UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

xAP as an open source communication protocol for Health Systems Engineering

An application in the telemedicine environment

Erich Paul Andrag

15382435

Final year project presented in partial fulfilment of the requirements for the degree of Bachelors of Industrial Engineering at Stellenbosch University.

Study Leader: Dr AF van der Merwe

October 2011

ACKNOWLEDGEMENTS

I wish to acknowledge the following people for their help and guidance during the project:

- Dr Andre van der Merwe, the study leader of this project, for the initiation of this project as well guidance throughout. The problems provided proved a challenge and creative solutions were constantly pursued.
- Mr Andries Venter, University Stellenbosch IT Department Head Network Engineer, for his careful explanation of how a network operates.

These two mentors contributed immensely to my quest for learning. I will always be thankful for the knowledge acquired through the study and execution of this project.

DECLARATION

I, the undersigned, hereby declare that the work contained in this final year project is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

……………………………….   ……………………………

EP Andrag   Date

ECSA EXIT LEVEL OUTCOMES REFERENCES

Exit level outcome	Section(s)	Page(s)
1. Problem solving	1.2 Problem Statement 5 Developing a Network Architecture 7 Conclusions and Recommendations	2-3, 27-39, 48-49
5. Engineering methods, skills & tools, incl. IT	5.1Components within the Telemedicine Network 5.4 Application Platform	29-32, 35-39
6. Professional & Technical communication	Entire Report	
9. Independent learning ability	2 Healthcare Standardisation Initiatives 3 Communication over a Network medium 4 Evaluation of the eXtensible Automation Protocol	6-7, 8-17, 18-26
10. Engineering professionalism	1 Introduction 7 Conclusions and Recommendations	1-5 48-49

SYNOPSIS

Engineering initiatives to standardise the communication of health related systems are ineffective and uncoordinated. The extensive advantages of such standardisation could benefit both quality of service and patient turnaround time. Standardisation becomes critical once information and communication technologies (ICT) are implemented. ICT system interoperability is core to ensure the success of telemedicine.

Current standardisation of telemedicine systems is led by two standardisation organisations. *Health Level Seven* (HL7) and the *International Organisation for Standardisation* (ISO) are both leading comprehensive standardisation initiatives. Access to documentation for development of systems according to these standards is restricted, inhibiting third party development and contribution.

Telemedicine in Africa requires an open source development platform where privileged users can develop their own extension without the restrictions associated with current standards. Properties of the platform should address specific problems faced in an African context. Problems could include i) a lack of network infrastructure, ii) costly data transmission and iii) a lack of devices able to access the Internet. Recent widespread adoption of mobile devices compatible with cellular networks also provides an opportunity to develop standards supporting telemedicine use on cellular networks.

Organisations are already capitalising on the benefits mobile phones offers. Applications for mobile phones, which provide medical related services, are popular. Services include general medical information as well as using the technology of the mobile phone to perform basic diagnoses. A simple heart rate monitoring is one such example. In Africa, a prime example for mobile initiatives is *EPurse*, capitalising on a successful implementation of mobile banking at the point of sale.

This project investigates the application of the eXtensible Automation protocol (xAP) as a communication protocol suitable to the telemedicine environment. The properties of xAP prove favourable for application in the African context. Requirements of a system able to support xAP integrations are determined in relation to the protocol specifications. xAP is further integrated with the Internet Protocol Suite to facilitate Internet communication.

A network configuration, representative of a real world operation, is tested in order to determine xAP suitability for telemedicine networks. The network configuration strives to represent telemedicine implementations, where data is communicated between a remote device and an interested party, over the Internet. Restrictions of the telemedicine systems communicating over the Internet were assessed.

Internet Service Providers (ISPs) often restrict xAP applications based on the underlying Internet structure it utilises. It is thus suggested that a secondary method of communication, conforming to Hyper Text Transfer Protocol data transfer, is required if a successful communication session cannot be realised given the properties of xAP.

Protocols are prescriptive on how communication should be done, as does xAP, whereas standards are an agreed way of operation. In order for telemedicine to be implemented sustainably, a standard for telemedicine networks should be created supporting the xAP framework. Simply put, xAP enabled network communication should be promoted as a standard and not just a protocol. This argument provides guidance to the execution of this project.

OPSOMMING

Ingenieursinisiatiewe vir die standardisering van kommunikasie van gesondheidstelsels is oneffektief en ongekoördineerd. Die voordele wat sulke standardiseerings inisiatiewe kan inhou is uitgebreid beide vir kwaliteit van die dienslewering so wel as pasiënt omkeer tyd. Standardisasie raak selfs meer krities wanneer inligting en kommunikasie tegnologie in gebruik geneem word. Gemak van oorskakeling in inligting en kommunikasie tegnologie stelsels is kern tot die sukses van telemedisyne.

Huidige standaardisasie van telemedisyne stelsels word gelei deur twee organisasies vir standaardisasie. *Health Level Seven* (HL7) en die *International Organistion for Standardisation* lei beide omvattende standaardiserings inisiatiewe. Toegang tot die dokumentasie vir die ontwikkeling van stelsels voldoende aan die standaarde is beperk, wat ontwikkeling en bydraes deur derde partye verhinder.

Telemedisyne in Afrika vereis 'n oop bron ontwikkelings platform waar voorkeur gebruikers hul eie uitbreidings kan ontwikkel sonder die beperkinge geassosier met huidige standaarde. Eienskappe van die platform moet spesifieke probleme wat Afrika in die gesig staar, aanspreek. Probleme kan opgesom word as i) die gebrek aan infrastruktuur, ii) die hoë koste van data oordrag en iii) gebrek aan toestelle met toegang to die Internet. Toename in die gebruik van selfone bied 'n ook geleentheid vir die ontwikkeling van standaarde geskik vir telemedisyne gebruik op selfone.

Organisasies kapitaliseer al reeds op die voordele gebied deur selfone. Sagteware, wat mediese dienste op selfone verskaf, is gewild. Dienste sluit in algemene mediese inligting sowel as toenemende gevorderde toepassings waar tegnologie van die selfoon gebruik word vir basiese diagnose. 'n Voorbeel hiervan is 'n eenvoudige hartklop monitor. In Afrika is a goeie voorbeel van selfoon verwante inisiatiewe, *EPurse*, 'n toepassing van bank dienste by die verkoopspunt.

Die projek ondersoek die *eXtensible Automation protocol* (xAP) as 'n kommunikasie protokol vir geskiktheid in die telemedisyne omgewing. Die eienskappe van die protokol bleik gunstig te wees vir implementeering in 'n Afrika konteks. Hierdie studie ondersoek 'n stelsel ondersteunend vir die integrasie van xAP in die lig van die protokol spesifikasies. xAP word verder geïntegreer met die *Internet Protocol Suite* om kommunikasie oor die Internet te vergemaklik.

'n Netwerk konfigurasie, verteenwoordigend van algemene gebruik, word getoets om xAP geskiktheid vir telemedisyne netwerke te bepaal. Die netwerk konfigurasie maak staat op die Internet as kommunikasie medium. Dit verteenwoordig telemedisyne kommunikasie tussen 'n

afgeleë toestel en ‘n ander geintereseerder toestel. Beperkinge op telemedisyne kommunikasie oor die Internet word ook geasseseer.

Internetdiensverskaffers beperk gereeld xAP toepassings, as gevolg van die onderliggende Internet struktuur wat xAP gebruik. Dus word dit voorgestel dat ‘n sekondêre kommunikasie metode daargestel word, wat die *Hyper Text Transfor Protocol* gebruik, indien ‘n kommunikasie sessie nie realiseer gegewe die xAP eienskappe nie.

Protokols dien as reëls vir kommunikasie, soos ook xAP, teenoor standaarde wat ‘n ooreengekomde manier van iets doen is. Vir die volhoubare implementering van ‘n xAP ondersteunende netwerk word dit voorgestel dat ‘n standaard rondom die xAP raamwerk ontwikkel word. Eenvoudig gestel, xAP netwerk kommunikasie moet as ‘n standaard bevorder word, nie net ‘n protokol nie. Die argument lei die uitvoering van hierdie projek.

Table of Contents

List of Figures

List of Tables

Glossary

API	Application Programming Interface
Communication	In terms of this project communication represents a session where two entities are able to send data between one another and both are able to understand and interpret the data
GPRS	General Packet Radio Service
HL7	Health Level Seven (7)
IANA	Internet Assignment Numbers Authority
ICT	Information and Communication Technology
IP	Internet Protocol
IP Network	A network based on the Internet Protocol
ISO	International Organisation for Standardisation
MTU	Maximum Transmission Unit
NIC	Network Interface Controller
Packet	A unit of data for transport, size measured in bytes
Parse	The ability of a device to separate a packet into understandable pieces
PID	Process Identification Number
PLC	Programmable Logic Controller
Protocol	Rules governing communication
Standard	An agreed-upon way of operation
Telemedicine	For the goal of this project telemedicine involves the sending of data over a network for medical applications
UPnP	Universal Plug-and-Play
xAP	eXtensible Automation Protocol
xAPp	eXtensible Automation Protocol application
xHCP	xPLHal Control Protocol
XML	eXtensible Markup Language
xPL	eXtremely simPle protocoL - A similar protocol to xAP
xPLHal	A service running on a computer to manage xPL devices on a network

1 Introduction

Specific applications of healthcare in South Africa are often unsuccessful due to the lack of information and communication infrastructure. Funding and knowledge for such infrastructure is absent or is applied ineffectively. Challenges unique to the South African and African context further hamper the successful application of healthcare initiatives. Overcoming these challenges provides for a multifaceted opportunity. This project focuses on the specific data and information sharing validity in a healthcare environment through the use of an open source protocol. Initiatives such as telemedicine will be investigated as the primary application. Telemedicine has gained widespread support and is considered to be a strategic tool for South Africa to improve healthcare delivery (Benatar, 2004).

1.1 Background

Africa, being known for its rural state and lack of infrastructure, is a fast developing continent in terms of information and communication technology (ICT). With a population of around 1 billion people, projected to double before the end of the century, the demand for technologies will continue to grow (United Nations, 2010). Mobile phones are a standard medium and have gained widespread integration for use in rural Africa. By the end of 2008, Africa had 32 million Internet users and an astonishing 246 million mobile cellular subscription (International Telecommunications Union, 2009), considering that most of Africa’s inhabitants are still considered poor.

The global rise in interest for telemedicine applications is driven by several factors. Factors are summarised below (Istepanian et al., 2006):

- Improved access to medical information and data
- Improved patient care and healthcare services
- Better specialist care and enhanced medical productivity
- To reduce costs associated with information and data sharing
- Automation of medical data capturing and monitoring thereof

These factors are as important in an African context as internationally. Considering additional challenges related to an African context, such as remote locations, high mobility of users and unreliable data networks, special care will have to be taken when designing a network for data and information sharing.

With the growth of technology in South Africa and Africa medical documentation and record keeping are slowly progressing to being created and kept electronically. Considering the wide variety of devices and mediums used to collect data it is important that data remains portable between medical systems. Absence of a protocol definition to facilitate inter-portability hampers quality of healthcare.

The current state of standards for health information technology is limited due to the high input requirements of such standards. Implementation of telemedicine systems are closely related to the adoption of an open standards environment (Kifle et al., 2008). Current standards that are generally accepted include Health Level 7 (HL7) and International Organisation for Standardisation (ISO) 11073-91064:2009.

HL7's focus is on the interoperability of health information technology. The current arrangement consists of 55 member countries. It is also the most widely used standard in America for such applications and forms part of the American National Standards Institute's portfolio of standards (Health Level Seven International, 2011). ISO 11073, defining a standard communication protocol for healthcare, is subset of and managed by ISO Technical Committee (ISO TC) 215 focusing on the standardisation of Health ICTs. It is closely related to HL7 and continued efforts exist to harmonize the standards.

In the South African context efforts exist to standardise the communication of healthcare technologies. Efforts in the private sector are managed by the Private Healthcare Information Standards Committee (PHISC). In 2009 Medi-Clinic and Discovery Health showed interest and eventually partially adopted HL7 based software solutions (Spronk, 2009). Public sector efforts lack direction and do not consist of the necessary resources to standardise ICTs (Matshidze & Hanmer, 2007).

The standardisation and conceptualisation of telemedicine ICT protocols is limited in terms of the supportive projects. ISO TC 215 has launched an effort to standardise the telemedicine procedure by providing documentation regarding the interoperability of telehealth systems (ISO, 2004). Unfortunately the standards produced by ISO TC 215 have restricted access and is thus not considered to be open source.

1.2 Problem Statement

In order for medical personnel to communicate and share patient data effectively a common platform needs to be called into existence to support such a goal. Considering the application thereof in an African context provides further challenges. Communication technologies, especially mobile communication technologies, are growing in capability and popularity. The effective use and full utilisation of these technologies in the medical environment provides for an improvement opportunity in the telemedicine environment.

Supporting systems for telemedicine activities is scarce due to the high complexity of conducting a diagnosis in a remote location. Costs related to commercial systems are steep and insufficient for rural African applications. Further integration of the telemedicine system with existing systems must be seamless, easing the volatile process of telemedicine diagnosis.

Data availability for telemedicine applications is often an undervalued aspect. To enhance the integrity of a system, accurate data must readily be available to its users. Considering the minimalistic nature of the technology used in telemetry this data should be simple in nature, providing only the most needed patient attributes for diagnosis.

Currently no standardised platform is widely accepted. The standard procedures and communication platforms vary from site to site. Commercialised platforms able to support widespread implementation are expensive and provide little in the sense of adaptability and open source development. Standardisation of a protocol to effectively transmit Electronic Health Records (EHR), Electronic Medical Records (EMR), relevant device data and status updates is thus cardinal. The protocol must be robust in the sense of being network agnostic, suitable for a wide variety of applications, adaptable and extendable for continual future use. Considering the application in an African context this protocol must strive to achieve minimal data size and complexity.

1.3 Objectives

The objective of this project, with reference to the above, is dual fold. Firstly to evaluate the xAP protocol, as used in home automation as a suitable protocol for telemedicine diagnosis and data sharing. The xAP protocol must be evaluated according to these aspects:

- Scalability of the xAP-protocol based system
- Integration with existing systems
- Network agnostic nature
- Simplistic and lightweight (bandwidth) properties

Secondly, network architecture capable of supporting telemedicine activities is suggested and tested based upon the xAP-protocol nature of the system. The network model will demonstrate the utilisation of the Internet as a central for data transmission. Robust network architecture is required where the effective operation of the network does not rely on a single device and can thus operate if one or more devices are offline.

The protocol and network platform should be open source as far as possible. This will provide for easier integration with existing platforms and open ended development.

1.4 Roadmap to Project Report

To place the contents in prospective questions were asked to guide the project study. Questions served to continuously compare the direction of the project to the problem statement and objectives. The questions are:

- Why xAP?
- What specifications should xAP fulfil?
- What format should xAP packages assume?
- What technology is already out there?
- What properties influence successful xAP communication?
- What tools are needed for communication?
- How can xAP be integrated?

With the questions as guideline a roadmap to this project was created. The roadmap can be seen in Figure 1.

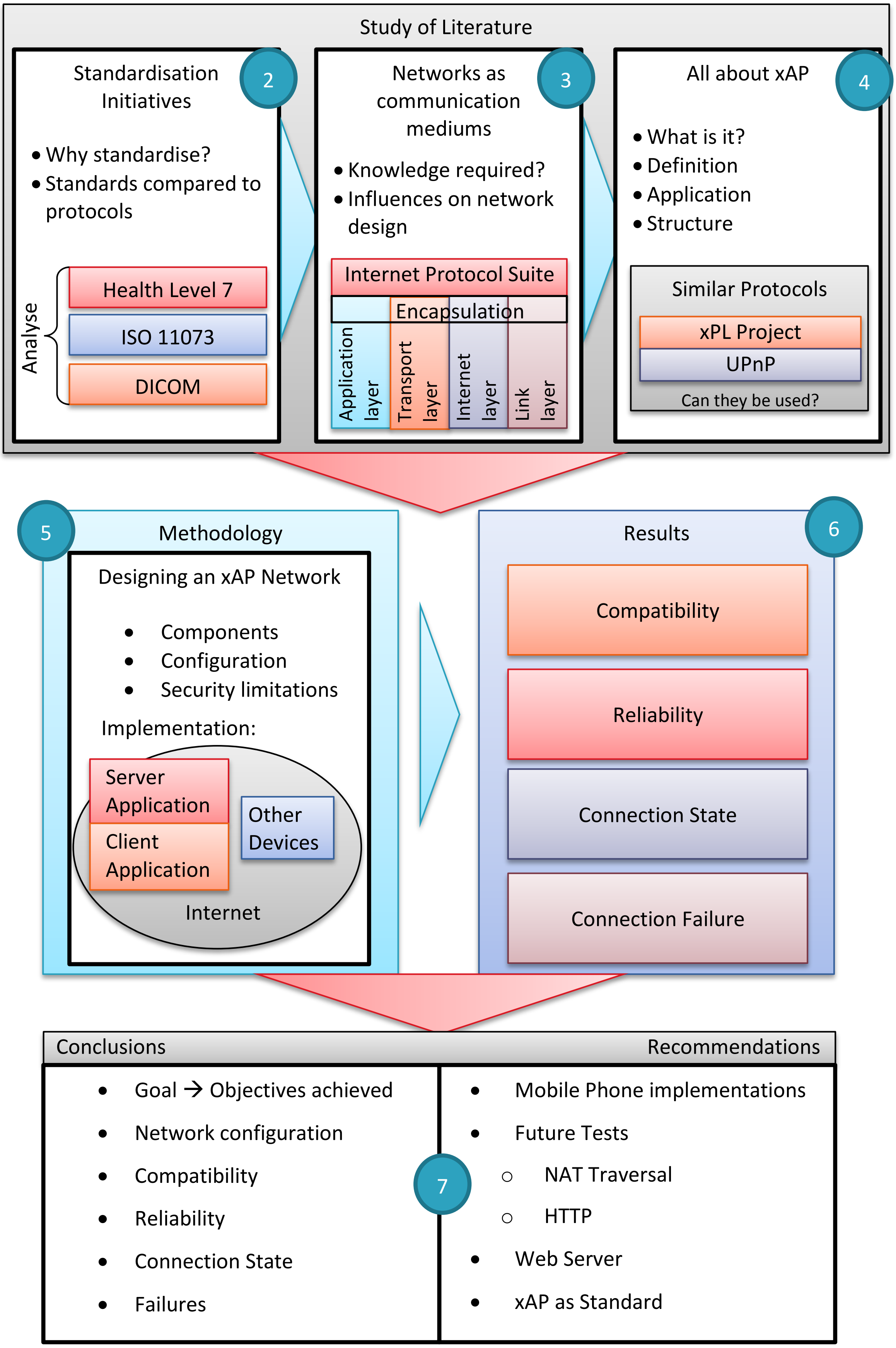


FIGURE 1: ROADMAP TO THIS PROJECT

2 Healthcare Standardisation Initiatives

Several efforts have been made to standardise healthcare systems. Standardisation centres on the principle of creating a widely accepted and proofed way of execution, implementation, design or measurement. It improves cooperation and, by making it publicly available, is beneficial to all parties. Standards differ from protocols by not explicitly prescribing rules, but rather prescribing a framework within which to interact. Protocols are thus a more specialised set of instructions or rules that govern interaction. Often a standard contain several subsets of protocols based upon that specific standard.

The standardisation efforts are continuous as new devices, terminology and operating procedures are developed every day. The largest and most supported standards applicable to the health sector are Health Level 7 (HL7), Digital Imaging and Communications in Medicine (DICOM) and ISO/IEEE 11073 (Schmitt et al., 2007).

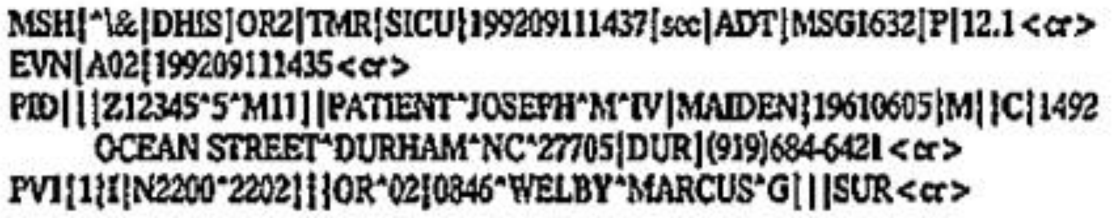
2.1 Communication Standardisation

The more widely accepted a standard becomes the more expensive it becomes for an organisation not to adopt the standard. Standards discussed here have all made efforts to become interoperable with each other as this is essential to survival of the specific standard. For most cases the word standards are often exchanged for the word protocol as the structure of the intercommunication is loosely defined compared to the formal specifications of a protocol. The core objectives and specifications of each standard are discussed to provide more perspective of inter-device communication in the healthcare industry works.

