparaatadres en een SCN kan men verbinding maken met de geassocieerde dienst.

## Implementatie Transparante Communicatielaag

Bij de implementatie van de Transparante Communicatielaag hebben we in grote mate gesteund op de ondersteuning van BlueTools voor diensten.

Voor de realisatie van het Router-component, hebben we gebruik gemaakt van een extensie van een BlueTools basisklasse voor het implementeren van diensten. Eén dergelijke dienst komt overeen met één Transparant Netwerk, dat beheerd wordt door de betreffende Router. We hebben ervoor gekozen om de dienst te adverteren op een dynamisch gekozen SCN. De benaming van de dienst komt overeen met de benaming van het corresponderende Transparant Netwerk. Voor de dienstklasse hebben we geopteerd voor de *Network Access Point (NAP)* klasse.

De Terminal klasse gebruikt de SDP-voorzieningen van BlueTools om te bepalen welke toestellen een Router-dienst aanbieden die correspondeert met een gewenst Transparant Netwerk. Om na te gaan welke Routers een interessant Transparant Netwerk aanbieden, vraagt de Terminal de diensten van de NAP-klasse op bij elk van de potentiële Routers. Vervolgens wordt gebruik gemaakt van een reguliere expressie om te bepalen of de naam van de dienst al dan niet correspondeert met die van een gewenst Transparant Netwerk. Deze reguliere expressie wordt ingesteld bij het initialiseren van de Terminal.

# 2.3.3 Contextgevoelige selectie van de Router

In 2.3.1 *Zoeken naar potentiële Routers*, hebben we de mogelijkheden om tot een minimale lijst van nabijgelegen apparaten te komen besproken; we creëerden met de aldaar vernoemde zoekprocedures een eerste notie van context. In 2.3.2 *Selectie van potentiële Routers*, hebben we vervolgens gezien hoe we de lijst met potentiële Routers kunnen reduceren, op basis van hun mogelijkheid om een rol als Router te vervullen, en het Transparant Netwerk dat ze in deze rol aanbieden. Elk van de Routers van de gereduceerde lijst komt qua eigenschappen en vanuit technische standpunt, evenveel in aanmerking voor een uiteindelijke selectie; we willen bijgevolg graag nagaan of er een criterium bestaat waarmee we de context van het gewenste Transparant Netwerk kunnen versmallen. Op die manier proberen we de mogelijke Transparante Netwerken nog beter te

laten aansluiten bij de reële context, om zo – in het beste geval – volledig autonoom een netwerk te kiezen.

We zullen in dit onderdeel onderzoeken of we een indicator voor de ontvangen signaalsterkte kunnen aanwenden om een bepaalde Router al dan niet de voorkeur te geven boven alle andere. We willen in dit onderdeel dus graag proberen om de granulariteit van de contextgevoeligheid te verfijnen van het volledige zendbereik van de Terminal naar een fijnere resolutie. Het is meteen duidelijk dat de nauwkeurigheid van de contextherkenning bepalend is voor de uiteindelijke inzetbaarheid van de transparante netwerkoplossing.

De aanleiding voor dit onderzoeken is een vermoeden dat de ontvangen signaalsterkte in verband kan gebracht worden met de *relatieve* afstand tussen de communicatiepartners. We vermoeden met andere woorden dat het mogelijk is om te achterhalen welke de dichtste mogelijke communicatiepartner is.

# Signaalsterkte in de SIG Standaard

De Bluetooth standaardspecificatie legt drie vermogensklassen vast, gebaseerd op het uitgangsvermogen van de zender. De standaard vereist bovendien de implementatie van de controle van het uitgangsvermogen in alle toestellen die behoren tot vermogensklasse 1. De implementatie in toestellen van klasse 2 en 3 is optioneel maar wordt momenteel door de meeste Bluetooth chipoplossingen ondersteund.

Het doel van de vermogensregeling is het minimaliseren van de energieconsumptie, het maximaliseren van de kanaalkwaliteit en het optimaliseren van het algemene interferentieniveau. We hebben de Bluetooth vermogensklassen reeds kort besproken bij het overzicht van de protocol stack in 2.2.3.

De controle van het uitgangsvermogen wordt gestuurd door de Link Manager en is gebaseerd op een stapsgewijze aanpassingen van het uitgangssignaal. De grootte van de stappen wordt door de Bluetooth standaard niet nader gedefinieerd, zodat we een verschillend gedrag kunnen verwachten wanneer we verschillende toestellen vergelijken.

Voor de omvang van de vereiste aanpassingen steunt de Link Manager op indicaties die doorgegeven worden door communicatiepartners van het betreffende toestel. Concreet meten twee communicerende toestellen elk de ontvangen signaalsterkte afkomstig van de zender van de andere partner en wisselen ze, op basis van deze metingen, verzoeken uit over het zendvermogen van de communicatiepartner. De Link Manager van elke partner kan op basis van de ontvangen verzoeken beslissen of er al dan niet wijzigingen van het uitgangsvermogen nodig zijn.

De indicator voor de ontvangen signaalsterkte krijgt in de standaard de benaming *Received Signal Strength Indication (RSSI)*. De mate waarin de Link Manager het vermogen kan verlagen of opdrijven wordt beperkt door het minimale en maximale zendvermogen; deze begrenzing is afhankelijk van de hardware en de vermogensklasse waartoe het toestel behoort.

Om een optimale werking te garanderen moet de RSSI zich bij voorkeur bevinden in het optimale interval, dat bekend staat als het *Golden Receive Power Range* (GRPR). De grenzen van het GRPR zijn hardwareafhankelijk en niet universeel gedefinieerd; de grenzen van het optimale interval variëren bovendien in functie van de toepassing. De grenzen van het GRPR en de huidige waarde van de RSSI worden niet rechtstreeks beschikbaar gesteld aan lagen hoger dan de Host to Controller Interface. Het is echter wel mogelijk om een indicatieve waarde op te vragen aan de HCI, namelijk het verschil tussen de RSSI en de grenzen van het optimale interval. In wat volgt zullen we naar dit verschil refereren als *de indicator*. De indicator varieert binnen het interval [-128,+127]. Wanneer de RSSI zich binnen het optimale interval bevindt, is de indicator gelijk aan nul. Positieve en negatieve indicatorwaarden geven respectievelijk een ontvangen signaalsterkte boven en onder het optimale interval aan.

Het is belangrijk om op te merken dat een Bluetooth toestel er steeds naar zal streven om zijn zendvermogen aan te passen zodat de signaalsterkte, gemeten bij een communicatiepartner, zich binnen het optimale interval bevindt. Bluetooth zal dus, in de mate van het mogelijke, proberen om de indicatorwaarde tot nul te herleiden. Uit een onderzoek in [24] is reeds gebleken dat precies deze eigenschap de indicatorwaarde ongeschikt maakt voor precieze plaatsbepaling.

# De indicator in Franson BlueTools

Franson BlueTools biedt de indicator aan zoals gedefinieerd in de Bluetooth standaard. Conform de definitie in de standaard is de waarde van de indicator 0 indien de gemeten signaalsterkte zich binnen het optimale interval bevindt. Indien dit niet het geval is, en het uitgangsvermogen van de zender bijgevolg te hoog of te laag is, is deze waarde respectievelijk positief of negatief.

In principe kan de waarde van de indicator schommelen binnen [-128,+127]; de BlueTools handleiding vermeldt evenwel richtwaarden in het interval [-10,+10]. Waarden buiten dit interval zijn mogelijk, maar geven aan dat de communicatielink mogelijk zal blokkeren of zelfs verbroken worden. Gezien het doel van de indicatorwaarde, is deze enkel beschikbaar wanneer er een fysieke link bestaat tussen twee toestellen.

Jammer genoeg is de indicator niet beschikbaar voor toestellen die gebruik maken van de Microsoft protocol stack; deze toestellen doen wel zichtbaar aan vermogenscontrole, maar

de API's bieden niet de mogelijkheid om een indicatorwaarde op te vragen. Bovendien is deze indicator evenmin beschikbaar op een aantal Broadcom gebaseerde toestellen. In het geval van de Broadcom stack is de ondersteuning van de indicator afhankelijk van het toestel en van de softwareversie; volgens de BlueTools handleiding is ten minste versie 1.5 vereist. Hoewel bijvoorbeeld de *HP iPAQ hx2490* Pocket PC aan deze vereiste voldoet, is de indicator toch niet beschikbaar. Op de *Dell Axim X51* Pocket PC is de indicator evenmin beschikbaar maar nu als gevolg van het gebruik van de Microsoft stack. Indien het opvragen van de indicator niet ondersteund wordt door de stack, geeft BlueTools waarde 0 terug.

# Bruikbaarheid van de indicator

Om wat meer voeling te krijgen met de indicatorwaarde en de automatische vermogensregeling zullen we een experiment uitvoeren met twee toestellen, namelijk één router en één mobiele terminal. Het doel van dit experiment is een studie naar het bestaan van een verband tussen de indicatorwaarde enerzijds en de *relatieve* afstand tussen de communicatiepartners, en de oriëntatie van de mobiele terminal anderzijds.

## Beschrijving experiment

Het experiment zal bestaan uit het opstellen van één Router en één Mobiele Terminal in een obstakelvrije omgeving. Beide communicatiepartners bevinden zich op een hoogte van 110 cm; dit is bij benadering de hoogte waarop een Pocket PC zich bevindt bij normaal gebruik door een persoon van gemiddelde lengte. De positie van de Router is vast tijdens de duur van het experiment. De Mobiele Terminal is met het scherm naar de gebruiker gericht en maakt een constante hoek van 45° met de horizon. Tijdens het experiment brengen we de Terminal in positie op respectievelijk 5 m, 10 m, 15 m en 20 m van de Router. Bovendien herhalen we op elke positie het experiment voor verschillende rotaties van de Mobiele Terminal om de verticale as. Concreet bekijken we rotaties van 0°, 90°, 180° en 270° met de klok mee. De rotatiehoeken zorgen ervoor dat respectievelijk de achter-, linker-, voor- en rechterzijde van de Mobiele Terminal naar de Router gericht worden.

De Router en de Mobiele Terminal zijn beide voorzien van de Transparante Communicatielaag, en maken deel uit van hetzelfde Transparant Netwerk. De testsoftware op de Mobiele Terminal zorgt dat er verbinding gemaakt wordt met het Transparant Netwerk geassocieerd met de Router; deze verbinding wordt gedurende 30 s in stand gehouden. Tijdens de verbinding legt de router met een resolutie van 20 metingen per seconde de indicatorwaarde vast in een bestand. Een dergelijk experiment wordt vier keer uitgevoerd voor elke van de afstanden en de rotatiehoeken. Bovendien herhalen we het experiment voor de *HP iPAQ hx2490* Pocket PC en de *Dell Axim X51* Pocket PC.

# Resultaten

In Figuur 16 en Figuur 17 worden de resultaten weergeven van de eerste uitvoering van het experiment met een afstandsparameter van 20 m en constante rotatiehoek 0°. Beide grafieken geven voor respectievelijk de HP Pocket PC en de Dell Pocket PC de exacte meetwaarden, het tijdsgemiddelde over 1 s en het cumulatief gemiddelde weer.

Uit een eerste analyse van de grafieken en de meetgegevens blijkt onmiddellijk dat er een belangrijk verschil bestaat tussen beide toestellen, zowel wat betreft de grootte van de gemeten waarden als de schommelingen die deze waarden vertonen. In het geval van de HP Pocket PC schommelen de meetwaarden aanvankelijk tussen -14 en 0, om al na enkele seconden uit te doven tot het interval [-4,0]. De meetwaarden voor de Dell Pocket PC schommelen binnen het interval [-14,0] en gaan bovendien onverminderd door tot het einde van de meting.

Vooral Figuur 16 geeft een mooie illustratie van de automatische vermogensregeling. Onmiddellijk na het opzetten van de verbinding is de ontvangen signaalsterkte duidelijk te laag. Tijdens de daaropvolgende seconde wordt het vermogen in de mate van het mogelijke verhoogt, zodat de gemiddelde waarden dichter tot 0 naderen.

Merk op dat, na de eerste 5 seconden, het cumulatief gemiddelde relatief constant blijft tijdens het verdere verloop van de meting. Deze observatie zal ons toelaten het gemiddelde van de waarden, gemeten over 5 seconden, als representatief te beschouwen voor de gemiddelden berekend over een langer tijdsinterval. Gezien de beperkte tijd die we wensen te spenderen aan het verzamelen van meetgegevens voor de selectie van de Router, is het belang van deze vaststelling niet te onderschatten. We zullen deze observatie toepassen in de hierop volgende paragrafen voor het uitvoeren van een onderzoek naar het verband tussen de rotatiehoek en de waarde van de meetgegevens.

In Figuur 18 en Figuur 19 worden de gemiddelden van de meetwaarden voor de afstanden 5 m, 10 m, 15 m en 20 m over de eerste 5 seconden van de meting voor beide mobiele terminals weergegeven. Deze gemiddelden zijn gegroepeerd per rotatiehoek.

De gemiddelde waarden voor de Dell Mobiele Terminal bevinden zich in Figuur 19. We stellen onmiddellijk vast dat er voor een gegeven rotatiehoek van 90°, 180° en 270° een duidelijk verband bestaat tussen de afstand en het gemiddelde van de meetgegevens voor die afstand. Dit geldt echter niet voor de gemiddelden gemeten bij een hoek van 0°, als gevolg van de optimale indicatorwaarde wanneer de Mobiele Terminal zich met deze hoek op 10 m van de router bevindt. Hoewel er voor de rotatiehoeken van 90°, 180° en 270° een verband bestaat tussen de onderlinge afstand en het gemeten gemiddelde, wegen de wijzigingen van de indicatorwaarden als gevolg van een verschillende onderlinge afstand, niet op tegen de verschillen geïntroduceerd door het wijzigen van de rotatiehoek. We illustreren het gevolg van deze vaststelling met het volgende voorbeeld.

Stel dat er twee Routers op een afstand van 25 m uit elkaar opgesteld staan. De Mobiele Terminal bevindt zich ergens op de rechte tussen de twee Routers, zodat er zich één Router links en één Router rechts van de Mobiele Terminal bevindt. Dit komt in feite neer op een rotatie van 90° en 270° met de klok mee, ten opzichte van respectievelijk de linker en de rechter Router. Vervolgens meten we de indicatorwaarde gedurende 5 s in beide routers en berekenen we hieruit de gemiddelde waarden. Als gevolg van de opstelling bekomen we telkens één gemiddelde uit groepering 90° en één gemiddelde uit groepering 270°, waarbij de som van de bijhorende afstanden 25 m bedraagt. Indien de mobiele terminal zich bijvoorbeeld op 10 m en 15 m van respectievelijk de linker en de rechter router bevindt, bekomen we bijvoorbeeld gemiddelden van ongeveer -4 en -2; de rechter router wordt dus met andere woorden verkeerdelijk als de dichtste router aanzien. Uit de gemiddelden in Figuur 19 kunnen we bovendien besluiten dat de rechter router steeds aanzien wordt als de dichtste, behalve indien de afstand tot de linker router minder dan 5 m bedraagt.

Intuïtief verwachten we de beste indicatorwaarden te verkrijgen wanneer de Mobiele Terminal in de richting van de Router gericht wordt, met andere woorden bij een rotatie van 0°. Uit Figuur 17 blijkt echter dat een rotatie van 180° over het algemeen de beste oplevert, ongeacht de onderlinge afstand. Een dergelijke rotatie treedt op bij normaal gebruik van de Mobiele Terminal, wanneer de gebruiker de Router de rug toekeert.

Uit de analyse van de overeenkomstige gegevens in Figuur 18 voor de HP Mobiele Terminal, besluiten we dat de oriëntatie van het toestel opnieuw belangrijkere wijzigingen introduceert op de meetwaarden, dan een wijziging van de onderlinge afstand. Vooral rotaties van 180° en 270° introduceren een opmerkelijke daling van de indicatorwaarden. Een rotatie van 0° levert hier over het algemeen de beste indicatorwaarden op, ongeacht de onderlinge afstand. Deze vaststelling laat vermoeden dat de ontwerpers van dit toestel rekening gehouden hebben met de natuurlijke reflex om het toestel te richten naar de gewenste communicatiepartner.

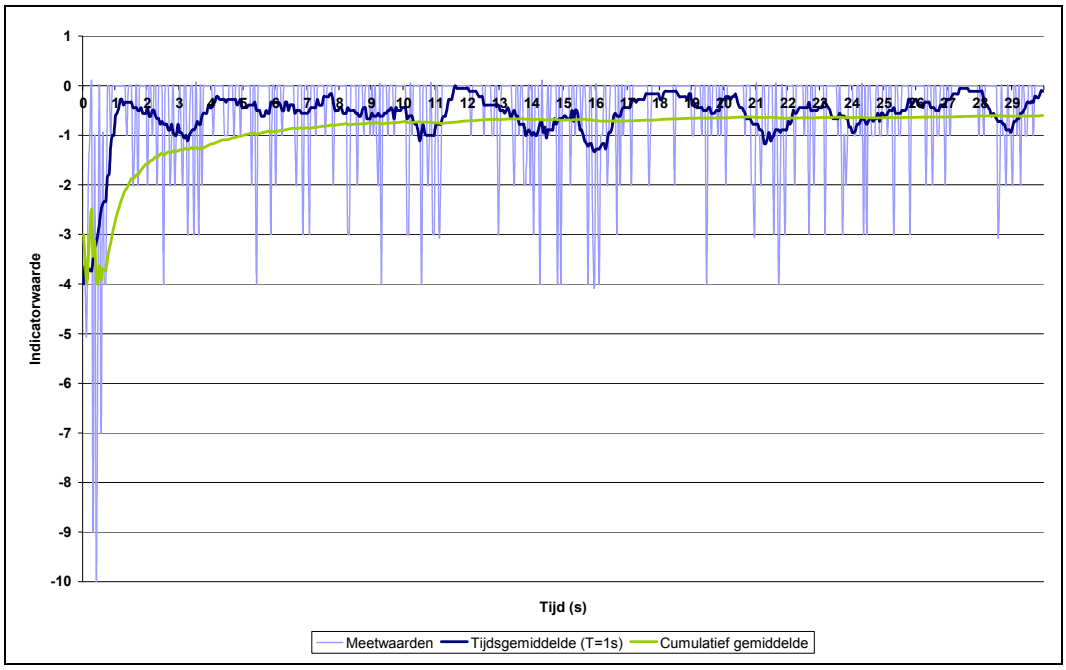
# Conclusie

Uit de bovenstaande resultaten blijkt duidelijk dat de indicator voor de ontvangen signaalsterkte ongeschikt is om zondermeer de relatieve afstanden tot de beschikbare Routers in te schatten. Het is echter wel mogelijk om met behulp van de bekomen resultaten het aantal keuzemogelijkheden verder te reduceren, wanneer we veronderstellen dat de gebruiker van de Mobiele Terminal deze op een welbepaalde manier zal richten naar de gewenste Router. Dit hangt evenwel af van het gebruikte toestel.

