Remote Desktop

Integrating multiple devices

DAVID SABATÉ MOGICA

KTH Information and Communication Technology

Bachelor of Science Thesis
Stockholm, Sweden 2008

COS/CCS 2008-15

Remote Desktop

Integrating multiple devices

David Sabaté Mogica
dsm@kth.se

Report for 1F1421:
Project in Computer Communications

29 November 2008

Examiner and Academic Supervisor
Prof. Gerald Q. Maguire Jr.

Department of Communication Systems
School of Information and Communication Technology
Royal Institute of Technology (KTH)
Stockholm, Sweden 2008

Abstract

Electronic devices have acquired an increasingly important role in our society and are integrated in our lives making both the users and their devices more accessible. Currently in the western world most families have at least one computer. This computer is generally equipped with multimedia accessories and an Internet connection. Portable devices, such as mobile phones and PDAs, are part of this technological and social environment. One might think about using a hands free Bluetooth headset together with a mobile phone, to obtain better sound quality, using a keyboard such as the Handykey Corp. Twiddler in order to dial/type quicker or send SMS messages in an easier way, watching a video on a large computer screen that had previously been downloaded to your mobile phone, etcetera. However, there is a problem when it comes to the interconnectivity between all these devices.

Today users face many difficulties when attempting to use what should be the aggregated possibilities of their devices, rather than simply the functionality of each device. The hypothesis of this project is that the user's difficulties could be overcome if their devices could be internetworked.

For example, even though mobile phones and PDAs often have a USB interface, unlike typical desktop or laptop computers these devices have been designed to only be USB slaves -- hence other USB devices cannot be directly attached to them. There are some signs of this changing with the introduction of USB On-The-Go - but we believe that this is a short-sighted evolutionary step.

The obvious solution is to internetwork these devices. For example, by attaching these various USB devices to a computer that is a USB bus master (host) - we can enable the user to use their USB Twiddler with a USB phone. In this way, a user could remotely access the functions of the set of all of their portable devices - without worrying about how to directly interconnect them in pairs. This could enable new functionality, such as the user being able to answer an incoming call to their cellular phone with the keypad of the Twiddler, while using the audio input and output functions of their Bluetooth headset.

We begin by examining a number of means to establish and use remote connections to access systems remotely. We have focused on the most popular desktop sharing systems, specifically those that use remote desktop protocols. Initially we require manual configuration or use of a discovery protocol to identify the different devices. Later we will examine additional protocols, along with some potentially automatic configuration mechanisms.

Sammanfattning

Elektroniska apparater har fått en att starkare position i vårt samhälle, integerationen i våra liv gör dem mer tillgängliga. Nu för tiden har de flesta familjer i västvärlden, minst en dator per hushåll. Datorerna har allt som oftast flera multimedia tillbehör och en internett uppkopling. Handhållna aparater så som mobiltelefoner och PDA'er är också en del av teknologiska och sociala miljön. Kanske vill man andvända en handsfree med blue tooth för att förbättra ljudkvaliteten, eller ett tangentbord t.ex. ett Handykey Corp. En twiddler för att ringa eller skicka SMS. Se på högkavitets video på en stor skärm som du tidigare lastat ner till din mobiltelefon, etc. Hur som hälst uppstår det problem vid sammankopplingen mellan olika tillbehör.

Dagens användare får en stor utmaning i användade av den kombinerade funktionen i stället för den ensamstående apparatens funktioner och förmåga. Hypotesen av detta projekt är att detta problem kan övervinnas om alla apparater var ihopkoplade via internetwork.

Till exempel, även om de flesta mobiltelefoner och PDA'er har USB gränssnitt, har dem tillskillnad från vanliga stationära datorer och laptops bara designats för att vara USB-slavar. Detta betyder att USB-tillbehör inte kan kopplas direkt till apparaten. Det finns tecken på tekniker som t.ex. USB On-The-Go men detta ser ut som en kortsiktig lösning.

En uppenbar lösning är att parkoppla dessa apparater via internetwork. Om man t.ex. kopplar alla dessa USB-tillbehör till en stationär eller bärbar dator, kan vi låta användaren komma åt dessa via sin telefon med USB. På så sätt kan användaren trådlöst komma åt alla tillkopplade tillbehör, utan att oroa sig över att para samman dem fysiskt. Möjligheten för nya funktioner och användningsområden visar sig då, som t.ex. att svara på inkommande samtal med den tillkoplade twiddler'n medans samtidigt tala via det trådlösa headsettet.

Vi börjar med att undersöka olika tekniker för att upprätta trådlös tillkoppling till olika system. Vi har focuserat på de populäraste datasystemen, framför allt de som använder trådlösa protokoll. Sådana som kräver manuell installation eller använder upptäkande protokoll för att identifiera olika tillbehör. Senare visar vi exempel på andra protokoll med några potensial att automatisk konfigurera tillkopplingen.

Table of contents

List of acronyms and abbreviations

A2DP	Advanced Audio Distribution Profile
AP	Access Point
API	Application Programming Interface
CoRRE	Compact RRE
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FAQs	Frequently Ask Questions
GPS	Geographic Positioning System
HTTP	Hypertext Transfer Protocol
IEEE	Institute for Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPsec	Internet Protocol Security protocol
L2CAP	Logical Link and Adaptation Protocol
LAN	Local Area Network
MAC	Media Access Control
N/A	Not applicable
NX	Next-generation X
ORL	Oracle Research Laboratory
PC	Personal Computer
PDA	Personal Digital Applicance
PIN	Personal Identifier Number
RC4	Rivest Cipher 4
RDP	Remote Desktop Protocol
RFB	Remote Frame Buffer
RFC	Request for Comments
RRE	Rise-and run-length encoding
RSA	Rivest-Shamir-Adeleman (the name of both an algorithm and a company)
RTP	Real-time protocol
SIM	Subscriber Identification Module
SMS	Short Message Service
SSH	Secure Shell
SSID	Service Set Identifier
TLS	Transport Layer Security
URL	
USB	Universal Serial Bus
UUID	Universally Unique Identifier
VNC	Virtual Network Computer
VOIP	Voice over IP
VPN	Virtual Private Network
WEP	Wire Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access version 2
Wi-Fi	Wireless Fidelidy
Win32	Microsoft’s Windows API
ZRLE	Zlib run-length-encoding

1 Introduction

Today, a typical user can have at their disposal portable devices such as the latest generation mobile phones and/or PDAs. These portable devices are able to offer a vast number of applications that would have been hard to imagine not so long ago, such as music, video and Internet connectivity, amongst others. Aggregating other devices to this scenario, for example a Handykey Twiddler [40] or a Bluetooth headset, a user could use these device together, taking advantage of every device; resulting in device *aggregation.*

This aggregation can be done physically, for example connecting the Twiddler to a computer via USB cable or remotely, via internetworking (taking advantage of wireless technologies, for example using a remote access system to communicate a PDA and a computer). Figure 1 shows an example of accessing a shell running on a remote linux PC from a PDA using VNC (see section 2.2.1).

Figure 1: Screenshot from a PDA running VNC.

Device aggregation offers several different scenarios. For example, a user working on a computer receives a mobile phone call while the user's mobile phone is a few meters of distance away. This user might be able to see this incoming call on the computer that they are using via a remote desktop client application running on this computer connected to the corresponding desktop server running on the user's phone. Thus the user might answer the call up by clicking on an icon on the computer's screen, then communicate using a Bluetooth headset connected to (paired with) their cellular phone.