2.1.1 Health Level 7

Health Level 7 is a standard that strives to be all-encompassing for medical applications. It provides for the storage of Electronic Health Records to communication of low level devices. A primary strategy of HL7 is to “develop coherent, extendible standards that permit structured, encoded healthcare information of the type required to support patient care, to be exchanged between computer applications, while preserving the meaning” (Basu, 2009). The “7” is with reference to the seven layers in the Open Systems Interconnection Reference Model (see chapter 3.4). This signifies the network integration of HL7.

HL7 consists of various protocols; of particular interest is the Application layer protocols (see chapter 3.3.1) used for software implementation. An example in Figure 2 shows how a typical message between two HL7 compatible systems is expressed. This particular message, containing a patient’s information, is according to version 2.2 of the standards. The standards have been updated to version 3. The message consists of a header, event type description and a patient information section (in this case).



```
MSH|^\&|DHIS]OR2|TMR{SICU}199209111437{scc|ADT|MSG1632[P|12.1<cr>
EVN|A02{199209111435<cr>
PID|||Z12345^5^M11]|PATIENT^JOSEPH^M^IV|MAIDEN}19610605|M||C|1492
        OCEAN STREET^DURHAM^NC^27705|DUR](919)684-6421<cr>
PV1{1}I|N2200^2202]||OR^02|0846^WELBY^MARCUS^G|||SUR<cr>
```

FIGURE 2: HEALTH LEVEL 7 MESSAGE BETWEEN SYSTEMS (MOOR, 1993)

HL7 is incorporated as an American National Standards Institute standard and enjoys wide use in the American health care industry. To implement the standards requires documentation to be acquired from HL7 and often consultation as well. This provides a flow of income to HL7.

2.1.2 ISO/IEEE 11073

The International Organisation for Standardisation (ISO) in cooperation with the Institute of Electrical and Electronic Engineers (IEEE) has developed and proposed the ISO/IEEE 11073 standards for medical device interoperability. The objectives of the standard are related to providing device plug-and-play (no setup requirements) capability and facilitate exchange of data. Access to these standards is restricted and little full system integration has been realised. Despite this the standards are defined comprehensively and provide a detailed architecture for the whole system, including integration with other standards supported by ISO. Compatibility with HL7 is also supported.

2.1.3 Digital Imaging and Communications in Medicine

As the name suggests DICOM is focused on the standardisation of imaging formats in the health-sector. DICOM is used to “produce, store, display, process, send, retrieve, query or print medical images and derived structured documents” (Digital Imaging and Communications in Medicine , [s.a]). Both products and information systems are developed conforming to the DICOM standards for images and is used in almost all medical environments. The DICOM standard documentation is freely available. The standard was expanded to be all encompassing for medical imaging applications and is optimized for use in a networked environment based on the Internet Protocol Suite (see Chapter 3.3.1). The protocols defined in the DICOM standard mostly interact in the Application layer of the Internet Protocol Suite but because it is defined for the purpose of image processing is of little application to the interconnection of other devices.

3 Communication over a Network medium

Networks are the key element in information sharing for the modern computing age. The physical medium by which data is communicated over a network varies and can influence the network type. Typical physical media are described as air (wireless) or wired (power lines or fibre optic cables).

Creating a network, by manipulating and connecting physical media in a communicative manner, has resulted in our ability to communicate over a distance. Once a network is created, a common protocol, which facilitates communication between endpoints, is needed. Standardisation organizations, such as the International Organization for Standardisation (ISO) and the Internet Engineering Task Force (IETF) labour continuously to create and update these protocols, but any individual or organisation can also create their own protocol. Creation of a nonstandard (not supported by a standardisation organisation) may result in incompatibility with standard protocols. If compatibility is required extra measures have to be incorporated to counteract this.

The eXtensible Automation Protocol (xAP), which is the focuses of this project, is defined, among others, for radio, RS232, Ethernet and Wireless-Fidelity physical networks. In order to transmit data between the different networks a protocol translator (bridge) is required (Lidstone et al., 2002). Current proposed architecture of the telemedicine network utilises smart devices, such as mobile phones, programmable logic controllers and computers, to collect and manipulate data.

Generally stated a network is several compatible devices connected together, not specifying the structure of the network (Elahi & Elahi, 2006). This project focuses on networks used for data communication and will discuss architecture relative to such networks. To communicate via a network using an interface, the interface requires a network interface controller (NIC), which enables Ethernet or wireless communication.

Data being communicated over a network is formatted into packets. A Packet is one transmission unit. Protocols format these packets according to rules. To allow for ease of handling such packets limits are given to the allowed size of a packet. This is called the Maximum Transmission Unit (MTU) and is generally accepted to be 1500 bytes (Postel, 1983).

To facilitate communication to the reader, the Internet, as the most accessible and relatable network, is used in examples. The Internet also plays a critical role in the developed of the telemedicine network architecture and is clarified in chapter3.3.

3.1 Network Models

Network models, not to be confused with network topology, describe the specific way in which devices are related on a network (Elahi & Elahi, 2006). For the application of this project two network model types are discussed below, namely client/server and peer-to-peer.

3.1.1 Client/Server Model

The client/server model is applicable to almost all networking environments (Hall, 2009). A server is used to store and process data which is of interest to one or more clients. Servers can also be used to manage the interconnection of devices on a network. For a client to be able to reach a server it is required that the server's address is known by the client. On the Internet servers, or a collection of servers, are often known by an alias such www.example.com. Clients are devices requesting a resource that the server may be able to provide. A typical demonstration of a client/server situation is when opening the website www.example.com, you, as the client, sends a request to the server and the server, represented by www.example.com, responds appropriately.

In a client/server model a client submits a task to the server; the server processes the task and returns the result to the client (Elahi & Elahi, 2006). Figure 3 depicts a typical Client/Server model. Note that the server rarely initiates the connection and this is especially true for the Internet.

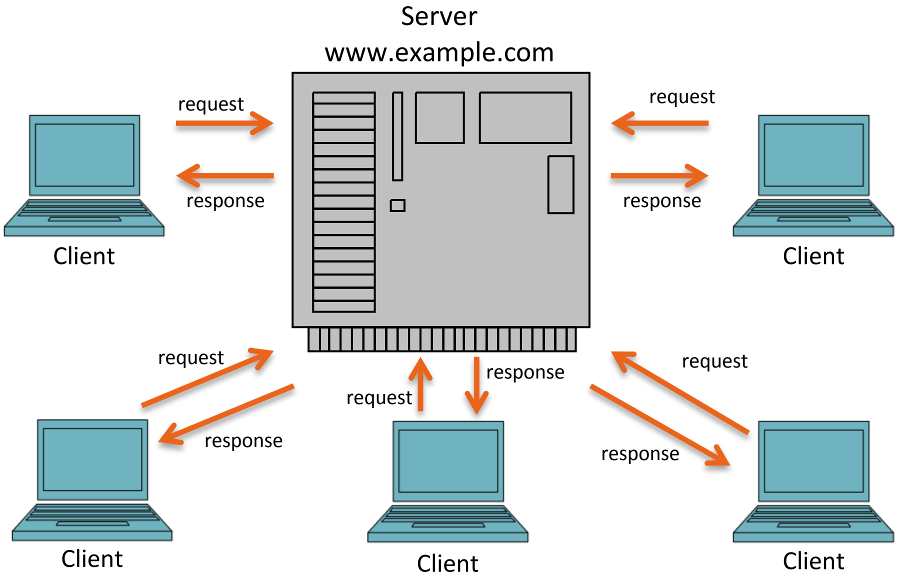


FIGURE 3: CLIENT/SERVER NETWORKING MODEL

3.1.2 Peer-to-Peer model

In a peer-to-peer model there is no central server which handles all connections. Every user station can connect to any other user station. Individual stations can be either a client or a server (send or receive respectively). An advantage of the peer-to-peer model is every station is responsible for its own administration (Elahi & Elahi, 2006). In peer-to-peer networking stations are also known as nodes. See Figure 4 for a depiction of a typical peer-to-peer network model. True peer-to-peer networking is only achievable when a target node is routable (see chapter 3.2.23.2.1). Peer-to-peer connections are typically found in Voice-over-Internet-Protocol applications where one node is able to communicate directly with another node. Such connections are often deemed not to be true peer-to-peer connections as the connection state to both nodes had to be initialized by an appropriate server.

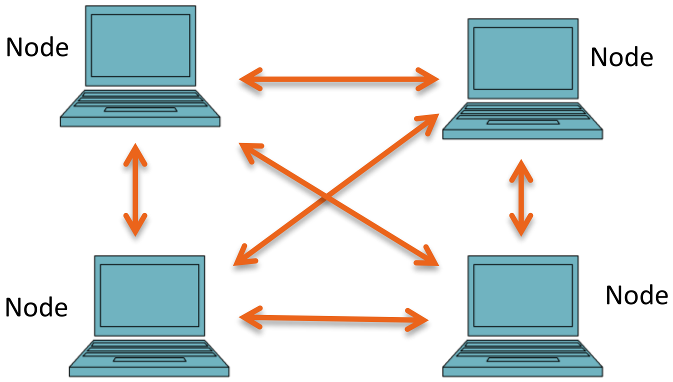


FIGURE 4: PEER-TO-PEER NETWORKING MODEL

3.2 Network Operation and Devices

To set up a network requires devices which facilitates the movement of data packets from one endpoint to another. If these packets are unwanted or do not conform to the required standards they must be blocked. Physical devices perform such functions on networks and provide the advance functionality needed to allow packets to transpire the Internet. Information regarding the devices which perform such functions on the network is readily available and is summarised to provide a background for discussion. Data packets transpiring a network are assigned information which can be referred to by a device which needs to decide what to do with the packet. Information usually included is the recipient's address, sender's address as well as protocols used to encapsulate the packets.

3.2.1 Routers

The first device in question is called a router and provides a wide range of functions. Routers, as the name implies, is responsible for the *routing*, or rather directing, of a packet to its intended recipient through the network to which it is connected. Routers are also used to connected different networks to one another, thus a router can be connected directly to another router. If a router compares the address of data packet to the addresses of all the devices on the connected network and finds no match, the packet is directed to another router. Routers are common to any network and are the principle device for connecting to the Internet.

A Router, connected to multiple other routers and controlling access from other routers to its own network is said to provide gateway functionality. Gateways, as such devices are called, thus allow or reject packets intending to transpire from one network to another. Gateways check for packet integrity and, if needed, transform the packet to standards acceptable by the forthcoming network. Almost all routers include gateway functionality and gateway devices are thus rarely used as a standalone device.

Other functionality provided by routers includes firewalls. Firewalls prevent unwanted access to a network. This is different from a gateway as firewalls block malicious attempts to harm the network or devices on the network.

To allow for scalability of a network routers are provided the functionality of assigning addresses to devices connected to the network on which the router is located. Devices can thus be connected or disconnected from the network without affecting other devices on the network.

3.2.2 Network Address Translation

Network Address Translation (NAT) is a process of modifying a packet’s address to a compatible address format for the intended network. Private networks usually have their own addressing scheme or address range which is compatible only for devices connected to the private network. Once a packet is intended for a device outside the network the packet passes through a router which determines if translation is needed and directs the packet through NAT if needed. The NAT modifies the sender’s address value appropriately. Modification is needed in order for the recipient to be able to send a reply packet to the original sender. A private network connected through a router to the Internet is often identified by a single address on the Internet side of the router.

In Internet terminology an address is often referred to as *routable.* This indicates that a packet sent to that *routable* address over the Internet will be able to reach the address because the address is an

Internet compatible address. Further explanation of compatible addressing schemes is discussed in chapter 3.3.1.

3.2.3 Connection State

The state of a connection between devices is often of interest. Due to a client not having a routable address, servers cannot initiate a connection to a client. A client is thus required to initiate the connection. A connection is automatically initiated if a client sends a packet to a server. Once a client initiates a connection to the server the server has an allotted time period to respond. During this time period routers will allow packets intended for the client and exhibiting the properties of the server to pass through. The time period is determined by various routers along the path to the server. Exact values for the time period vary, but 60 seconds is a generally accepted value (Venter, 2011). If the time period has not yet expired "connection state" is said to be true. To maintain connection state a client often sends packets called *keepalives* to the server allowing the server to communicate with a client when it is required.

Take note that connection state is only applicable for a unique client/server communication session. Another device cannot use the connection state established between a client/server pair to communicate with the client. Routers will block this packet based on it not exhibiting the required properties. If a device, which is not the server, sends packets mimicking the server's properties to the client it is called *spoofing*. Spoofing is a serious threat to systems as such packets are usually sent with malicious intent and measures should be taken to identify and block such packets (compare chapter 5.3.2).

3.3 Internet Protocol Suite

The Internet Protocol Suite describes the architecture of the Internet and the protocols used at different layers of the Internet architecture. The Internet Protocol Suite is often also referred to as the TCP/IP Model, the actual difference in stature and function is trivial for the purpose of this project. The Internet is generally stated as being a network of networks. It enables the communication between two hosts located at different endpoints of a network, or in this case, the Internet. Hosts are the ultimate consumers of communication services and execute processes on behalf of the user (Braden, 1989). A router acting as a gateway to the Internet is typically used to regulate and configure packets designated for Internet transmission. Figure 5 shows the network topology of a communication session involving a packet sent from Host A to Host B.

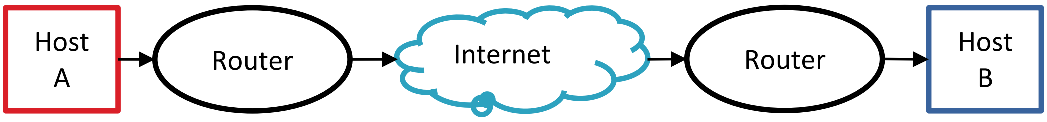


FIGURE 5: COMMUNICATION OVER THE INTERNET, HOST A TO B

As mentioned the layering of the Internet architecture plays a key role in the transmission of packets. The Internet Protocol consists of four layers, known as the Application Layer, Transport Layer, Internet Layer and Link Layer (Braden, 1989). Each layer contains a set of protocols and, depending on the application of the communication between Host A and B; a packet is encapsulated or *wrapped* in an applicable protocol at each ensuing layer. This enables a process executing on Host A to communicate with a process on Host B. Figure 6 graphically represent this process. Layering of the Internet architecture provides for the generalisation of communication thus standardising and reducing the variance in communication sessions between hosts.

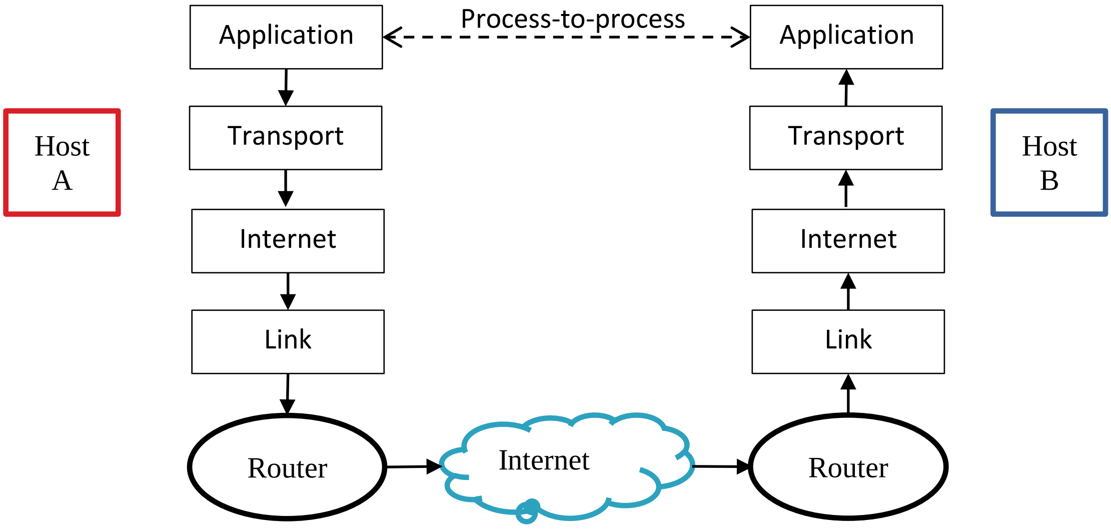


FIGURE 6: PACKET WRAPPING FOR COMMUNICATION OVER THE INTERNET

3.3.1 Internet Suite Protocols

Protocols establish the rules of encapsulation at each layer of the Internet architecture. Several standard protocols for each layer are maintained by the IETF. Protocols are chosen based on the properties required for a communication session between two hosts. The user, using a service on a host, is often unaware of the encapsulation process as the protocol to be used for encapsulation is determined at the development stage of the service. Packets encapsulation is done by adding a protocol specific header (and footer if needed) to the packet passed down from the preceding layer, see Figure 7 for an example of how a packet is encapsulated. Each packet transmitted over the

Internet contains information that informs the recipient which protocol to use at each layer in order to interpret the message. For the goal of this project the Link layer will be automatically configured by the host platform, but protocols used for the Internet, Transport and Application layers need to be specifically configured by the developed telemedicine application and thus requires further discussion.

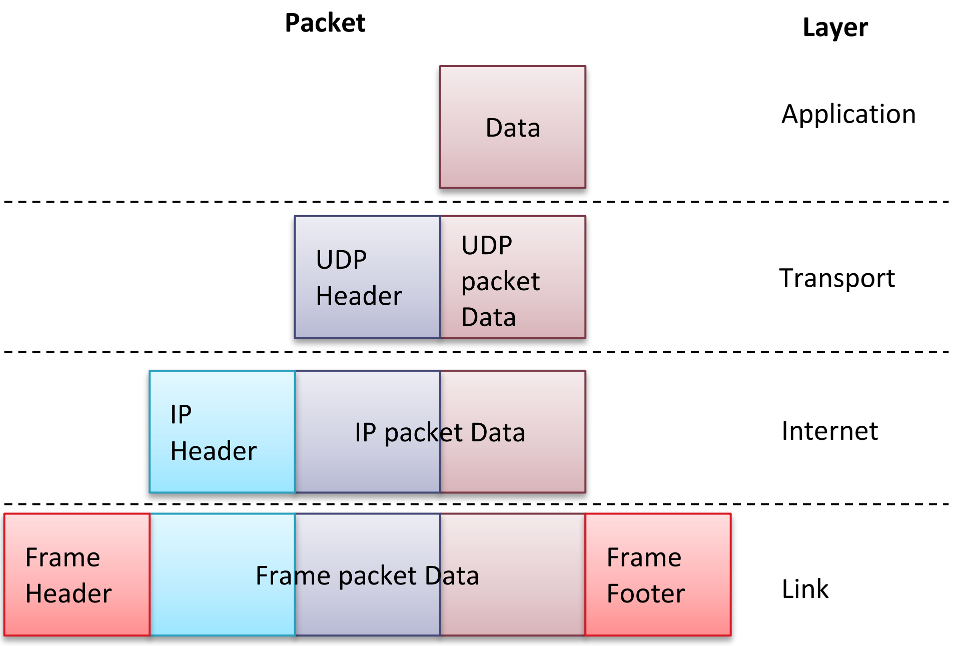


FIGURE 7: INTERNET PROTOCOL SUITE PACKET ENCAPSULATION

Internet Layer Protocol

The protocol of interest in the Internet layer is the Internet Protocol (IP). The Internet Protocol forms the basis of the Internet and is responsible for addressing and fragmentation of packets intended for transmission and received by the network interface (Information Sciences Institute University of Southern California, 1981). The IP is responsible for the encapsulation of all outgoing packets with an applicable header section. This allows other hosts on the network to identify the packet and its intended target. All IP packets are datagrams, that is, there is no guarantee that a packet will arrive at its intended target and no confirmation if it does.