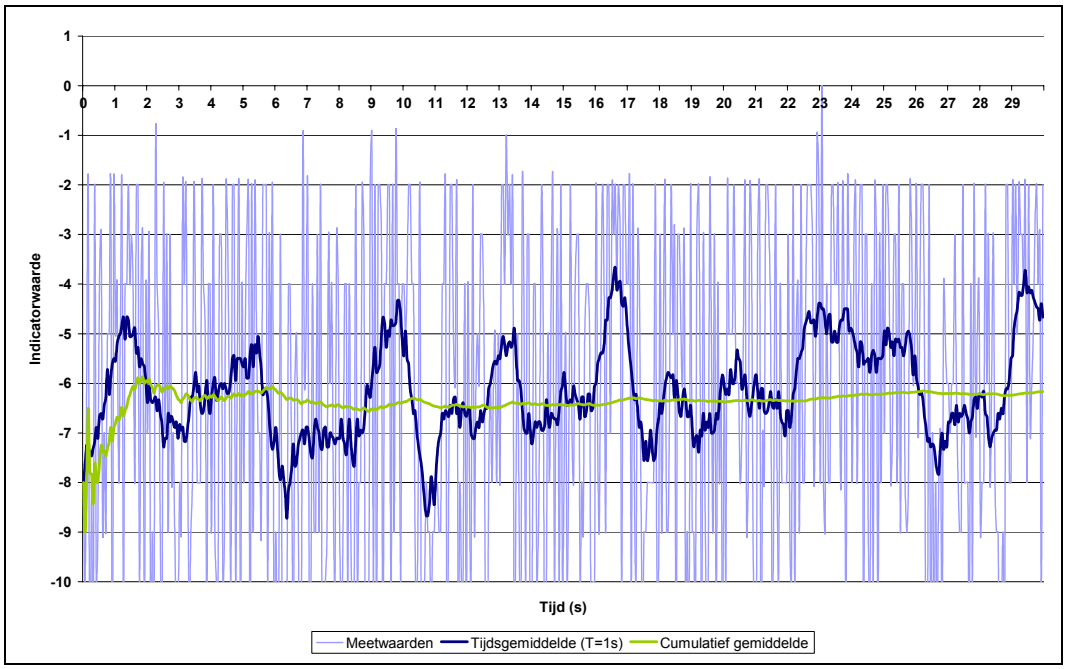
Voor de HP Pocket PC veronderstellen we dat de gebruiker deze intuïtief naar de gewenste Router richt. Uit Figuur 18 blijkt dat we in dit geval alle Routers, die een gemiddelde meetwaarde opleveren onder -1, buiten beschouwing mogen laten. Indien de Routers op

uniforme wijze verspreid opgesteld staan in de omgeving, betekent dit dat we ongeveer de helft van de keuzemogelijkheden kunnen elimineren.

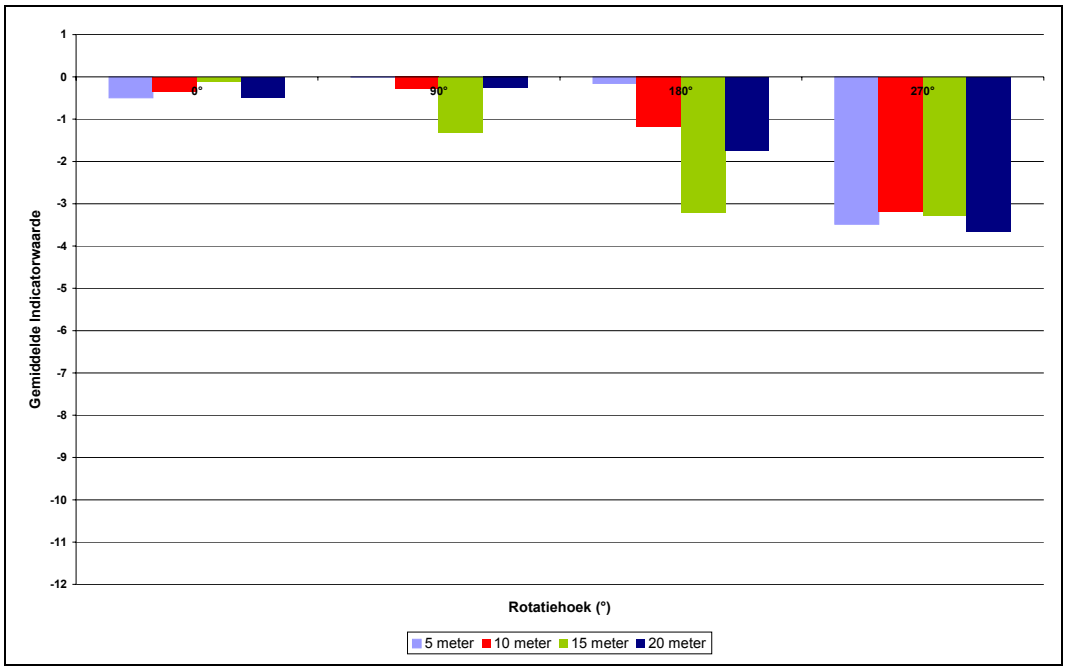
Voor de Dell Pocket PC zouden we kunnen veronderstellen dat het mogelijk is om de bekomen gemiddelden over 180° te verschuiven; met deze veronderstelling bekomen we de resultaten van een rotatiehoek van 180° wanneer de gebruiker het toestel naar de Router toe richt, in plaats van de rug toekeert. We zouden dit bijvoorbeeld kunnen realiseren door het veranderen van de inwendige lay-out van het toestel. Rekening houdend met de gemaakte veronderstelling, kunnen we nu een gelijkaardige eliminatie bekomen zoals bij de HP Pocket PC. We elimineren in dit geval echter elke Router die een meetwaarde oplevert lager dan -2. Uit Figuur 19 blijkt dat we, in het geval van een uniforme distributie van de Routers in de omgeving, opnieuw ongeveer de helft van de keuzemogelijkheden kunnen elimineren.



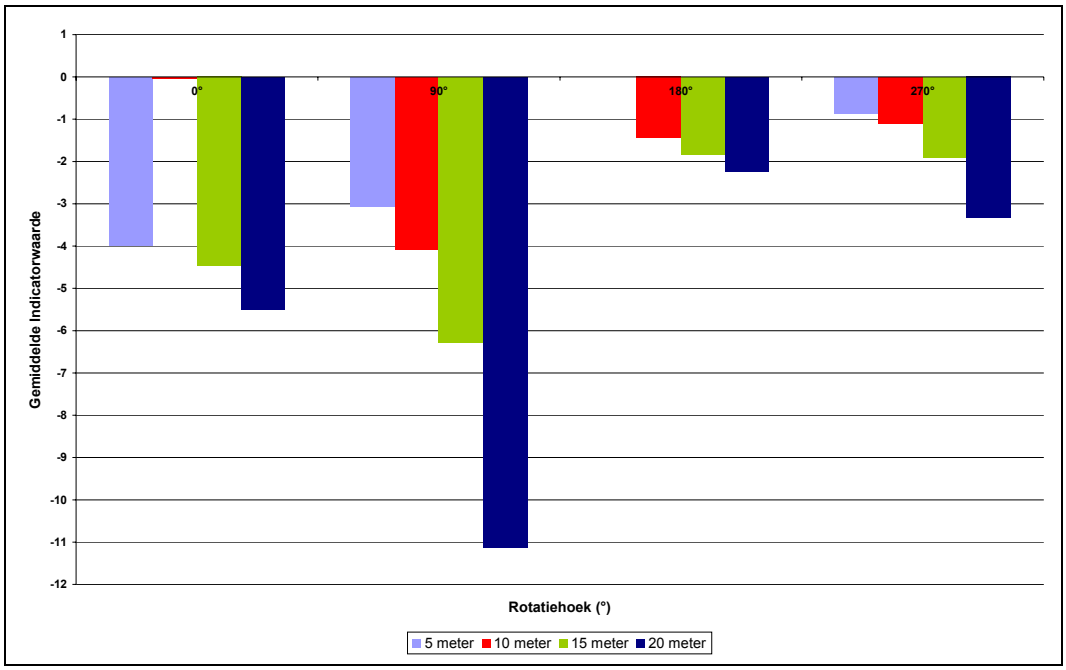

Figuur 16: Indicatorwaarden voor de HP iPAQ hx2490 in functie van de tijd (20 m, 0°, meting 1)




Figuur 17: Indicatorwaarden voor de Dell Axim X51 in functie van de tijd (20m, 0°, meting 1)




Figuur 18: Gemiddelde indicatorwaarden over 5 s voor de HP iPAQ hx2490 in functie van de rotatiehoek




Figuur 19: Gemiddelde indicatorwaarden over 5 s voor de Dell Axim X51 in functie van de rotatiehoek

# Implementatie Transparante Communicatielaag

Gezien het karakter van de bekomen resultaten, mogen we veronderstellen dat het praktisch niet haalbaar is om een volledig geautomatiseerde keuze van de Router te realiseren, uitsluitend gebaseerd op een indicator voor de ontvangen signaalsterkte. In heel wat gevallen zal een korte interactie met de gebruiker noodzakelijk blijven. Verder moet men er rekening mee houden dat de bekomen resultaten afhankelijk zijn van het gebruikte type Mobiele Terminal, zodat er voor elk toesteltype een specifieke configuratie van een aantal parameters vereist zal zijn.

De Transparante Communicatielaag laat aan de bovenliggende toepassing de keuze of er, voor de automatische partnerselectie, al dan niet gebruik dient gemaakt te worden van de indicator voor de ontvangen signaalsterkte. Bovendien kan elke Terminal een eigen drempelwaarde instellen voor gebruik bij de eliminatie van Routers.

Indien de toepassing ervoor opteert om deze mogelijkheden te benutten, maakt de Mobiele Terminal gebruikt van het algoritme in Figuur 20 voor de partnerselectie.

```
routerList := discover_routers();

for each router in routerList do
   connect_to_router(router);
   send_indicator_request(router);
   indicator_value := receive_indicator_value(router);

   if indicator_value < thresshold then
      routerList.remove(router);
   end if;
end loop;

if routerList.length() = 0 then
   schedule_discovery_and_sleep();
elsif routerList.length() = 1 then
   connect(routerList[1]);
else
   sort_by_indicator_value(routerList);
   request_user_interaction(routerList);
   router := get_user_choice;
   connect(router);
end if;
```

Figuur 20: Algoritme voor contextgevoelige Router selectie

De Mobiele Terminal bepaalt eerst de minimale lijst met potentiële Routers. Daarna wordt er achtereenvolgens verbinding gemaakt met elk van de routers. Tijdens de duur van de verbinding meet de router elke 50 ms de indicatorwaarde. Voor onze toepassing zullen we volstaan met 100 metingen, aangezien we eerder aantoonden dat het gemiddelde over 5 s beschouwd kan worden als voldoende representatief voor het gemiddelde over een lang interval (bijvoorbeeld 30 s). Na 5 s stuurt de router dit gemiddelde naar de mobiele terminal en wordt de verbinding afgebroken. Een router wordt alvast verwijderd uit de lijst indien het ontvangen gemiddelde lager is dan een vooraf ingestelde drempelwaarde, bijvoorbeeld -1 voor de Dell Pocket PC en -2 voor de HP Pocket PC. Wanneer de mobiele terminal alle meetwaarden verzameld heeft, wordt de lijst met mogelijke

communicatiepartners gesorteerd op de gemiddelde indicatorwaarde, en voorgelegd aan de gebruiker voor verdere selectie. Indien de lijst leeg is of slechts één element bevat, ligt de te nemen actie uiteraard voor de hand en is er geen tussenkomst van de gebruiker vereist.

Merk op dat de potentiële verhoging van de transparantie een verlenging van de Router-selectiefase impliceert. Concreet bedraagt deze verlenging een vijftal seconden per router binnen bereik. De lineaire toename van de duur van de selectiefase, wordt echter gerechtvaardigd door het vergemakkelijken van de moeizame invoer of selectie op het kleine scherm van de Mobiele Terminal. De moeilijkheid van de invoer kan immers eveneens als lineair in het aantal keuzemogelijkheden beschouwd worden, zodat het reduceren van het aantal keuzemogelijkheden zeker het overwegen waard is.

### 2.3.4 Lidmaatschap van een Transparant Netwerk

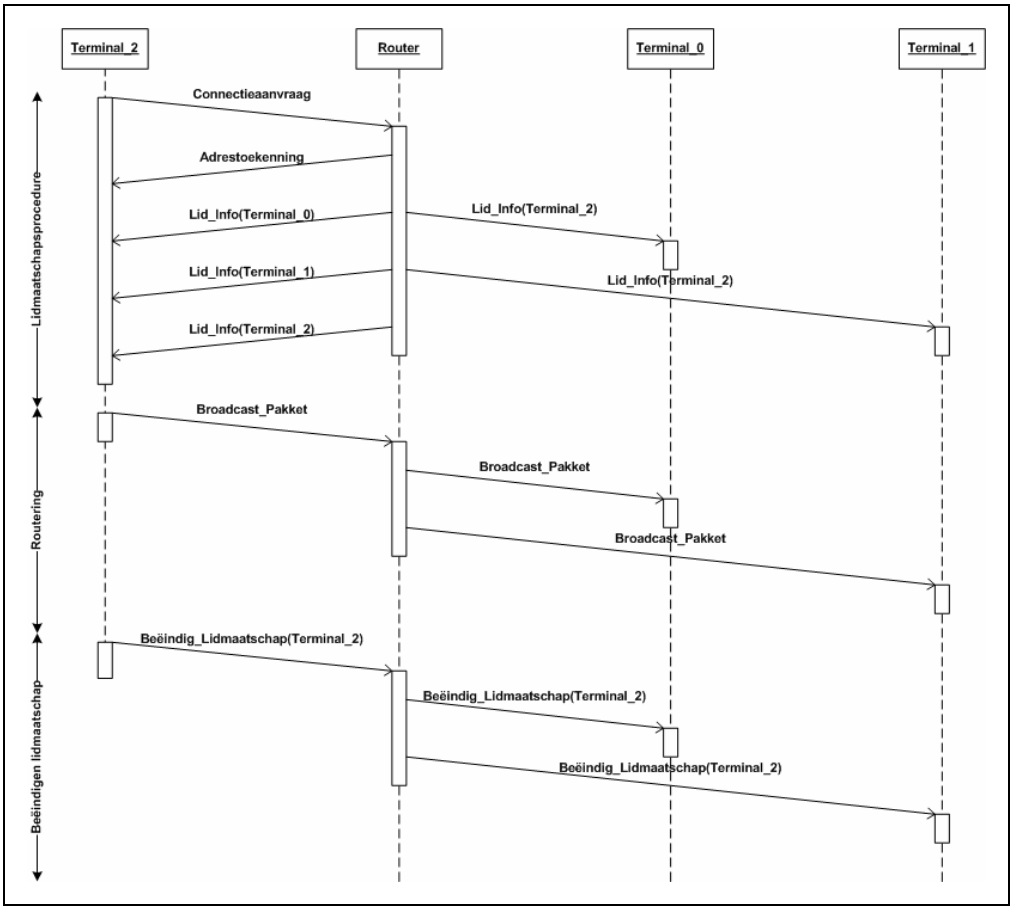
In 2.3.1 *Zoeken naar potentiële Routers* en 2.3.2 *Selectie van potentiële Routers*, hebben we besproken hoe een Terminal een minimale lijst met potentiële Routers kan bekomen, die een gewenst Transparant Netwerk aanbieden. Vervolgens zijn we in 2.3.3 *Contextgevoelige selectie van de Router* nagegaan hoe we, aan de hand van een indicator voor de ontvangen signaalsterkte, de lijst met potentiële Routers verder kunnen reduceren; op deze manier zorgen we dat de resterende keuzemogelijkheden beter aansluiten bij de reële context van de Mobiele Terminal. Indien er uiteindelijk meerdere potentiële Routers overblijven, is een interactie met de gebruiker echter niet te vermijden. We veronderstellen in deze sectie dat er – al dan niet met de tussenkomst van de gebruiker – een Router gekozen is, en dat de Terminal bijgevolg lid wenst te worden van het Transparant Netwerk geassocieerd met de Router.

De eerste stap van de verbindingsprocedure bestaat uit het realiseren van een gegevensverbinding tussen de Terminal en de geselecteerde Router. De gegevensverbinding wordt geïnitieerd door de Terminal; deze maakt gebruik van het apparaatadres van de Router en het Service Channel Number van de dienst, geassocieerd aan het geselecteerde Transparant Netwerk. Het resultaat is de realisatie van een RFCOMM transportkanaal tussen beide apparaten. Het bestaan van een gegevensverbinding impliceert echter niet dat de Terminal reeds een volwaardig lid is van het netwerk.