1.1 Discovering devices

In order to communicate with new devices, users need these devices to know about each other. Two different alternatives are available: traditional manual configuration and use of a discovery protocol.

In a manual configuration, the details will depend upon the transport layer used. In the case of local access, the device might be accessible by name, while in a non-local access such as via the Internet, the user must enter the IP address (or the fully qualified domain name) of the device that they want to communicate with.

2 Background

2.1 Device Aggregation

We have focused our work on remote control of a device. More specifically, we have focused on the Graphical User Interface, for example: using a cording keyboard, such as the Handikey Twiddler [40] with a cell phone.

Why is device aggregation difficult?

- Although the cell phone has a USB interface it is designed to be only a USB client
- Meanwhile, the Handikey Twiddler is also another USB client

We require a USB master to interconnect these devices

2.1.1 Wired device aggregation

We can use wires to connect the various devices together. For example, using a laptop we can connect devices with client USB interfaces to a laptop as it is capable of being a USB master. An example of the devices that we might aggregate this way is shown in Figure 2.

An advantage of this approach is that it is simple for the user and most modern operating systems support automatic detection and configuration of USB devices when they are first attached to the computer. The disadvantages are that we need to make physical connections between all of the devices and we need a USB host (master). It is this later aspect which is the most vexing problem as few PDAs, cellular phones, etc. implement being a USB host.

Figure 2: Devices for wired device aggregation. Note that the two items in the top row (a PDA and a cellular phone) are USB clients only, while the PC is a USB host. Combining the cellular phone with the laptop enables the display of the cellular phone as shown in the figure on the bottom row on the right.

2.1.2 Wireless Device Aggregation

We can also use wireless links to connect multiple devices together. The advantages of this are that we do not need to use cables to connect the different components, but the disadvantages are that we now have to address the problems of (1) Device discovery and (2) Addressing.

As an example of wireless device aggregation, Figure 3 shows the screen of a cellular phone being operated from a laptop. The details of this example will be explained later in this thesis.

Figure 3: Cellular phone's screen as viewed on a laptop

An additional disadvantage of wireless device aggregation is that the cost of developing a wireless version of a device can be expensive. For example, Chris George of Handikey Corporation on 23 July 2007 estimated that developing a Bluetooth version of the Handykey Twiddler would cost US$60,000 [40].

2.2 Remote Desktop Applications

One of the possibilities for device aggregation is to internetwork devices via the use of remote desktop applications [21]. These applications give a use the possibility to connect to and use their devices remotely through the Internet or some other kind of network. Remote desktop applications provide remote control of applications that normally utilize the system's graphical user interface (typically referred to as a desktop), virtually adding new capabilities and services to a device that is executing the remote desktop client application. While remote desktop applications are frequently used for troubleshooting – by enabling a remote technician to see what the user's sees and to enable them to provide input and interact with the programs, but without need to be physically present at the desktop; such remote desktop application

can also be used by a client running on a PDA to enable control of a powerful program that is not available for a PDA and for which the PDA would not be able to provide sufficient local resources to run (even if a version of the application were available for it). Similarly such a remote desktop application could be used by a user to write a SMS using a computer's full size keyboard, making text entry much easier, but with the SMS application actually running the user's cellular phone.

Remote desktop applications protocols work on the application layer, generally over TCP/IP, and they are responsible for communication between the remote desktop server and the remote desktop client. Schematically this can be seen in Figure 4.

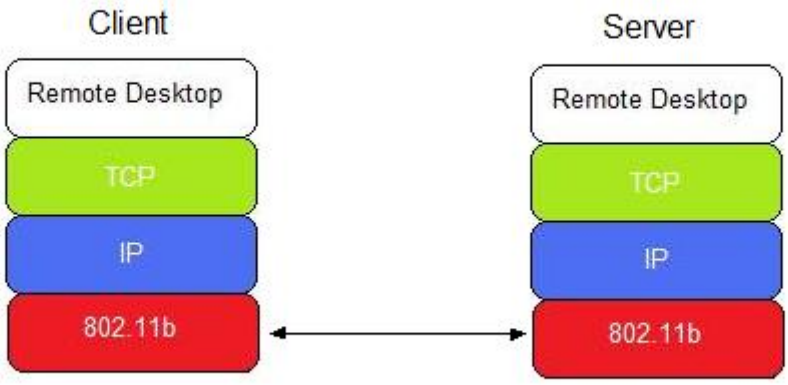

Figure 4: TCP/IP stack on a typical remote desktop scenario

A variety of protocols can be used below the application layer in order to best support this application protocol. In our example we illustrate a typical remote desktop scenario, which utilizes TCP/IP using a wireless local area network (WLN) wireless link. However, we could use a remote desktop protocol such as VNC (see next section) together with RTP[1] providing multimedia – both protocols could be running over UDP – in order to enable multicasting of the contents to multiple devices or to provide reduced delay in the event of lost packets.

With regard to security, what security is necessary (or desirable) and what security protocols should be used and on what layer or layers should they be applied. For example, is be better to secure the application layer using SSH or to provide security at the transport layer using TLS[2]? Should we create a virtual private network (VPN) between our devices using an IP layer security protocol, such as IPsec? Or should be limit access to only those devices a known MAC[3] address, by using an access control list.

---

[1] The Real-time Transport Protocol provides transport functions suitable for applications transmitting real-time data, such as audio and video.[18]

[2] The Transport Layer Security Protocol that provides secure communications using cryptography [20]

[3] A Medium Access Control layer address in the case of IEEE 802 interface is a 48-bit identifier for a network interface.[19]. Note that this interface may or may not claim be unique – depending upon whether the unique bit is set or not. It is important to note that many devices allow the user to set a

2.2.1 VNC

Virtual Network Computing (VNC) is a technology that allows users to view and interact remotely with one computer's desktop (via a server program) using a client-program (called a "viewer") on another computer. VNC is mainly used for system administration (e.g. troubleshooting), flexible hot-desking, and educational purposes.[22]

It is possible to use VNC on a wide variety of different types of computers and operating systems [24]. One of the first activities in this project was to examine its behaviour on our test devices when performing tests to understand its possibilities and limitations. The following paragraphs will describe how the protocol works and what security it offers.

2.2.1.1 RFB protocol

VNC uses the Remote FrameBuffer protocol [23], a so-called "thin client" protocol; as most of the responsibilities are delegated to the server, therefore clients can run in devices with very limited resources and it is generally very easy to implement clients. Because the protocol works at the framebuffer level there is no need for the client to understand any of the semantics of the graphics operations or to perform any sophisticated rendering. Additionally, this means that the protocol is independent of the operating system, the windowing system, and the applications! The protocol simply describes updates that are to be made to rectangles of a remote framebuffer to give it the appearance of the framebuffer at the server.

The client is stateless, thus users can disconnect from a session and reconnect to same session at the point that they left (i.e., before disconnecting) – without needing to preserve any state information. The use of such a stateless protocol enables the session to easily be shifted from one device to another (allowing session mobility) or to allow the session to utilize multiple devices.

As a result of the stateless nature of the protocol, updates are demand-driven by the client; in response the server sends display updates represented by a sequence of rectangles after every input event. Note that a benefit of this is this naturally adapts to the bandwidth between the server and the client and to the performance of the client. Additionally, since a high performance display system at the server might make many updates to its local framebuffer – a slow client or link will simply see the effect of the aggregate of changes and will not have to perform operations at the speed of the server's local display subsystem.