Each IP address is used to uniquely identify a host. Addressing in the Internet Protocol is done using a representation of bits. Two versions of addressing schemas exist, IP version 4 (IPv4) and IP version 6(IPv6). IPv4 contains four bit octets, an octet represented in decimal form as a value between 0 and 255, for example 192.0.2.128. The IPv4 standard, although still the most common form of addressing is slowly being replaced by the newer IPv6 schema, the reason being the limited number of possible addresses IPv4 allows (2^{32}). Devices on a private network are assigned address in the range of 192.168.*x.x* (where *x* is any value between 0 and 255). These addresses are not suitable for the Internet and packets from such address devices needs to be translated by a NAT to be compatible with the Internet.

Due to the limited number of IPv4 addresses and the fact that more devices require an address than available, a host is often represented by a domain name such as www.example.com. This allows for an IP address to be assigned to more than one host at alternating periods. A host is thus typically only assigned an IP address for a limited time period. IP addresses assigned in this manner are called dynamic IPs and are rarely routable (see chapter 3.2.2). Records of which IP address is assigned to domain name (and thus also a host) at a specific time is stored of a Domain Name Server (DNS). DNSs perform the function of comparing the domain name of a request (from a client) to an IP address stored in its memory and directing the request to that address. Domain Name Servers are updated automatically, a discussion of how this is performed is beyond the scope of this document.

Transport Layer Protocols

Transport layers form the connection between Internet layer and services running in the Application layer. Protocols are divided into two subcategories: connection-oriented and connectionless. Connection-oriented protocols form a reliable end-to-end connection between two hosts by acknowledging every packet. It thus rectifies the problem of Internet Protocol datagram non-guarantee delivery. If a packet is not delivered an acknowledge packet (ACK packet) is not received and the packet is resent by the sender (Information Sciences Institute University of Southern California, 1981). The primary protocol used for connection-oriented applications is the Transmission Control Protocol (TCP).

Connectionless protocols provide a non-reliable service as delivery is not guaranteed. In most cases the datagram service of the IP is used directly (Braden, 1989). The primary connectionless protocol is the User Datagram Protocol (UDP). When using a connectionless protocol packets may be duplicated, thus arriving more than once, as well as arrive out of order if more than one packet is sent (Postel, 1980). These properties limit the application of connectionless communication sessions to a session requiring no guaranteed delivery and order of arrival is trivial. The advantage of using a connectionless protocol is the speed at which a session can proceed as no ACK packets are required. Often rectifications to non-guarantee delivery are implemented at the application layer by allowing the service to resend if no custom service acknowledgement was received from the intended recipient.

The Transport layer also introduces the concept of ports. Ports allow multiple channels of communication to a host simultaneously. A service executing on a host is thus assigned its own available port by the host for communication purposes and only that service may communicate on the assigned port. Some port numbers are used for standard Internet applications. These port numbers are deemed restricted by the Internet Assigned Numbers Authority (IANA). Currently ports 1 to 1024 are for restricted use. A communication session running on a specified port to a specified host can be identified by the following combination of IP address and port: 192.0.2.45:3639.

Application Layer Protocols

At the Application layer data created by a user undergoes its first encapsulation. Application layer provides intercommunication between services of processes running on different machines. Although more complex data constructions are allowed as more resources are available to parse data at the Application layer, normal written languages are quite difficult to be interpreted by a computer and thus protocols providing rules for how data should be encoded are still required. Application layer protocols are less restricted than lower levels and any individual or organisation can create or customise a protocol for a specific use. Creating a custom protocol will restrict interoperability between systems but may enhance security and fulfil other needs of the system.

Standard Application layer protocols are maintained by the IETF and most of the protocols are assigned a restricted port number from the Transport layer. The most widely used protocol is the Hyper Text Transfer Protocol (HTTP) and utilises port 80. HTTP is used to transfer text files, such as web pages, over the Internet and it is deemed standard for smart devices to be able to communicate using HTTP (Venter, 2011).

In conclusion, communication to a host requires standardisation at different abstraction layers using a protocol serving the purposes of the application most appropriately. Packets intended to a host are often identified in the following format: http://www.example.com:80. Where “http” specifies the Application layer protocol, “:80” specifies the port at the intended recipient (this is generally omitted as HTTP is assigned port 80 by default) and “www.example.com” represents the domain name which is linked to a varying IP address. Not specified in this identification is the Transport layer protocol used, this is because HTTP uses TCP by convention.

3.4 OSI Reference Model

The Open Systems Interconnection Reference Model is proposed by the ISO to facilitate network communication between devices. It servers the same purpose as the Internet Protocol Suite with the added goal of implementation on any network, not just IP-based networks. If a device complies with the ISO standard it should able to communicate with any other ISO compliant device. The model specifies seven layers for a network which is used to develop custom networks and implementation provides a structured environment in which the network operates. An open system is a set of protocols which allows effective communication between two devices regardless of their design, manufacturer or other properties (Elahi & Elahi, 2006). The seven layers from the bottom up are Physical layer, Data Link layer, Network layer, Transport layer, Session layer, Presentation layer and the Application layer. HL7 based their standardisation procedure on the seven layers of the OSI Reference Model (Health Level Seven, [s.a]). The OSI Model is often compared to the TCP/IP model although situational requirements for implementation are different.

3.5 Cellular Networks

Cellular networks are an extension of wireless communication networks and are primarily used for the interconnection of mobile devices. The widespread use of mobile devices for communications provides an opportunity in terms of data collection and monitoring as well as providing alerts to designated persons if needed. The increase in capability of cellular networks, especially the addition of General Packet Radio Service (GPRS) and development of Third generation (3G) standards has led to development of applications suitable for deployment on mobile devices. Through the use of GPRS and 3G technologies these devices are able to communicate over the Internet.

If a mobile device starts a session to communicate over the Internet the cellular network operator assigns the device a temporary IP address. This address allows the device to seamlessly integrate with the Internet providing the correct protocols are used.

Due to the high security risk associated with Internet connectivity incoming connections to a cellular device is blocked. When initiating a connection to the Internet a mobile device uses an Access Point Name (APN) to inform the cellular network what type of connection is to be initiated. This includes what IP address is assigned to the mobile device, what security parameters apply to the connection and how the connection is to operate (Digi, 2006). APN names may be used for advanced functionality to connect to other networks than the standard connection provided by the cellular network.

4 Evaluation of the eXtensible Automation Protocol

The eXtensible Automation Protocol (xAP) is proposed to be used in conjunction with the telemedicine data distribution network as a standardisation principle. Careful evaluation of the protocol is required for its suitability and implementation in a practical environment. The protocol described in this chapter is as outlined on the xAP group web site, www.xapautomation.org.

4.1 Protocol Definition

4.1.1 Introduction to xAP

xAP is an open standard and can be implemented by any knowledgeable individual or organisation for any purpose. It is originally intended for the purpose of home automation. The primary design objectives, as stated by the developers (Lidstone et al., 2002) are:

- Minimalist, elegant and simple, easy to implement/retrofit
- Suitable for use with a wide range of processing capabilities, from embedded controllers to fully fledged PC's
- Operating system agnostic
- Programming language agnostic
- Network agnostic

The objectives provide for an attractive architecture when implemented on a telemedicine network, especially considering implementation in African context, where connectivity is unreliable and data transfer is expensive.

Current implementation focuses on IP based networks (see chapter 3.3.1); although other network types such as RS232, RS485 serial and wireless networks are also supported (Lidstone et al., 2002) . xAP operates based on broadcasting, that is, a device *pushes* a packet at the network where it is received by all willing devices. Willing, in this case, is defined by a device choosing whether to accept a particular packet from a particular host or having other properties of interest to the recipient device. Although packets directed at a single recipient are discouraged in the xAP definition, broadcasting is not suitable for large IP networks and is discussed in chapter 5.2.1.

xAP enabled applications (xAPp) are responsible for the sending, receiving and processing of xAP data packets. A xAPp can be implemented on any supporting platform in a structure which is acceptable on the specific platform. A xAPp is represented on the application layer of the network and utilises the appropriate native platform API's to communicate with other devices on the network. Even though a xAPp is not strictly required for a xAP-based network to be operational, xAPps provide an easy way to monitor a network of interest.

4.1.2 xAP on Ethernet networks

Primary application of this project relates to the implementation of the xAP protocol on Ethernet networks, including and extending to wireless radio networks. Ethernet compatibility is thus of key concern. For the distribution of xAP packets on a network the xAP group has been allocated a dedicated UDP port, port 3639, for use on Ethernet networks. The port number was assigned by the Internet Assignment Numbers Authority (IANA). xAP packets require no further encapsulation in the application layer in order to be distributed on the network. Computers running multiple xAPp's require a hub to distribute traffic. Hubs are responsible for receiving and sending packets on the network side and distributing the packet to the applicable xAPps running on the same platform as the hub. Hubs are needed because only one application can communicate on a designated network port at a time; this is due to the workings of the API, see chapter 5.4.1**Error! Reference source not found.**.

4.1.3 Packet Structure

Irrespective on any additional network related packet wrapping a xAP packet always consists of a:

- header section
- message body

These are each configured out of lines of text. The message body may consist of multiple message blocks, but care should be taken not to exceed the MTU (see chapter 3). See Figure 8 for a graphical representation of a xAP packet.

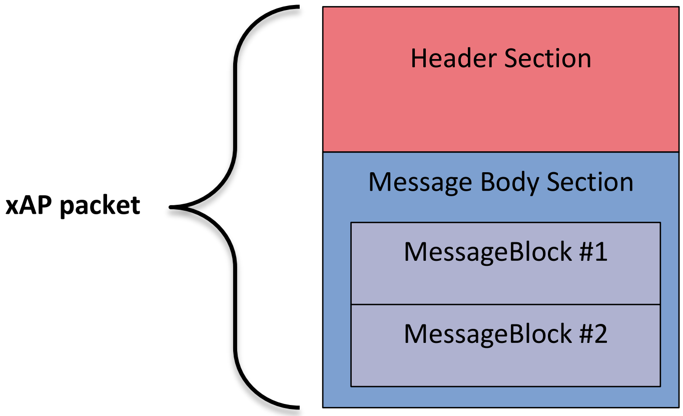
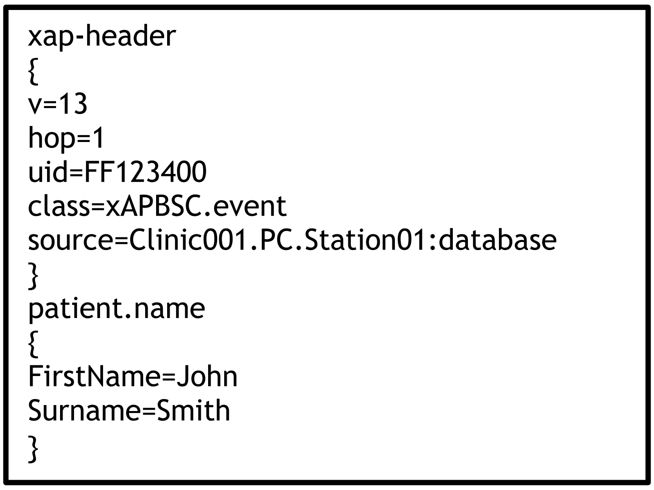


FIGURE 8: XAP PACKET REPRESENTATION

Each section of the xAP packet starts with a keyword line, followed by the section contents enclosed with curly braces { }. Termination of a line is done with a <LF> (linefeed, ASCII character 10 decimal). Every message block consists of a name-value pair, such as “status=off”, which indicates that the “status” variable of a device is currently described by the term “off”. See Figure 9 for a typical xAP packet. Notice how every block (enclosed with { }) is preceded by a header.



```
xap-header
{
v=13
hop=1
uid=FF123400
class=xAPBSC.event
source=Clinic001.PC.Station01:database
}
patient.name
{
FirstName=John
Surname=Smith
}
```

FIGURE 9: TYPICAL XAP PACKET

Packet header

The packet header is used to identify and describe the xAP packet. A keyword line introduces the header with the contents as usual enclosed in curly braces. Characters allowed for use in keywords are alpha-numeric characters, _ (underscore), - (dash) and embedded _ (space). Keywords are not case sensitive. No leading or trailing white spaces are allowed.

It is strongly recommended that the following information is contained in the header (see Figure 9):

- The schema version used.
- Hop counts, used to indicate the number of checkpoints passed.
- Unique identifier, which is a hexadecimal number, is used to identify a particular device. It is in the form nn dd dd ss. The first two digits is FF by default, the 2nd and 3rd pair is used to uniquely identify the device and the last pair is used to define and endpoint.
- I applied a class, consisting of a class name and a class type, to tell the recipient what type of schema (see chapter 4.2) is expected and additionally a subclass indicated by class type.
- The source address, used for filtering received messages or directing a message to a specific device.

Additional information that can be included based on a specific implementation or application:

- A targeted device indicated by “target=AVendor.Adevice.AnInstance:AnEndpoint”

Message grammar

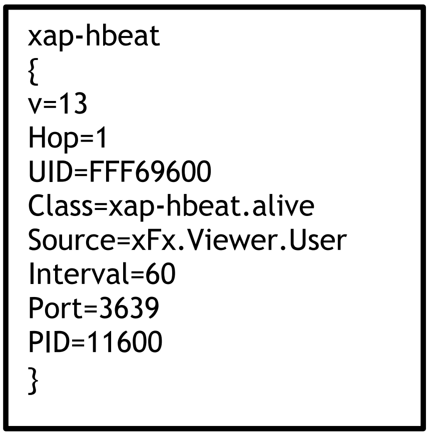
The actual message follows after the header section and is represented in one or more message blocks. Again each message block is introduced by a keyword line. The same range of allowed characters for the packet header keyword line applies for the message block keyword line. The contents of a message block are enclosed in curly braces.

The following properties apply to messages:

- A message consists of name-value pairs.
- Name-value pairs can appear in any order.
- The value part of a name-value pair is considered case sensitive and literal.
- White space contained within the value is significant and affects the output.
- An “=” sign between the name and value indicates the value is encoded as an ASCII string
- A “!” sign indicates the value is encoded as an ASCII hex representation. Hex representation is discouraged as the meaning might become opaque and is rarely used (Lidstone et al., 2002).

4.1.4 xAP Heartbeats

Heartbeat packets are used on a network to monitor the health of device and create an alert if a device does not produce the required heartbeat. Heartbeats by a xAP enabled device or xAPp are a form of a *keepalive* (see chapter 3.2.3). A specific structure exists for a heartbeat packet as shown in Figure 10.



```
xap-hbeat
{
v=13
Hop=1
UID=FFF69600
Class=xap-hbeat.alive
Source=xFx.Viewer.User
Interval=60
Port=3639
PID=11600
}
```

FIGURE 10: XAP HEARTBEAT PACKET

The heartbeat header is identified by the “xap-hbeat” keyword line. In addition to the standard header structure discussed, the heartbeat header also includes an interval indicator and, optionally, the port number and process identification number (PID). The “interval” value is in seconds and indicates the elapsed time between heartbeats from the source. The “port” value indicates the port on which the source is ready to receive xAP data, the default value being 3639. Lastly the “PID” value can have one or both of the following attributes: <ip_address>, <process number>.

4.1.5 Address Wildcarding

Wildcarding enables a message to be sent or received by several targets with similar property values of interest. Wildcarding can be implemented in the header or in the body section of a xAP packet. The “source” or “target” fields are usually wildcarded. The character “*” indicates that any value may be substituted for a field. A “>” character indicates that all subsequent fields are matched.

Given devices identified by address a.b.c.d and a.b.h.d. If the intended target device(s) of a packet is a.b.*.d, both devices will receive the message. Similarly, both devices will respond to a packet addressed to a.b.>, but only the first device will receive a packet identified by a.*.c.d.

4.2 xAP Standard Schema

Schemas are used to provide a recipient with the information needed to be able to process a packet. A schema tells the recipient how the xAP packet will be represented. Several standard schemas exist which is maintained by the xAP group. These standard schemas are intended for interoperability of networks. As mentioned previously a schema applied to a given message is identified by the “class” keyword in the packet header.

Schema typically identifies the collection of message blocks to be expected in a packet. It also determines the prevalent semantics to be used in a message of schema type <Aclass.Aclasstype>. Standard schemas approved by the xAP group are identified as starting with “xAP”. For example the Basic Status and Control schema is identified by “xAPBSC.*”

4.3 xAP Framework for implementation on a platform

The xAP group provided several frameworks for development purposes. These frameworks provide packaged contents for a specific programming language. Frameworks typically represent a library for xAP functions such as encoding information in the “Basic Status and control” schema. Such frameworks provide a basis for developers, lessening the amount of code that needs to be created in order to represent a message in xAP format; it also serves the purpose of standardisation among different xAP applications by reducing variability.

Frameworks provided include:

- .Net frameworks for C# and VB.Net development
- Visual Basic and Webserver framework
- Visual Basic Active-X control framework
- C libraries
- Java software development kit (SDK) for xAP
- Python scripting engine for xAP
- Perl frameworks

This project will utilise the Java SDK for xAP, reason being the cross-platform compatibility of java and the readily available knowledge for java development.

4.4 Related Protocols

There is two protocols with which xAP is often compared. The first is a similar lightweight version of the protocol, named xPL. xPL shares its origins with xAP and applications usually support both of the protocols (Openremote, 2009). The second protocol is the universal Plug-and-Play protocol (UPnP). The UPnP protocol is aimed at easing device discovery on an IP network (Sherwin, 2009).

4.4.1 xPL protocol

The xPL project has similar design goals to that of xAP. It is also intended for use in home automation. It is considered a simpler protocol to xAP (Lowe & Tofts, 2011). The structure of an xPL packet consists of a header block and one message block. Only three packet schemas are allowed, namely *xpl-cmnd*, *xpl-stat* or *xpl-trig*. Similar grammar restrictions, compared to xAP packets, apply to both the header and message section. xPL utilises an xPLHal server running on a computer to monitor all devices.

xPLHal

xPLHal is a service type application, implying that it runs passively on a computer. xPLHal server can be interacted with using another application called a manager which is not necessarily executing on the same computer. The xPL project provides xPLHal Manager as its default application to manage the xPLHal service. The xPLHal service is managed using the xPLHal control protocol (xHCP) on TCP port 3865 (Bent et al., 2007).

xPL on Ethernet Networks

xPL implementation on an Ethernet network is similar to that of xAP. A hub is also required to distribute incoming xPL packages to all clients running on the same computer (see chapter 5.4.1).

xPL is allocated UDP and TCP port 3865 by the IANA. No further application layer encapsulation is required to transmit an xPL packages on the network.

4.4.2 Universal Plug-and-Play

Universal Plug-and-Play (UPnP) is a collection of network protocols to ease the discovery and control of devices connected to a network (UPnP Forum, 2011). UPnP devices can under most circumstances only operate on IP network. UPnP protocols use the extensible mark-up language (XML) to communicate. The UPnP Forum is responsible for maintaining the standard which is supported and actively promoted by the Digital Living Network Alliance (DLNA) (Sherwin, 2009), the DLNA being an alliance of electronic equipment manufacturers such as Sony and Microsoft.

UPnP Devices

When an UPnP enabled device is connected to a network the first step is to discover other devices on the network which could provide it with a relevant IP address. In the case where none is found the device assigns itself an IP address. Once a device is "discovered" it broadcasts its capabilities to the rest of the network in XML. Subsequent interactions are also done in XML. XML describes a set of rules for encoding documents to be readable by capable devices. XML is maintained by the World Wide Web Consortium (W3C) and is considered an open language. The goals of XML as expressed by the W3C are many; shortly summarised for the context of this project the following goals (Bray et al., 2008):

- XML shall be straightforwardly usable over the Internet.
- XML shall support a wide variety of applications.
- It shall be easy to write programs which process XML documents.

Due to the variety of goals and widely used nature of XML it has become quite extensive and thus proves difficult to parse by low-level devices.

4.4.3 Comparison of Protocols

The standards for communication discussed in this chapter each serve a purpose deemed important by the supporting group. To determine which protocols is best suited for the application of this project the standards must be evaluated against the project objectives. Although xAP is currently deemed the standard to be used for the telemedicine network, comparison of similar standards can help improve implementation of the standard. Table 1 compares the three standards for the fields of interest.

xAP still proves to poses the most modifiable structure. The modifiable structure allows for the creation of telemedicine application specific schemas. These schemas are not currently documented and a full evaluation and creation of a library describing of schemas is required for implementation. Although UPnP is considerably better supported than the other standards, the complexity of parsing XML, considering low-level devices, and the nature of the protocols deem it unsuitable for the primary implementation of a telemedicine network. xPL is gaining support due to its simple structure but the limited application scope is unsuited to the purposes of this project.