Gedurende het bestaan van de gegevensverbinding tussen een Terminal en een Router, worden drie grote fases doorlopen:

- *Lidmaatschapsprocedure*. Deze procedure zorgt dat de Terminal een volwaardig lid wordt van het Transparant Netwerk; dit houdt ondermeer in dat de Router een Transparant Netwerkadres toekent aan de Terminal.
- *Routering*. Tijdens zijn lidmaatschap van een Transparant Netwerk, kan een Terminal gebruik maken van de routeringsfunctionaliteit van de Router voor het afleveren van zijn berichten aan andere Terminals.
- *Beëindiging lidmaatschap*. De procedure voor het beëindigen van het lidmaatschap van een Terminal van een Transparant Netwerk.

In Figuur 21 wordt een overzicht getoond van de drie grote fases; we zullen deze in wat volgt kort bespreken.




Figuur 21: Overzicht van de fases in een Transparant Netwerk

# De lidmaatschapsprocedure

Onmiddellijk na het realiseren van een gegevensverbinding, initieert de Terminal de lidmaatschapsprocedure met een connectieaanvraag aan de Router. Deze aanvraag bevat ondermeer de naam en het type van de betreffende Terminal; de beschikbare gegevens zijn afhankelijk van de precieze toepassing waarin de Transparante Communicatielaag ingezet

wordt. Men kan de Terminalgegevens bijvoorbeeld gebruiken voor het implementeren van een toepassingsgebonden beveiligingspolitiek.

Wanneer de Router de verbindingsaanvraag ontvangt, reserveert deze een Transparant Netwerkadres voor de Terminal; het adres wordt aan de Terminal meegedeeld via een adrestoekenningsbericht.

Eenmaal er een Transparant Netwerkadres toegekend is aan de Terminal, brengt de Router de reeds aanwezige leden van het Transparant Netwerk op de hoogte van het nieuwe lidmaatschap; hiertoe wordt naar elk reeds aanwezig lid een bericht gestuurd met informatie over de nieuwe Terminal. De Router brengt de nieuwe Terminal eveneens op de hoogte van de reeds aanwezige leden.

Tot slot wordt de nieuwe Terminal op de hoogte gebracht van zijn eigen lidmaatschap van het Transparant Netwerk; voortaan wordt deze Terminal beschouwd als een volwaardig lid van het netwerk.

# Routering

De Router biedt aan de leden van zijn Transparant Netwerk een routeringsdienst aan; met behulp van deze dienst kunnen de Terminals onderling communiceren. Er is ondersteuning voor routering van unicast-, multicast- en broadcastberichten.

Om de doorvoercapaciteit van het Transparant Netwerk eerlijk te verdelen onder de aanwezige leden, en om de latentietijd beperkt te houden, maakt de Router gebruik van één uitgaande wachtrij per Terminal. De berichten in de wachtrijen worden behandeld volgens het *round robin* verwerkingsregime. Jammer genoeg biedt dit geen garanties voor een goede doorstroming binnen het netwerk. We zullen in de volgende sectie namelijk met een experiment aantonen dat het Transparant Netwerk gedurende een aantal seconden volledig blokkeert, alvorens een communicatielink breekt; de linkbreuk kan bijvoorbeeld optreden ten gevolge van het uit bereik gaan van een Terminal.

# Het beëindigen van het lidmaatschap

Er zijn diverse mogelijke redenen waarom een Terminal zijn lidmaatschap van het Transparant Netwerk beëindigt. Het is bijvoorbeeld mogelijk dat de applicatie aangeeft dat het lidmaatschap niet langer nodig is aan het onderliggende Terminal-component. De Terminal verstuurt een aanvraag naar de Router om het lidmaatschap te beëindigen, en deze laatste propageert de mededeling naar de overige aanwezige leden van het netwerk. De uitgewisselde berichten worden getoond in Figuur 21. Dit is de meest gecontroleerde manier om het lidmaatschap te beëindigen, en moet bijgevolg de voorkeur krijgen. Een applicatie is echter niet altijd in staat om autonoom een dergelijke beslissing nemen.

In het kader van de Transparante Communicatielaag wensen we na te gaan of we het lidmaatschap kunnen beëindigen wanneer een gebruiker de context van het Transparant Netwerk verlaat. Een eerste notie van context bestaat uiteraard uit het bereik van de Mobiele Terminal; wanneer de Router zich niet meer binnen bereik bevindt, wordt de communicatielink tussen beide uiteraard verbroken. We zullen echter in het volgende experiment aantonen dat een dergelijke linkbreuk gepaard gaat met de tijdelijk blokkering van het volledige Transparant Netwerk.

## Breken van een communicatielink

Uit de bespreking van de Bluetooth protocol stack in 2.2.3, volgt dat we voor de gegevensverbinding in feite steunen op een ACL link. Als gevolg van het eenvoudige 1-bit herverzendingsschema van deze link, kan de gegevenstransmissie vrij snel blokkeren indien er botsingen optreden of wanneer de communicatiepartner uit bereik gaat. Aangezien de pakketnummers voor herverzending praktisch onmiddellijk uitgeput zijn, en er geen ontvangstbevestiging komt van de communicatiepartner, kan de interface onmogelijk verdere communicatie binnen het piconet ondersteunen; er moet immers eerst een bevestiging komen of een effectieve linkbreuk optreden. De lagen boven de ACL link voegen zelf ook een eigen herverzendingsschema toe; bijgevolg kan de protocol stack in praktijk tot een vijftal seconden blokkeren alvorens de link daadwerkelijk verbroken wordt. In het geval van de Transparante Communicatielaag, zorgt dit niet alleen voor het blokkeren van de link tussen de Router en een Terminal, maar blokkeert in de praktijk alle communicatie in het Transparant Netwerk.

We illustreren de problematiek als gevolg van de linkblokkering met het onderstaande experiment. Vervolgens stellen we een oplossing voor die steunt op de indicator voor de ontvangen signaalsterkte; we introduceerden deze indicator reeds eerder in 2.3.3 *Contextgevoelige selectie van de Router*.

## Beschrijving experiment

Met het onderstaande experiment willen we de blokkering van een communicatielink tussen een Mobiele Terminal en de Router van naderbij bestuderen; concreet willen we nagaan wat de invloed van de blokkering op de communicatie tussen de overige leden van het Transparant Netwerk is.

We beschikken voor het experiment over een opstelling met één Router en twee Mobiele Terminals, namelijk een *Dell Axim X51* Pocket PC en een *HP iPAQ hx2490* Pocket PC. Aanvankelijk bevinden de Router en de Mobiele Terminals zich allemaal binnen eenzelfde kamer. De beide Mobiele Terminals registreren zich bij de Router, en worden lid van hetzelfde Transparant Netwerk. Het toestel met het Router-component zal in dit experiment eveneens optreden als Terminal; deze Terminal genereert een constante stroom van broadcastberichten aan ongeveer 17 KBps. De Router zorgt voor de aflevering van deze

berichten aan de beide Mobiele Terminals. De Router legt, gedurende de verbinding met de Mobiele Terminals, de corresponderende waarden van de indicator voor de ontvangen signaalsterkte vast. Bovendien registreren beide Mobiele Terminals de omvang van de ontvangen berichten per tijdsinterval.

Gedurende het experiment behoudt de Dell Mobiele Terminal zijn positie op ongeveer 2 m van de Router. We zullen echter in de loop van de proefneming de afstand tussen HP Mobiele Terminal en de Router op een gelijkmatige manier verhogen. Zo gaan we na wanneer er een blokkering optreedt tussen de HP Mobiele Terminal en de Router, en wat de invloed van deze blokkering is op de communicatie tussen de Router en de Dell Mobiele Terminal.

# Resultaten

De resultaten van het experiment worden getoond in Figuur 22. In de grafiek tonen we voor beide Mobiele Terminals de omvang van de ontvangen berichten per tijdseenheid. Bovendien geven we voor beide Mobiele Terminals de waarden van de indicator voor de ontvangen signaalsterkte weer. Zoals reeds vermeld wordt deze indicator gemeten in de Router.

Aanvankelijk benaderen de gemeten waarden voor beide Mobiele Terminals vrij nauwkeurig de vooropgestelde waarden; voor de doorvoer verwachten we inderdaad ongeveer 17 KBps, terwijl de indicatoren voor de ontvangen signaalsterkte een optimale waarde aannemen als gevolg van de beperkte onderlinge afstand.

We zien duidelijk dat het verhogen van de afstand tussen de HP Mobiele Terminal en de Router, een daling van de indicator voor de ontvangen signaalsterkte impliceert. Een eerste sterkte daling van deze indicator treedt op rond de twintigste seconde van het experiment, wanneer de HP Mobiele Terminal de kamer verlaat. We blijven de afstand tussen de HP Mobiele Terminal en de Router verhogen tot de communicatielink met de Router uiteindelijk breekt. De linkbreuk treedt op rond de vijfenveertigste seconde van het experiment.

Uit Figuur 22 blijkt dat het breken van de communicatielink tussen de HP Mobiele Terminal en de Router, gepaard gaat met drie belangrijke nevenverschijnselen. De breuk impliceert uiteraard de scherpe val van de doorvoer van de communicatielink tussen de HP Pocket PC en de Router. Het is eveneens duidelijk te zien, dat ook de andere communicatielink – tussen de Router en de Dell Pocket PC – in belangrijke mate verstoord wordt door de blokkering; ondanks de kleine afstand tussen de Dell Pocket PC en de Router, is deze link gedurende maar liefst vijf seconden onbruikbaar. Een laatste observatie betreft de waarde van de indicator voor de gemeten signaalsterkte van de HP Mobiele Terminal; vlak voor het blokkeren en breken van de link, vertoont deze een erg scherpe val tot ongeveer -6.

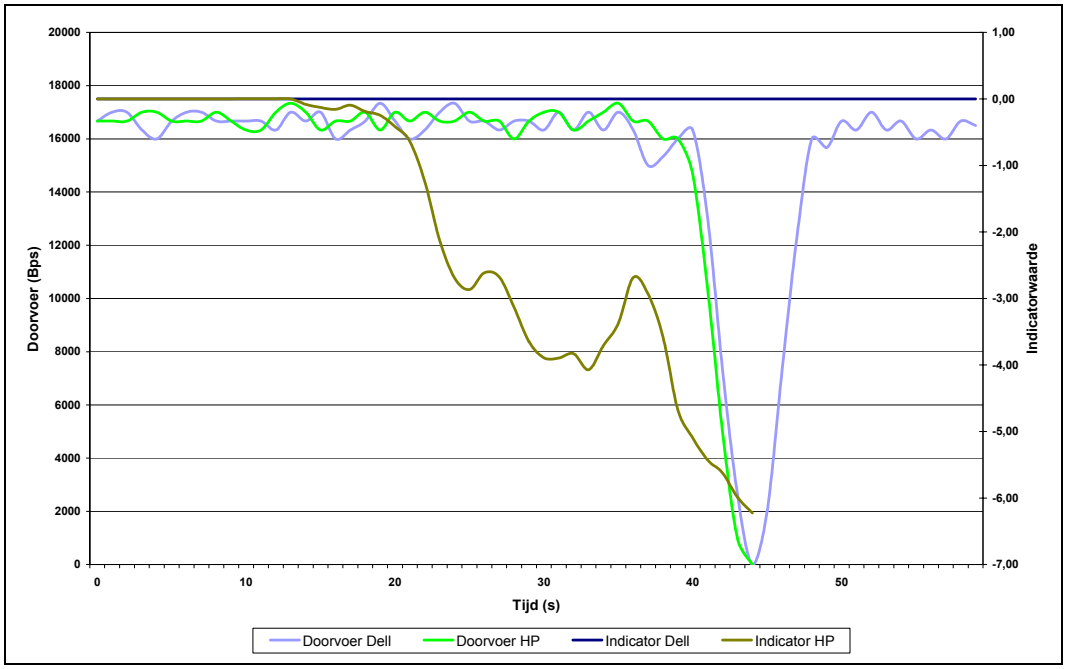
# Conclusie

We hebben met de voorgaande proefneming aangetoond dat een linkbreuk tussen de Router en een Mobiele Terminal een sterke invloed heeft op alle overige communicatielinks in de context van een Transparant Netwerk. De omvang van de hinder voor de overige links zal voor heel wat toepassingen onaanvaardbaar zijn.

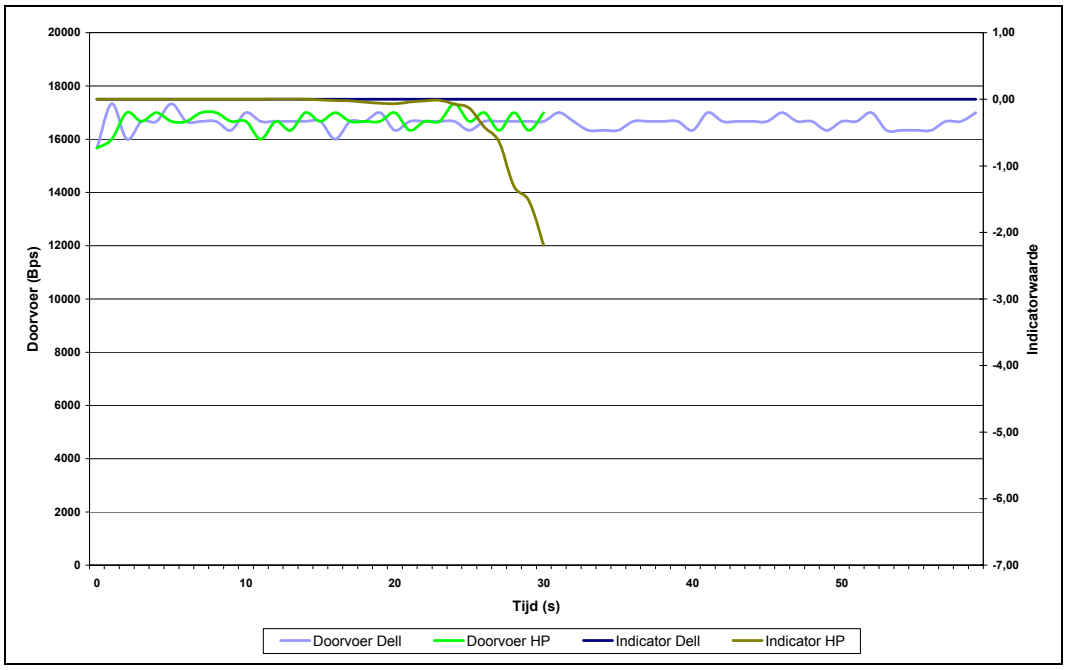
Gelukkig is het mogelijk gebleken om het blokkeren van een link te voorspellen aan de hand van de indicator voor de ontvangen signaalsterkte. We stellen dan ook voor om het Router-component van de Transparante Communicatielaag te voorzien van een *link monitor*; deze berekent het tijdsgemiddelde van de indicator voor de ontvangen signaalsterkte voor elke link, en verbreekt een link wanneer dit gemiddelde beneden een instelbare ondergrens zakt.

Wanneer we voorgaand experiment herhalen met de voorgestelde link monitor, bekomen we de resultaten in Figuur 23. In het weergegeven geval gebruiken we -2 als ondergrens; deze waarde komt in de praktijk overeen met het verlaten van de kamer waarin de Router zich bevindt. Merk op dat dit de contextgevoeligheid van de Transparante Communicatielaag in grote mate ten goede komt; het lidmaatschap met het Transparant Netwerk wordt immers in heel wat gevallen verbroken als gevolg van het verlaten van de context geassocieerd met dat netwerk.

Het is duidelijk dat we het blokkeren van een link in heel wat gevallen kunnen vermijden door deze preventief te verbreken. We merken evenwel op dat de prestaties van de link monitor afhankelijk zijn van de specifieke omgeving waarin de Transparante Communicatielaag ingezet wordt; een hoog interferentieniveau kan bijvoorbeeld zorgen dat een link toch blokkeert, ondanks een goede indicatorwaarde.




Figuur 22: Blokkering communicatielinks zonder link monitor




Figuur 23: Blokkering communicatielinks met link monitor

# 2.3.5 Uitbreiding naar Linux

Zoals blijkt uit de bespreking van de Bluetooth API's in 2.2.4, was de implementatie van de Transparante Communicatielaag aanvankelijk uitsluitend voorzien voor PC's en Pocket PC's met respectievelijk Windows XP en Windows Mobile 5. Het is echter mogelijk om de Transparante Communicatielaag uit te breiden naar Linux. In het kader van dit eindwerk kwam geen volledige implementatie voor Linux tot stand; we onderzochten echter wel de mogelijkheid om een dergelijke implementatie te realiseren.

Bij de implementatie van de Transparante Communicatielaag met BlueTools steunden we enkel op het RFCOMM protocol, voor het realiseren van een punt-naar-puntverbinding tussen een Terminal en zijn Router. De standaard Bluetooth protocol stack implementatie voor Linux is BlueZ[25]; deze implementatie biedt uiteraard eveneens ondersteuning voor het RFCOMM protocol. We hebben dus goede redenen om er van uit te gaan dat we de Transparante Communicatielaag kunnen uitbreiden naar Linux. Jammer genoeg is het enkel mogelijk om de BlueZ protocol stack te programmeren met C/C++. Momenteel is er slechts één open source initiatief voor de implementatie van een BlueZ Java klassenbibliotheek, namelijk JBlueZ[26]. De implementatie van de bibliotheek is echter helemaal nog niet ver gevorderd en biedt momenteel nog niet de nodige voorzieningen voor het implementeren van de Transparante Communicatielaag. Om aan te tonen dat het echter wel mogelijk is om een uitbreiding naar Linux te realiseren, hebben we gebruik gemaakt van het Linux shell script in Figuur 24, in combinatie met een Java toepassing. We schetsen hieronder de werking van het script en de bijhorende toepassing.

```
sdptool add --channel=1 SP
while true;
do
   rfcomm listen /dev/rfcomm0 1
done
```

Figuur 24: Script voor het ontvangen van binnenkomende Bluetooth verbindingen

In de eerste lijn van het script wordt een SDP service record gemaakt voor het adverteren van een seriële poort op Service Channel Number 1; vervolgens wordt een lus doorlopen voor het ontvangen van binnenkomende verbindingen op dat SCN. Het commando *rfcomm listen /dev/rfcomm0 1* blokkeert tot er een RFCOMM verbinding binnenkomt op SCN 1; de binnenkomende verbinding wordt vervolgens met behulp van het Bluetooth seriële poort profiel aangeboden via de virtuele seriële poort */dev/rfcomm0*. De Java toepassing zal gebruik maken van de virtuele seriële poort voor de verdere communicatie.