---

new MAC address to be used with this interface, thus one can not count on this identifier actually being unique or that it can not be counterfeited by an attacker.

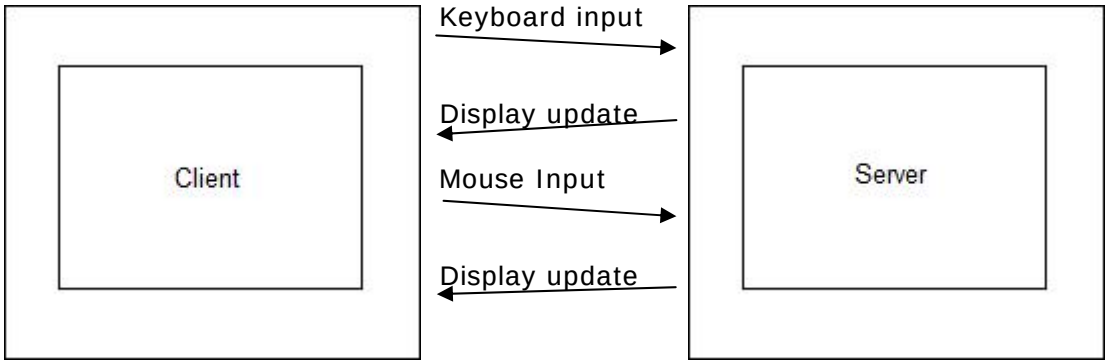

Figure 5: Update sequence

2.2.1.1.1 Representation of pixel data

Interaction between the RFB client and server involves a negotiation of the format and encoding of the pixel data, i.e., how the pixel data within a rectangle will be sent. The server must provide pixel data, following client requests.

The client can decide amongst 24-bit, 16-bit, or 8-bit pixel formats and 5 different types of encoding (Raw, Copy Rectangle, RRE, CoRRE and Hextile). Raw encoding is useful in scenarios where the client and the server are located on the same machine, Copy Rectangle is useful for repetitive patterns, a two dimensional version of run-length encoding (RRE) is useful for large blocks using same colour, CoRRE is an improvement of RRE, and Hextile & ZRLE offer the best performance on high-speed networks. Some programs include their own encoding format, such as Tight which is implemented by TightVNC.[23]

2.2.1.2 RFB Protocol Messages

There are two steps to the protocol: the initial handshaking and the actual interaction. After initializing a communication session with the server (shown in Figure 6) the client can send messages and may receive messages from the server. These messages begin with a message-type byte, followed by message-specific data (if any).

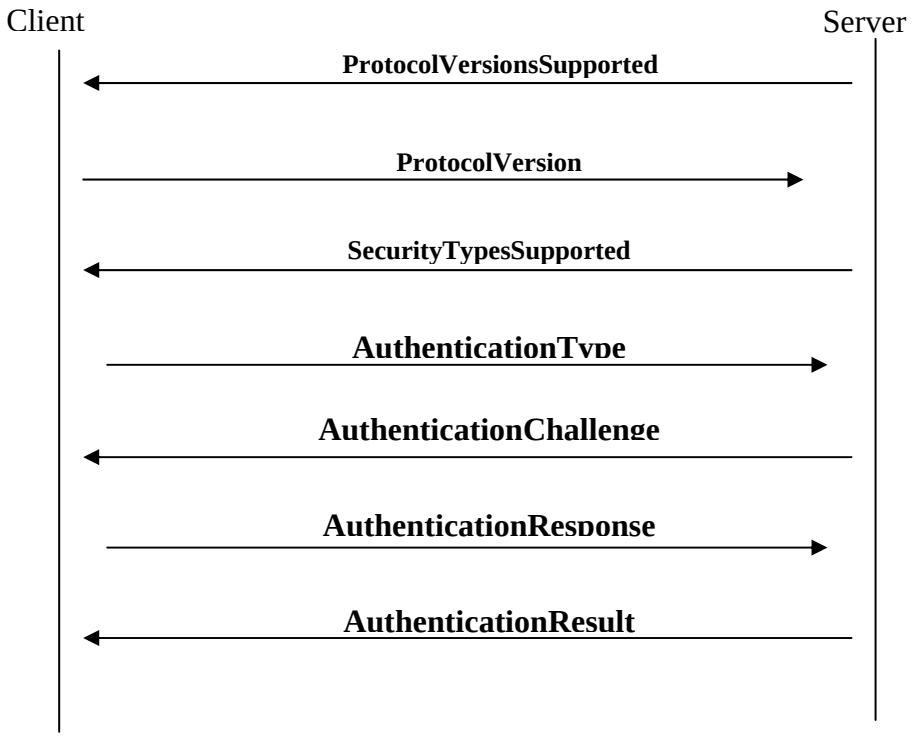

Figure 6: Initial Handshaking

2.2.1.3 VNC Security

Generally, VNC applications do not provide secure, thus communications is not encrypted. However, it is possible to tunneling the communications over Secure Shell (SSH) (see section 2.2.4.1). This is a typical means to increase the security of such communications – without the need to incorporate encryption into the application itself.

2.2.2 Terminal Services

Terminal services are part of Microsoft® Windows operating systems. For example, a client application for Terminal Services is installed by default on Microsoft® Pocket PC 2003 on the PDA we have used in our tests.

Similar to VNC, this technology allows users to access to applications and files stored on another computer over a thin client terminal connection to a network attached server. Clients running on a variety of different operating systems can connect to servers using the Remote Desktop Protocol.[25][27] It should be noted that the terminal services client is not called the Remote Desktop Connection.

Other interesting features present in RDP that are potentially interesting for use are: sound support and remote printing.

2.2.2.1 Remote Desktop Protocol

Terminal Services uses Remote Desktop Protocol (RDP) protocol, a multiple-channel capable protocol that allows for separate virtual channels for carrying device communication and presentation data from the server, as well as encrypted client mouse and keyboard data.[26]

On the server, RDP uses its own on-screen keyboard and mouse driver to receive the input events. A video driver is responsible for translating these events in the display output to convert into a network packet by using the RDP protocol and sending them over the network to the client. On the client, the data is received by RDP and interpreted into Microsoft® Win32 Graphics Device Interface API calls. This data sent by the server machine has been compressed in order to provide faster transmission.

2.2.2.2 Terminal services security

For security, RDP uses encryption based on the widespread RSA[4] RC4 software stream cipher.[32] Although an efficient encryption algorithm, it may be vulnerable to attacks.[21] Again forwarding the connection over SSH can be used to increase the security of the communication between client and server.

2.3 Client-server networked windowing systems

2.3.1 X Window System

The X Window System [28] was the first remote desktop environment to be released in the 1980s. It is a network transparent windowing system that has been implemented on a large variety of computer architectures and operating systems. The X Windowing system went through a number of revisions, the most well know is the 11th version, commonly referred to as X11. The X windowing system reverses the role of client and server from the system mentioned above. In the X windowing system the server is running on the computer where the user (and screen) is, while the client runs on the remote machine and emits X windowing protocol requests. The X server receives these requests and renders them on the screen.

One of the main differences from the above applications is the ability to use the X windowing system with independent window managers. The window manager is responsible for the screen layout and management.

Another major difference is that the X Windowing protocol is quite complex and has a very large number of functions, along with a number of optional extensions. Thus implementing the X Windowing system requires considerably more effort that the systems described above. However, the source code for the systems is freely available, along with ports for nearly all platforms. Thus despite its complexity it is widely available.