.

TABLE 1: COMPARISON OF STANDARDISATION PRINCIPLES AND PROTOCOLS SIMILAR TO XAP

Field of Interest	xAP	xPL	UPnP
Network Agnostic	Yes	Yes	No, only suited for IP based networks
Industry Support	Weak	Weak	Strong, supported by the DLNA consisting of more than 245 members. Built into Microsoft Windows.*
Native device capability	Suitable for almost all devices from low-level to fully fledged	Suitable for almost all devices from low-level to fully fledged	Only suitable for device capable of parsing XML
Consumer Acceptance	Low, user has to integrate network by developing own software or using limited existing software	Low, user has to integrate network by developing own software or using limited existing software	High with more than 440 million UPnP enabled devices sold. **
Packet Language	Native xAP format	Native xPL format	XML
Operation Layer (IP based networks)	Application Layer	Application Layer	Primarily layers between the Transport Layer and Application Layer
Objective of standard	Home Automation	Home Automation	Device discovery and control over network
Adaptability	High, custom schemas can easily be created	Medium, only three schema types are used and one message block allowed	Medium, devices must be compatible with other UPnP enabled devices and source code cannot be modified
Inherent capability to transfer data	High, due to highly modifiable schemas. Limited by MTU	Low, standard schemas. Size limited by MTU	Dependant on XML and compatibility with other devices
Telemetry capability	Medium, standard schemas exist which allows for implementation by user	Low, no support within standard schemas	High, numerous supported devices

* (Anonymous, 2011)

** (Sherwin, 2009)

5 Developing a Network Architecture

To create a telemedicine network capable of achieving the desired objectives, technologies have to be integrated seamlessly at alternating levels of complexity. The possibilities of available and suitable technology for the use in telemedicine networks are endless and thus provide a challenge. Scalable architecture is thus essential. Complexity of a network tends to increases with the size of the network as well as differing platforms. A network consisting of hundreds of stations to manage requires more effective management than a single station. The configuration designed strives to accommodate such complexity and provide a structure for future similar projects. Following the proposed network structure, the relevant equipment is then selected to aid in testing core feasibility.

Figure 11 show the methodology followed to design a network structure. An iterative process was follow to develop a feasible system for integration with existing technologies. Factors such as the Internet structure, application development for client and server interfaces and router compatibility played a key role in the design process.

As demonstration of compatibility, xAP encoded packets (generated by the developed applications) are sent from host to host over the Internet. To provide comprehension of the challenge in developing a xAP-friendly platform the development process and factors influencing development is elaborated upon.

Potential security risks are briefly discussed in order provide a frame of reference for the limitations placed upon the development of a network. The sources of these limitations are typically external to the project and a part of normal Internet architecture. Existing alternative solutions are used as far as possible to provide network feasibility.

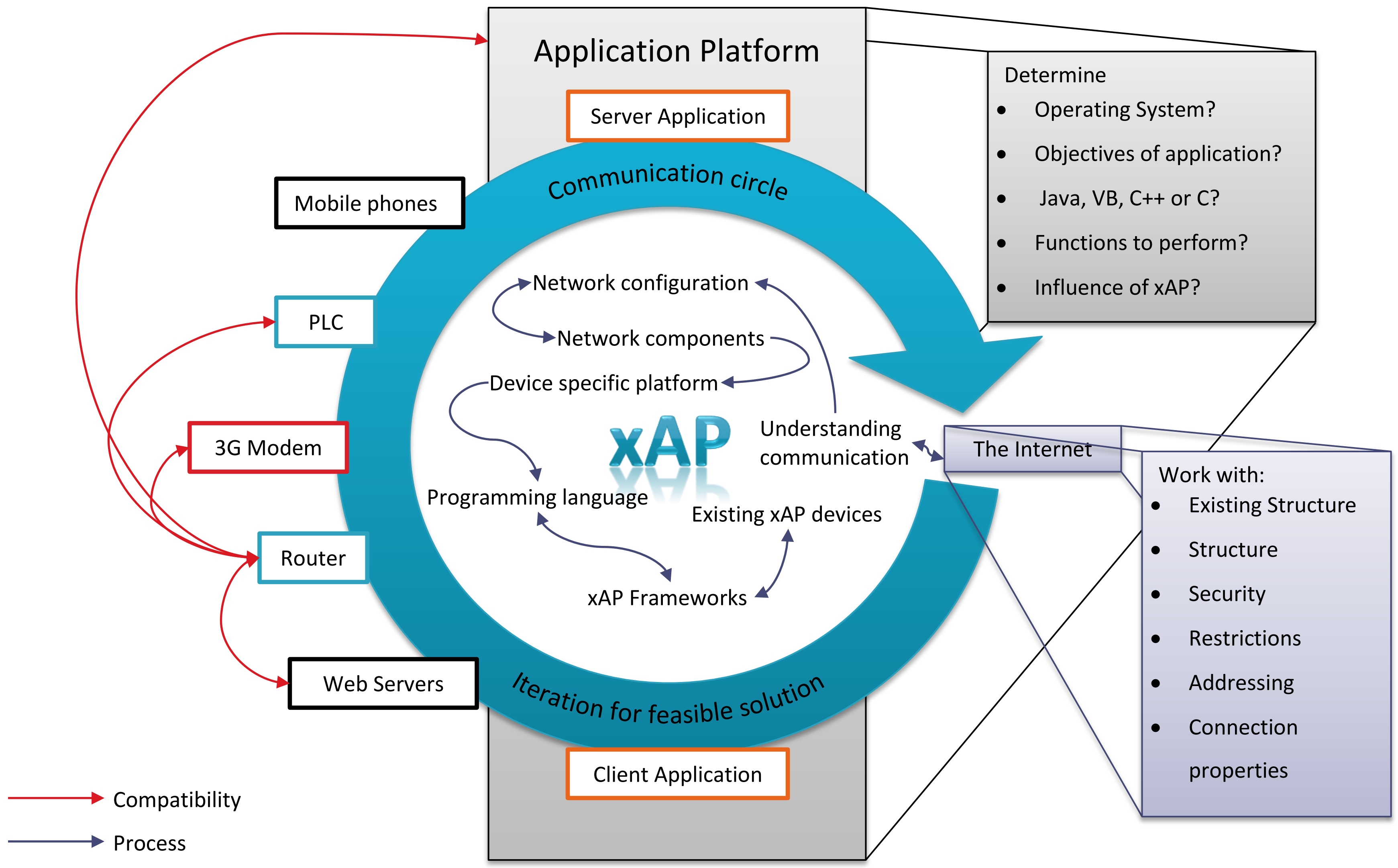


FIGURE 11: METHODOLOGY ITERATION PROCESS

5.1 Components within the Telemedicine Network

To assist in the development of a representative architecture, components are categorised to represent a typical information network. Categorisation validates the testing of selected components according to a real world implementation. Categories are loosely defined for devices having similar properties. Sub categories can be created within a category. Sub categories are specialisations of general categories. For the current applications, it is suggested to restrict categorisation to general categories with one layer of specialisation. The suggested categorisation is discussed in Table 2: Categorisation of TELEMEDICINE NETWORK components.

Selected equipment for conducting a feasibility test does not cover all categories according. This is due to the extensive development required to be able to test xAP-related operations on specific devices and a general lack of device-specific frameworks.

5.1.1 Equipment used to test core feasibility

Devices used to conduct the test were selected from available technologies with the specific intention of providing functionality suited to the testing environment without sacrificing real world applicability.

Low –level Device

To represent a low-level device the Barix Barionet 100 was used. The Barionet is equipped with a NIC as well as several input and output ports suited to a variety of applications. Refer to Appendix A.1 for a full list of the available ports. Being a programmable logic controller (PLC) the Barionet 100 provides a modifiable web interface as well as a specific development environment which can be uploaded to the device. This specific Barionet 100 is programmed to be compatible with xAP.

Router

A router is required to connect to the Internet. A TP-Link model MR-3420 is used for this specific application. See Appendix A.2 for more details. The router provides connection to wireless as well as Ethernet connected devices. Additional features include the capability to connect a 3G modem compatible with cellular networks which is the configuration used throughout the tests.

Station

A station is represents the functionality of being able to communicate with a server and other devices on the network. To create such functionality a personal computer with Microsoft Windows 7 operating system is used as platform. The actual functionality is implemented through Java application code for the respective platform, as is discussed later. Stations are also referred to as clients.

TABLE 2: CATEGORISATION OF TELEMEDICINE NETWORK COMPONENTS

General Category	Description of general category	Specialised category	Description of specialised category
Low-Level device	A device with no hard drive. Examples: Sensors or PLC's	Low-level device with no NIC	A device that must be connected to station or any other device with a NIC. Example: temperature sensor
		Low-level device with NIC	A device capable of connecting to the network without a facilitating station. Example: Barix Barionet 100 PLC
Router	See chapter 3.2.1	N/A	N/A
Station	A device with more advanced capabilities, such as telemetric data collection and processing. Example: computer	Stationary Station	A station that is not mobile and connected with a Ethernet cable to a server
		Mobile Station	A station connecting via wireless connection. This excludes mobile devices intended for use on a cellular network
Server	A device capable of handling requests, stations and other servers. Facilitates network creation, communication as well as data storage	Facility Server	A server with one or several stations connected to it. One or more other servers can also be interconnected with the server. The facility server typically manages a facility network.
		Web Server	A server with one or more facility servers connected to it. This category of servers does not relate to geographical location.
Cellular device	A device primarily connected to a cellular network	Cellular Phone	A standard phone operating on a cellular network.
		Modem	A device intended to connect a station, low-level device, server or router to the Internet via a cellular network.

Server

Server functionality is achieved through the same process as with the station, the difference being in how the Java code is applied. A discussion of the complete process is done later.

Cellular device

In the cellular device category two specialisations exist. Only a modem was used for the tests. As stated previously the modem allowed for Internet connectivity through the respective cellular network. A Huawei K3770 modem, see Appendix A.3, operating on the Vodacom cellular network is used. 3G modems use an APN (refer chapter 3.5) to indicate to the network what type of authorisation is requested.

In this case both the “internet” and “unrestricted” APN was used. The “internet” APN is for standard connections and all Vodacom enabled cellular devices use these APN’s. IP addresses provided for devices using the “internet” APN are not routable and thus have limited functionality.

In order to use the “unrestricted” APN, the operator needs to enable the device for such use. Once the APN is activated, incoming and outgoing communication is not restricted it terms of being routable and the type of connections allowed. Typical situations required for the server to be connected to the Internet through the “unrestricted” APN and the client using the standard “internet” APN.

The compatibility of cellular phones with the xAP-enabled network was not tested. This is due to no existing xAP-enabled framework with which to conduct such tests. To develop a compatible framework requires device-specific development as operating systems for cell phones differ, with no standard development language.

5.1.2 Network Configuration

The test configuration is chosen based on a robust representation of real world applications. The basic configuration is shown in Figure 12. The server connects to one of the routers by using a wireless connection. The router utilises the connectivity provided by the cellular modem to connect to the Internet. Packets can then be sent or received by the server through the Internet. The red waves indicate packets sent over a cellular network while blue waves indicate normal wireless transmission.

A low-level device which is xAP enabled, was connected to the same private network as the server to test compatibility with the server. Communication between clients and servers are tested using a variety of connections from the client side. As shown in Figure 12 a client connected through a router-cellular modem combination was used in tests as well as a client using the services of a commercial Internet service provider (ISP). This configuration provides a basis to test:

- Compatibility with existing xAP tools
- Reliability of the connection
- Required interval for "heartbeat" packets
- Ability of stations to communicate with one another over the Internet

Configuration should support the ideal of data always being available on request. This implies that stations and servers should be able to store data locally as well as keep track of where other data of interest might be found. The storage and tracking of data involves the creation of a database, which is beyond the scope of this project, but functionality to support such future implementations should be accommodated.

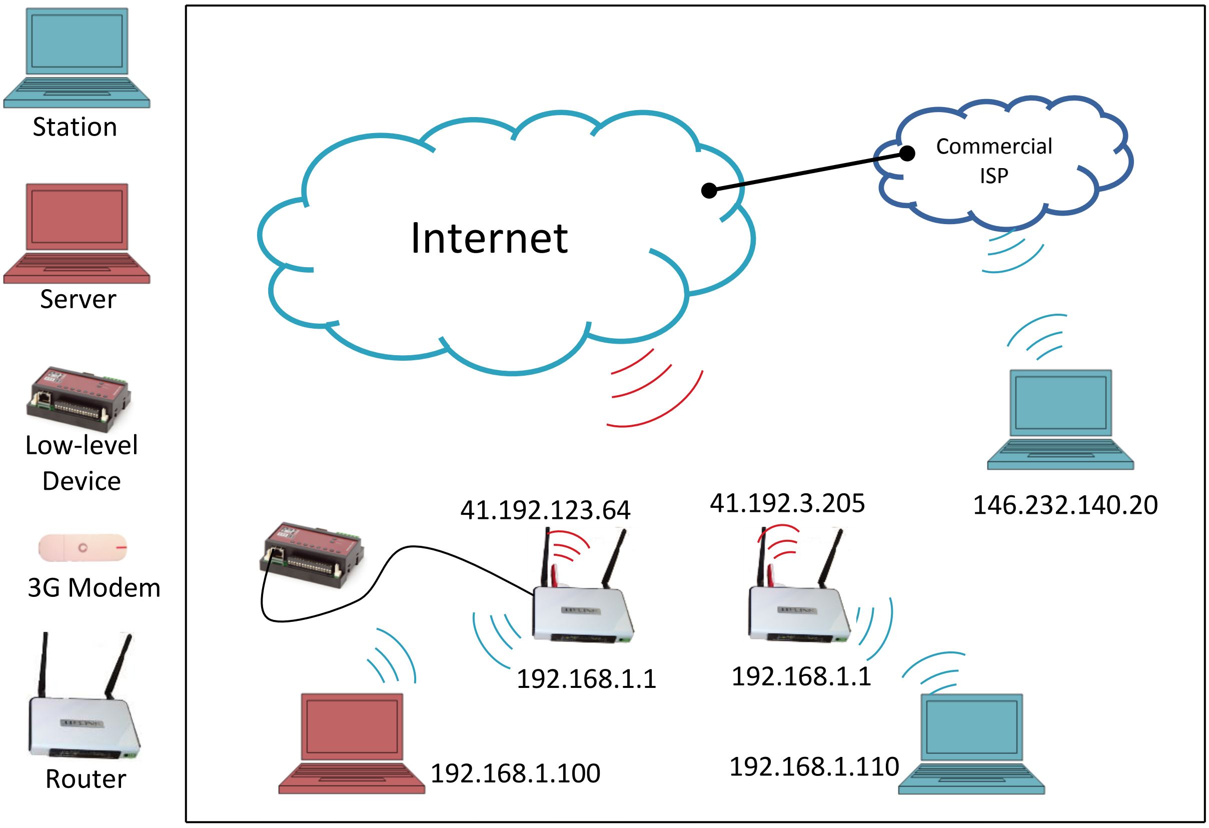


FIGURE 12: CONFIGUARATION TO TEST THE XAP NETWORK

5.2 Assigning Addresses in the Network

Current system functionality allows for IP addresses to be assigned automatically. Assigning of addresses is discussed with reference to Figure 12. Routers assign all devices connected on the private network an IP address value, depending on setup parameters, between 192.168.1.100 and 192.168.1.255. The router itself is assigned the IP address of 192.168.1.1 by default. 192.168.1.1 is the IP address by which devices on the LAN will recognize the router. If the system is looked at from the Internet viewpoint (as opposed to the LAN viewpoint), routers, and by reference the private network connected to it, is assigned an Internet compatible IP address by the service provider. A router is thus assigned two addresses, one for inside the network and one for outside. The methodology used by ISPs to assign IP addresses to a router endpoint can vary and does not necessarily follow the process as described. Depending on the connection type the service provider can either be a cellular network operator or a commercial ISP.

5.2.1 Broadcasting

For broadcasting purposes the IP address 255.255.255.255 is used. If a packet is sent to this address the router administering the LAN sends the packet to all the devices on the network. Almost all routers are configured to prevent broadcasted packets from transpiring networks and some even prohibits broadcasting on the LAN (Venter, 2011). Broadcasting packets on large complex networks with multiple routers connected to the network can cause the network to become futile due to all the network resource being consumed to transport the packets. Broadcasting should thus only be used on small, simple networks and only if needed.

5.2.2 Routing to an Endpoint

As mentioned not all addresses are routable over the Internet, either due to ISPs blocking access to that address or because the IP address is dynamically assigned (see chapter 3.3.1). To overcome lack of this, a domain name is given or a special routable IP addresses is assigned to a device (or device representing a network). This allows a packet to be sent to that specific address. Depending on the router's settings (representing a network), the router may decide to discard the incoming packet due to no connection being established from the inside of the LAN to the sender of the incoming packet. This is a general security measure to prevent unsolicited access to a network.

To prevent an incoming packet being blocked *port forwarding* has to be enabled on the router. Port forwarding requires three variables to be set up. The first is the IP address (representing a device) to which the packet has to be *forwarded* on the inside. As in Figure 12 the server has an IP address of 192.168.1.100 and this is the IP address which could be used. The second field is the port to which packets should be forwarded. Only packets addressed to the port (or port range) specified will be *forwarded* to the device. The device will have to be waiting for a packet on the specified port otherwise the packet will be lost. The last field is the Transport protocol for which *port forwarding* should be enabled, either TCP or UDP. Some router manufacturers call *port forwarding* “virtual servers”, as is the case with the TP-Link routers. Client devices do not require *port forwarding,* as they initiate the connection.

If it is required to send a packet to a client without a connection being established by the client, an external service has to be used. Of course this is not applicable to clients connected to a routable address. These external services have two components. The first executes on the client computer and at specified intervals sends an update to a server. The server is the second component. It receives and update from the first part of the service and extracts the IP address of the client from the packet. It then assigns a registered domain name (see chapter 3.3.1) to the IP address. If another device then sends a packet to the domain name, the packet is forwarded to the IP address of the client. DynDNS is an organisation offering such services (Dyn, [s.a]). It allows for the registration of one free domain name after which a fee is charged. Applying this service requires *port forwarding* to be set up on the client’s side with the appropriate variables. A service such as DynDNS cannot be applied to a low-level device if it is the only device on a network as a part of the service must execute on a platform inside the network.

5.3 Security Considerations

Security Considerations for a network is of paramount importance especially in the healthcare industry and requires extensive research to provide feasible solutions for security problems. It is unethical to be callous with patient medical information and legislation prevents unauthorised access or distribution. Although actual security implementations are beyond the scope of this document, all networks have inherent security measures in place to prevent unsolicited access to a system. This is especially true for a network connected to the Internet. When designing system’s architecture, security considerations must be accommodated in order to prevent deteriorating existing security parameters. Three areas of security are of importance; packet security, database security and network security. If security is breached in one of these areas it may influences the integrity of all other areas. Network security risks are typically the largest threat to ISPs as it can influence all their other operations as well. ISPs thus typically implement their own measures to minimize the risk. Database Security is of no concern in this project and will not be discussed

5.3.1 Packet security

Packet security considers the actual packet being sent over the network. A packet may be intercepted along the path to its intended host, resulting in the information becoming available to persons not privileged to it. Packet security is implemented by encrypting the contents of the packet providing only the recipient with the tools to decrypt the contents.

5.3.2 Network Security

Security related to the network involves simply applying parameters to prevent unsolicited access through the network interface. UDP packets are known to cause security problems. A UDP packet can be imitated and thus misrepresented by an attacker. This allows malicious packets to enter through the network firewall damaging the unprotected local network (Srisuresh & Holdrege, 1999). Another threat is due to UDP requiring no ACK packet (see Chapter 3.3.1) a host can flood an unsuspecting recipient host with UDP packets. This is a notable security risk and if left unchecked could essentially consume the bandwidth allocated to the recipient and prevent the recipient from performing any other network related operations, as all network resources will be occupied dealing with the flood of packets. ISPs thus frequently limit UDP operations by either blocking packets intended for non-standard ports or blocking packet with contents not conforming to an acceptable standard. This project determines in general to what extent ISPs limit such functionality and proposes an alternative in the conclusion for cases where limitations are applied.