Bij de implementatie van de Java toepassing constateerden we dat de Java Virtuele Machine standaard geen mogelijkheden bezit voor het gebruik van seriële poorten; onder Linux kan de Virtuele Machine hiertoe echter uitgebreid worden met de Sun Comm API[27]. Om de communicatielink effectief te openen, controleert de toepassing periodiek of

de virtuele seriële poort geopend kan worden. Wanneer het Router-component van de Transparante Communicatielaag verbinding maakt via de Bluetooth interface van de Linux computer, ontvangt de Java toepassing de binnenkomende connectie via de virtuele seriële poort en is er communicatie mogelijk.

Het is duidelijk dat de geschetste methode niet meteen toepasbaar is in een commerciële toepassing. We toonden echter wel aan dat het mogelijk is om met Bluetooth op een transparante manier een communicatielink te realiseren onder Linux. Het implementeren van een Java klassenbibliotheek voor het programmeren van de BlueZ protocol stack, en de Java implementatie van de Transparante Communicatielaag voor Linux, vormen ongetwijfeld een interessant onderwerp voor verder onderzoek. We merken tot slot op dat de mogelijkheid om de Transparante Communicatielaag uit te breiden naar Linux een voorwaarde is voor slagen van het scenario in 1.2.4 *Communicatie Digibox/IVT*.

## 2.4 Conclusie

We hebben in dit hoofdstuk de haalbaarheid aangetoond van een transparante netwerkoplossing die een antwoord kan bieden op de problemen uit het scenario in 1.2.1.

Na een korte studie van de beschikbare communicatietechnologieën, hebben we gekozen voor Bluetooth voor het ondersteunen van de netwerkoplossing. We bekeken deze draadloze communicatietechnologie van naderbij, en concludeerden dat ze omwille van haar karakteristieken uitermate geschikt is voor onze doeleinden. Het bleek evenwel dat de technologie standaard niet in staat is om op voldoende transparante wijze een ad-hoc netwerk te vormen.

Een eerste probleem van softwareontwikkeling met Bluetooth, betreft het kiezen van een API voor het programmeren van de protocol stack; er zijn op dit vlak immers geen *open source* initiatieven beschikbaar voor de besturingssystemen van Microsoft. Er is bovendien nog relatief weinig ervaring met het schrijven van softwaretoepassingen, die steunen op Bluetooth voor hun draadloze communicatie. We hebben echter aangetoond dat er voldoende technisch en commercieel haalbare oplossingen voor handen zijn.

We hebben met de Transparante Communicatielaag geïllustreerd dat we de vorming van ad-hoc netwerken met Bluetooth volledig transparant kunnen maken voor de gebruiker met behulp van een softwareoplossing. De transparantieproblematiek verplaatst zich hiermee van de technische vorming van het netwerk, naar de contextgevoelige selectie van de apparaten die eraan zullen deelnemen. We maakten dit probleem handelbaar door het op te splitsen in vier deelproblemen.

Het eerste deelprobleem van de contextgevoeligheid is het bepalen van de mogelijke communicatiepartners die zich in de huidige context bevinden. We onderzochten in welke mate Bluetooth hier voorzieningen voor bezit, en stelden een alternatief voor; dit alternatief biedt in een aantal specifieke gevallen een antwoordt op de tekortkomingen van de standaardprocedure. We toonden eveneens aan dat er vrij goede criteria beschikbaar zijn, voor het reduceren van het aantal ongewenste communicatiepartners.

Wanneer we beschikken over een minimale lijst met mogelijk communicatiepartners, wensen we na te gaan welke van deze toestellen optreden als Router binnen een gewenst Transparant Netwerk. We maakten gebruik van de dienstgeoriënteerde architectuur van Bluetooth om een dergelijk selectie door te voeren.

Het is mogelijk dat er zich binnen eenzelfde context meerdere overlappende Transparante Netwerken bevinden; de hierdoor geïntroduceerde keuzemogelijkheid geeft aanleiding tot het derde deelprobleem. We hebben een oplossing voorgesteld die in staat bleek om het

aantal keuzemogelijkheden in belangrijke mate te reduceren, steunend op de indicator voor de ontvangen signaalsterkte.

Een laatste deelprobleem betreft het beëindigen van het lidmaatschap van een Transparant Netwerk. Analoog met het deelprobleem in de vorige paragraaf, steunden we op een indicator voor de ontvangen signaalsterkte om tot een oplossing te komen. We gebruiken deze indicator om het verlaten van de context te detecteren, en aldus op contextgevoelige wijze het lidmaatschap van het Transparant Netwerk te beëindigen.

De effectieve transparantie van de netwerkoplossing hangt in grote mate af van de prestaties van elk van de voorgestelde oplossingen voor de genoemde deelproblemen. We mogen bijgevolg niet stellen dat het mogelijk is om met Bluetooth een netwerkoplossing te bouwen die in elke individuele situatie gegarandeerd transparant is. De diverse voorgestelde maatregelen zorgen evenwel dat in de praktijk de interactie met de gebruiker in dergelijke mate gereduceerd wordt, dat de oplossing zeker bruikbaar is een reële situatie zoals geschetst in het genoemde scenario. Een belangrijk aspect van het realiseren van de gewenste prestaties, is het precies configureren van de Transparante Communicatielaag voor de omgeving waarin deze ingezet zal worden.

Tot slot toonden we aan dat het mogelijk is om de Transparante Communicatielaag uit te breiden naar Linux, en op die manier tot een oplossing te komen voor het scenario in 1.2.4 *Communicatie Digibox/IVT*. Deze oplossing werd in het kader van dit eindwerk niet volledig uitgewerkt, maar kan aanleiding geven tot een interessant vervolg op het onderzoek van deze thesis.

# Hoofdstuk 3
# De Click Machine

We bespreken in dit hoofdstuk de problematiek en oplossingsmethode voor scenario's 1.2.2 *Communicatie Digibox/internet* en 1.2.3 *Communicatie Digibox/glucosemeter*. We zullen voor het realiseren van de communicatielinks uit beide scenario's gebruik maken van een compacte Linux computer tussen de Digibox en de kabelmodem, namelijk de *Click Machine*. In beide gevallen zal de Click Machine bepaalde delen van het netwerkverkeer tussen Digibox en kabelmodem onderscheppen, en aldus de gewenste communicatielinks creëren.

We bekijken in 3.1 het creëren van een communicatielink tussen de Digibox en het internet van naderbij; we motiveerden een dergelijke communicatielink eerder in het Digibox/internet scenario. We gaan eerst na welke mogelijkheden de Digibox bezit voor het onderhouden van een netwerkverbinding. Vervolgens proberen we de structuur van het Telenet netwerk, en daarmee de precieze oorzaken van de connectiviteitsproblematiek, te achterhalen; we doen dit aan de hand van een analyse van het netwerkverkeer tussen de Digibox en de kabelmodem. De kennis die we met dit onderzoek opdoen zal ons toelaten om een Click Modular Router script te ontwerpen dat de Digibox toepassing toelaat om het internet te benaderen, terwijl we de goede werking van interactieve televisie ongemoeid laten. Het ontwerp van het Click script zal eveneens communicatie tussen de Digibox toepassing en de software op de Click Machine mogelijk maken.

In het tweede deel van dit hoofdstuk, benutten we de mogelijkheden van het Click script voor communicatie tussen de Digibox toepassing en software op de Click Machine; aan de hand van deze communicatiemogelijkheid zal de Click Machine een webservice aanbieden voor toegang tot randapparaten. Op deze manier proberen we in 3.2 een antwoord te bieden op de problematiek geschetst in het Digibox/glucosemeter scenario. Net zoals dit het geval was in Hoofdstuk 2, zal de Bluetooth communicatietechnologie opnieuw een belangrijke rol spelen; in dit geval zetten we de technologie echter in als kabelvervanger, om zo een uniforme aansluitmogelijkheid te realiseren voor het verbinden van randapparaten met de Click machine.

# 3.1 Communicatie Digibox/internet

In dit onderdeel bespreken we een oplossing voor de problematiek die we introduceerden in het scenario in 1.2.2. We toonden in het genoemde scenario aan dat de zorgtoepassing op de Telenet Digibox nood heeft aan een verbinding met het internet. Aanvankelijk verwachtten we geen bijzondere problemen met deze verbinding; de Digibox is immers voorzien van een standaard 10 Mbps netwerkadapter en wordt – net als een PC – aangesloten op de Telenet kabelmodem. Bovendien maakt Telenet voor zijn interactieve televisietoepassingen gebruik van dit zogenaamd *terugkeerkanaal* (Eng. *return channel*) voor het verwerken van de invoer van de gebruiker. In de praktijk is echter gebleken dat het terugkanaal uitsluitend toegang biedt tot de Telenet servers ter ondersteuning van interactieve televisieactiviteiten.

We onderzoeken in 3.1.1 welke de precieze oorzaken zijn van de connectiviteitsproblematiek van de Digibox. Vervolgens gaan we in 3.1.2 na op welke manier we de gestelde problemen kunnen oplossen aan de hand van de Click Machine; we zullen de Click Machine hiertoe voorzien van de Click Modular Router software, en deze configureren met een Click script.

## 3.1.1 Problematiek en conceptuele oplossing

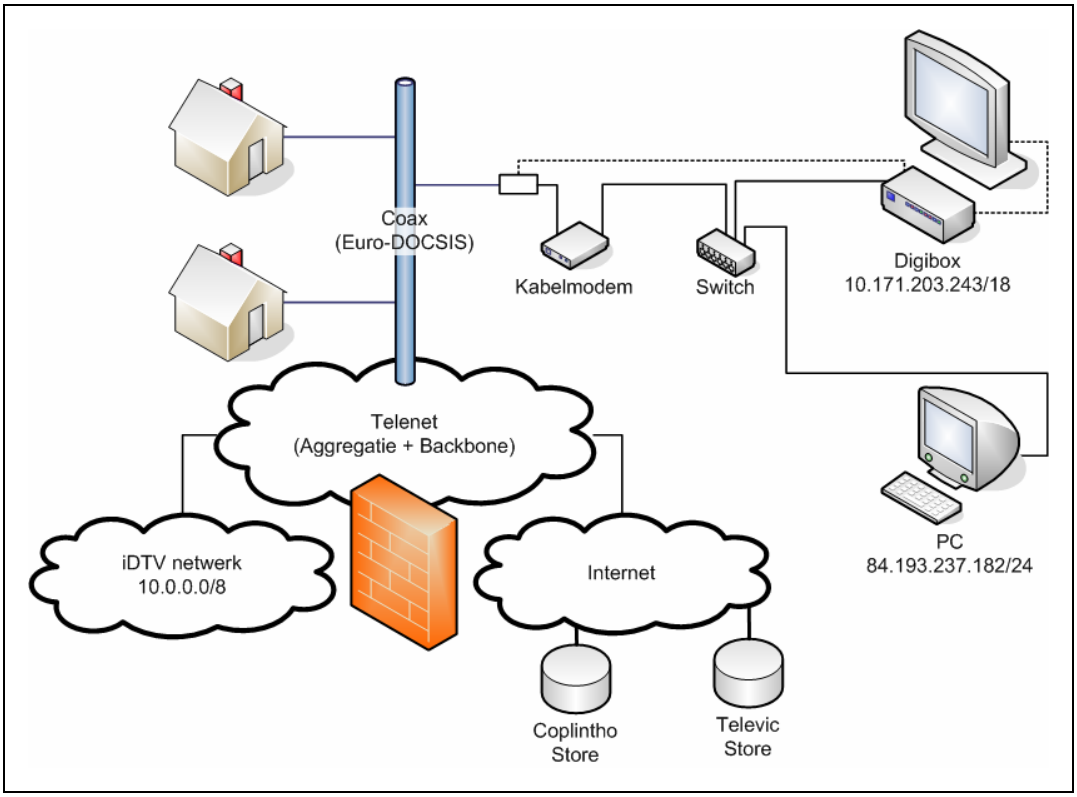
We onderzoeken in deze sectie de connectiviteitsproblematiek van naderbij. We bekijken daartoe kort de structuur van het Telenet netwerk voor interactieve televisie en internettoegang. Vervolgens analyseren we het gegevensverkeer tussen de Digibox en de kabelmodem. Tot slot stellen we een conceptuele oplossing voor.

Voor meer informatie over de netwerkprotocollen die aan bod komen tijdens deze bespreking verwijzen we naar [28].

### Het Telenet netwerk

In Figuur 25 wordt een overzicht gegeven van het Telenet netwerk. Elk aangesloten huishouden beschikt over een verbinding met het coax netwerk; de bandbreedte van dit netwerk wordt verdeeld tussen het klassieke TV-signaal en gegevenstrafiek voor interactieve digitale televisie en internettoegang via Euro-DOCSIS[29]. Rechtsboven in de figuur wordt een typische thuisopstelling getoond; het betreffende gezin beschikt over een

computer voor toegang tot het internet en een Digibox voor digitale televisie. Beide toestellen zijn via een switch verbonden met de kabelmodem; deze laatste zorgt dat er communicatie mogelijk is met het Telenet backbone via een verbinding met het Euro-DOCSIS kabelnetwerk en het aggregatienetwerk.




Figuur 25: Overzicht Telenet netwerk

Wanneer we de Digibox en de computer via de switch aansluiten op de kabelmodem, wordt er via DHCP een netwerkadres toegekend aan beide toestellen. In het voorbeeld in de figuur verwerven de Digibox en de computer respectievelijk de adressen 10.171.203.243/18 en 84.193.237.182/24. Het valt onmiddellijk op dat beide adressen behoren tot verschillende subnetwerken. Bovendien behoort 10.171.203.243 tot het subnet met bereik 10.171.192.0-10.171.207.255; dit bereik is een onderdeel van het gereserveerde *private subnet* 10.0.0.0/8. Er zijn in totaal drie private adresreeksen gereserveerd door de *Internet Assigned Numbers Authority (IANA)*[30]. De bereiken van de private subnetten zijn bedoeld voor het creëren van private netwerken; op deze manier moet men niet voor elke entiteit binnen dat netwerk een wereldwijd uniek IP-adres reserveren. Aangezien een privaat netwerkadres niet gegarandeerd wereldwijd uniek is, is het evenwel niet mogelijk om zich met behulp van een dergelijk adres op het publieke internet te begeven. Bedrijven die zich bedienen van private netwerkadressen voor het realiseren van hun intranet, gebruiken meestal een netwerkadresomzetter (Eng. *Network Address Translation*, NAT) om het internet toch beschikbaar te maken vanuit dat private netwerk[28]. Telenet heeft zijn interactieve televisieactiviteiten ondergebracht in een dergelijk privaat netwerk, het iDTV-netwerk; er

zijn in dit geval echter geen voorzieningen voor toegang tot het publieke internet via netwerkadresomzetting.

# Verkeersanalyse van het terugkeerkanaal

Om wat meer zicht te krijgen op het karakter van het gegevensverkeer verbonden met digitale televisie, hebben we de trafiek die passeert via het terugkanaal van de Digibox vastgelegd en geanalyseerd.

We geven hieronder een overzicht van het verkeer dat we observeerden tussen de Digibox en het interactieve televisienetwerk van Telenet; we zullen dit doen aan de hand van enkele fragmenten, die we vastlegden tijdens het bestellen van een aflevering van *De Parelvissers*. Voor het vastleggen van de waarnemingen hebben we gebruik gemaakt van Ethereal[31].

We starten met een reset van de Digibox en schakelen deze vervolgens in; het toestel lanceert vrijwel onmiddellijk een aanvraag voor het verwerven van een netwerkadres met DHCP. Het toegekende adres is 10.171.203.243/18. De Digibox verneemt eveneens het adres van zijn gateway; dit is 10.171.192.1/18. Het geobserveerde fragment is een standaard DHCP interactie; we zullen dit fragment dan ook niet tonen. Merk op dat de Digibox zich in een subnet van het iDTV-netwerk bevindt. Uit het DHCP verkeer dat we observeerden bij het aansluiten van een Digibox en een computer, kunnen we afleiden dat de Telenet DHCP-servers adressen toekennen aan de hand van het MAC-adres van de aanvrager. Men beschikt blijkbaar bij Telenet over een databank met de MAC-adressen van alle verkochte Digiboxen. Wanneer een netwerkapparaat een IP-adres aanvraagt via DHCP, gaat men na of het MAC-adres hoort bij een Telenet Digibox; het toestel krijgt vervolgens al dan niet een privaat netwerkadres. Wanneer we het MAC-adres van de netwerkadapter van een computer wijzigen in dat van de Digibox, verkrijgt deze computer eveneens een netwerkadres uit het private bereik; bovendien is het toegekend netwerkadres identiek aan dat van de betreffende Digibox.

| **Aanvraag door** | **Antwoord door** | **Protocol** |
|---|---|---|
| Digibox [**10.171.203.243**] | **10.16.2.6** | TCP/HTTP |

```
GET /vod/voditem_ZZZZ5000000000112114_nld.txt HTTP/1.1
[…]

HTTP/1.1 200 OK
[…]
De parelvissers 05-03
vrt_parelvissers_05_03.ts
[…]
Zesdelige fictiereeks over de op- en ondergang van het televisieproductiehuis 'De
Parelvissers', de op- en ondergang van de vriendschap van dertigers en de zoektocht naar
de ware toedracht over de verdwijning van een van hen: Jan Deridder.
[…]
```

Figuur 26: Fragment – opvragen van informatie over een aflevering

Nadat de Digibox opgestart is, bekijken we in het menu voor video op aanvraag welke items we kunnen bestellen. Het opvragen van meer informatie over de nieuwste aflevering van *De Parelvissers*, zorgt voor de HTTP-trafiek in het fragment in Figuur 26.