The X Window System does not provide any encryption mechanism, thus tunnelling the connection over SSH will be again a solution to avoid malicious users sniffing transmitted packets. SSH is so commonly used together with the X Windowing System that one of the command line options to SSH is the “-X” option, which indicates that you want SSH to offer pass-through of X commands from the remote system to the local X Windows server.

2.3.1.1 NX

X Window connections can be improved using NX, that compresses the X11 protocol enabling it to be used on low bandwidth links. Additionally, information that

[4] RSA, the Security Division of EMC Corporation. **www.rsa.com**

has already been accessed by a user is shown instantly due to special caching techniques. Furthermore, this information is tunnelled over a SSH channel with data flow control, providing a faster and more secure channel.[29] [30]

2.4 Secure Shell tunnelling

As we have mentioned before, remote desktop applications do not normally offer reliable security. Secure Shell (SSH) [31] is a protocol/application that provides security to remote login and other network services over a non-secure network tunnelling[5] data.
SSH consists of:

- Transport Layer Protocol: provides server authentication, confidentiality and integrity, and optionally compression. It normally runs over a TCP/IP connection.
- User Authentication Protocol: authenticates the client to the server. It runs over the transport layer protocol.
- Connection Protocol: multiplexes the encrypted tunnel into several logical channels. It runs over the user authentication protocol.

2.5 Bluetooth

Handheld devices often utilize Bluetooth technology to avoid wires. Bluetooth adapters are also available for laptop and desktop computers. Bluetooth allows 2 to 7 devices to exchange messages and files in a secure way. It uses a globally unlicensed ISM 2.4 GHz short-range radio frequency link, whose bandwidth that can reach up to 3 Mbps in some versions of the protocol.[9] Typically Bluetooth communication will be less than 720 kbps with 10 m of distance between the devices.

Such a short range wireless connection simplifies configuration and the process of device location (since due to the short range, the pair of devices can not be far away from each other). Its key features are robustness, low power, and low cost.[2]

2.5.1 Bluetooth physical layer

Bluetooth uses Frequency Hopping Spread Spectrum, a signal transmission technology switches a narrowband carrier signal over a wide band of frequencies using a pseudorandom hopping sequence.[3][4] This frequency switching reduces interference with other radio signals.[3][5][6]

2.5.2 Discovering devices. Inquiry Procedure

Initially, a Bluetooth device is a peer unit *listening* to the network. One Bluetooth device, known as a master, starts to establish a connection by sending an inquiry to all the devices within range. Bluetooth devices acting as slaves or access points each answer with their own address. Up to 7 active slave devices and up to to 255 devices[6] can be connected in stand-by mode to a master device.[2][6][8]

---

[5] Tunneling is the process of forwarding selected TCP ports through an authenticated and encrypted tunnel

[6] This is due to devices in a Piconet having a 3 bit logical address, so the maximum number of devices is 8.Up to 255 devices can be connected in stand-by mode

The Personal Area Network [7] formed by these devices is called Piconet. An example of such a piconet is shown in Figure 7.

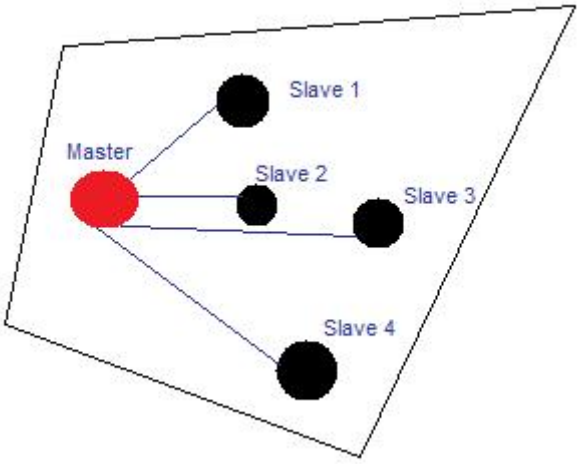

Figure 7: Piconet

Up to 10 Piconets can be in the same physical space. Two or more Piconets can form an *ad hoc* network called scatternet. The device participating in both Piconets will relay data between members of both networks.[6][8]. An example of a scatternet is shown in Figure 8. However, we should note that now devices support scatternets and they are not relevant to our project, so no further mention will be made of them in this thesis.

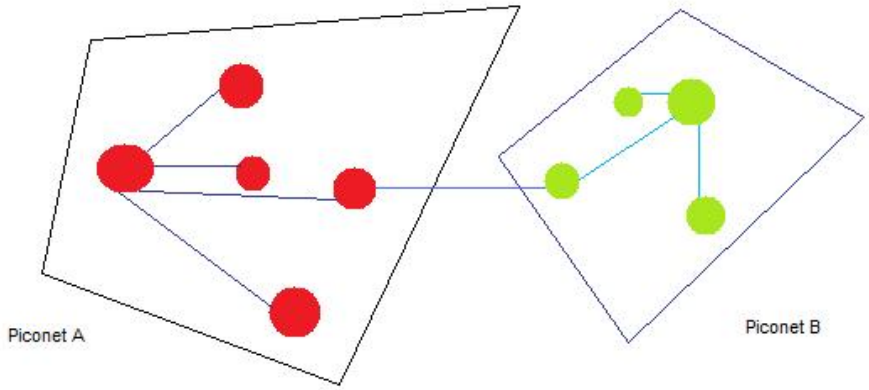

Figure 8: Scatternet

2.5.3 Discovering services

After the master has chosen a slave, it will try to find out which services the slave device offers. The Service Discovery Protocol, as its own name states, is responsible for this service discovery task.[6][8]

Bluetooth services include positioning (location identification), networking (LAN, Ad Hoc,…), rendering (printing, speaker, …), capturing (microphone, scanner), object transfer (v-Inbox, v-Folder,…), audio (speaker, microphone, headset, …), telephony (cordless telephony, modem, headset service, …), information (WEB-server, WAP-server, …) and the temporary use service “Limited Discoverable Mode”.[10]

These services might be freely accessible from other devices or can require authorization and or authentication.

Note that one of the disadvantages of the Bluetooth discovery method is that it takes a very long time to discover a device (up to ~10 seconds), thus in the case of moving devices due to the limited range the user might pass a device before discovering it. Thus for the purpose of this thesis project we will assume that the user (and their device(s)) remain stationary while discovering a device. A solution to increase the speed of device and service discovery is described in Cécile Ayrault's masters thesis[41].

2.5.4 Identification and authorization

Services are represented by profiles, with a 128-bit Universally Unique Identifier (UUID). A profile defines a set of procedures and messages for a specific use of a Bluetooth device. The most frequently used profiles [9] are: Advanced Audio Distribution Profile), Dial-Up Networking Profile, Generic Access Profile, Hands-Free Profile, Headset Profile, Human Interface Device Profile, Object Exchange Profile, Object Push Profile and Serial Port Profile.

Once the discovery process is done, the master device is ready to create a communication channel between the slave device and itself, using the Logical Link and Adaptation Protocol (L2CAP) protocol. It is also possible to use another channel that works over the L2CAP channel, to provide a virtual serial port.