5.4 Application Platform

Several factors influence the selection of platform to develop xAP applications. As discussed in chapter 4.3 several frameworks are presented for xAP development purposes. These frameworks require knowledge of and compatibility with a provided platform in order to develop applications using the framework. In order to create functions able to test the network configuration as required by the objectives it was decided to create applications without the use of the framework, allowing for more modifiability. Using the tools discussed below two applications was developed, each representing the functionality of a station (client) and server respectively.

Computers with Microsoft Windows 7 operating systems are readily available and development of an application is done using this operating system. To accommodate the limited compatibility of applications developed on this operating system Java, as an operating system agnostic programming language, was used to develop the application. Java executable files contain all the libraries required to execute the files in on supported platforms, unlike other development platforms. Java *.class files needs to be compiled for a specific platforms (Windows, Mac OSX, Linux, etc.).This allows any system equipped with the required Java tools to execute the developed application (Oracle, [s.a]). To facilitate development of Java applications, an Integrated Development Environment (IDE) is used. For this project Netbeans IDE was used. The applications as well as the code for the applications can be found in Appendix B.

5.4.1 Application Programming Interface

An Application Programming Interface (API) provides an interface for different applications to communicate with each other. All operating systems come with standard APIs implemented to help with the development of applications. If an application utilises a specific API, it utilises the libraries, functions or application with the API represents.

To send or receive a packet over a network from an operating the sockets API of an operating system is used. A socket is thus an endpoint in a communication session. Sockets API perform further encapsulation or unpacking according to the Internet Protocol Suite (see chapter 3.3) as required by the parameters passed to the socket API. Applications either sends a packet to the socket, which in turn sends the packet out over the network via a NIC, or checks the socket for any packets that have been received from the network (Oracle, [s.a]). In the case of a packet being sent over the network the socket requires the IP address of the recipient and the port on which the recipient is expecting the packet. For receiving a packet the socket expects a port number variable on which to check if a packet has been received.

The sockets API provide the basis on which an application able to communicate over a network operates. The java.net library implements the functionality of sockets by providing the developer with the necessary functions.

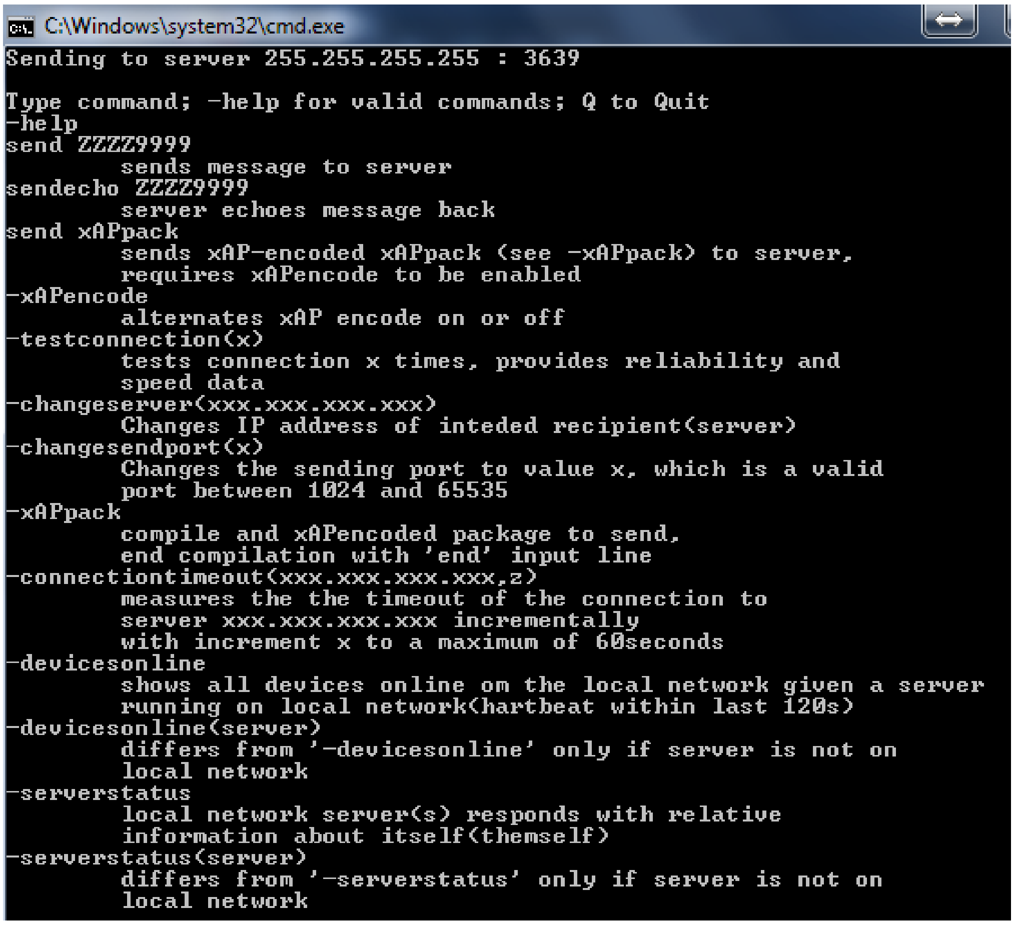
5.4.2 Implementing xAP

In order to implement xAP formatting a function is created which receives the data and the schema as input and then applies the formatting required to comply with the specified schema to the data. The function thus encapsulates the data in xAP format.

The “source” field in the header of the xAP packet indicates the origin of a packet and is assigned either a “server” or “client” value concatenated with the name of the host computer. The recipient uses the “source” and “class” values to determine how a packet should be handled.

5.4.3 Client Application

The Client application is intended to execute on a station and is tasked with initiating the connection to the server. A client instance provides the user with functions which will test the network configuration. It implements the xAP format when needed. By typing the “-help” command a list of available functions and commands is given as shown in Figure 13. Client applications are equipped with a heartbeat at intervals of 60 seconds which is sent to the current server. The functions used to test the network are discussed in greater detail below. For information regarding the other commands and functions refer to Appendix B.



```
Sending to server 255.255.255.255 : 3639

Type command; -help for valid commands; Q to Quit
-help
send ZZZZ9999
        sends message to server
sendecho ZZZZ9999
        server echoes message back
send xAPpack
        sends xAP-encoded xAPpack (see -xAPpack) to server,
        requires xAPencode to be enabled
-xAPencode
        alternates xAP encode on or off
-testconnection(x)
        tests connection x times, provides reliability and
        speed data
-changeserver(xxx.xxx.xxx.xxx)
        Changes IP address of inteded recipient(server)
-changesendport(x)
        Changes the sending port to value x, which is a valid
        port between 1024 and 65535
-xAPpack
        compile and xAPencoded package to send,
        end compilation with 'end' input line
-connectiontimeout(xxx.xxx.xxx.xxx,z)
        measures the the timeout of the connection to
        server xxx.xxx.xxx.xxx incrementally
        with increment x to a maximum of 60seconds
-devicesonline
        shows all devices online om the local network given a server
        running on local network(hartbeat within last 120s)
-devicesonline(server)
        differs from '-devicesonline' only if server is not on
        local network
-serverstatus
        local network server(s) responds with relative
        information about itself(themself)
-serverstatus(server)
        differs from '-serverstatus' only if server is not on
        local network
```

FIGURE 13: COMMANDS AND FUNCTIONS IMPLEMENTED IN CLIENT APPLICATION

Test connection function

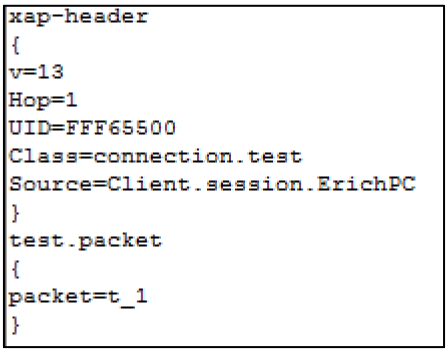
This function tests the reliability and speed data of a given connection to a server. This function is comparable to the standard Packet Internet Gopher (PING) function native to most operating systems. A custom function had to be developed in order to test specific xAP packets. The command to invoke the function is expressed as follows:

-testconnectection(x)

The variable *x* represents the number of packets that will be used to test communication. The client initiates the communication session by sending a xAP packet to the server asking the server to respond appropriately. When the response is received by the client the elapsed time is calculated and the instance is deemed a success. If no response is received within 3 seconds the application expresses “Request timed out”. This continues for *x* repetitions.

After all repetitions are accounted for the function computes the maximum, minimum and average elapsed time for a round trip to the server and back. Reliability data is also presented as number of packets sent, number of packets received and the percentage of packets lost.

A xAP schema for this specific function was created. Figure 14 shows the developed schema.



```
xap-header
{
v=13
Hop=1
UID=FFF65500
Class=connection.test
Source=Client.session.ErichPC
}
test.packet
{
packet=t_1
}
```

FIGURE 14: XAP SCHEMA TO TEST CONNECTION

Connection Timeout function

In order to evaluate the period for which the connection state is true a function was developed which tests the connection states at increasing intervals. The syntax of the command is:

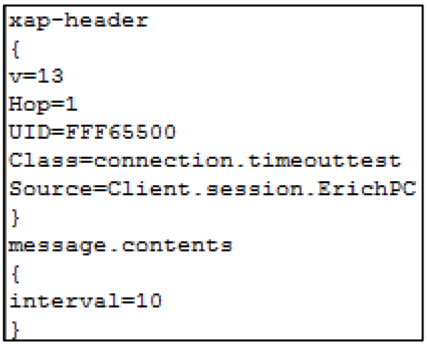
-connectiontimeout(xxx.xxx.xxx.xxx,z)

The value “xxx.xxx.xxx.xxx” represents a valid IP address of the server with which connection state is tested. “z” is the interval value. The client sends packet to the server commanding the server to respond appropriately in “z” seconds. If the test is successful for the current interval value a new interval value of is assigned according to:

New interval value = old interval value + z

The interval value is incremented until an interval value of 60 seconds is achieved. To prevent interference heartbeats from the client is temporarily switched off.

The schema developed to test connection state is shown in Figure 15.



```
xap-header
{
v=13
Hop=1
UID=FFF65500
Class=connection.timeouttest
Source=Client.session.ErichPC
}
message.contents
{
interval=10
}
```

FIGURE 15: XAP CONNECTION TIMEOUT TEST SCHEMA

5.4.4 Server Application

The server application was developed in a similar way to the client application. The server application executes on the computer deemed to be the server in the network configuration. The goal of the server application is to respond appropriately to the requests of the client. A server application also monitors all traffic on port 3639 for the network.

Heartbeats are produced at 60 second intervals and are broadcasted to the LAN. This indicates other devices which port the server is functioning on what its current status is, allowing other devices to discover the server.

Roaming Server Application

The roaming server application is a variation of the standard server application. The roaming server does not broadcast its heartbeats; rather it directs them at another server. This allows the testing of advanced network functionality.

5.5 Web server development

Web servers require special mention as a telemedicine network will be unable to reach its full potential without the incorporation of one or several web servers. A web server provides key functionality to the network. Web servers describe the hardware and software of computers accessible over the Internet. They are used to mainly host web sites, but they can also be used for data storage or administering enterprise applications. They are usually known by a domain name, but a routable IP address is also applicable. The idea is to provide web servers with the same functionality as normal servers, operating according to xAP, but with the added functionality of supporting standard protocols. This will help devices unable to access xAP enabled networks to retrieve information despite their disposition. Web servers can thus manage both xAP networks as well as web sites accessible through the HTTP. A dedicated computer platform with required software and knowledge of PHP or Active Server Page coding is required to set a web server.

6 Results

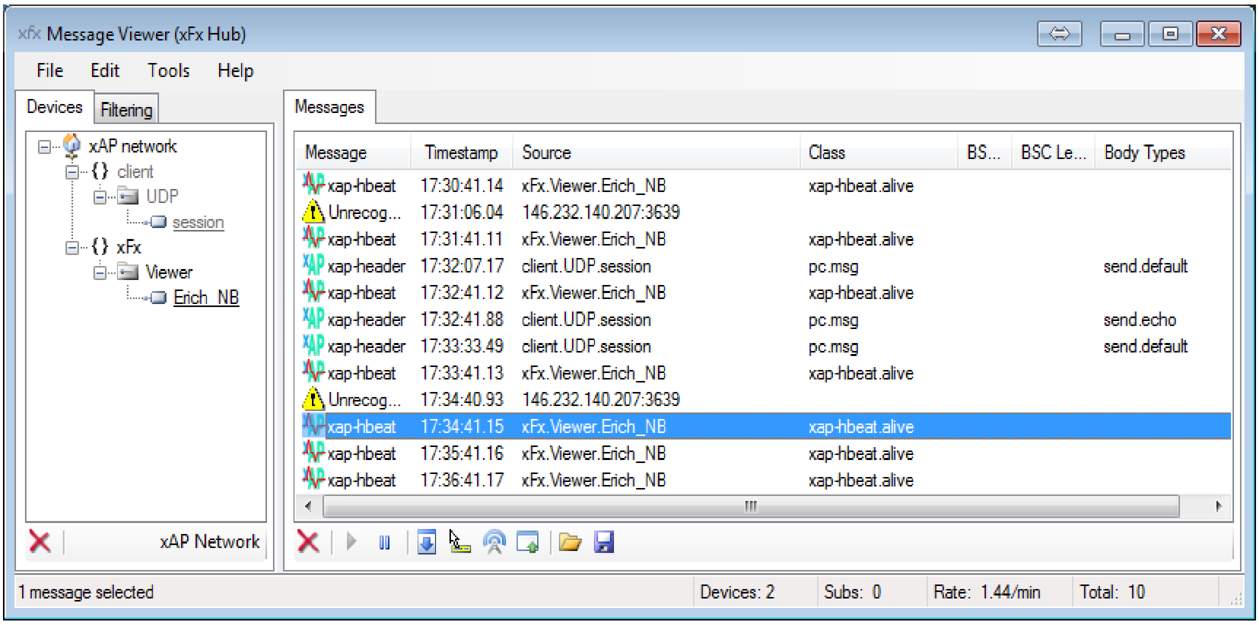
Tests were conducted to investigate feasibility of the network. The configuration was adapted as needed to analyse potential problems for implementation. xAP enabled communication between the client, server in different configurations were tested as well as compatibility with existing xAP tools. All tests were conducted with the xAP protocol encapsulation as well as without. The configurations tested were:

- Local Area Network; all devices on the same private network
- Client to server, both utilising their own 3G modem connections
- Client to server, where the client communicates from a commercial ISP

To demonstrate advance functionality of the xAP network configuration was set up where a client and server was connected with their own 3G connections; additionally a server running on a ISP connection was set up to broadcast heartbeats to the other server running on its 3G connection.

6.1 Compatibility with existing xAP tools

On a Local Area Network (LAN) broadcasting can be used effectively to distribute packets to all willing recipients. Compatibility of the developed applications with the xFx Viewer provided conclusive results. The xFx Viewer could identify packets on the network and whether they are in acceptable xAP format. Figure 16 shows the log screen of an xFx Viewer executing on a computer connected to the LAN. Message that is not in the xAP format is marked as “Unrecognised”.

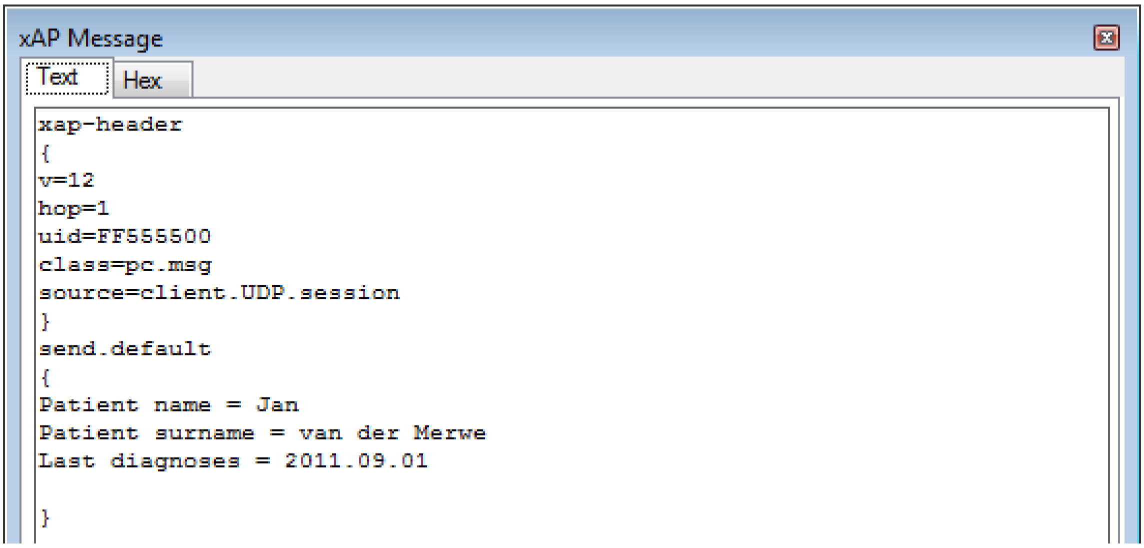


Message	Timestamp	Source	Class	BS...	BSC Le...	Body Types
xap-hbeat	17:30:41.14	xFx.Viewer.Erich_NB	xap-hbeat.alive			
Unrecog...	17:31:06.04	146.232.140.207:3639				
xap-hbeat	17:31:41.11	xFx.Viewer.Erich_NB	xap-hbeat.alive			
xap-header	17:32:07.17	client.UDP.session	pc.msg			send.default
xap-hbeat	17:32:41.12	xFx.Viewer.Erich_NB	xap-hbeat.alive			
xap-header	17:32:41.88	client.UDP.session	pc.msg			send.echo
xap-header	17:33:33.49	client.UDP.session	pc.msg			send.default
xap-hbeat	17:33:41.13	xFx.Viewer.Erich_NB	xap-hbeat.alive			
Unrecog...	17:34:40.93	146.232.140.207:3639				
xap-hbeat	17:34:41.15	xFx.Viewer.Erich_NB	xap-hbeat.alive			
xap-hbeat	17:35:41.16	xFx.Viewer.Erich_NB	xap-hbeat.alive			
xap-hbeat	17:36:41.17	xFx.Viewer.Erich_NB	xap-hbeat.alive			

1 message selected | Devices: 2 | Subs: 0 | Rate: 1.44/min | Total: 10

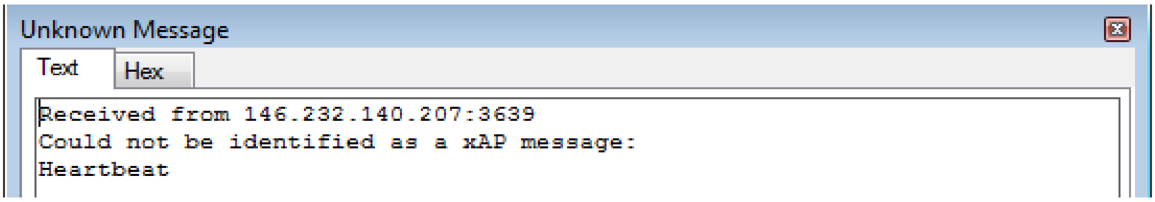
FIGURE 16: XFX VIEWER LOG SCREEN

Message contents are verified by opening the message. Figure 17 shows a packet recognised by xFx Viewer as xAP encoded packet. Figure 18 shows a packet containing just the words “Heartbeat”. As this is packet is not encoded according to the xAP format xFx Viewer deems it incompatible with the xAP devices. Figure 19 shows a message encoded in the xAP format but with a syntax error in the message contents. The message is still received by all willing devices but will be rejected due to the error in syntax.