Wanneer we beslissen om daadwerkelijk over te gaan tot het bestellen van de aflevering, vraagt de Digibox informatie op bij de server over het *huishouden* dat bij het kabelabonnement hoort. Bij het bekijken van de verstrekte gegevens in het fragment in Figuur 27, stellen we ons ernstige vragen bij het versturen als klare tekst van dergelijke gevoelige informatie.

| **Aanvraag door** | **Antwoord door** | **Protocol** |
|---|---|---|
| Digibox [**10.171.203.243**] | **10.16.2.5** | TCP/HTTP |

```
POST /IDTV/FrontController HTTP/1.0
[…]
applicationService=ACCOUNT&methodName=retrieveHousehold&[…]

HTTP/1.1 200 OK
[…]
HouseholdName=IBBT | expenditureLimit=3 | easyRecording=false | HouseholdStreet=Gaston
Crommenlaan | HouseholdHouseNumber=8 | HouseholdAreaCode=9050 | HouseholdTown=Ledeberg
(Gent) | […] | dayTotalExpenditures=0.00 | monthTotalExpenditures=0.00
```

Figuur 27: Fragment – uitwisseling van gebruikersinformatie

Zoals getoond in Figuur 28, gebeurt het starten en de verdere controle van de videostroom met behulp van het *Real Time Streaming Protocol (RTSP)*[32]. Merk op dat de verdeling en de controle van de videostroom blijkbaar regionaal gebeurt; in ons geval is dit bijvoorbeeld de regio Gent.

| **Aanvraag door** | **Antwoord door** | **Protocol** |
|---|---|---|
| Digibox [**10.171.203.243**] | **10.16.70.4** | TCP/RTSP |

```
SETUP rtsp://10.16.70.4/?[…]&NodeGroupId=VD58GENT112 RTSP/1.0
[…]
Transport:  MP2T/DVBC/QAM;unicast
[…]

RTSP/1.0 200 OK
[…]

PLAY rtsp://10.16.70.4/?[…]&NodeGroupId=VD58GENT112 RTSP/1.0
[…]

RTSP/1.0 200 OK […]
```

Figuur 28: Fragment – effectief bestellen en starten van de aflevering

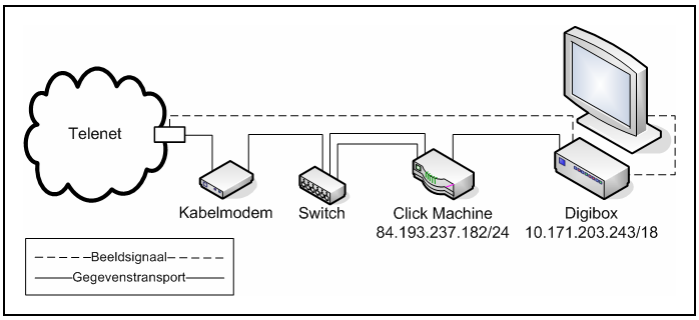
We merken tot slot op dat de Digibox in de praktijk uitsluitend communiceert met netwerkentiteiten binnen het private iDTV-netwerk. Deze netwerkentiteiten bevinden zich echter niet in hetzelfde subnet als de Digibox; de Digibox maakt bijgevolg uitgebreid gebruik van de diensten van de gateway van zijn subnet.

We besluiten uit de verkeersanalyse dat de Digibox uitsluitend gebruik maakt van standaard internetprotocollen; meer specifiek gaat het over IP- en ARP-verkeer. Bovendien is de trafiek gebonden aan het gebruik van interactieve televisie vrij eenvoudig te herkennen aan de hand van de bron- en bestemmingsadressen.

# Conceptuele oplossing

Uit een analyse van de netwerkstructuur en het gegevensverkeer, besluiten we dat de connectiviteitsproblematiek veroorzaakt wordt door het gebruik van het private iDTV-netwerk. Als gevolg van het uitreiken van netwerkadressen op basis van het MAC-adres, behoort de Digibox eveneens tot dit private subnet. Aangezien het private netwerk niet over een netwerkadresomzetter beschikt, is het internet niet bereikbaar voor de Digibox.

We hebben reeds eerder vermeld dat men in het geval van een privaat netwerk vaak netwerkadresomzetting inzet om toch toegang te krijgen tot het publieke internet. Bovendien ervaart men in de gegeven opstelling geen enkel probleem wanneer men zich met een computer toegang wenst te verschaffen tot het internet. We verwachten dan ook dat we het internet bereikbaar kunnen maken van op de Digibox door de aanvragen naar publieke servers te onderscheppen, en voor deze aanvragen netwerkadresvertaling toe te passen. We moeten hierbij uiteraard het verkeer voor het interactieve televisienetwerk ongemoeid laten; we zagen eerder dat dit verkeer steunt op standaardprotocollen, en bovendien vrij eenvoudig te herkennen is aan de hand van de bron- en bestemmingsadressen. Om de aanvragen naar publieke servers te onderscheppen plaatsen we een toestel – de Click Machine – tussen de Digibox en de switch. De nieuwe opstelling wordt weergegeven in Figuur 29.

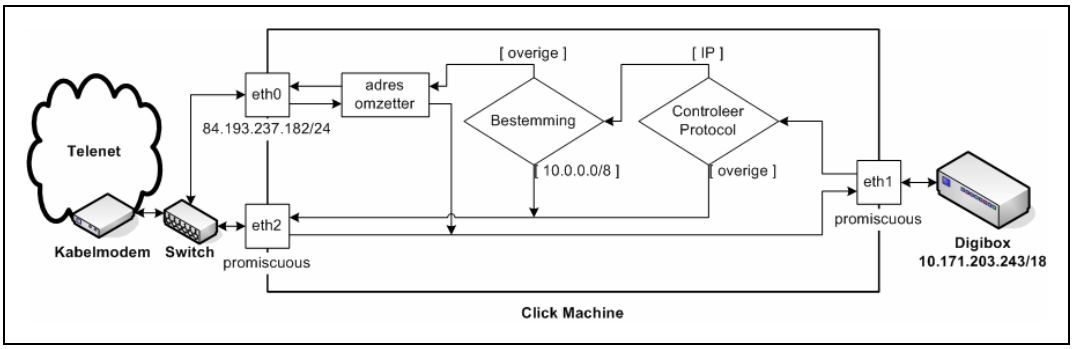



Figuur 29: Opstelling met de Click Machine

Zoals weergegeven beschikt de Click Machine over drie netwerkadapters; één van deze adapters wordt verbonden met de Digibox, terwijl we de overige verbinden met de switch. Aangezien de Click Machine geen Digibox is, kan deze met behulp van DHCP een netwerkadres verwerven dat toelaat om het publieke internet te benaderen. In het weergegeven voorbeeld krijgt de Click Machine het IP-adres toegekend dat eerder gebruikt werd door de computer. Het is de bedoeling om al het verkeer dat verband houdt met interactieve televisie ongemoeid te laten. Indien de Digibox echter een aanvraag naar het publieke internet verstuurt, zal de Click Machine deze onderscheppen en adresomzetting toepassen; op deze manier lijkt het van buitenaf alsof de aanvraag afkomstig was van een computer, en kan deze beantwoord worden door de publieke internet servers. Het

corresponderende antwoord zal uiteraard eveneens onderschept dienen te worden door de Click Machine; in dit geval wordt een omgekeerde adresvertaling toegepast.

In Figuur 30 geven we de adresomzetting in de Click Machine in meer detail weer. We zullen het weergegeven schema hieronder kort bespreken.




Figuur 30: Adresomzetting Click Machine

We bekijken eerst de pakketten afkomstig van de Digibox. Deze komen binnen in de Click Machine via de eth1 netwerkinterface. De pakketten afkomstig van de Digibox zijn in de praktijk altijd bestemd voor het MAC-adres dat hoort bij de gateway van het subnet van het private netwerk, waarbinnen de Digibox zich bevindt. Opdat de Click Machine deze pakketten zou ontvangen, dient de eth1 netwerkinterface in *promiscuous* mode te werken; in deze modus ontvangt de netwerkadapter eveneens pakketten die niet voor zijn eigen MAC-adres bestemd zijn. Wanneer een pakket binnenkomt bekijken we eerst het bijhorende protocol. Indien het om een IP-pakket gaat, wensen we het verder te behandelen. Is dit niet het geval, dan sturen we het pakket ongewijzigd via de eth2-interface naar de kabelmodem. Voor een IP-pakket gaan we na welke de bestemming is; indien het pakket bestemd is voor een server binnen het iDTV-netwerk, laten we het opnieuw ongemoeid en sturen we het via de eth2-interface naar de kabelmodem. Indien het pakket echter voor het publieke internet bestemd is, vertalen we het oorspronkelijk bronadres 10.171.203.243/18 naar 84.193.237.182/24; dit laatste adres is het publieke adres van de eth0-interface van de Click Machine. Op die manier kan het pakket probleemloos de server in het internet bereiken.

De pakketten die terugkomen via de kabelmodem, kunnen gericht zijn aan de Click Machine of aan de Digibox. De eth2 promiscuous interface zorgt dat de Digibox elk pakket ontvang dat binnenkomt via de kabelmodem; op die manier emuleren we een rechtstreekse connectie tussen de Digibox en de kabelmodem. De eth0-interface ontvangt uitsluitend pakketten die gericht zijn aan het bijhorende MAC-adres; in de praktijk zijn dit de antwoorden op de internetaanvragen van de Digibox die we eerder vertaald en doorgestuurd hebben. We sturen deze pakketten dan ook in de omgekeerde richting door de adresomzetter zodat het oorspronkelijke bestemmingsadres 84.193.237.182/24 vertaald

wordt in 10.171.203.243/18. Tot slot bezorgen we het antwoordpakket via eth2 aan de Digibox.

### 3.1.2 Implementatie

We bespreken in deze sectie kort de hardware en software voor de Click Machine, en gaan vervolgens dieper in op de implementatie van de routeringsfunctionaliteit van het toestel aan de hand van een Click script.

#### Hardware en software

Zoals weergegeven in Figuur 29, zullen we de connectiviteitsproblematiek oplossen door de Click Machine tussen de Digibox en de kabelmodem te plaatsen; dit toestel zal instaan voor het onderscheppen en aanpassen van bepaalde pakketten. De Click Machine is een standaard PC platform voorzien van voldoende netwerkadapters. Hoewel er in de conceptuele oplossing in Figuur 30 drie netwerkadapters getoond worden, volstaat men in de praktijk met slechts twee adapters. We kunnen immers de twee adapters die we in de figuur via een switch met de kabelmodem verbinden, vervangen door slechts één adapter die rechtstreeks verbinding maakt met de modem. De Click machine is verder voorzien van het Debian GNU/Linux[33] besturingssysteem en de Click Modular Router[34] software.

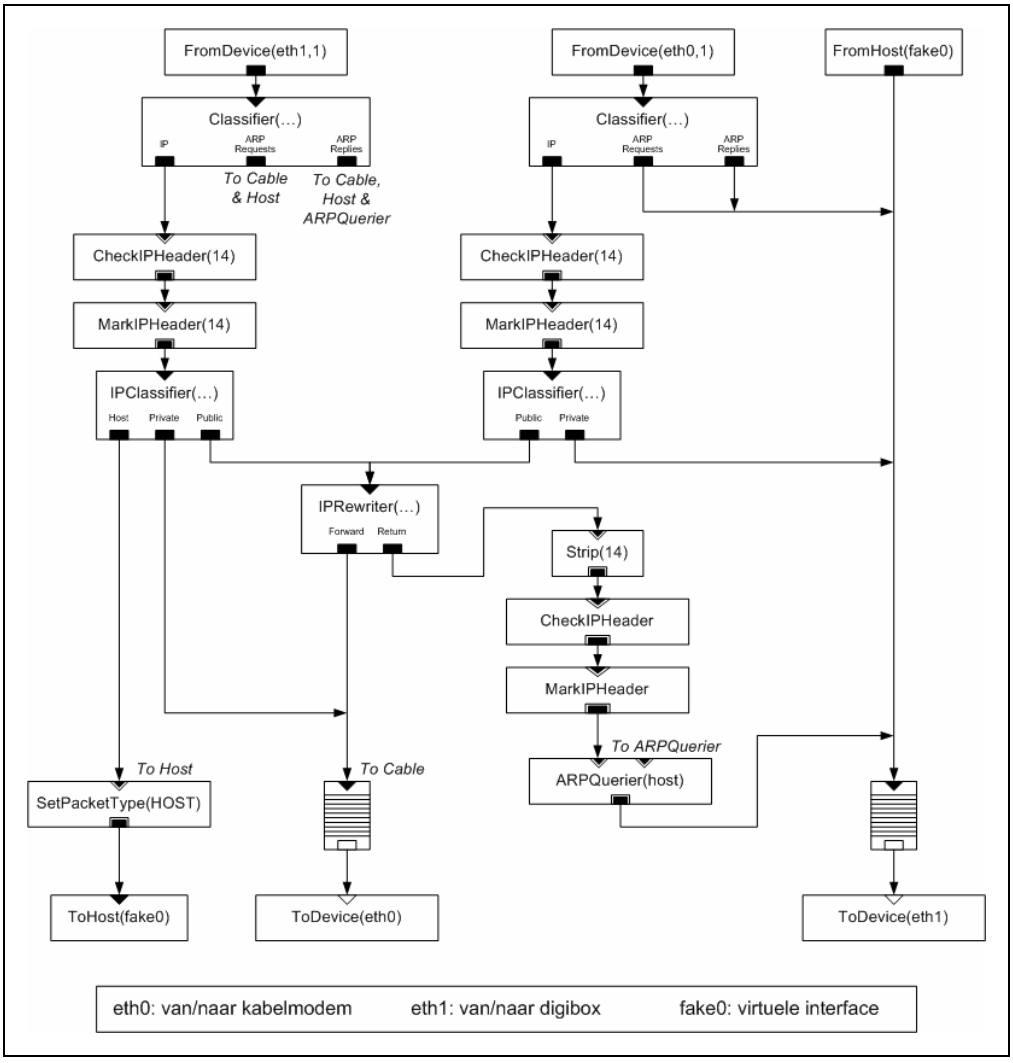
De Click Modular Router is een softwarearchitectuur voor het bouwen van flexibele en configureerbare routers. Deze architectuur laat toe om een complexe routeringsfunctionaliteit op te bouwen aan de hand van eenvoudige elementen. Een element is een kleine functionele eenheid met een goed gedefinieerde en meestal vrij eenvoudige routeringsfunctie. Er zijn bijvoorbeeld elementen beschikbaar de voor classificatie van pakketten of het implementeren van een wachtrij. Een Click router configuratie is een gerichte graaf die bestaat uit dergelijke elementen en de doorstromingsrelaties daartussen. Voor meer informatie over de Click Modular Router software verwijzen we naar [34]. Een goede inleiding tot Click bevindt zich in [35].

#### Het Click script

We voegen het Click script voor de implementatie van de routeringsfunctionaliteit bij in Appendix A, en geven het hier schematisch weer in Figuur 31; we gebruiken hiervoor de grafische voorstelling uit [35].

Alvorens de werking van het Click script te verduidelijken aan de hand van enkele voorbeelden, formuleren we eerst een aantal veronderstellingen die van belang zijn voor de goede werking van het script.

Ten eerste halen we aan dat de Digibox aangesloten is op netwerkadapter eth1, terwijl de kabelmodem verbonden is met eth0; dit is in overeenstemming met wat weergegeven wordt in Figuur 30. Aangezien de eth2 interface uit Figuur 30 in de promiscuous modus werkt, en bovendien zelf geen netwerkadres bezit, kunnen we deze interface weglaten en alle verkeer van en naar de kabelmodem via de eth0-interface behandelen. Bij gebruikt van het Click script in Figuur 31 is het dus de bedoeling dat eth0 een netwerkadres bezit en tegelijk de promiscuous modus aanneemt. De eth0-interface verkrijgt dit netwerkadres via DHCP; we veronderstellen dat dit gebeurt is vóór het installeren van het Click script.




Figuur 31: Schematische voorstelling Click script

Merk op dat er zich in het schema een extra virtuele netwerkadapter bevindt, namelijk fake0. Door gebruik te maken van deze virtuele netwerkadapter is de Linux protocol stack bereikbaar van op de Digibox; op die manier kan de software op de Digibox rechtstreeks communiceren met software op de Click Machine. De fake0-interface bezit het vaste netwerkadres 10.0.0.1/8. Bij het uitvoeren van een *ping*-test, kwam er geen antwoord van een eventuele host op dit adres. Bovendien stelden we tijdens de verkeersanalyse geen communicatie met een host op dit adres vast. De mogelijkheid om van op de Digibox de

Click Machine te adresseren met een adres uit het private adresbereik levert een aantal voordelen op die we later zullen behandelen. Het MAC-adres van de virtuele interface is 00:01:02:03:04:05. De mogelijkheid om te communiceren met software op de Click Machine zal nuttig blijken in 3.2, wanneer we de machine zullen inzetten als aansluitpunt voor randapparaten voor gebruik door de Digibox.

## Verkrijgen van een netwerkadres

Na het opstarten, vraagt de Digibox vrijwel onmiddellijk een IP-adres aan met behulp van DHCP. Het DHCP protocol maakt voor het transport van zijn berichten gebruik van broadcast UDP/IP-pakketten op poorten 67 en 68[30].