The access point can include a PIN based key security mechanism to restrict access to authorized users. If the safe mode is activated, then the PIN code will be sent encrypted with a second key in order to reduce risk or interception of the PIN code and it exploitation by an attacker. Once the pairing is done (i.e., the master and slave have agreed to communicate and have exchanged the relevant security information), then the master device use the newly established communication channel.[6]

2.5.5 Security

Three security modes are defined by device manufacturers. These are:

- Security mode 1: Non-secure mode.
- Security mode 2: Service level enforced security. Two devices can establish a non-secure Asynchronous Connection-Less link. Security procedures (authentication, authorization and optional encryption) are initiated when a L2CAP channel request is made.
- Security mode 3: Link level enforced security. The Bluetooth device initiates security procedures before the channel is established.[11][12]

2.6 Wi-Fi

Wireless Fidelity (Wi-Fi) technology is a branding and marketing name for interoperable IEEE 802.11 devices – that have been designed to provide wireless network connections for home computers and nearly all laptop computer; and it is also increasingly incorporated into PDAs and the latest generation cellular phones. Much faster connections than Bluetooth offers are provided – up to 54 Mbps – along with a greater range, reaching 140 meters under ideal conditions, making Wi-Fi a common solution for wireless services that require fast connections.[13][14]

2.6.1 Signal Transmission. Carrier-Sense Multiple Access/Collision Avoidance

Wi-Fi uses Carrier-Sense Multiple Access with Collision Avoidance. Therefore, devices will not transmit unless the channel is idle, reducing the number of collisions. Because collisions are harder to detect on a wireless link than in a wired Ethernet network, Wi-Fi provides two other mechanisms to avoid them: the Distributed Foundation Wireless MAC protocol and optionally the Point Coordination Function.

Currently IEEE 802.1g, which is compatible with the IEEE 802.11b version, are the standards for Wi-Fi devices. These two versions of the IEEE 802.11 standard work in the same frequency as Bluetooth (2.4 GHz) but using 11 Mbps bandwidth and Direct Sequence Spread Spectrum technology to limit interferences with other devices. The IEEE 802.11b/g standards specify how Wi-Fi devices communicate with each other.[17]

2.6.2 Operation Modes: Infrastructure and Peer-To-Peer

Two modes can be used to interconnect Wi-Fi devices: Infrastructure and Peer-To-Peer (*ad hoc*) mode. In infrastructure mode a devices needs to connect to an Access Point which forwards traffic to other devices (in the wired or wireless network), while in *ad hoc* mode devices communicate directly between themselves without an access point – thus offering a highly portable solution.[17] These two modes are illustrated in Figure 9 and Figure 10. While it might seem that ad hoc mode would be most useful for establishing wireless network connections with nearby devices, if there is an access point present and the user's device has associated with this access point, then it may be appropriate to use infrastructure mode to communicate with the other nearby devices.

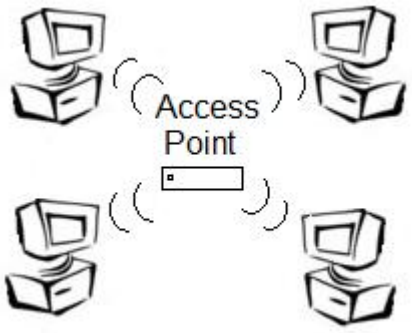

Figure 9: Infrastructure mode

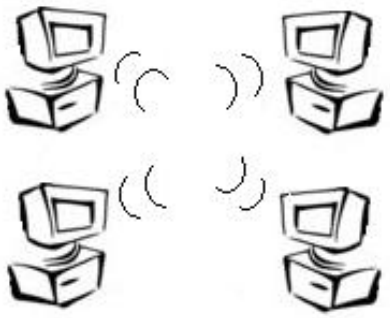

Figure 10: Peer-To-Peer (Ad Hoc) Mode

2.6.3 Security

WLAN security has focused on access control and data privacy. Strong access control is desired to deny authorized users the ability to communicate through access points. The primary focus of data privacy has been to prevent users from eavesdropping on a device's communications.

Wireless networks are identified by their Service Set Identifier (SSID). This SSID is generally broadcast by an access point so that devices can choose to which access point they wish to associate. Some have suggested that for security purposes that this

broadcast should be disabled, but this only provides a false sense of security - as an attacker can listen to other communications to learn the SID.

In this thesis we will only consider the user of access control lists for denying unknown devices from communicating (see section 2.6.3.3). Other techniques for controlling access via access points in a fixed infrastructure have been or are being addressed in other concurrent theses (such as [42]).

There are two types of authentication: open network and shared-key. Any Wi-Fi device can connect to an access point running in open authentication mode (see Figure 11).

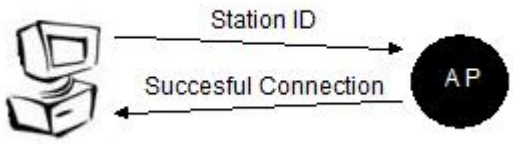

Figure 11: Open Network Authentication.
Both stations are authenticated at the end of the sequence.

2.6.3.1 Wired Equivalent Privacy

Wired Equivalent Privacy (WEP) was intended to provide encryption to Wi-Fi devices, but its weak use of the RC4 cryptographic algorithm was quickly found not to be a secure solution[43]. Although Wi-Fi Alliance released later Wi-Fi Protected Access (WPA) and declared WEP deprecated, many Wi-Fi devices continue to use WEP as their only security.

WEP can be used for encrypting the data frames after authentication and association. Then, only clients that try to associate to the AP using the right WEP key will be able to send and receive data.

Shared-key authorization mode is based on the fact that the two stations that take part in the authentication process share a previously configured key.

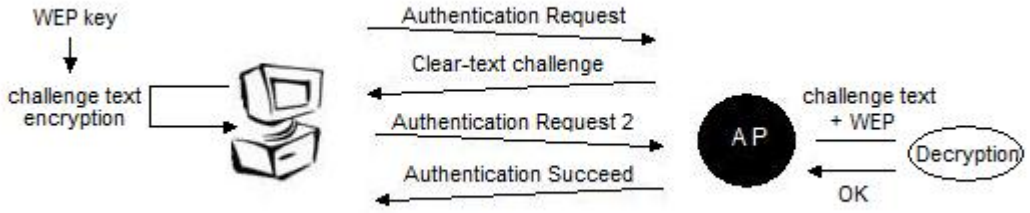

Figure 12: Successful Shared-Key Authentication

Although open network authentication mode might seem at first sight more insecure than shared-key, this is not actually true. The key used for the shared-key authentication could be decrypted by a user by capturing the challenge frames, thus obtaining the key for the whole WLAN.[34]

2.6.3.2 WPA/WPA2

Wireless Protected Access is based on the IEEE 802.11i standard. It was designed to be an improvement over WEP in terms of access control and data protection, by

providing stronger data encryption and user authentication. The goal was to enable users to forget about the need to deploy complementary security solutions in a Wi-Fi environment, such as configuring a Virtual Private Network.

WPA encryption utilizes a Temporal Key Integrity Protocol that uses 128-bit dynamic keys, in contrast with WEP's static keys. These keys are generated and distributed by the server such that the key that will be used by a node to communicate is no longer predictable. In addition to this, a Message Integrity Code will protect the integrity of the encrypted information.

In infrastructures where an authentication server is available, IEEE 802.11x together with an Extensible Authentication Protocol framework is increasingly employed for authentication (for details see [42]). In scenarios where an authentication server is not affordable or not available, a Pre-Shared Key (PSK) can be utilized, to act as a shared password that must be entered in the device and the AP – thus this share secret can be used for authentication.