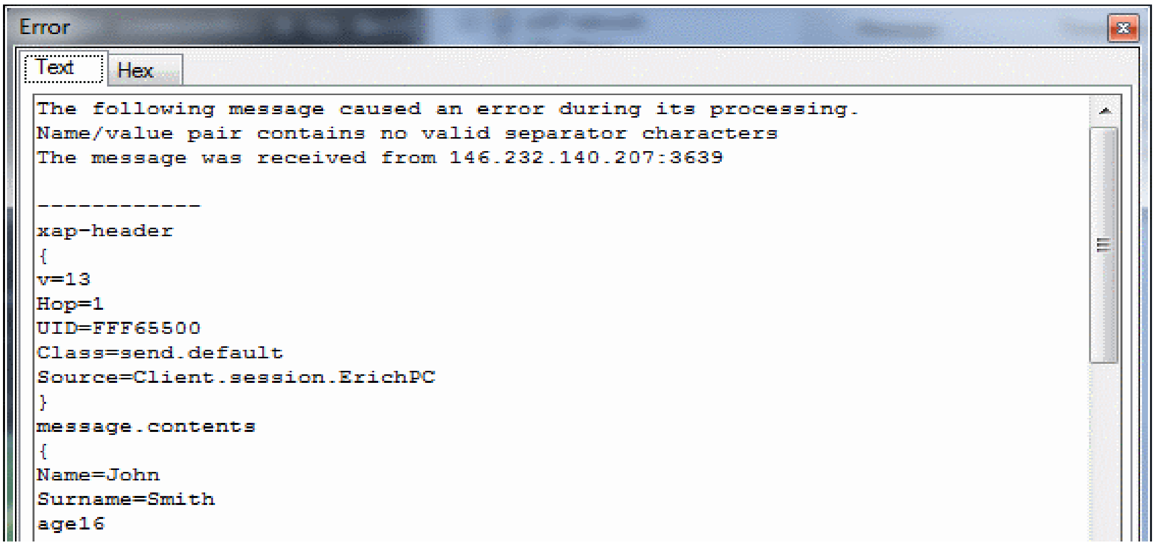
```
xap-header
{
v=12
hop=1
uid=FF555500
class=pc.msg
source=client.UDP.session
}
send.default
{
Patient name = Jan
Patient surname = van der Merwe
Last diagnoses = 2011.09.01

}
```

FIGURE 17: XAP MESSAGE IN XFX VIEWER



```
Received from 146.232.140.207:3639
Could not be identified as a xAP message:
Heartbeat
```

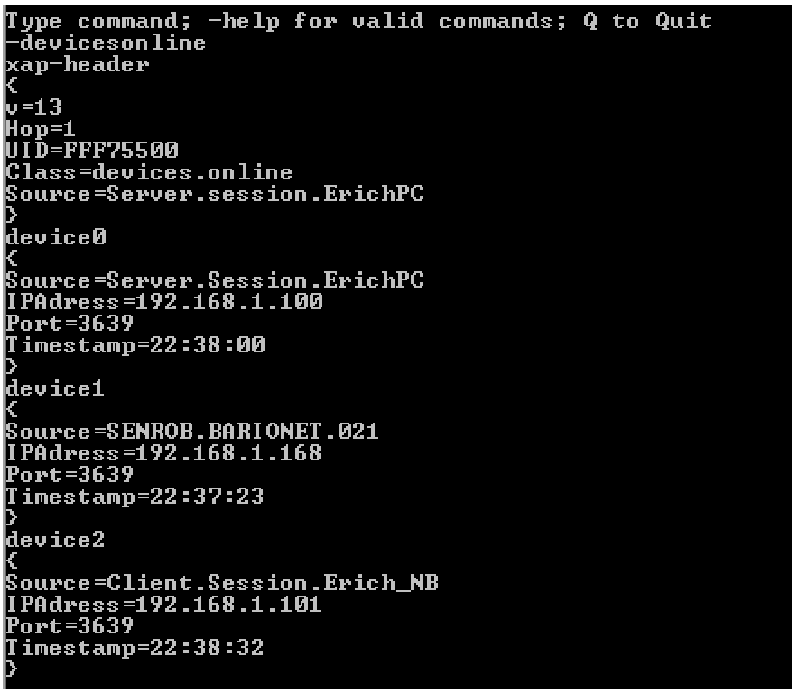
FIGURE 18: UNRECOGNISED MESSAGE IN XFX FIEWER



```
The following message caused an error during its processing.
Name/value pair contains no valid separator characters
The message was received from 146.232.140.207:3639

------------
xap-header
{
v=13
Hop=1
UID=FFF65500
Class=send.default
Source=Client.session.ErichPC
}
message.contents
{
Name=John
Surname=Smith
age16
```

FIGURE 19: ERROR IN XAP MESSAGE

Compatibility of a low-level device within the configuration was tested using a Barix Barionet 100. The device is loaded with xAP-enabled firmware. Results show compatibility on a local network with the server. Considering other results, the low-level device should definitely be able to communicate over the Internet with a server given the correct set up parameters. See Figure 20 below, a client session requested a list of devices online at the server, and the server responded as shown.



```
Type command; -help for valid commands; Q to Quit
-devicesonline
xap-header
{
v=13
Hop=1
UID=FFF75500
Class=devices.online
Source=Server.session.ErichPC
}
device0
{
Source=Server.Session.ErichPC
IPAdress=192.168.1.100
Port=3639
Timestamp=22:38:00
}
device1
{
Source=SENROB.BARIONET.021
IPAdress=192.168.1.168
Port=3639
Timestamp=22:37:23
}
device2
{
Source=Client.Session.Erich_NB
IPAdress=192.168.1.101
Port=3639
Timestamp=22:38:32
}
```

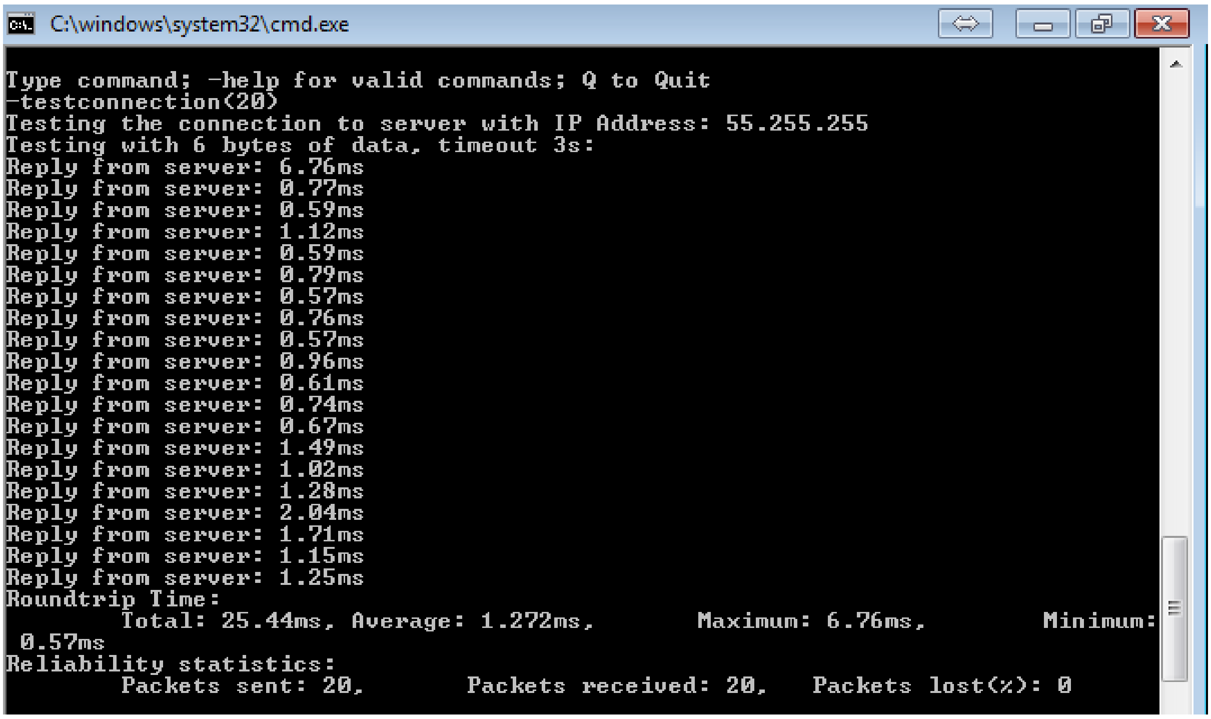
FIGURE 20: LOW-LEVEL DEVICE COMPATIBILITY

6.2 Reliability of connections

Due to all packets formatted in the UDP format there is no guaranteed arrival of the packet as discussed in chapter 3.3.1. Evaluation of the networks is tested for the standard configurations. Packets communicated over a LAN were found to be very reliable with a packet rarely not completing the full route. A seen in Figure 21, 100% packet arrival is common. The packet loss might increase with network complexity and it is still recommended to implement acknowledgement of packet arrival. Take note of the time elapsed of route completion for a LAN connection, which provides a response capable of performing continuous operations on such a connection

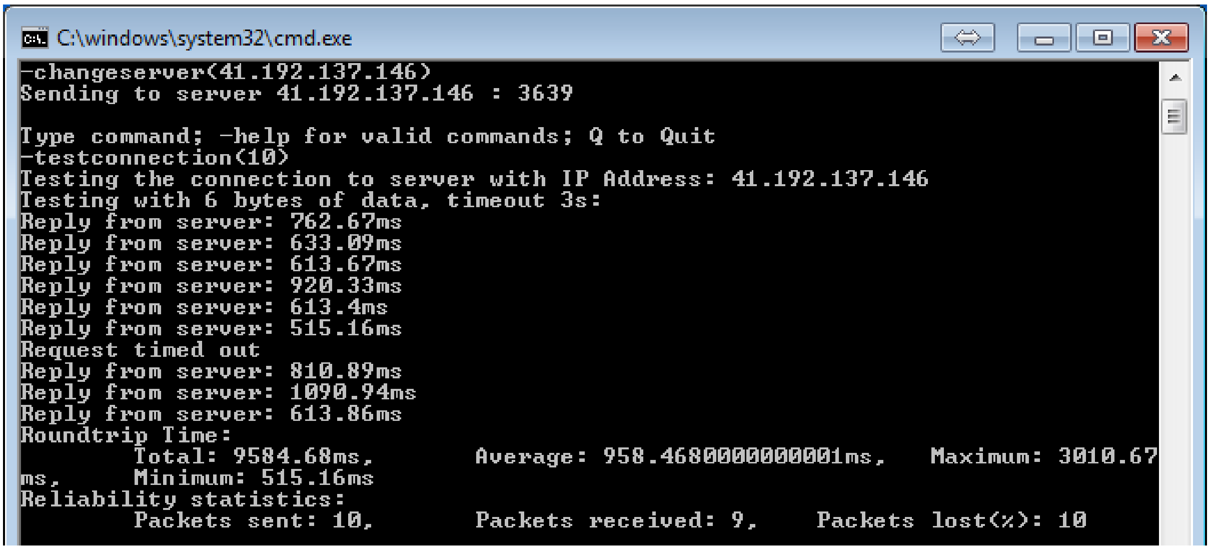
Figure 22 and Figure 23 depicts a reliability test conducted where the client and server is communicating through the Internet each utilising its own separate 3G connection. As seen from comparison of the figures reliability as well as the speed of the connection varies. Connection reliability of less than 80% is rare.

Connection reliability of an ISP connected client to a 3G connected server is shown in Figure 24. Again a good reliability is noted with 100% packet arrival being common. Less variability in the elapsed time for a round trip is noted.



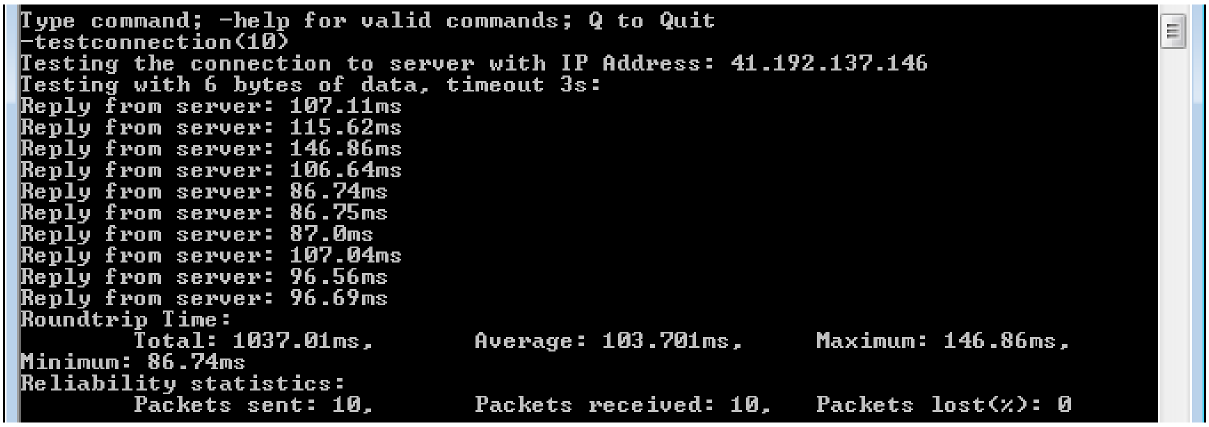
```
Type command; -help for valid commands; Q to Quit
-testconnection(20)
Testing the connection to server with IP Address: 55.255.255
Testing with 6 bytes of data, timeout 3s:
Reply from server: 6.76ms
Reply from server: 0.77ms
Reply from server: 0.59ms
Reply from server: 1.12ms
Reply from server: 0.59ms
Reply from server: 0.79ms
Reply from server: 0.57ms
Reply from server: 0.76ms
Reply from server: 0.57ms
Reply from server: 0.96ms
Reply from server: 0.61ms
Reply from server: 0.74ms
Reply from server: 0.67ms
Reply from server: 1.49ms
Reply from server: 1.02ms
Reply from server: 1.28ms
Reply from server: 2.04ms
Reply from server: 1.71ms
Reply from server: 1.15ms
Reply from server: 1.25ms
Roundtrip Time:
        Total: 25.44ms, Average: 1.272ms,       Maximum: 6.76ms,        Minimum:
 0.57ms
Reliability statistics:
        Packets sent: 20,       Packets received: 20,   Packets lost(%): 0
```

FIGURE 21: LAN CONNECTION RELIABILITY TEST



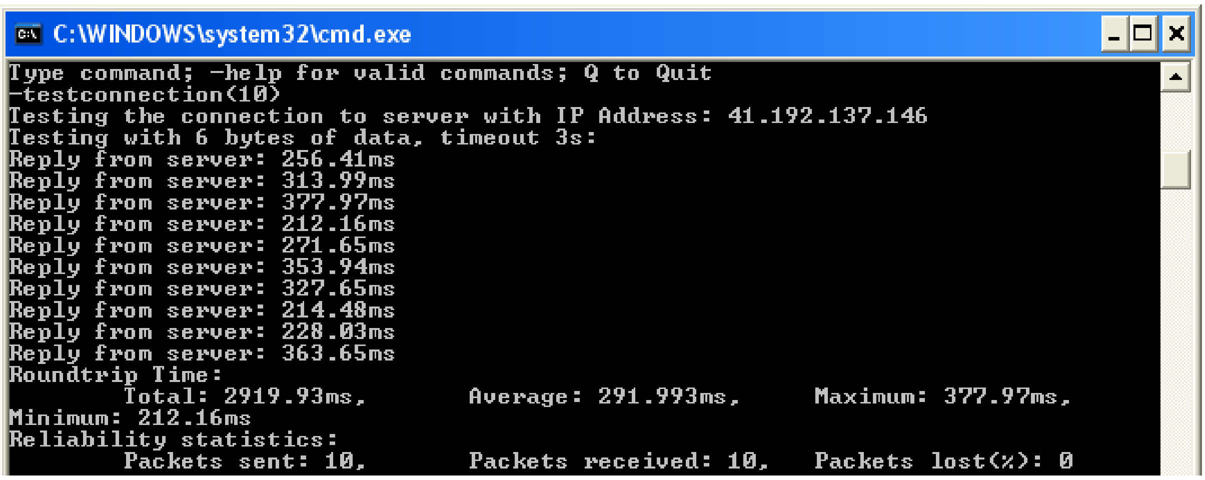
```
-changeserver(41.192.137.146)
Sending to server 41.192.137.146 : 3639

Type command; -help for valid commands; Q to Quit
-testconnection(10)
Testing the connection to server with IP Address: 41.192.137.146
Testing with 6 bytes of data, timeout 3s:
Reply from server: 762.67ms
Reply from server: 633.09ms
Reply from server: 613.67ms
Reply from server: 920.33ms
Reply from server: 613.4ms
Reply from server: 515.16ms
Request timed out
Reply from server: 810.89ms
Reply from server: 1090.94ms
Reply from server: 613.86ms
Roundtrip Time:
        Total: 9584.68ms,       Average: 958.4680000000001ms,   Maximum: 3010.67
ms,     Minimum: 515.16ms
Reliability statistics:
        Packets sent: 10,       Packets received: 9,    Packets lost(%): 10
```

FIGURE 22: 3G CONNECTION RELIABILITY TEST 1



```
Type command; -help for valid commands; Q to Quit
-testconnection(10)
Testing the connection to server with IP Address: 41.192.137.146
Testing with 6 bytes of data, timeout 3s:
Reply from server: 107.11ms
Reply from server: 115.62ms
Reply from server: 146.86ms
Reply from server: 106.64ms
Reply from server: 86.74ms
Reply from server: 86.75ms
Reply from server: 87.0ms
Reply from server: 107.04ms
Reply from server: 96.56ms
Reply from server: 96.69ms
Roundtrip Time:
        Total: 1037.01ms,       Average: 103.701ms,     Maximum: 146.86ms,
Minimum: 86.74ms
Reliability statistics:
        Packets sent: 10,       Packets received: 10,   Packets lost(%): 0
```

FIGURE 23:3G CONNECTION RELIABILITY TEST 2

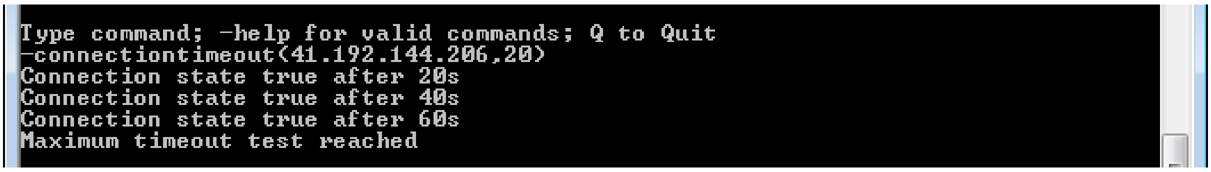


```
Type command; -help for valid commands; Q to Quit
-testconnection(10)
Testing the connection to server with IP Address: 41.192.137.146
Testing with 6 bytes of data, timeout 3s:
Reply from server: 256.41ms
Reply from server: 313.99ms
Reply from server: 377.97ms
Reply from server: 212.16ms
Reply from server: 271.65ms
Reply from server: 353.94ms
Reply from server: 327.65ms
Reply from server: 214.48ms
Reply from server: 228.03ms
Reply from server: 363.65ms
Roundtrip Time:
        Total: 2919.93ms,       Average: 291.993ms,     Maximum: 377.97ms,
Minimum: 212.16ms
Reliability statistics:
        Packets sent: 10,       Packets received: 10,   Packets lost(%): 0
```

FIGURE 24: ISP CONNECTION RELIABILITY TEST

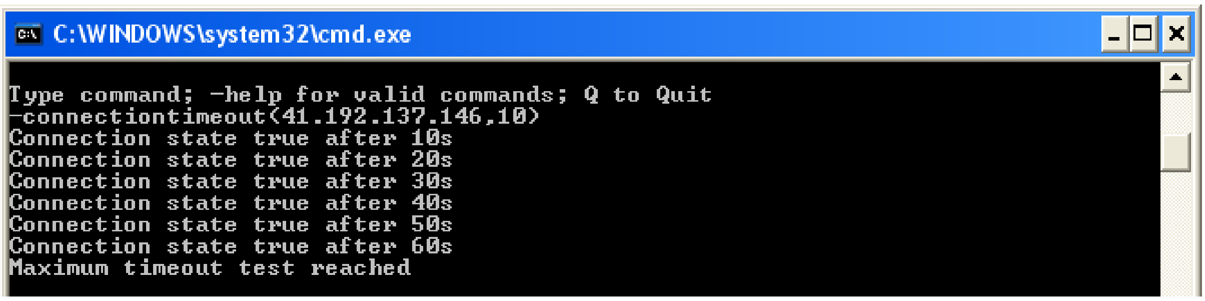
6.3 Connection Timeout

Timeout are only applicable to connections over the Internet and thus no connection timeout tests were done for devices connected over a LAN. The tests are conducted in order to determine if the proposed “heartbeat” interval of sixty seconds is sufficient for xAP networks. Connection timeout tests often proved to be inconclusive as they are dependent on the arrival of a packet. Tests using the same parameters were repeated to accommodate this problem. Configurations where a client and server utilising their own 3G connections to communicate as well as a client connected over an ISP connection to a 3G connected server were set up. The tests provided no reason for “heartbeat” intervals to be less than sixty seconds. Figure 25 and Figure 26 show the results for the two configurations respectively.



```
Type command; -help for valid commands; Q to Quit
-connectiontimeout(41.192.144.206,20)
Connection state true after 20s
Connection state true after 40s
Connection state true after 60s
Maximum timeout test reached
```