De Digibox initieert de toekenningsprocedure met het versturen van een DHCP-discoverbericht; de Click machine ontvangt dit bericht via de eth1-interface en stuurt het vervolgens in de Click configuratie, zodat het terecht komt in de Classifier links in Figuur 31. Aangezien het om een IP-pakket gaat, verlaat dit pakket de Classifier via de corresponderende uitgang. Vervolgens wordt de IP-header gecontroleerd en gemarkeerd; deze header bevindt zich op een offset van 14 bytes ten opzichte van het begin van het Ethernet-frame. Eenmaal de IP-header gemarkeerd is, kan de IPClassifier het IP-pakket verder classificeren. Omwille van de waarden van het protocolveld (UDP) en het poortnummer (67 of 68) van het IP-pakket, wordt het onderverdeeld bij de *private* klasse en via de overeenkomstige uitgang naar de kabelmodem gestuurd via eth0. Het DHCP-discoverbericht wordt bijgevolg ongewijzigd doorgestuurd naar het iDTV-netwerk.

Wanneer de DHCP-server het discoverbericht ontvangt, beantwoordt deze de aanvraag met een DHCP-offerbericht. Het antwoordbericht komt de Clickconfiguratie binnen via de eth0-interface en wordt op een gelijkaardige manier behandeld als het aanvraagbericht. De rechter Classifier brengt het pakket onder in de IP-klasse. Vervolgens wordt het verder behandeld door de IPClassifier en op basis van het protocolveld en het poortnummer ongewijzigd doorgestuurd naar de Digibox via de eth1-interface.

## Address Resolution Protocol

Het Address Resolution Protocol (ARP) zorgt voor de koppeling van een IP-adres aan een MAC-adres, en is bijgevolg essentieel voor de werking van een IP-netwerk. De IP-pakketten ter ondersteuning van DHCP waren steeds broadcast-pakketten en behoefden bijgevolg geen koppeling door het ARP. Indien de Digibox echter een server wenst te bereiken aan de hand van een IP-adres, zal deze gebruik maken van het ARP om het MAC-adres van de volgende hop naar de gewenste bestemming te bepalen; daartoe verstuurt de Digibox een ARP-aanvraag voor die bestemming. Deze aanvraag komt binnen in het Click script langs de eth1-interface. De linker Classifier onderschept deze aanvraag en stuurt een kopie naar de kabelmodem en naar de Linux protocol stack. Het versturen van een exemplaar naar de kabelmodem zorgt ervoor dat de Digibox het MAC-adres van de volgende hop naar een

bestemming in het internet of het iDTV-netwerk kan aanvragen; in de praktijk is dit uiteraard het MAC-adres van de gateway. Het verzenden van een exemplaar naar de Linux protocol stack, zorgt ervoor dat de Digibox een antwoord krijgt wanneer deze een ARP-aanvraag verstuurt voor het virtuele adres 10.0.0.1/8. Aangezien we deze pakketten toch zullen onderscheppen op IP-adres, is de inhoud van het antwoord niet zo belangrijk; de Digibox moet echter *een* antwoord krijgen alvorens er een pakket verstuurt kan worden. De protocol stack zorgt ervoor dat een dergelijke antwoord – namelijk 00:01:02:03:04:05 – verstuurd wordt naar de Digibox.

De ARP-antwoorden op de aanvragen vanwege de Digibox komen de Click-configuratie binnen via de eth0 interface. De Classifier selecteert deze antwoorden en verstuurt ze ongewijzigd naar de Digibox via eth1. Hetzelfde geldt voor de ARP-aanvragen die via de eth0-interface aankomen en gericht zijn naar de Digibox.

Wanneer de Digibox een ARP-aanvraag beantwoordt, komt dit antwoord binnen via de eth1-interface. De Classifier selecteert deze aanvragen en stuurt kopieën naar de kabelmodem, de Linux protocol stack en de ARPQuerier onderaan in Figuur 31. Het nut van de eerste twee kopieën is onmiddellijk duidelijk. We komen bij de bespreking van de communicatie met het internet terug op de ARPQuerier.

## Communicatie met het iDTV-netwerk

Uit de analyse van het verkeer op het terugkeerkanaal van de Digibox, is gebleken dat voor de communicatie met servers in het iDTV-netwerk uitsluitend gebruik gemaakt wordt van TCP/IP en UDP/IP; dit verkeer bestaat bijvoorbeeld uit HTTP- en DNS-berichten De pakketten worden op vrijwel dezelfde manier behandeld als die voor het transport van de DHCP-berichten, uitgezonderd voor wat betreft de classificatie in de IPClassifiers. De pakketten worden in dit geval ondergebracht in de private klasse omwille van hun bestemmingsadres binnen het private bereik 10.0.0.0/8. De pakketten zetten op die manier hun weg ongewijzigd verder.

## Communicatie met het internet

Net zoals dit het geval was voor de communicatie met het iDTV-netwerk, zal de communicatie met het internet voornamelijk bestaan uit TCP/IP- en UDP/IP-pakketten. In tegenstelling tot het iDTV-netwerkverkeer, wensen we in dit geval een adresomzetting uit te voeren; op die manier kunnen de pakketten van de Digibox het internet bereiken, en omgekeerd.

Wanneer de Digibox een IP-pakket verstuurt naar een publieke server, komt dit pakket zoals steeds de Click-configuratie binnen via de eth1-interface. De Classifier zorgt voor het duursturen van dit pakket naar de IPClassifier links in Figuur 31; deze brengt het pakket onder in de *public* klasse, aangezien het noch bestemd is voor de Click Machine op 10.0.0.1/8, noch voor het iDTV-netwerkbereik 10.0.0.0/8. De IPClassifier stuurt het IP-

pakket vervolgens naar de IPRewriter. De IPRewriter vervangt het bronadres van het pakket door het publieke adres dat eth0 eerder verkreeg, en stuurt het pakket vervolgens verder naar de kabelmodem via eth0. Telkens wanneer de IPRewriter een vertaling uitvoert voor een *(bron_oorspronkelijk, bron_nieuw)*-paar, maakt dit element een overeenkomstig record aan voor het uitvoeren van de omgekeerde vertaling wanneer er een antwoord komt met als bestemming *bron_nieuw*. Merk op dat het Ethernet-frame nog steeds geadresseerd is aan het MAC-adres van de gateway van de Digibox; wanneer deze gateway het pakket ontvangt, stuurt deze het verder naar het internet. Het is opmerkelijk dat de gateway van de Digibox blijkbaar toch zelf een connectie met het internet bezit.

De antwoordpakketten van een publieke server zijn gericht aan het netwerkadres van eth0, en komen bijgevolg via deze interface de Click-configuratie binnen. De rechter IPClassifier in Figuur 31 brengt een dergelijk pakket onder in de public klasse aan de hand van het bestemmingsadres; dit ligt nu immers niet binnen het private bereik 10.0.0.0/8. Het gevolg van deze classificatie is het doorsturen van dit pakket naar de IPRewriter. De IPRewriter maakte bij het uitvoeren van de omzetting van de aanvraag een (*bron_oorspronkelijk*, *bron_nieuw*)-record aan, en voert op basis daarvan een omgekeerde adresomzetting uit op het bestemmingsadres; concreet wordt het bestemmingsadres van het IP-pakket hierdoor gewijzigd in het netwerkadres van de Digibox. De pakketten waarop een omgekeerde adresomzetting uitgevoerd werd, verlaten de IPRewriter via de twee de uitgang van dit element. Als gevolg van de adresomzetting is het bestemmingsadres van het Ethernetframe niet langer consistent met het bestemmingsadres van het IP-pakket; we verwijderen dan ook de Ethernet-header aan de hand van het Strip-element en sturen het IP-pakket vervolgens verder naar de ARPQuerier. De ARPQuerier bekomt het MAC-adres van de volgende hop naar het IP-bestemmingsadres door het versturen van een ARP-aanvraag; in de praktijk is dit bestemmingsadres het netwerkadres van de Digibox, en bekomt men bijgevolg het MAC-adres van dit toestel. Eenmaal het MAC-adres voor handen is, kan de ARPQuerier het Ethernet-frame opbouwen en het IP-pakket doorsturen naar de Digibox.

## Communicatie met de Click Machine

Zoals reeds eerder vermeld, kan een toepassing op de Digibox communiceren met software op de Click Machine. De toepassing op de Digibox gebruikt daartoe het vaste netwerkadres 10.0.0.1/8 voor het adresseren van de Click Machine. Merk op dat de communicatie steeds geïnitieerd dient te worden door de Digibox, aangezien deze zelf geen vast IP-adres bezit.

Wanneer de Digibox een IP-pakket naar de Click Machine stuurt, komt dit pakket de Click configuratie binnen via de eth1-interface. Het pakket wordt vervolgens omwille van het geassocieerde protocol door de Classifier naar de IPClassifier gestuurd, waar het ondergebracht wordt in de *host* klasse op basis van het bestemmingsadres 10.0.0.1/8.

Vervolgens wordt het pakket geannoteerd met het pakkettype *HOST*, en afgeleverd aan de Linux protocol stack.

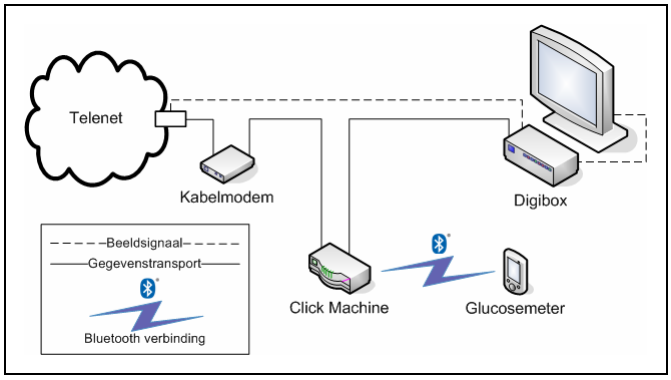
Het belang van de keuze van het netwerkadres 10.0.0.1/8 voor het adresseren van de virtuele interface van de Click Machine, wordt duidelijk wanneer de software op de machine een antwoord wenst te versturen naar de Digibox. Als gevolg van de adreskeuze zijn we namelijk zeker dat het netwerkadres van de Digibox zich binnen hetzelfde subnet bevindt als dat van de virtuele interface; bijgevolg tracht Linux nooit beroep te doen op een gateway om de Digibox te bereiken, en worden de antwoorden probleemloos verstuurd. Indien we een adres in een ander bereik zouden kiezen, zou Linux problemen ondervinden bij het bepalen van een route naar de Digibox; er is in dat geval immers een gateway nodig om deze te bereiken.

## 3.2 Communicatie Digibox/Randapparaat

We bespreken in dit onderdeel de problematiek van het aansluiten van een randapparaat aan de Digibox. We motiveerden de nood aan een dergelijke verbindingsmogelijkheid reeds eerder in het scenario in 1.2.3 *Communicatie Digibox/glucosemeter*. We toonden in het genoemde scenario aan dat het communiceren met bijvoorbeeld een glucosemeter, heel wat mogelijkheden biedt voor het realiseren van een efficiënte en gebruiksvriendelijke zorgtoepassing. We zullen het voorbeeld van de glucosemeter aangrijpen om de problematiek van naderbij te analyseren, en vervolgens een concrete oplossing te realiseren.

### 3.2.1 Problematiek en conceptuele oplossing

De Telenet Digibox is een vrij eenvoudig toestel dat vooral ontwikkeld werd met het oog op het aanbieden van interactieve digitale televisie. De Digibox bezit dan ook standaard geen mogelijkheden voor het aansluiten van randapparaten.



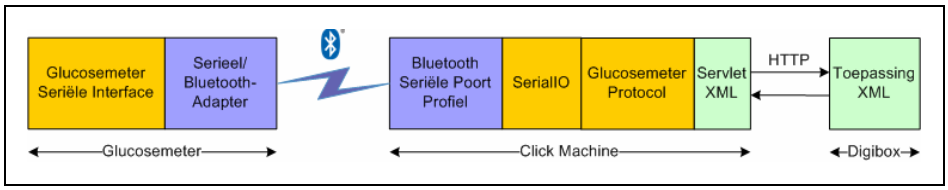

Figuur 32: Opstelling met de Click Machine en de glucosemeter

We zagen eerder in 3.1 dat de Digibox beschikt over een netwerkinterface die momenteel uitsluitend gebruikt wordt voor interactieve televisietoepassingen. We introduceerden in hetzelfde onderdeel de Click Machine, en toonden aan dat het mogelijk is om een dergelijk toestel tussen de Digibox en de kabelmodem te plaatsen, zonder de interactieve televisiefunctionaliteit te schaden. Bovendien bleek het mogelijk om vanuit de Digibox te communiceren met software op de Click Machine. We stellen dan ook voor om de randapparaten aan te sluiten op de Click Machine, en deze beschikbaar te maken met behulp van een webservice via het terugkeerkanaal van de Digibox. Een dergelijke

opstelling wordt getoond in Figuur 32. We bespreken de concrete implementatie van de conceptuele oplossing in de volgende sectie.

# 3.2.2 Implementatie

We bespreken in deze sectie de implementatie van de conceptuele oplossing uit 3.2.1. We splitsen het connectiviteitsprobleem hiertoe op in twee deelproblemen, namelijk het realiseren van een communicatielink tussen de Click Machine en een randapparaat, en het aanbieden van de diensten van dit apparaat aan de Digibox via een webservice. We bekijken eerst de communicatielink tussen de Click Machine en het randapparaat van naderbij. Vervolgens stellen we een implementatie van een apparaatspecifieke webservice voor, die de apparaatspecifieke details afschermt van de Digibox. Tot slot geven we aan hoe we de apparaatspecifieke webservice kunnen uitbreiden naar een algemene webservice, waarbij de apparaatspecifieke details afgehandeld worden door de Digibox. We geven alvast een overzicht van de nodige software en communicatielinks in Figuur 33; we bespreken zo meteen de elementen uit de figuur en de manier waarop deze samenwerken.




Figuur 33: Overzicht communicatie Digibox/glucosemeter

# Communicatie Click Machine/glucosemeter

Voor de communicatie tussen de Click Machine en de glucosemeter, zullen we gebruik maken van Bluetooth. We bespraken Bluetooth reeds eerder uitgebreid in Hoofdstuk 2. Hoewel medische randapparaten met een Bluetooth interface momenteel nog vrij schaars zijn, ziet het er naar uit dat deze in de toekomst in groten getale op de markt zullen komen[4].

Televic stelde voor dit eindwerk een *Lifescan OneTouch Ultra*[36] glucosemeter ter beschikking. Deze glucosemeter beschikt standaard uitsluitend over een seriële RS-232 interface voor interactie met computertoepassingen. Zoals weergegeven in Figuur 33, werd de seriële interface van de glucosemeter verbonden met een *BlueSerial Bluetooth/Serieel-adapter*[37]; deze biedt de seriële interface met de meter via het Bluetooth seriële poort profiel aan als een dienst. Bluetooth apparaten die eveneens het seriële poort profiel ondersteunen, kunnen via de adapter verbinding maken met de glucosemeter alsof deze met een seriële kabel aangesloten was op een standaard seriële poort.

BlueZ is de standaard Bluetooth protocol stack implementatie voor Linux; de Click Machine maakt dan ook gebruik van BlueZ voor de communicatie met de Bluetooth/Serieel-adapter. We gebruikten BlueZ reeds eerder in 2.3.5, en vermeldden toen dat er momenteel geen BlueZ klassenbibliotheek beschikbaar is voor Java; om die reden hebben we gebruik gemaakt van een manuele configuratie van de link tussen de Click Machine en de Bluetooth/Serieel-adapter. De manuele configuratie bestaat uit het toevoegen van de configuratieparameters uit Figuur 34 aan het bestand */etc/bluetooth/rfcomm.conf*, en het toevoegen van *rfcomm bind rfcomm0* aan het opstartscript van de Click Machine.

```
rfcomm0 {
  /* Apparaatadres van de BlueSerial adapter */
  device 00:0B:91:FF:F1:EC;

  /* Service Channel Number (SCN) van de seriële poort dienst */
  channel 1;
}
```

Figuur 34: OneTouchUltra configuratieparameters

In de praktijk is het uiteraard de bedoeling dat de gebruiker na de aankoop van een glucosemeter, de configuratieparameters eenmalig invoert in de zorgtoepassing op de Digibox. De zorgtoepassing dient de parameters vervolgens naar de Click Machine te sturen, waarna deze zichzelf kan configureren voor het gebruik van de meter. Na de configuratie van BlueZ voor het gebruik van de BlueSerial adapter, is het mogelijk om de glucosemeter te benaderen alsof deze met een seriële kabel aangesloten was op de virtuele seriële poort */dev/rfcomm0*.

Zoals reeds vermeld in 2.3.5, bezit de Java Virtuele Machine standaard geen mogelijkheden voor het benaderen van seriële poorten. We hebben hiervoor de *SerialIO SerialPort API*[38] gebruikt omdat deze – net als de rest van de implementatie van de webservice – platformonafhankelijk is. De SerialPort-interface is echter een stuk minder elegant dan die van de Sun Comm API die we eerder gebruikt hebben in 2.3.5 en is bovendien niet gratis. Door de SerialPort API te gebruiken, kunnen we echter garanderen dat de webservice even goed gebruikt kan worden onder Linux als onder Windows XP.

Voor de communicatie met de OneTouch Ultra glucosemeter, dient men gebruik te maken van een eenvoudig serieel protocol. In het kader van dit eindwerk werd hiervoor een Java klassenbibliotheek geïmplementeerd; deze zorgt voor het afschermen van de details van het seriële communicatieprotocol, en laat aan toepassingssoftware toe om op een eenvoudige manier gebruik te maken van de glucosemeter. We zullen deze klassenbibliotheek niet uitgebreid bespreken maar verwijzen in plaats naar [39] voor details over het protocol, en naar de uitgebreide documentatie in de broncode voor meer informatie omtrent de implementatie.