WPA2 uses a different encryption scheme, Advanced Encryption Standard, in order to provide 256-bit keys.[33][34]

2.6.3.3 MAC address filtering

In our tests using an access point that is not attached to the universities network, we utilized MAC filtering to identify a device based upon its MAC address. The network administrator configures the access point with a list of all the authorized MAC addresses. Only these devices will be able to access the WLAN and interact. This method improves security and offer a good solution for (very) small networks [35], but it is possible to by-pass this protection by simply sniffing the traffic and masquerading as one of the permitted devices. Thus providing better security relies on using another solution, for example enabling WPA encryption or utilizing a VPN between communicating devices.

2.7 Dynamic Host Configuration Protocol (DHCP)

Devices acting as clients make use of the DHCP protocol to get their network parameters such as an IP address, subnet masks, a gateway, address of a DNS server, and so on. In our tests, we have used a DHCP server installed on a SmartBadge (for details see [44]) to allocate fixed IP addresses to the devices that connect via the test access point. This facilitated creating the test configurations that were desired and might also represent the method of address allocation that could be used in some future scenarios. DHCP utilizes UDP port 67 (for the server) and 68 (for client).[36]

2.7.1 Client – Server Interaction

The client broadcasts a message on the interface to find available DHCP servers (see Figure 13). When a DHCP server receives this request (a DHCPDISCOVER message), it allocates an IP address for this client and sends an IP lease offer (a DHCPOFFER message) that includes the client's MAC address, the assigned IP address, the subnet mask, the lease duration, and the DHCP server's own IP address. After the client receives this message, it broadcasts a message to all the servers telling them that it has accepted an offer from a specific server (identified by the selected server's IP address), so that the other servers can offer the address previously offered to this client to other clients.

The configuration process ends with client sending a message (a DHCPREQUEST message) acknowledged by server (with a DHCPACK message) that includes the lease duration and information regarding configuration that the client requested.[36]

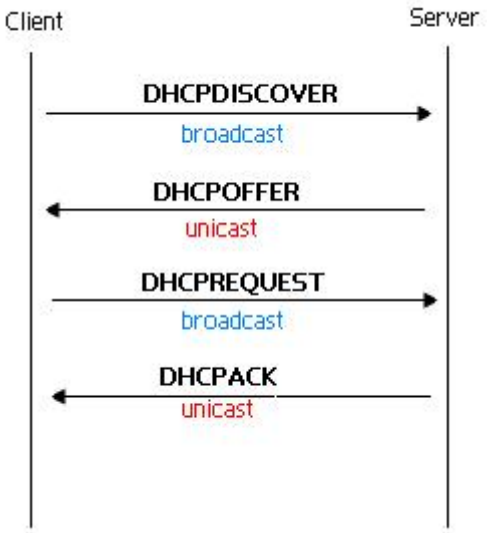

Figure 13: DHCP Client-Server interaction

2.7.2 IP address allocation modes

IP addresses can be assigned in a number of ways (see Table 2-1). Our DHCP server has used a static allocation to assign IP addresses to our test devices.

Table 2-1: Methods of assigning IP addresses via a DHCP server

Manual	the client is configured with a user-specified static IP address.
Static	The DHCP server only allocates addresses to a set of clients that are listed in a table with their MAC address paired with a defined IP address.
Dynamic	Dynamic allocation allows IP addresses to be reutilized dynamically. A range of IP addresses is for temporary assignments, forcing the client to request an address after the lease time is up.
Automatic	Automatic mode is similar to dynamic allocation, but in this case the server preferentially allocates to a client the same IP address that the client was using before. It is useful when the number of clients does not change at all.[36]

2.8 Comparison

In this section we will summarize and compare the three main remote desktop solutions which we have considered. Table 2-2 compares the three with regard to which host operating systems the software is available for and if the software is open source. We can briefly summaries the three with the following bullets:

- VNC
  - Free, Multi-platform and Open Source (original)
  - Good performance on the Internet
- Microsoft Terminal Services
  - Includes encryption based on RC4
  - Good performance on LANs
  - Sound support and support for remote printing
- X Window System
  - Free, Multi-platform and Open Source (original)
  - *NX more secure and faster*

Table 2-2: Comparison of different remote desktop applications

	Unix	Windows	Pocket PC	Symbian	Open source?
VNC	Server & client	Server & client	Server & client	Server & client	Yes
RDP	Server & client	Server & client	Client	Client	Partially yes
X Window System	Server & client	Server & client	Client	N/A	Yes

3 Method

3.1 Design

We have a number of choices to make at each of the layers, specifically these layers are:

- Application
- Transport
- Network
- Physical/MAC

Thus we can see that our remote desktop must utilize the support of the lower layers and its properties (in terms of performance, reliability, etc.) will partially derive from the choice of these lower layers. In addition, we need to consider what security will be used with each of the layers; some of these alternatives are shown in Figure 14

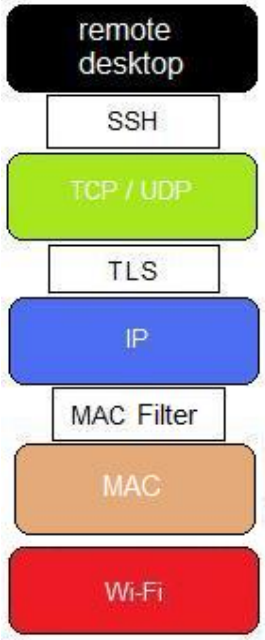

Figure 14: Protocol stack - with examples of security shims

VNC was selected as the basis for remote display of the desktop for the following reasons:

- Fast connections
- Open source
- Free solution
- Well supported (specifically, it was available for all the devices which we wished to use)

3.2 Implementation

On our first tests try to access the remote desktops via a Wi-Fi network. For example, we will use the KTH-Open network, which is an open WLAN that employs the captive portal technique for access control (i.e., the user's browser will be redirected to a login page – where the user can login with their university user name and password). In these tests after our devices associate with an access point, this access point will forward the data (using infrastructure mode) to the other Wi-Fi equipped devices that we are using. We have also performed some tests using a *private* WLAN, using a SmartBadge running a DHCP server; this server assigns new IP addresses to our known devices set on a MAC addresses list, providing only access to the devices listed (for details of this DHCP server and its configuration see [44]).

3.2.1 VNC configuration

3.2.1.1 Client

Generally, the only data initially required to connect from a VNC client to a server are the VNC server's fully qualified domain name (or IP address) and the number of the display that user would like to see; this information is given in the form of "[IP_address | computer_name]: [display_number]".

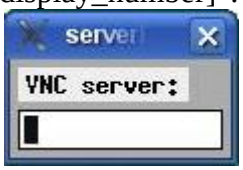

Figure 15: VNCViewer first screen

Other options such as preferred encoding, colour depth, user name and password, and protocol version may also be configured or passed as command line options.

3.2.1.2 Server

The VNC server is the most important and perhaps most difficult part of the software to set up. This occurs because of VNC's use of a "thin client" architecture, thus the server must be responsible for sending updates to the client based upon changes to the framebuffer. Configuration involves:

- Selecting the displays to be shown remotely;
- Set up a password and/or decide which devices can access this server, in order to avoid unwanted and undesired connections;
- Select an appropriate image quality, note that 32 bits mode is not supported by the iPAQ, and that 16 bits provides better image quality but is slower than 8 bit framebuffer depth
- Select the type of encoding to be used
- Change the display resolution; for example, they will be able to obtain optimal results in a small screen by lowering its resolution, thereby extracting information from the high resolution computer. If we use an iPAQ to access a desktop with a resolution of 1280x1024, approximately 1 minute is needed to connect and to display this desktop - rendering it practically useless. However, if we reduce the

resolution to 230x260, less than 10 seconds are needed to connect and the speed of updated is perceived as quite adequate.