FIGURE 25: 3G CONNECTION TIMEOUT TEST



```
Type command; -help for valid commands; Q to Quit
-connectiontimeout(41.192.137.146,10)
Connection state true after 10s
Connection state true after 20s
Connection state true after 30s
Connection state true after 40s
Connection state true after 50s
Connection state true after 60s
Maximum timeout test reached
```

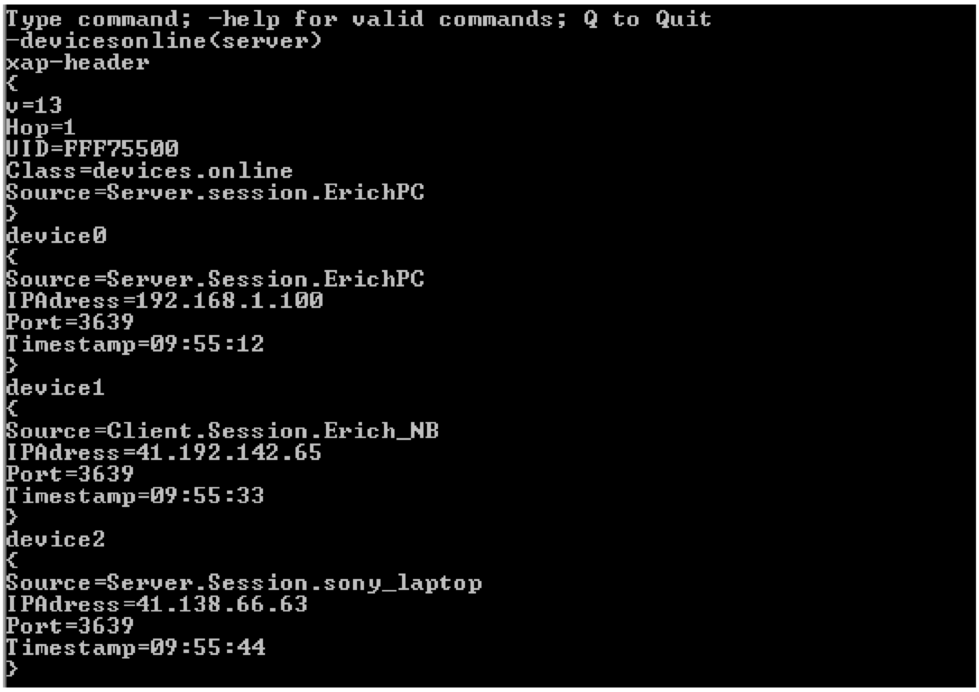
FIGURE 26: ISP CONNECTION TIMEOUT TEST

6.4 Advanced Network Functionality

Advanced functionality includes the ability of a client session to communicate with other devices registered as online at the server. A configuration used to test such functionality:

- A server connected to the Internet through a 3G connection
- A client connected to the Internet through a 3G connection
- A server connected to the Internet through an ISP connection and sending "heartbeats" to the other server

The primary server is thus connected to a 3G connection and has a routable IP address. The secondary server's IP Address is not routable. It was attempted to communicate between the client and the secondary server when both devices are registered as online by the primary server. To confirm that all three devices were online the client requested the primary server to provide a list of the devices online. Figure 27 shows the status of the request.



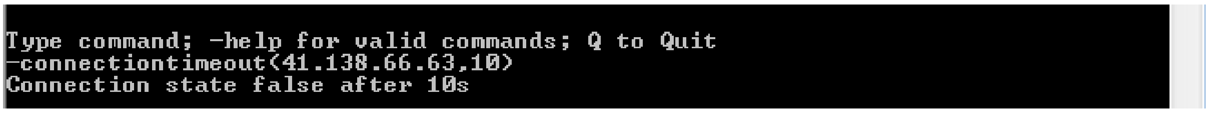
```
Type command; -help for valid commands; Q to Quit
-devicesonline(server)
xap-header
{
v=13
Hop=1
UID=FFF75500
Class=devices.online
Source=Server.session.ErichPC
}
device0
{
Source=Server.Session.ErichPC
IPAdress=192.168.1.100
Port=3639
Timestamp=09:55:12
}
device1
{
Source=Client.Session.Erich_NB
IPAdress=41.192.142.65
Port=3639
Timestamp=09:55:33
}
device2
{
Source=Server.Session.sony_laptop
IPAdress=41.138.66.63
Port=3639
Timestamp=09:55:44
}
```

FIGURE 27: DEVICES REGISTERED AS ONLINE AT THE SERVER

Using the information provided by the primary server, the client attempted to complete a connection timeout test to the secondary server. I present the result in Figure 28. No communication could be established between the client and the secondary server. Several possible reasons exist for the result:

- The packet is lost on the network
- The server failed to respond
- The packet was blocked by the server's ISP

The most probable conclusion is the ISP blocking the incoming packet due to no connection state existing between the client and secondary server. Connecting to another device which does not have a routable address, even if the information of that device is known, could not be confirmed and alternative methods should be used in future tests.



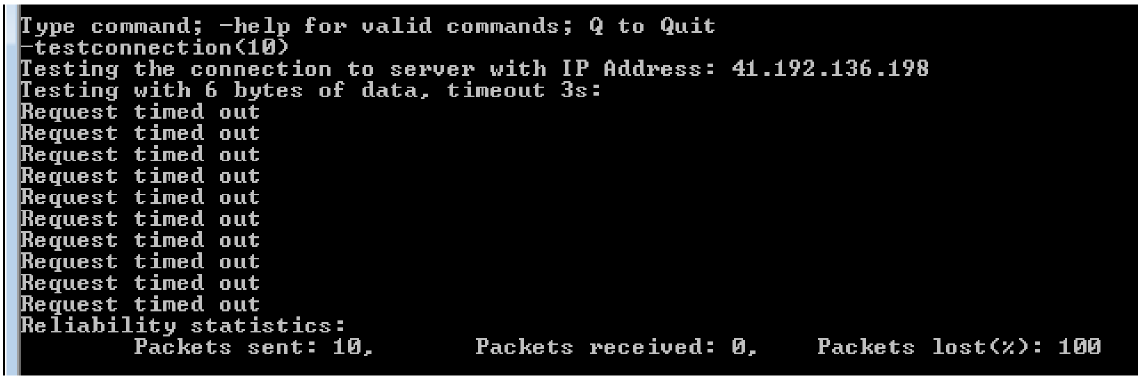
```
Type command; -help for valid commands; Q to Quit
-connectiontimeout(41.138.66.63,10)
Connection state false after 10s
```

FIGURE 28: CONNECTION TIMOUT TEST TO SECONDARY SERVER

6.5 Connection Failure

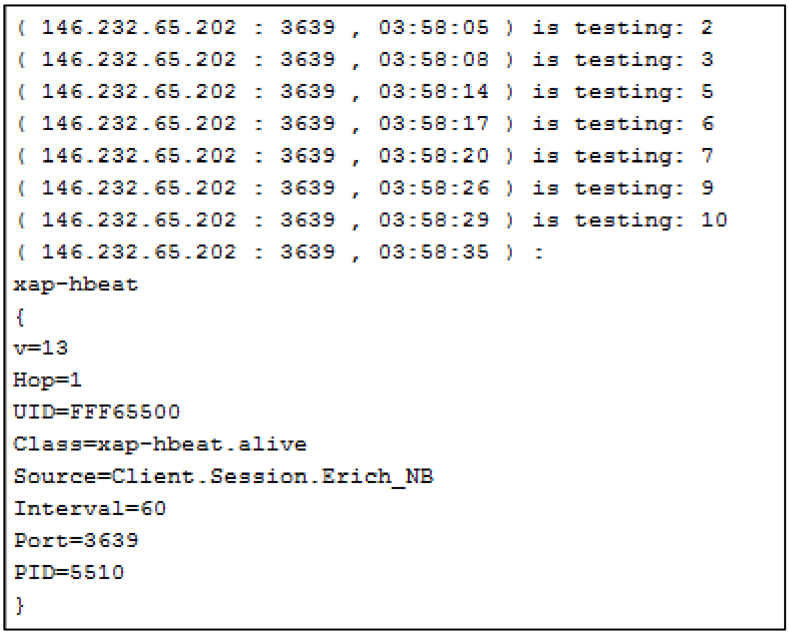
To demonstrate the restrictions ISPs can place on non-standard connections, a communication session was attempted through the University of Stellenbosch wireless network: MatiesWifi. High security measures are in place for this network and practically only allows standard Internet operations such as web browsing. The server side was set up according to the normal configuration.

Figure 29 shows the client trying to test the connection to the server, while Figure 30 shows the response on the server side. Even though the client is connected through a unreliable connection (notice only seven packets arrive at server), none of the packets complete the round-trip back to the client. This can be due to lost packets, but this is highly unlikely as seven packets arrived at the server. To communicate over such a restricted network a standard communication protocol with a standard dedicated port should be used.



```
Type command; -help for valid commands; Q to Quit
-testconnection(10)
Testing the connection to server with IP Address: 41.192.136.198
Testing with 6 bytes of data, timeout 3s:
Request timed out
Request timed out
Request timed out
Request timed out
Request timed out
Request timed out
Request timed out
Request timed out
Request timed out
Request timed out
Reliability statistics:
        Packets sent: 10,       Packets received: 0,    Packets lost(%): 100
```

FIGURE 29: CLIENT COMMUNICATING FROM RESTRICTED NETWORK



```
( 146.232.65.202 : 3639 , 03:58:05 ) is testing: 2
( 146.232.65.202 : 3639 , 03:58:08 ) is testing: 3
( 146.232.65.202 : 3639 , 03:58:14 ) is testing: 5
( 146.232.65.202 : 3639 , 03:58:17 ) is testing: 6
( 146.232.65.202 : 3639 , 03:58:20 ) is testing: 7
( 146.232.65.202 : 3639 , 03:58:26 ) is testing: 9
( 146.232.65.202 : 3639 , 03:58:29 ) is testing: 10
( 146.232.65.202 : 3639 , 03:58:35 ) :
xap-hbeat
{
v=13
Hop=1
UID=FFF65500
Class=xap-hbeat.alive
Source=Client.Session.Erich_NB
Interval=60
Port=3639
PID=5510
}
```

FIGURE 30: SERVER RESPONSE TO RESTRICTED NETWORK COMMUNICATION SESSION

7 Conclusions and Recommendations

The goal of the project was to investigate the suitability of xAP for Health Systems. The current status quo of such systems was investigated and network architecture to support xAP implementation was provided. To accomplish this goal the following objectives were demonstrated:

- Suitable network architecture for xAP environment
- Categorisation of network components to facilitate scalability
- Integration with existing systems
- Adaptability of xAP schemas for the telemedicine environment
- Evaluating the network medium in support of xAP communications

The network architecture to support xAP environment was defined with respect to the Internet Protocol Suite. Special care was taken to allow full communication capability over the Internet. The Internet restricts xAP implementation, as broadcasting, which is the primary way of communication for xAP enabled devices, is not permitted over the Internet and packets have to be directed at a routable address.

Components required to represent the xAP network were categorised according to the function they will fulfil on the network. A framework was developed for that specific category of components which could then be used to develop device specific applications or services. A device performing a task in a category could easily be replaced without affecting the health of the network. Categorisation also allows devices to be interoperable with any given device in a category, preventing system degradation if a device is disconnected.

Although existing tools provided for xAP systems are limited, compatibility could be confirmed with the xFx Viewer tool provided by the xAP support group. Correct encoding of a xAP packet allowed the xFx Viewer to interpret the packet, regardless of the schema and contents. Schemas were created to provide functionality to test network architectures. These schemas could easily be adapted to contain any information and still be recognized by other devices as a viable xAP packet. The lacking support for devices capable of parsing xAP packets proved a challenge. Future implementations will rely heavily on development of xAP specific applications, inhibiting integration with existing systems.

Various tests were conducted to test network mediums based a configuration chosen to represent real world implementations. 3G connections via a cellular network to the Internet were demonstrated successfully with suitable reliability. Commercial ISP compatibility was tested and again positive results in terms of reliability and general connection health was realised. Although some networks restricted the communication, alternative channels can be used to communicate over the Internet.

This resulted in the first recommendation for future implementation. Due to some ISPs restricting xAP traffic deemed malicious, accommodations should be made to provide a secondary method of communicating to a device. It is recommended to implement secondary compatibly for the Hyper Text Transfer Protocol. This protocol is the most widely used on the Internet and almost all devices with a NIC has the capability to parse HTTP packets. ISPs do not block this protocol or the standard port allocated to it as HTTP provides the core functionality of Internet browsing. A web server could easily be set up to handle secondary HTTP connections.

Further research is required for conclusive feasibility of xAP implementation for mobile phones. Given the wide range of development environments for operating systems suited for mobile phones, the research required is extensive. Currently no xAP applications suited for operation on mobile phones is available for communication over the Internet. To develop such applications require knowledge of the coding language applicable to a specific mobile phone operating system.

Medical records are protected by legislation and ethical principles. Applicable protocols must thus incorporate the necessary safety mechanisms to prevent unauthorised access to records. Security will need to be implemented on three layers; i) Database security, ii) Packet security and iii) network security.

Existing standards enjoy commercial integration and their use will become more widespread as organisations realise the benefit of standardisation. Although access to these standards is restricted they cannot be ignored as they provide complete documentation of health systems which could assist in implementation and development of future xAP initiatives. Compatibility with these standards should be strived for, which will result in xAP being adopted more readily.

Lastly, currently xAP is deemed a format for encoding data. It is proposed to promote xAP as a means to standardise device interoperability in telemedicine networks. This will provide complete network architecture for facilitators to promote. This project proved core feasibility of such standardisation using xAP tools. Objectives identified for xAP by the support group, such as ease of use and simplicity, prove favourable compared to the extensively defined existing standards. xAP as a standard will enhance sustainability of the protocol and increase manufacturer support, which is currently lacking.

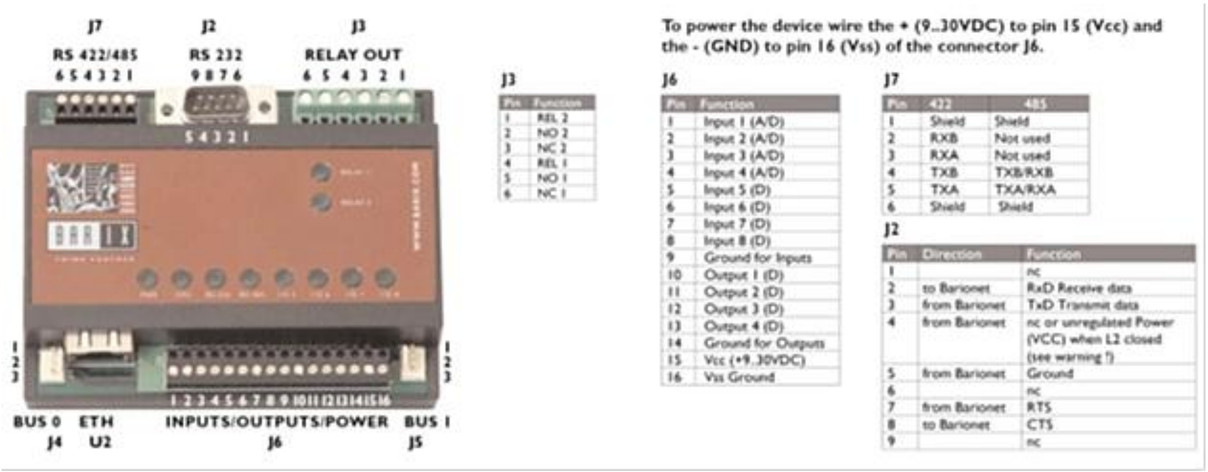
8 References

1. Anonymous, 2011. *About Digital Living Network Alliance*. [Online] Available at: http://www.dlna.org/about_us/about/ [Accessed September 2011].
2. Basu, S., 2009. *HL7 Strategies*. [Online] Available at: http://www.hl7.org/documentcenter/public_temp_BB08EBA6-1C23-BA17-0C6CD87394AFECFC/training/IntroToHL7/data/downloads/hl7%20strategies.pdf [Accessed September 2011].
3. Benatar, S.R., 2004. Health Care Reform and the Crisis of HIV and AIDS. *The New England Journal of Medicine*, 35(1), pp.81-92.
4. Bent, J., Tofts, T. & Jeffery, I., 2007. *The xPLHal Control Protocol*. [Online] (1.5) Available at: http://xplproject.org.uk/wiki/index.php?title=XHCP_Protocol [Accessed September 2011].
5. Braden, , 1989. *Requirements for Internet Hosts -- Communication Layers, RFC 1122*. [Online] Available at: http://tools.ietf.org/html/rfc1122 [Accessed September 2011].
6. Bray, T. et al., 2008. *Extensible Markup Language (XML)*. [Online] (5) Available at: http://www.w3.org/TR/2008/REC-xml-20081126/ [Accessed September 2011].
7. Digi, 2006. *Application Guide: Cellular IP Connections (Uncovered)*. [Online] Available at: http://ftp1.digi.com/support/documentation/appguide_connectcellular_ipconsiderations.pdf [Accessed September 2011].
8. Digital Imaging and Communications in Medicine , [s.a]. *DICOM Brochure*. [Online] Available at: http://medical.nema.org/dicom/geninfo/Brochure.pdf [Accessed September 2011].
9. Dyn, [s.a]. *DynDNS Free*. [Online] Available at: http://dyn.com/dns/dyndns-free/ [Accessed October 2011].
10. Elahi, A. & Elahi, M., 2006. *Data, Network & Internet Communications Technology*. New York: Thomson Delmar Learning.
11. Hall, B., 2009. *Using Internet Sockets*. [Online] Available at: http://beej.us/guide/bgnet/ [Accessed April 2011].
12. Health Level Seven International, 2011. *HL7*. [Online] Available at: www.HL7.org [Accessed September 2011].
13. Health Level Seven, [s.a]. *HL7*. [Online] Available at: http://www.hl7.org/about/FAQs/ [Accessed September 2011].
14. Information Sciences Institute University of Southern California, 1981. *INTERNET PROTOCOL, RFC 791*. [Online] Available at: http://tools.ietf.org/html/rfc791 [Accessed September 2011].
15. Information Sciences Institute University of Southern California, 1981. *Transmission Control Protocol, RFC 793*. [Online] Available at: http://tools.ietf.org/html/rfc793 [Accessed September 2011].
16. International Telecommunications Union, 2009. *Information Society Statistical Profiles 2009*. [Online] Available at: www.itu.int/pub/D-IND-RPM.AF-2009/en [Accessed September 2011].
17. ISO, 2004. *ISO/TR 16056-1:2004*. [Online] Available at: http://www.iso.org/iso/catalogue_detail?csnumber=37351 [Accessed September 2011].
18. Istepanian, R.S.H., Pattichis, C.S. & Laxminarayan, S., 2006. Ubiquitous M-Health Systems and the convergence towards 4G Mobile Technologies. In E. Mechell-Tzanakou, ed. *M-Health: emerging mobile health systems*. New York: Springer. pp.3-10.