# De webservice

Bij de implementatie van de webservice kan men opteren voor een apparaatspecifieke of een algemene benadering. In het geval van de apparaatspecifieke benadering, handelt de webservice alle apparaatspecifieke aspecten van de communicatie met het randapparaat af; de toepassing op de Digibox hoeft bijgevolg geen apparaatspecifieke klassen te bezitten om te communiceren met het randapparaat. De bedoeling van een algemene webservice is het aanbieden van een communicatiepoort van de Click Machine aan de toepassing op de Digibox; in dit geval dient deze toepassing de communicatie met het randapparaat zelf volledig te sturen, en heeft deze dus kennis nodig over apparaatspecifieke details.

We bespreken hieronder de implementatie van een apparaatspecifieke webservice voor de Lifescan OneTouch Ultra glucosemeter, en geven vervolgens aan op welke manier men deze kan uitbreiden naar een algemene webservice.

## Een webservice framework

Een eerste stap bij de realisatie van een webservice, is het kiezen van het framework waarbinnen deze service uitgevoerd zal worden. Voor de realisatie van de apparaatspecifieke webservice hebben we geopteerd voor *Apache Tomcat*[40]; dit is een compacte platformonafhankelijke servlet container met een omvang van slechts 15 MB. Hier bovenop komt echter wel de installatie van de *J2SE Runtime Environment*[41], die ongeveer 60 MB schijfruimte vereist. Het zal later mogelijk zijn om Tomcat uit te breiden voor het realiseren van de apparaatonafhankelijke webservice.

Voor de implementatie van de apparaatspecifieke webservice, zullen we gebruik maken van servlets. Een servlet is een Java klasse die voldoet aan de specificaties van de Java Servlet Technology[42]. In de praktijk is een servlet te vergelijken met een Java applet die binnen een container aan de serverzijde draait. De servlet container staat ondermeer in voor het onderhouden van de netwerkverbindingen met de clients, de beveiligingspolitiek en de realisatie van een correcte meerdradige uitvoering van de aanwezige servlets.

## Servletimplementatie

Er zijn verschillende basisklassen beschikbaar voor het implementeren van een servlet. De basisklasse bepaalt het protocol waarop de servlet zal steunen voor het ontvangen en beantwoorden van aanvragen. Voor de realisatie van de apparaatspecifieke servlet voor het aanbieden van de diensten van de OneTouch Ultra glucosemeter, opteerden we voor de HTTPServlet basisklasse; zoals de naam van de klasse laat vermoeden, zal de servlet steunen op het HTTP-protocol voor de communicatie tussen de toepassing op de Digibox en de servlet op de Click Machine. We kozen voor het HTTP-protocol omwille van de eenvoud in gebruik, en de standaardondersteuning van dit protocol door de Java Virtuele Machine van de Digibox.

Wanneer de toepassing op de Digibox een aanvraag wenst te sturen naar de glucosemeter, realiseert deze een HTTP-sessie met de OneTouchUltraServlet op de Click Machine; deze servlet is in de praktijk voor de Digibox toepassingssoftware beschikbaar op het adres *http://10.0.0.1:8080/onetouchultraservlet/OneTouchUltraServlet*. Wanneer de sessie geïnitieerd is, wordt gebruik gemaakt van het *GlucoMeter*-protocol voor de verdere communicatie. Het GlucoMeter-protocol is een eenvoudig XML-gebaseerd apparaatonafhankelijk protocol voor het uitwisselen van gegevens tussen de Digibox toepassing en bijvoorbeeld de OneTouchUltraServlet; het gebruik van het protocol vereist geen kennis van aspecten specifiek voor een bepaalde glucosemeter. Merk op dat de toepassing op de Digibox uitsluitend kennis moet hebben van het GlucoMeter-protocol, en dus geen apparaatspecifieke klasse moet bezitten voor de communicatie met een welbepaalde glucosemeter. De specificatie van het GlucoMeter-protocol bevindt zich in Appendix B.

Wanneer de servlet een aanvraag ontvangt, wordt deze ontleed; indien de aanvraag correct is, communiceert de servlet vervolgens – via de hiertoe geïmplementeerde Java klassenbibliotheek – met de glucosemeter om de aanvraag uit te voeren. Het resultaat van de communicatie met het apparaat wordt omgezet in een GlucoMeter-protocolbericht en via HTTP doorgestuurd naar de Digibox toepassing. Tot slot wordt de HTTP-transactie beëindigd.

## Uitbreiding naar een algemene webservice

De hierboven besproken oplossing vereist de implementatie van een apparaatspecifieke servlet voor elk type meter dat gebruikt zal worden. Het voordeel van deze benadering is de afscherming van de apparaatspecifieke aspecten van het randapparaat; de Digibox toepassing kan in dit geval gebruik maken van het eenvoudige GlucoMeter-protocol voor apparaatonafhankelijke toegang tot de glucosemeter. Het is echter niet ondenkbaar dat er na de ontplooiing van de zorgtoepassing nieuwe glucosemeters op de markt komen; in dit geval kan de Click Machine mogelijk niet overweg met het nieuwe toestel en moet men in een update van de webservices voorzien. De Click Machine kan zijn updates via het internet downloaden; hierbij loopt men echter het risico dat de update faalt en er eventueel een dure interventie ter plaatse nodig is.

Voor de distributie van de zorgtoepassing gebruikt men een carrouselsysteem; dit systeem stuurt de toepassing via de kabel mee met het televisiesignaal naar de Digibox, en zorgt dat de gebruiker steeds kan beschikken over de nieuwste versie. De distributie van software via een carrousel is volledig transparant voor de gebruiker, en bezit het grote voordeel dat het updatemechanisme aan de gebruikerskant niet stuk kan gaan. Dit in beschouwing genomen, is het in een aantal gevallen eenvoudiger om de apparaatspecifieke details van het randapparaat af te handelen in de Digibox; men kan de Digibox toepassing immers

eenvoudig updaten, terwijl men potentiële updateproblemen met de Click Machine kan vermijden. In dit scenario dient de Click Machine een algemene webservice aan te bieden voor het realiseren van een generische communicatielink tussen de Digibox en het randapparaat.

Voor het realiseren van een generische webservice zouden we in theorie kunnen volstaan met het verplaatsen van het glucosemeter protocolelement in Figuur 33 van de Click Machine naar de Digibox. De servlet hoeft dan niets anders te doen dan de protocolboodschappen ongewijzigd naar het apparaat te sturen, en de antwoorden rechtstreeks aan de Digibox toepassing te bezorgen. We merken echter op dat we te maken hebben met vrij eenvoudige randapparaten die relatief hoge eisen stellen aan de timing van de communicatie; het is duidelijk dat we met behulp van een servlet en het HTTP-protocol onmogelijk aan deze vereisten kunnen voldoen. Een eerste stap naar een algemene webservice is de uitbreiding van Tomcat met de Axis[43] SOAP[44]-implementatie. Door gebruik te maken van een Axis Java webservice kan de communicatie op een objectgeoriënteerd manier benaderd worden, en overschrijdt de levensduur van een communicatiesessie meerdere transacties; dit zal toelaten om meer controle uit te oefenen op de timing van de communicatie met het randapparaat. Axis gebruikt echter TCP voor de onderliggende communicatie, zodat er nog steeds geen garanties zijn omtrent de timing van de communicatie; willen we harde garanties, dan moeten we gebruik maken van een ware-tijdsprotocol zoals bijvoorbeeld het Real-time Transport Protocol (RTP)[45], of eventueel rechtstreekse communicatie met UDP.

De uitbreiding naar een algemene webservice werd niet verder uitgewerkt in het kader van dit eindwerk; het oplossen van deze problematiek kan evenwel aanleiding geven tot een interessant vervolg op deze thesis.

## 3.3 Conclusie

We behandelden in het eerste deel van dit hoofdstuk de problematiek uit het Digibox/internet scenario. Het is duidelijk dat deze connectiviteitsproblematiek het gevolg is van de structuur van het Telenet-netwerk.

Hoewel een Digibox op dezelfde manier aangesloten wordt op de kabelmodem als een standaard PC, verkrijgen beide toestellen toch netwerkadressen binnen verschillende adresbereiken; dit is het gevolg van DHCP-netwerkadrestoekenning op basis van het MAC-adres van de aanvrager. Als gevolg van de toekenning van een netwerkadres in een privaat bereik, kan de Digibox enkel het iDTV-netwerk bereiken.

Door te steunen op de kennis van de netwerkstructuur – en in de wetenschap dat de gateway het internet effectief kan bereiken – zijn we in staat, om aan de hand van de Click Machine, een communicatielink tussen de Digibox en het internet te realiseren. De Click Machine wordt hiertoe tussen de Digibox en de kabelmodem geplaatst, en verzorgt een selectieve netwerkadresomzetting. De oplossing met de Click Machine is bijgevolg uitermate geschikt voor testdoeleinden, tijdens de ontwikkeling van een toepassing voor de Digibox, die gebruik maakt van publieke servers.

We merken op dat men – ondermeer omwille van de gemaakte veronderstellingen – voor de ontplooiing van een commerciële toepassing, beter kan opteren voor een eigen toepassingsgerichte server binnen het iDTV-netwerk. Het is immers niet ondenkbaar dat Telenet zijn beveiligingspolitiek wijzigt, en de firewallregels overeenkomstig aanpast; in dit geval is de correcte werking van de Click-configuratie uiteraard niet langer gegarandeerd, en staat bijgevolg de goede functionering van de toepassing op het spel.

In de tweede helft van dit hoofdstuk hebben we de aanwezigheid van de Click Machine aangegrepen om een communicatielink te realiseren tussen de Digibox en een randapparaat.

We toonden eerst aan dat we met Bluetooth een uniform aansluitpunt kunnen creëren, waarmee we een randapparaat op een eenvoudige manier draadloos kunnen verbinden met de Click Machine. Voor de communicatie maakten we gebruik van het Bluetooth seriële poort profiel, en een Java implementatie van het eenvoudige seriële protocol van de OneTouch Ultra glucosemeter; daartoe breidden we de Java Virtuele Machine uit met de SerialIO bibliotheek voor seriële communicatie.

Eenmaal er een communicatielink beschikbaar was tussen de Click Machine en het randapparaat, implementeerden we een apparaatspecifieke webservice aan de hand van een servlet en de Tomcat servlet container. Als gevolg van deze benadering, konden we de glucosemeter via een eenvoudig XML-gebaseerd protocol aanbieden aan de Digibox.

Bovendien gebruikten we uitsluitend HTTP voor de communicatie tussen de Click Machine en de Digibox. Als gevolg hiervan, heeft de Digibox toepassing geen nood aan extra klassenbibliotheken om de webservice te kunnen gebruiken. We gaven evenwel aan dat het in een aantal gevallen eenvoudiger zal zijn om software-updates te verspreiden naar de Digibox dan naar de Click Machine; bijgevolg zal men voor een commerciële toepassing zeker overwegen om de apparaatspecifieke details onder te brengen in de Digibox toepassing. Deze laatste benadering vermijdt eventuele problemen met het updatemechanisme van de Click Machine en zorgt dat de toepassing ten allen tijde gegarandeerd beschikbaar is voor gebruik. Deze observatie gaf aanleiding tot een korte bespreking van de uitbreiding naar een algemene webservice, en vormt een interessant onderwerp voor een vervolg op deze thesis.

We merken tot slot op dat de combinatie van de ervaring met de toegang tot randapparaten via een webservice, en de in 2.3.5 voorgestelde uitbreiding naar Linux van de Transparante Communicatielaag, het mogelijk zal maken om het scenario in 1.2.4 *Communicatie Digibox/IVT* te realiseren.

# Hoofdstuk 4
# Conclusie

We kregen in dit eindwerk te maken met de heterogene infrastructuur waarbinnen Televic zijn zorgtoepassing wenst te realiseren. Binnen deze infrastructuur bevinden zich allerhande systemen en platformen die – afhankelijk van de context – de nodige ondersteuning bieden aan de zorgtoepassing. Elk van de systemen bleek uniek omwille van zijn karakteristieken en de mogelijkheden die het biedt aan de zorgtoepassing; we gebruikten  toestellen met een verschillende omvang en een mobiel of vaste karakter, voorzien van Windows XP, Windows Mobile 5 of Debian GNU/Linux. De doelstelling van het eindwerk was het realiseren van een aantal communicatielinks waarvoor we het bestaan motiveerden in de scenario's in Hoofdstuk 1. Ondanks het heterogene karakter van de infrastructuur, heeft één communicatietechnologie in het bijzonder een belangrijke rol gespeeld bij het realiseren van deze communicatielinks; zoals de titel van het eindwerk laat vermoeden was dit de Bluetooth technologie.

In Hoofdstuk 2 behandelden we de implementatie van een Transparante Communicatielaag; deze softwarelaag maakt het mogelijk om op een transparante manier ad-hoc Bluetooth netwerken te vormen, en leverde de nodige communicatievoorzieningen voor het realiseren van het IPT/IVT scenario. We behandelden uitgebreid de fases die van belang zijn voor de transparante netwerkvorming, en deden een onderzoek naar de prestaties van de verschillende procedures, die samen de transparante netwerkvorming realiseren.

Eenmaal de transparante netwerkvorming gerealiseerd was, hebben we bekeken we of het mogelijk is om op een transparante manier aan netwerkselectie te doen; het is immers mogelijk dat verschillende Bluetooth netwerken overlappen. Concreet gingen we na in welke mate de Bluetooth indicator voor de ontvangen signaalsterkte geschikt is voor het inschatten van de context. We toonden in dit onderzoek aan dat deze indicatorwaarde sterk afhankelijk is van de precieze oriëntatie van het toestel, en om die reden niet zondermeer inzetbaar is voor het kiezen van een netwerk. Indien men echter bereid is om in te leveren op het vlak van het transparant gebruik van het toestel, konden we een hogere mate van transparantie realiseren op het vlak van de netwerkselectie. Concreet toonden we aan dat het richten van het toestel naar de router van het gewenste netwerk, het aantal selectiemogelijkheden in heel wat gevallen sterk kan reduceren. Tot slot gebruikten we de

indicator voor de ontvangen signaalsterkte voor de detectie van het verlaten van de context; op die manier zijn we in staat om de blokkering van communicatielinks, of zelfs het blokkeren van het volledige Transparant Netwerk, te vermijden.

Het resultaat van het werk verricht in Hoofdstuk 2 is de realisatie van een softwarelaag voor Windows XP en Windows Mobile 5, die kan dienen als referentie bij de implementatie van het communicatiecomponent van een concrete toepassing; we concludeerden immers dat het niet mogelijk is om een softwarelaag te maken die een hoge mate van transparantie biedt voor elke toepassing en elke omgeving. Tot slot toonden we de mogelijkheid aan om de Transparante Communicatielaag uit te breiden naar Linux, om op die manier ondersteuning te bieden voor het Digibox/IVT scenario.

Voor de oplossing van de problematiek geïntroduceerd in de Digibox/internet en Digibox/glucosemeter scenario's, introduceerden we in Hoofdstuk 3 de Click Machine; we plaatsten deze compacte Linux computer tussen de kabelmodem en de Digibox.

Voor het creëren van een communicatielink tussen de Digibox en het internet, maakten we gebruik van de Click Modular Router software. Aan de hand van een analyse van het netwerkverkeer dat passeert via het retourkanaal van de Digibox, waren we in staat om een beeld te vormen van de structuur van het Telenet netwerk; met deze kennis was het mogelijk om de oorzaak van de problematiek te bepalen. We leerden dat Telenet een privaat netwerk gebruikt waarin het zijn servers voor interactieve televisie onderbrengt. Een digibox verkrijgt eveneens een netwerkadres binnen het adresbereik van dit private netwerk. We zagen dat de Digibox het internet onmogelijk kan benaderen omwille van zijn private internetadres, en het gebrek aan een netwerkadresomzetter in het iDTV-netwerk. We implementeerden dan ook een Click script dat deze beperking omzeilt door gebruik te maken van selectieve netwerkadresomzetting. Het script is een mooie illustratie van de kracht en de flexibiliteit van de Click Modular Router software.

Naast de routeringsfunctionaliteit van de Click Machine, hebben we deze eveneens ingezet voor de communicatie tussen de Digibox en een glucosemeter. We zagen dat het met Bluetooth mogelijk is om een randapparaat op een eenvoudige manier draadloos te verbinden met de Click Machine, en steunden op de Java Servlet Technology om de functionaliteit van het randapparaat via een webservice aan te bieden aan de Digibox.

We concludeerden dat we de implementatie van een webservice voor de toegang tot randapparaten op twee manieren kunnen benaderen; enerzijds kan men de apparaatspecifieke aspecten isoleren op de Click Machine en zo afschermen van de Digibox, terwijl het anderzijds mogelijk is om de toepassing op de Digibox rechtstreeks te laten communiceren met het randapparaat. Het voordeel van de eerste benadering is de mogelijkheid om gebruik te maken van een eenvoudig protocol voor de communicatie tussen de Digibox en de Click Machine. Bovendien zijn er dan geen apparaatspecifieke klassen

nodig op de Digibox. Het is echter wel zo dat deze benadering apparaatspecifieke servlets op de Click Machine vereist; in bepaalde gevallen zal men dan ook de voorkeur geven aan de tweede benadering, die het afhandelen van apparaatspecifieke details overlaat aan de Digibox toepassing. We schetsten kort de timingproblematiek van de tweede benadering, en gaven aan hoe hiervoor een oplossing gerealiseerd kan worden.

Zoals dit wel vaker het geval is, creëerde de oplossing van de problemen van dit eindwerk een aantal nieuwe vragenstukken; deze kunnen ongetwijfeld aanleiding geven tot een interessant vervolg op het onderzoek uit deze scriptie. We denken hierbij in het bijzonder aan het Digibox/IVT scenario en de uitbreiding van het Digibox/glucosemeter scenario naar een algemene webservice.