3.3 Testing

3.3.1 Establishing communication

First of all, the devices to be used as clients need to be able to reach the intended server machine, meaning devices need a TCP/IP network connection via a WLAN, LAN, or via the Internet. In this thesis, we will utilize a WLAN in infrastructure mode to provide WLAN access to our devices. More specifically we will use IEEE 802.11b compliant devices and access points.

MAC filtering was initially used as a security option when we tried to connect to a private access point in the lab. However, we quickly found that this is not a secure solution; as we can see in Figure 16, Wireshark displays the MAC address of the devices access this access point! Implementing some other form of Wi-Fi security, for example WPA2, will potentially increase our wireless security; although even though this may be decrypted by Wireshark in its last versions (if it can learn the key). However, our PDA can only support WPA – thus ruling this last option out.[39]

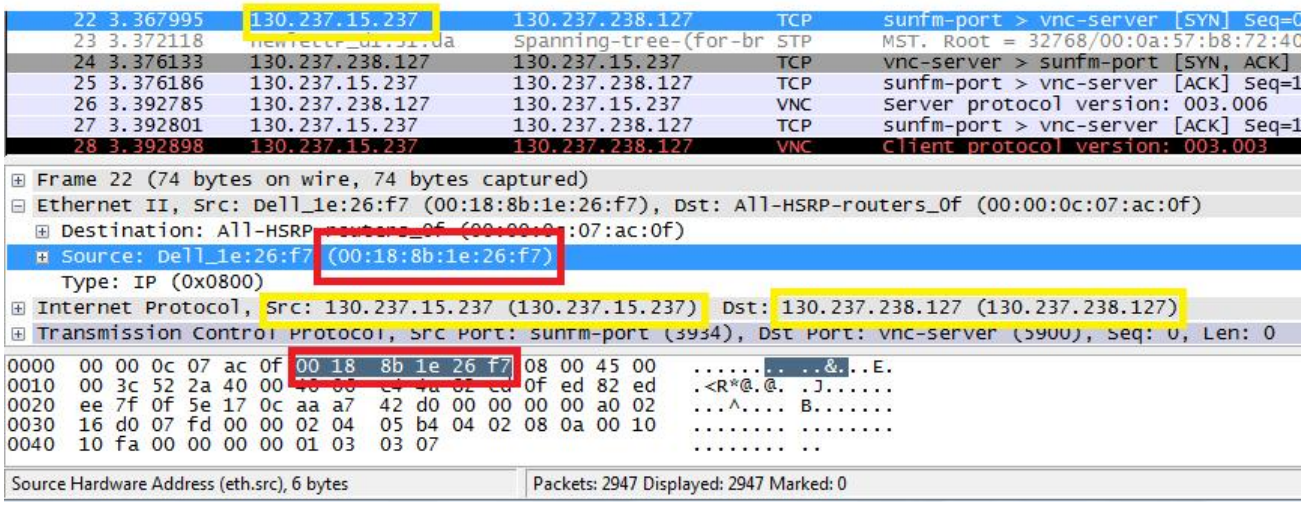

Figure 16: Wireshark capture of MAC and IP addresses

3.3.2 Addressing and connectivity

In order to communicate between our devices, it is necessary that they are able to identify themselves and locate each other. Every device is identified by its own IP address on the network layer and by a MAC address on the link/MAC layer. While the device's MAC address will be constant in every session, unless the user changes it manually, the IP may change, depending on the method used by the DHCP server to assign the IP address.

3.3.3 Discovery

Because we have configured our devices to use infrastructure mode, the devices are not able to discover other devices by themselves, they can only detect and associate

with access points[7]. Devices will connect to an Access Point either the KTH-Open or the one network with the SSID "ece8883" which is connected to the CSS laboratory. The main difference between them resides in the way their DHCP server works: while on KTH-Open DHCP server makes use of automatic address allocation, ece8883's (installed on the badge) assigns new well-known IP addresses to clients listed in a MAC/IP table.

When a user connects to KTH-Open network, the DHCP server will provide them a new temporary and pseudorandom obtained dynamic IP. This device will be able then to communicate with devices on its own Wi-Fi network but also with other devices from other networks connected to the Internet. The problem comes when trying to use several times this device: every time it connects to the Access Point it will obtain a different IP. Just a coincidence can make a device use the same IP; the DHCP server will create different IP addresses but these addresses range between a definite set of numbers. Automatic configurations then will be harder to set up than in the case we employ fixed IPs.

Every time devices in the DHCP server's MAC/IP table connect to an access point with the SSID "ece8883", they will receive a static IP address. The DHCP server only allows these devices (based upon these MAC addresses) to receive assignments[8]. The IP addresses that were used for our experiments are:

- PDA: 192.168.2.70
- Nokia: 192.168.2.71
- Laptop: 192.168.2.72

3.3.4 Establishing communication between the client and the server using VNC

Devices in this this private sub-network must allow connections on the port used by the remote desktop protocol. However, by default this port number is usually closed for security reasons. Therefore, it is important to configure our firewall settings before starting. In the case of VNC, the set of TCP ports meant to be opened range from 5900 to 6000, even though these ports may be forwarded, for example, to the TCP port 22 if we use SSH tunnelling. The VNC server is assumed to be properly set up with the correct number of ports opened in that range, which must equal the number of displays that can be seen. Figure 17 shows the Wireshark capture and decoding of a VNC client initiating a connection to the VNC server.

[7] This could be achieved using a protocol such as the Link Layer Topology Discovery Protocol (LLTD)[45].

[8] Note that all of the other devices which might utilize this access point have a statically assign IP address that is manually configured in the device. Those only our test devices were sending DHCP requests and only a single DHCP server was present on this network to reply. While the system is not limited to behaving this this fashion, it made it easier to examine and understand the traffic – as this private WLAN was essentially only being used for these tests. Note however, that other WLANs using the same IEEE 802.11 channels were in operations, but used different SSIDs.