19. Kifle, M. et al., 2008. A Telemedicine Transfer Model for Sub-Saharan Africa. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*. Waikoloa, 2008. http://doi.ieeecomputersociety.org/10.1109/HICSS.2008.41.
20. Lidstone, P., Harrison, M., Hawkins, K. & Tankard, J., 2002. *Protocol Definition*. [Online] (1.2-9) Available at: http://www.xapautomation.org/index.php?title=Protocol_definition [Accessed September 2011].
21. Lowe, I. & Tofts, T., 2011. *xPL Specification Document*. [Online] Available at: http://xplproject.org.uk/wiki/index.php?title=XPL_Specification_Document [Accessed September 2011].
22. Matshidze, P. & Hanmer, L., 2007. *HST Publications*. [Online] Available at: http://www.hst.org.za/uploads/files/chap6_07.pdf [Accessed September 2011].
23. Moor, G.J.E.D., 1993. Health Level 7. In *Progress in standardization in health care informatics*. IOS Press. pp.144-47.
24. Openremote, 2009. *Openremote.org*. [Online] Available at: http://www.openremote.org/display/knowledge/xAP+and+the+xPL+project [Accessed September 2011].
25. Oracle, [s.a]. *Java*. [Online] Available at: http://www.oracle.com/us/technologies/java/index.html [Accessed October 2011].
26. Oracle, [s.a]. *Java Tutorials: What is a Socket*. [Online] Available at: http://download.oracle.com/javase/tutorial/networking/sockets/definition.html [Accessed October 2011].
27. Postel, J., 1980. *User Datagram Protocol, RFC 768*. [Online] Available at: http://tools.ietf.org/html/rfc768 [Accessed September 2011].
28. Postel, J., 1983. *The TCP Maximum Segment Size The TCP Maximum Segment Size, RFC 879*. [Online] Available at: http://gamay.tools.ietf.org/html/rfc879 [Accessed September 2011].
29. Schmitt, L., Falck, T., Wartena, F. & Simons, D., 2007. Novel ISO/IEEE 11073 Standards for Personal Telehealth Systems Interoperability. In *High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP. Joint Workshop on*., 2007. [Online] Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4438177&isnumber=4438152.
30. Sherwin, L., 2009. *uPnP Forum*. [Online] Available at: http://www.upnp.org/news/documents/UPnPForum_02052009.pdf [Accessed September 2011].
31. Spronk, R., 2009. *Ringholm*. [Online] Available at: http://www.ringholm.de/column/hl7_south_africa_PHISC_private_healthcare.htm [Accessed September 2011].
32. Srisuresh, P. & Holdrege, M., 1999. *Internet Engineering Task Force*. [Online] Available at: https://tools.ietf.org/html/rfc2663 [Accessed September 2011].
33. United Nations, 2010. *World Population Prospects*. [Web site] Available at: http://esa.un.org/unpd/wpp/index.htm [Accessed September 2011].
34. UPnP Forum, 2011. *What is UPnP?* [Online] Available at: http://upnp.org/about/what-is-upnp/ [Accessed September 2011].
35. Venter, A. 2011. Personal Interview. 11 August, Stellenbosch (Job description:*Stellenbosch University IT Department Head Network Engineer*)

APPENDIX A: EQUIPMENT USED FOR NETWORK FEASIBILITY TESTS

Appendix A.1



To power the device wire the + (9..30VDC) to pin 15 (Vcc) and the - (GND) to pin 16 (Vss) of the connector J6.

J3

Pin	Function
1	REL 2
2	NO 2
3	NC 2
4	REL 1
5	NO 1
6	NC 1

J6

Pin	Function
1	Input 1 (A/D)
2	Input 2 (A/D)
3	Input 3 (A/D)
4	Input 4 (A/D)
5	Input 5 (D)
6	Input 6 (D)
7	Input 7 (D)
8	Input 8 (D)
9	Ground for Inputs
10	Output 1 (D)
11	Output 2 (D)
12	Output 3 (D)
13	Output 4 (D)
14	Ground for Outputs
15	Vcc (+9..30VDC)
16	Vss Ground

J7

Pin	422	485
1	Shield	Shield
2	RXB	Not used
3	RXA	Not used
4	TXB	TXB/RXB
5	TXA	TXA/RXA
6	Shield	Shield

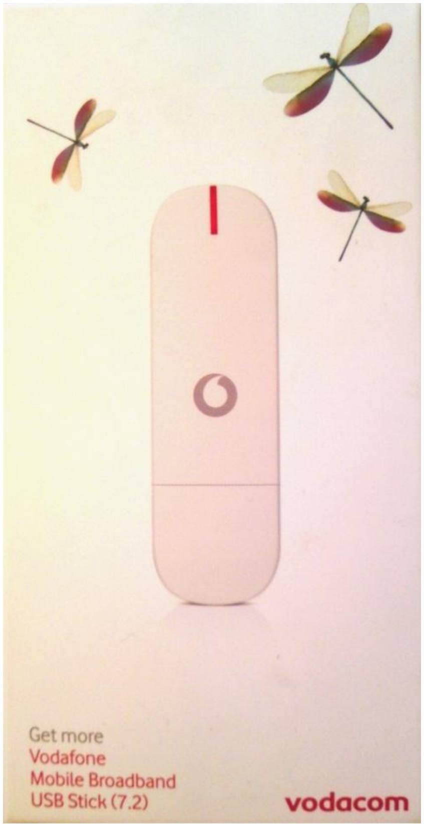
J2

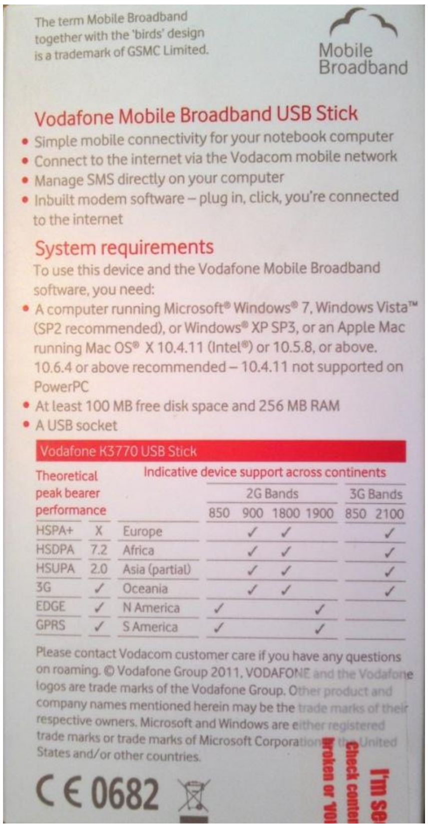
Pin	Direction	Function
1		nc
2	to Barionet	RxD Receive data
3	from Barionet	TxD Transmit data
4	from Barionet	nc or unregulated Power (VCC) when L2 closed (see warning !)
5	from Barionet	Ground
6		nc
7	from Barionet	RTS
8	to Barionet	CTS
9		nc

Appendix A.2

TP-LINK®
The Reliable Choice
Compatible With
Windows®7
3G
SHARE
3G Connectivity On-the-Go
Works with UMTS/HSPA/EVDO USB modems to instantly establish a mobile wireless 3G hotspot
3G/WAN Connection Back-up
Ensures that even if one connection fails, you remain connected to the Internet
Incredibly High Speed Wireless
Share your internet locally with 11N speed, at up to 300Mbps
3G/3.75G Wireless N Router
N
300Mbps
2.4GHz
3G
WAN
QSS™
Quick Secure Setup
WiFi
CERTIFIED
PWR SYS WLAN 1 2 3 4 WAN 3G QSS
3G/3.75G Wireless N Router
TL-MR3420
QSS
Model No. TL-MR3420

Appendix A.3





The term Mobile Broadband together with the 'birds' design is a trademark of GSMC Limited.

Mobile Broadband

Vodafone Mobile Broadband USB Stick

- Simple mobile connectivity for your notebook computer
- Connect to the internet via the Vodacom mobile network
- Manage SMS directly on your computer
- Inbuilt modem software – plug in, click, you're connected to the internet

System requirements

To use this device and the Vodafone Mobile Broadband software, you need:

- A computer running Microsoft® Windows® 7, Windows Vista™ (SP2 recommended), or Windows® XP SP3, or an Apple Mac running Mac OS® X 10.4.11 (Intel®) or 10.5.8, or above. 10.6.4 or above recommended – 10.4.11 not supported on PowerPC
- At least 100 MB free disk space and 256 MB RAM
- A USB socket

Vodafone K3770 USB Stick

Theoretical peak bearer performance		Indicative device support across continents	2G Bands				3G Bands	
			850	900	1800	1900	850	2100
HSPA+	X	Europe		✓	✓			✓
HSDPA	7.2	Africa		✓	✓			✓
HSUPA	2.0	Asia (partial)		✓	✓			✓
3G	✓	Oceania		✓	✓			✓
EDGE	✓	N America	✓			✓		
GPRS	✓	S America	✓			✓		

Please contact Vodacom customer care if you have any questions on roaming. © Vodafone Group 2011, VODAFONE and the Vodafone logos are trade marks of the Vodafone Group. Other product and company names mentioned herein may be the trade marks of their respective owners. Microsoft and Windows are either registered trade marks or trade marks of Microsoft Corporation in the United States and/or other countries.

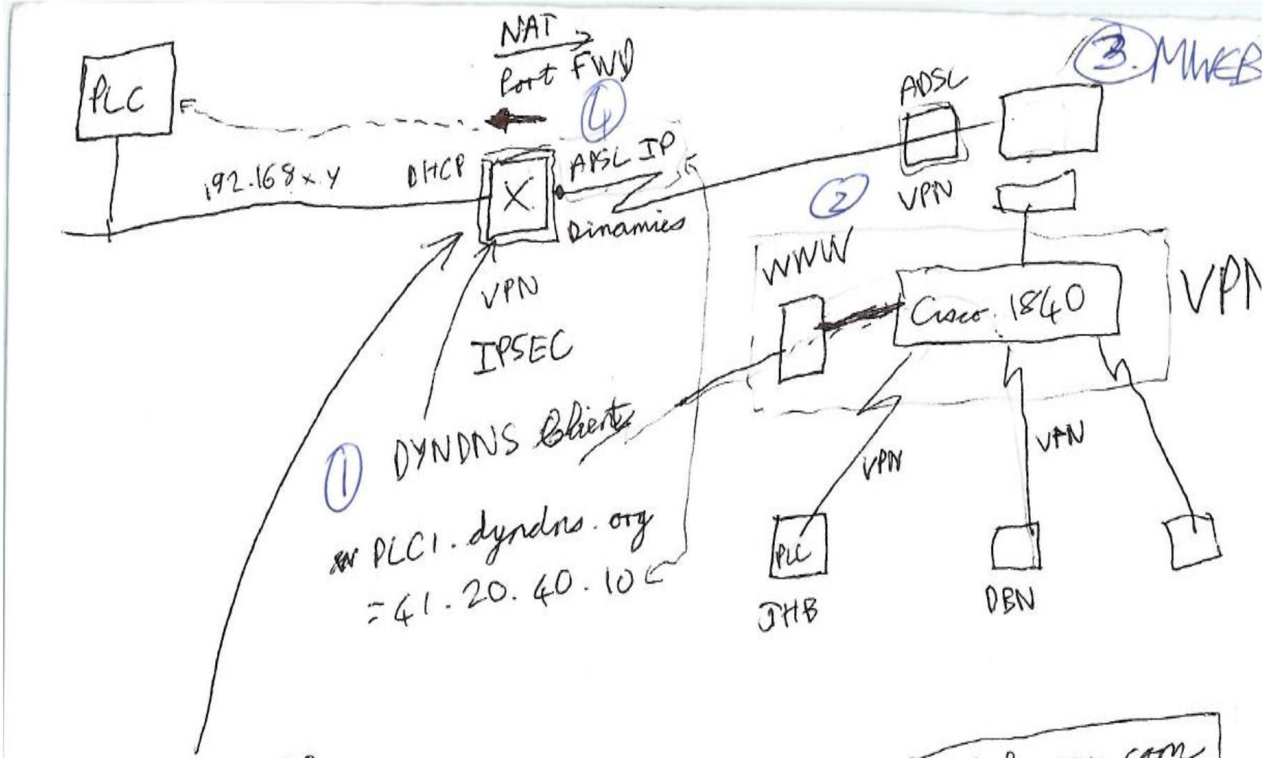
CE 0682

I'm se
Check conten
broken or VO

APPENDIX B: CLIENT AND SERVER APPLICATION

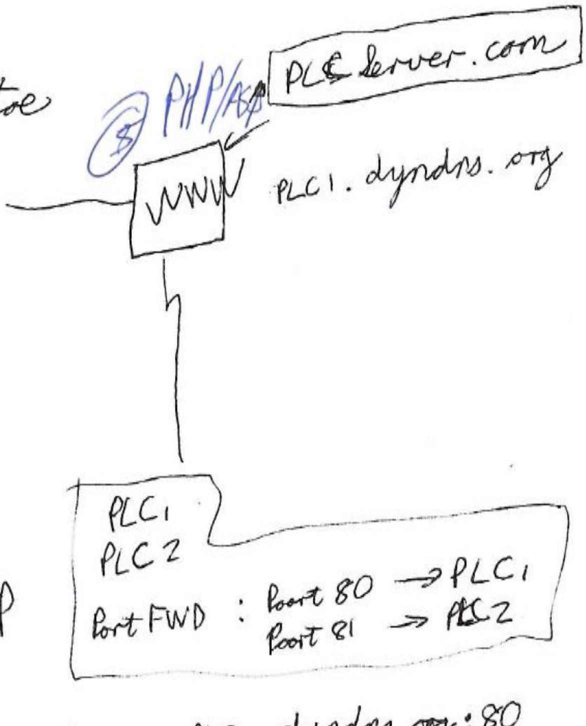
PLEASE REFER TO ATTACHED DISK

APPENDIX C: INTERVIEW NOTES



ADSL . Firewall

- Laat slegs sekere IP's toe
- Laat sekere poorte toe



ADSL DHCP

- Ken IP's toe

DNS

- Vertaal naam na IP

1)

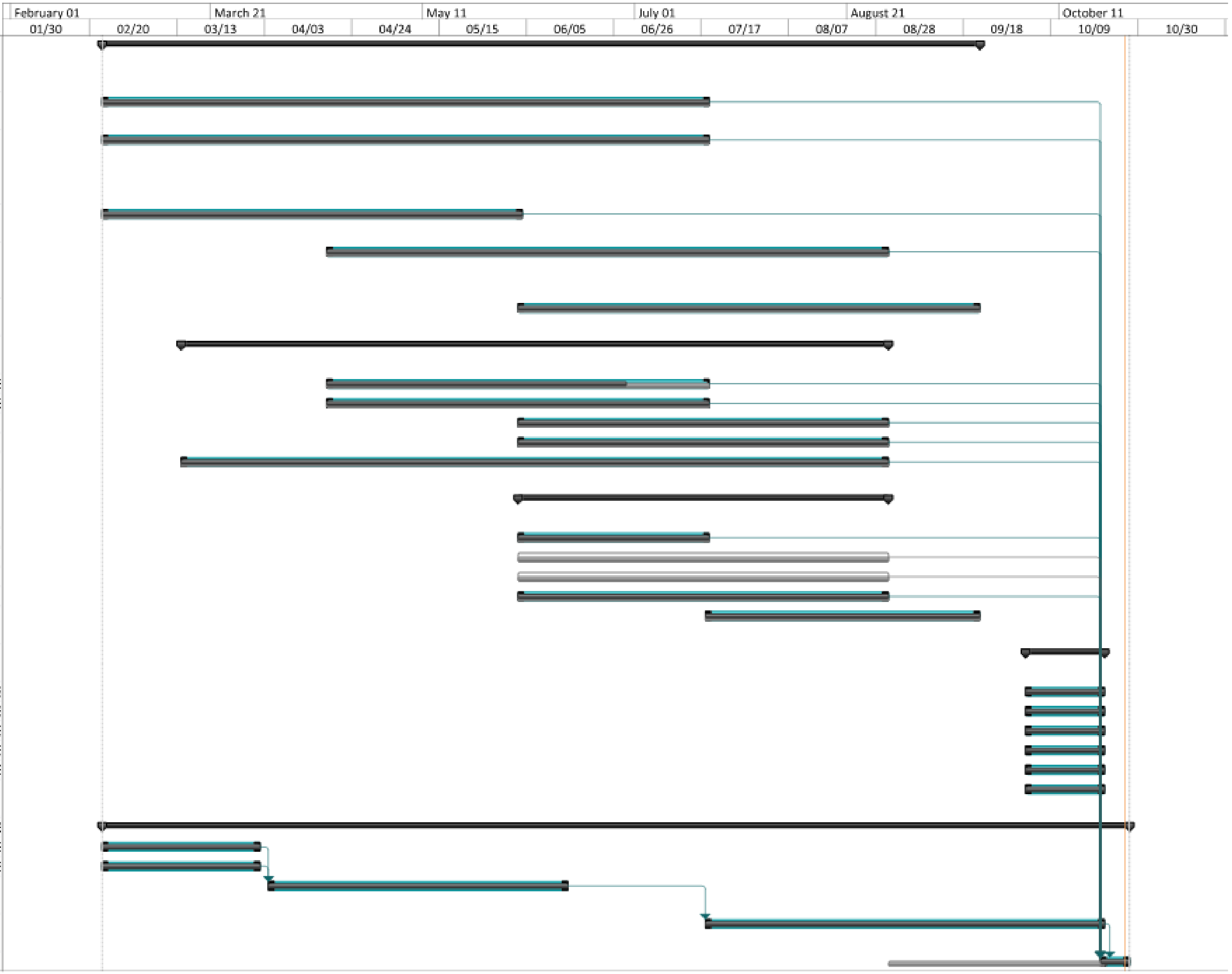
PLC1 = PLC1.dyndns.org:80
PLC2 PLC1.dyndns.org:81

6. EMAIL: Andries
Oor port forwarding

7 Ping leeftyd

APPENDIX D: PROJECT PLAN

ID		Task Mode	Task Name	Duration	Start
1	✓		**Determine Requirements for Transmission Network**	**151 days**	**Wed 11/02/23**
2	✓		Requirements of Telemedicine network	104 days	Wed 11/02/23
3	✓		Research API for OS's and Structure of data transmission services over internet	104 days	Wed 11/02/23
4	✓		Research xAP structures/limitations	73 days	Wed 11/02/23
5	✓		Procedures for network implementation	97 days	Mon 11/04/18
6	✓		Determine limitations of network	79 days	Fri 11/06/03
7			**Determine Possible structure and method**	**122 days**	**Mon 11/03/14**
8			Ethernet network	66 days	Mon 11/04/18
9	✓		Transmission over IP	66 days	Mon 11/04/18
10	✓		Cell phone network	63 days	Fri 11/06/03
11	✓		Other devices	63 days	Fri 11/06/03
12	✓		Testing of transmissions	122 days	Mon 11/03/14
13	✓		**Select/create adequate application for:**	**63 days**	**Fri 11/06/03**
14	✓		Windows	32 days	Fri 11/06/03
15			Cell phones	63 days	Fri 11/06/03
16			~~Other devices~~	~~63 days~~	~~Fri 11/06/03~~
17	✓		Unix or Apple OS's	63 days	Fri 11/06/03
18	✓		Determine adequacy for Telemedicine network	48 days	Mon 11/07/18
19	✓		**Do experiments with xAP packets**	**15 days**	**Mon 11/10/03**
20	✓		Test LAN	15 days	Mon 11/10/03
21	✓		Test ISP to 3G	15 days	Mon 11/10/03
22	✓		3G to 3G	15 days	Mon 11/10/03
23	✓		University network to 3G	15 days	Mon 11/10/03
24	✓		PLC on 3G to 3G	15 days	Mon 11/10/03
25	✓		Server response by hole punching	15 days	Mon 11/10/03
26	✓		**Deliverables**	**177 days**	**Wed 11/02/23**
27	✓		Promblem Statement	28 days	Wed 11/02/23
28	✓		Project Plan	28 days	Wed 11/02/23
29	✓		Project Progress Report	52 days	Mon 11/04/04
30	✓		100% First draft of Project	70 days	Mon 11/07/18
31	✓		Compile final Report	5 days	Fri 11/10/21



Project: ProjectPlan_Telemedicine
Date: Thu 11/10/27

Task	▬	Project Summary	▬	Inactive Milestone	◇	Manual Summary Rollup	▬	Deadline	⇩
Split	┈┈┈	External Tasks	▬	Inactive Summary	▭	Manual Summary	▬	Baseline	▬
Milestone	◆	External Milestone	◆	Manual Task	▬	Start-only	[	Progress	▬
Summary	▬	Inactive Task	▭	Duration-only	▭	Finish-only	]		