Het realiseren van het Digibox/IVT scenario vereist de implementatie van een Java klassenbibliotheek voor het programmeren van de BlueZ Bluetooth protocol stack. Vervolgens kan de C# implementatie van de Transparante Communicatielaag geporteerd worden naar Java voor gebruik op de Click Machine. Een laatste stap is het ontwerpen van een webservice die de communicatie met de IVT aanbiedt aan de Digibox.

De uitbreiding van het Digibox/glucosemeter scenario zal verder onderzoek vergen naar een geschikt webservice framework en een bruikbaar ware-tijdsprotocol; men dient hierbij vooral rekening te houden met de ondersteuning die de Digibox hiervoor biedt. Tot slot kan aan de hand van de Java klassenbibliotheek voor BlueZ een generische webservice voor toegang tot Bluetooth randapparaten geïmplementeerd worden.

# Appendix A

# Het Digibox/internet Click script

```
/**
 * Click script for providing the Telenet Digibox with internet access.
 * eth0: cable
 * eth1: digibox
 */
AddressInfo(public_address eth0 eth0:eth);
AddressInfo(host_address 10.0.0.1/8 00:01:02:03:04:05);
AddressInfo(private_net 10.0.0.0/8);

/* From Linux */
FromHost(fake0, host_address)
  -> to_digibox::Queue
  -> ToDevice(eth1);

/* From digibox */
FromDevice(eth1,1)
  -> cl_from_digibox::Classifier(
       12/0800,              // IP traffic
       12/0806 20/0001,      // ARP requests
       12/0806 20/0002)      // ARP replies
   -> CheckIPHeader(14)
   -> MarkIPHeader(14)
   -> ipcl_from_digibox::IPClassifier(
       dst host_address,                                    // IP traffic for host
       dst net private_net or udp port 67 or port 68,       // IP traffic for private net
       -)                                                   // IP traffic for public net
   -> to_host::SetPacketType(HOST)
   -> ToHost(fake0);

ipcl_from_digibox[1] // IP traffic for private net
  -> to_cable::Queue
  -> ToDevice(eth0);

ipcl_from_digibox[2] // IP traffic for public net
  -> iprw::IPRewriter(pattern public_address - - - 0 1)
  -> to_cable;

cl_from_digibox[1] // ARP requests
  -> t0::Tee(2)
  -> to_cable;
t0[1]
  -> to_host;

cl_from_digibox[2] // ARP replies
  -> t1::Tee(3)
  -> to_cable;
t1[1]
  -> to_host;
t1[2]
  -> [1]arpq_to_digibox::ARPQuerier(host_address)
   -> to_digibox;

/* From cable */
FromDevice(eth0,1)
  -> cl_from_cable::Classifier(
       12/0800;              // IP traffic
       12/0806 20/0001,      // ARP requests
       12/0806 20/0002)      // ARP replies
```

```
   -> CheckIPHeader(14)
   -> MarkIPHeader(14)
   -> ipcl_from_cable::IPClassifier(
       dst net private_net or udp port 67 or port 68,      // IP traffic to digibox
              dst public_address)                          // IP traffic to public iface
   -> to_digibox;

ipcl_from_cable[1] // IP traffic addressed to the public interface
  -> iprw[1]
  -> Strip(14)
  -> CheckIPHeader
  -> MarkIPHeader
  -> arpq_to_digibox;

cl_from_cable[1] // ARP requests
  -> to_digibox;

cl_from_cable[2] // ARP replies
  -> to_digibox;
```

# Appendix B

# Het GlucoMeter-protocol

## XML schema aanvraag

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="qualified">
  <xs:element name="get_version_number" />
  <xs:element name="get_serial_number" />
  <xs:element name="get_date_and_time" />
  <xs:element name="set_date_and_time" >
    <xs:complexType>
      <xs:attribute name="date" type="xs:date" use="required" />
      <xs:attribute name="time" type="xs:time" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="get_blood_records" />
  <xs:element name="set_glucose_units" >
    <xs:complexType>
      <xs:attribute name="value" type="glucose_units" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="toggle_glucose_units" />
  <xs:element name="get_glucose_units" />
  <xs:element name="set_time_format" >
    <xs:complexType>
      <xs:attribute name="value" type="time_format" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="toggle_time_format" />
  <xs:element name="get_time_format" />
  <xs:element name="zero_datalog" />
  <xs:simpleType name="glucose_units">
    <xs:restriction base="xs:string">
      <xs:enumeration value="MG/DL" />
      <xs:enumeration value="MMOL/L" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="time_format">
    <xs:restriction base="xs:string">
      <xs:enumeration value="AM/PM" />
      <xs:enumeration value="24:00" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

## XML schema antwoord

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="qualified">
  <xs:element name="version_number">
    <xs:complexType>
      <xs:attribute name="value" type="version_number" />
    </xs:complexType>
  </xs:element>
  <xs:element name="serial_number">
    <xs:complexType>
```

```
      <xs:attribute name="value" type="serial_number" />
    </xs:complexType>
  </xs:element>
  <xs:element name="date_and_time">
    <xs:complexType>
        <xs:attribute name="date" type="xs:date" />
        <xs:attribute name="time" type="xs:time" />
    </xs:complexType>
  </xs:element>
  <xs:element name="blood_records">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="blood_record" type="blood_record" minOccurs="0"
maxOccurs="150" />
      </xs:sequence>
      <xs:attribute name="serial_number" type="serial_number" use="required" />
      <xs:attribute name="glucose_units" type="glucose_units" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:complexType name="blood_record">
    <xs:attribute name="date" type="xs:date" />
    <xs:attribute name="time" type="xs:time" />
    <xs:attribute name="control_solution" default="false" type="xs:boolean" />
    <xs:attribute name="high" default="false" type="xs:boolean" />
    <xs:attribute name="value" type="xs:integer" use="required" />
  </xs:complexType>
  <xs:element name="glucose_units" >
    <xs:complexType>
      <xs:attribute name="value" type="glucose_units" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="time_format" >
    <xs:complexType>
      <xs:attribute name="value" type="time_format" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="zero_datalog">
    <xs:complexType>
      <xs:attribute name="success" type="xs:boolean" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:element name="set_date_and_time">
    <xs:complexType>
      <xs:attribute name="success" type="xs:boolean" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="version_number">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9][0-9]\.[0-9][0-9]\.[0-9][0-9]" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="serial_number">
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9]{8}T" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="test_result_value">
    <xs:restriction base="xs:integer">
      <xs:minExclusive value="0" />
      <xs:maxInclusive value="600" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="glucose_units">
    <xs:restriction base="xs:string">
      <xs:enumeration value="MG/DL" />
      <xs:enumeration value="MMOL/L" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="time_format">
    <xs:restriction base="xs:string">
      <xs:enumeration value="AM/PM" />
      <xs:enumeration value="24:00" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

# Voorbeeldtransacties

## Ophalen versienummer

### Aanvraag

```
<?xml version="1.0" encoding="utf-8"?>
<get_version_number/>
```

### Antwoord

```
<?xml version="1.0" encoding="utf-8" ?>
<version_number value="01.00.00"/>
```

## Ophalen serienummer

### Aanvraag

```
<?xml version="1.0" encoding="utf-8"?>
<get_serial_number/>
```

### Antwoord

```
<?xml version="1.0" encoding="utf-8" ?>
<serial_number value="DUMMYDUMT"/>
```

## Ophalen meterklok

### Aanvraag

```
<?xml version="1.0" encoding="utf-8"?>
<get_date_and_time/>
```

### Antwoord

```
<?xml version="1.0" encoding="utf-8" ?>
<date_and_time date="2006-05-30" time="10:10:05"/>
```

## Instellen meterklok

### Aanvraag

```
<?xml version="1.0" encoding="utf-8"?>
<set_date_and_time date="2006-05-30" time="10:10:00"/>
```

### Antwoord

```
<?xml version="1.0" encoding="utf-8" ?>
<set_date_and_time success="true"/>
```

## Ophalen van meetgegevens

### Aanvraag

```
<?xml version="1.0" encoding="utf-8" ?>
<get_blood_records/>
```

### Antwoord

```
<?xml version="1.0" encoding="utf-8" ?>
<blood_records glucose_units ="MG/DL" serial_number="DUMMYDUMT">
<blood_record time="10:30:00" date="2006-05-21" value="130" high="false"
control_solution="false"/>
<blood_record time="10:44:00" date="2006-05-22" value="135" high="false"
control_solution="false"/>
<blood_record time="10:43:00" date="2006-05-23" value="125" high="false"
control_solution="false"/>
</blood_records>
```

# Bronnen


[1] Het Generatiepakt. *http://premier.fgov.be/nl/051011_generatiepact.pdf*. (Oktober, 2005).

[2] Voorstel van resolutie betreffende een toekomstgericht beleid dat meer perspectieven biedt aan verpleegkundigen. R. Van Cleuvenbergen et al. Vlaams Parlement. *http://jsp.vlaamsparlement.be/docs/stukken/2000-2001/g504-1.pdf*. (December, 2000).

[3] Coplintho – Innovative Communication Platform for Interactive eHomeCare. IBBT. *https://coplintho.ibbt.be*. (2005-2006).

[4] MobiHealth – Shaping The Future Of Healthcare. *http://www.mobihealth.org*.

[5] Televic NV. *http://www.televic.com*.

[6] Contextbewuste beveiliging van vertrouwelijke informatie binnen ziekenzorgtoepassing. O. Christiaens. Universiteit Gent. (Mei, 2006).

[7] Digitale tv in Vlaanderen – De visie van de Vlaamse overheid. Kabinet Bourgeois. *http://www2.vlaanderen.be/ned/sites/media/eflanders/kenniswijzer/jaargangen/2005nummer09/digitale%20tv.pdf*. (November, 2005)

[8] Studie van MHP-ontwikkeltraject aan de hand van een iDTV-toepassing voor e-thuiszorg. E. Cant. Universiteit Gent. (Mei, 2006).

[9] The Computer for the 21st Century. M. Weiser. Scientific American, 265. (September, 1991).

[10] IrDA.org – The Infrared Data Association. *http://www.irda.org*.

[11] IEEE 802.11 – The Working Group Setting the Standards for Wireless LANs. *http://grouper.ieee.org/groups/802/11/*.

[12] Bluetooth.org – The Official Bluetooth Membership Site. *http://www.bluetooth.org*.

[13] IrDA Global Market Report 2005. The Infrared Data Association. *http://www.irda.org/displaycommon.cfm?an=1&subarticlenbr=17*. (2005).

[14] Bluetooth Core Specification Version 2.0 + EDR. Bluetooth Special Interest Group. *http://www.bluetooth.org/foundry/adopters/document/Core_v2.0_EDR/en/1/Core_v2.0_EDR.zip*. (November, 2004).

[15] Bluetooth Application Developer's Guide. Jennifer Bray et al. Syngress Publishing, Inc. (2002).

[16] Mobile Communications, Second Edition. Jochen Schiller. Pearson Education Limited, United Kingdom. (2003).

[17] IEEE OUI and Company_id Assignments. IEEE Registration Authority. *http://standards.ieee.org/regauth/oui/index.shtml*.

[18] Programming Microsoft Windows CE .NET, Third Edition. Douglas Boling. Microsoft Press, Redmond, USA. *http://bolingconsulting.com/programmingwindowsce.html*. (2003).

[19] 32Feet.Net – Personal Area Networking with .NET. *http://32feet.net/*.

[20] High Point Software. *http://www.high-point.com/*.

[21] Franson Technology AB. *http://www.franson.com/*.

[22] Bluetooth Assigned Numbers Workgroup – Baseband. Bluetooth Special Interest Group. *https://www.bluetooth.org/foundry/assignnumb/document/Baseband*.

[23] Bluetooth Assigned Numbers Workgroup – Service Discovery Protocol. Bluetooth Special Interest Group. *https://www.bluetooth.org/foundry/assignnumb/document/service_discovery*.

[24] Vergelijkende studie van ZigBee en Bluetooth voor context-aware ziekenzorgtoepassingen. P. De Mil. Universiteit Gent. (Mei, 2005).

[25] BlueZ – Official Linux Bluetooth protocol stack. *http://www.bluez.org/*.

[26] JBlueZ – Java API for BlueZ. *http://jbluez.sourceforge.net/*.

[27] Sun Comm API. Sun Microsoft Systems. *http://java.sun.com/products/javacomm/*.

[28] Computer Networks – A Top-Down Approach Featuring the Internet, Second Edition. James F. Kurose and Keith W. Ross. Pearson Education Inc. (2003).

[29] Data Over Cable Interface Specification (DOCSIS). CableLabs. *http://www.cablemodem.com/*.

[30] Internet Assigned Numbers Authority (IANA). *http://www.iana.org/*.

[31] Ethereal – Network Protocol Analyzer. *http://www.ethereal.com/*.

[32] Real Time Streaming Protocol (RTSP). *http://www.rtsp.org/*.

[33] Debian GNU/Linux. *http://www.debian.org/*.

[34] The Click Modular Router Project. *http://www.read.cs.ucla.edu/click*.

[35] The Click Modular Router. Eddie Kohler et al. Laboratory for Computer Science, MIT. *http://pdos.csail.mit.edu/papers/click:tocs00/paper.pdf*. (Augustus, 2000).

[36] Lifescan. *http://www.lifescaneurope.com/benl/*.

[37] BlueSerial Bluetooth Products and RS232 Cable Replacement. *http://www.blueserial.com/*.

[38] SerialIO SerialPort API. *http://www.serialio.com/*.

[39] OneTouch Ultra Meter RS-232 Communication Specification. Lifescan.

[40] Apache Tomcat. *http://tomcat.apache.org/*.

[41] Java 2 Platform Standard Edition (J2SE). Sun Microsoft Systems. *http://java.sun.com/j2se/1.5.0/index.jsp*.

[42] Java Servlet Technology. Sun Microsystems. *http://java.sun.com/products/servlet/index.jsp*.

[43] Apache Axis. *http://ws.apache.org/axis/*.

[44] Apache Simple Object Access Protocol (SOAP). *http://ws.apache.org/soap/*.

[45] Real-time Transport Protocol (RTP). Internet Engineering Task Force (IETF). *http://www.ietf.org/rfc/rfc1889.txt*.

# Figuren

Figuur 1: Overzicht infrastructuur ..... 2
Figuur 2: Opstelling zorginstelling – IPT/IVT ..... 6
Figuur 3: Opstelling thuiszorg – Digibox/internet ..... 8
Figuur 4: Opstelling thuiszorg – Digibox/glucosemeter ..... 9
Figuur 5: Opstelling thuiszorg – Digibox/IVT ..... 10
Figuur 6: Overzicht Transparant Netwerk ..... 14
Figuur 7: Overlappende Transparant Netwerken ..... 17
Figuur 8: Bluetooth authenticatieprocedure onder Windows XP en Windows Mobile 5 ..... 22
Figuur 9: Problematiek Seriële Bluetooth-poort dienst ..... 23
Figuur 10: De operationele modi van Bluetooth ..... 26
Figuur 11: Formaat Bluetooth apparaatadres ..... 28
Figuur 12: Een weergave van de Bluetooth protocol stack ..... 31
Figuur 13: De Bluetooth gegevenstransportarchitectuur ..... 32
Figuur 14: De Microsoft Bluetooth protocol stack ..... 39
Figuur 15: Formaat Bluetooth Class Of Device ..... 51
Figuur 16: Indicatorwaarden voor de HP iPAQ hx2490 in functie van de tijd (20 m, 0°, meting 1) ..... 67
Figuur 17: Indicatorwaarden voor de Dell Axim X51 in functie van de tijd (20m, 0°, meting 1) ..... 67
Figuur 18: Gemiddelde indicatorwaarden over 5 s voor de HP iPAQ hx2490 in functie van de rotatiehoek ..... 68
Figuur 19: Gemiddelde indicatorwaarden over 5 s voor de Dell Axim X51 in functie van de rotatiehoek ..... 68
Figuur 20: Algoritme voor contextgevoelige Router selectie ..... 69
Figuur 21: Overzicht van de fases in een Transparant Netwerk ..... 71
Figuur 22: Blokkering communicatielinks zonder link monitor ..... 76
Figuur 23: Blokkering communicatielinks met link monitor ..... 76
Figuur 24: Script voor het ontvangen van binnenkomende Bluetooth verbindingen ..... 77
Figuur 25: Overzicht Telenet netwerk ..... 83
Figuur 26: Fragment – opvragen van informatie over een aflevering ..... 84
Figuur 27: Fragment – uitwisseling van gebruikersinformatie ..... 85
Figuur 28: Fragment – effectief bestellen en starten van de aflevering ..... 85
Figuur 29: Opstelling met de Click Machine ..... 86
Figuur 30: Adresomzetting Click Machine ..... 87
Figuur 31: Schematische voorstelling Click script ..... 89
Figuur 32: Opstelling met de Click Machine en de glucosemeter ..... 94
Figuur 33: Overzicht communicatie Digibox/glucosemeter ..... 95
Figuur 34: OneTouchUltra configuratieparameters ..... 96

# Tabellen

Tabel 1: Overzicht Pocket PC's en ondersteunde communicatietechnologieën ..... 19
Tabel 2: Bluetooth vermogensklassen ..... 32
Tabel 3: Toestellen in eenzelfde kamer – aantal succesvolle zoekacties naar een toestel per 5 pogingen ..... 47
Tabel 4: Toestellen in verschillende kames – aantal succesvolle zoekacties naar een toestel per 5 pogingen ..... 48
Tabel 5: Toestellen binnen eenzelfde kamer – gemiddelde oproeptijd ..... 53
Tabel 6: Toestellen  in verschillende kamers – gemiddelde oproeptijd ..... 54