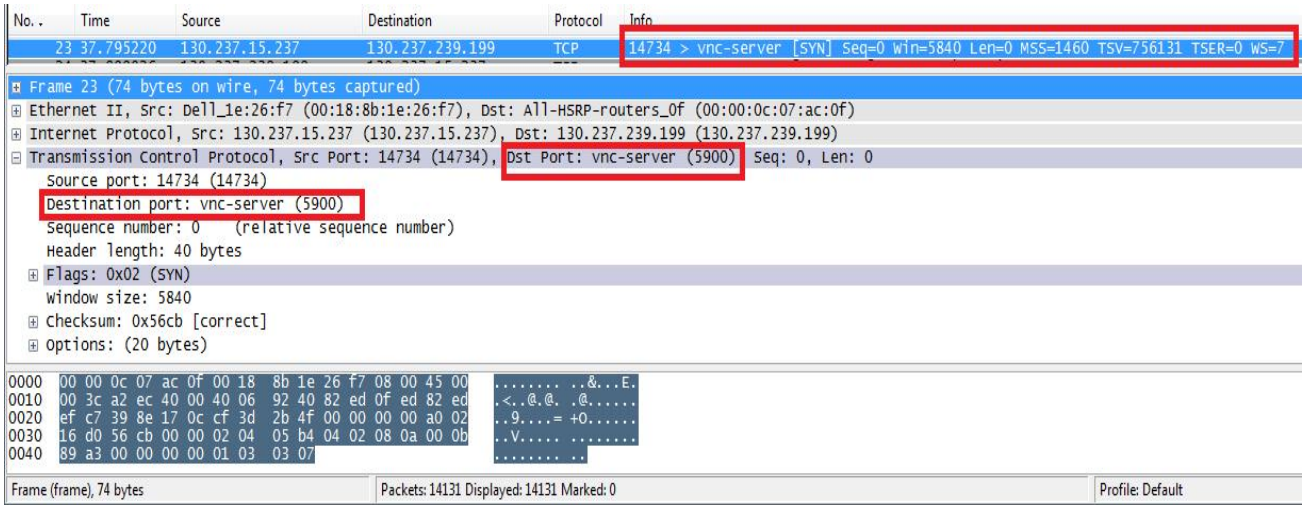

Figure 17: Client requests VNC connection

Authentication is optional in VNC, but authentication is necessary if users need secure connections. Passwords must be at least 8 characters long to prevent brute-force cracking[36].

One of the advantages of using remote desktop protocols is that data exchanged during the interaction between devices is simply input events from the client and graphical output from the server. If a sniffer captures this traffic it will actually show what is being displayed, but it will not be able to show which *operations* are being performed.

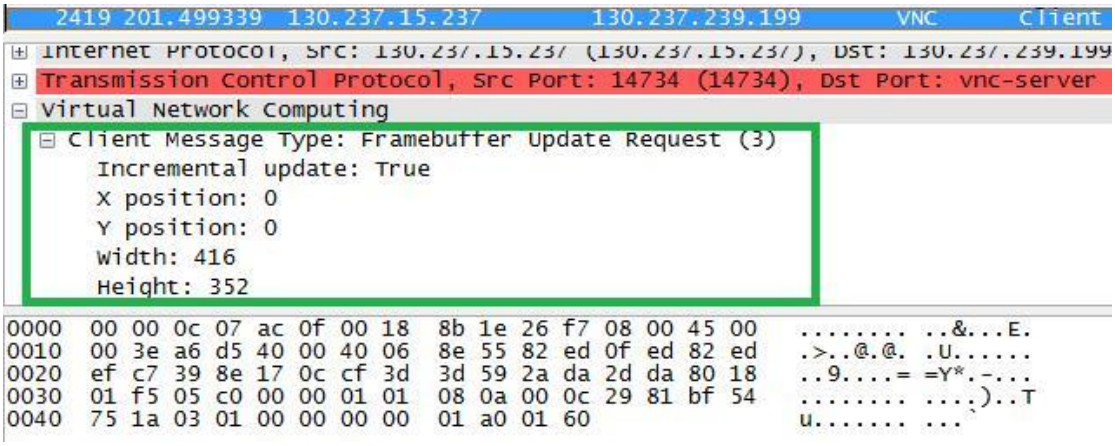

Figure 18: Client Framebuffer Update Request.

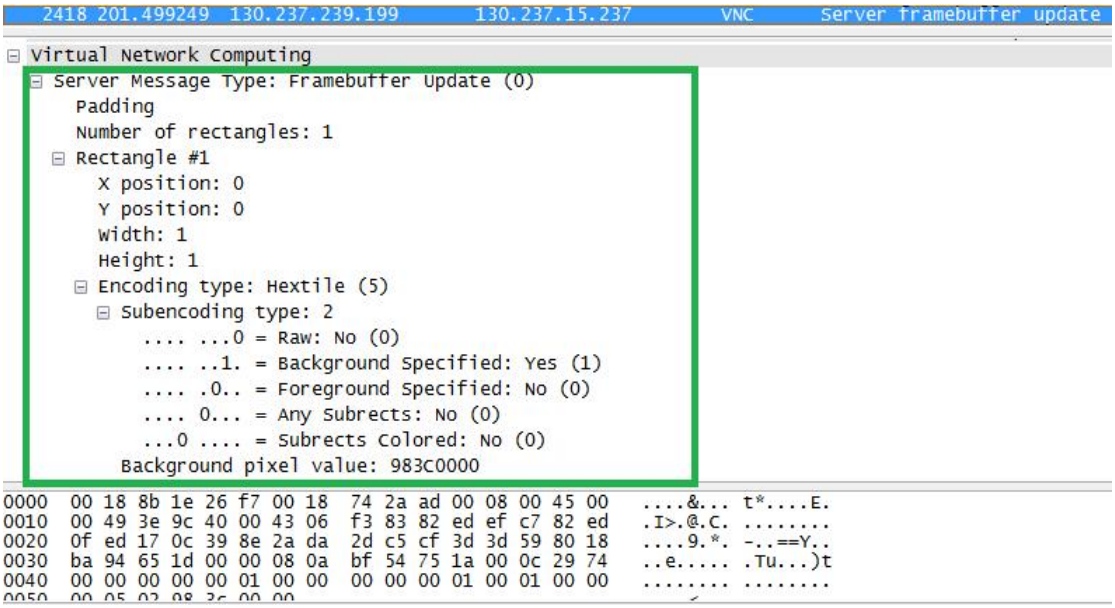

Figure 19: Server Framebuffer Update answer.

3.3.5 Importance of compatible protocol versions

During the initial handshaking, the server makes sure that it is compatible with the version of the protocol used on the client (viewer). For example, VNCViewer 3.3.2[9], works with the version 3.x of the VNC protocol and is not compatible with VNC servers using the latest protocol versions (4.0) unless these include an option for older protocol versions, as vncserver[10]. In this later case the earlier version of the protocol can be used by specifying the protocol version to be used on the command line, as:

username@ccsright:~$ vncserver -Protocol3.3

In vncserver. the option *-Protocol3.3* enables backward compatibility for VNC clients that use this version, such as Pocket PC's VNCViewer 3.3.2. Otherwise, the iPAQ will not be able to connect to the VNC server. Although the server will query the client for what protocol versions it supports (as described in section 2.2.1.1), it will not automatically utilize an earlier version of the protocol unless this is explicitly permitted via configuration or on the command line (when the server program is started).

The following message shows a server that has been configured to allow HTTP connections. A HTTP connection offers another potential way to connect to the VNC server and it enable an Internet browser to connect to the VNC server using as a URL the VNC server's address and the port assigned to the VNC. In this case, the desktop number 1 has been initialized, so the server will communicate through TCP port 5901 to (in our example) to the computer named "*ccsright*". If a client wants to connect to this virtual desktop, the client must specify the computer name (ccsright) if they belong to the same local network and there is a DNS entry for this host or if not, then the IP can be specified, followed by a colon sign, and the number of the desktop.

[9] VNCViewer 3.3.2 is a VNC client program available for Pocket PC
[10] vncserver is a VNC server program available for Linux

Found /usr/share/vnc-java for http connections.

New 'X' desktop is ccsright: 1

Starting applications specified in /etc/X11/Xsession
Log file is /home/username/.vnc/ccsright:1.log

It is very interesting to note that the server supports a web browser as a client, as browsers are increasingly commonly used as the interface to nearly all web based applications. This remote desktop access simply becomes yet another web application!

4 Analysis

4.1.1 Should all clients have Full Access

The action of connecting to a remote desktop server suggests other implicit issues that need to be considered by administrators regarding privacy and security. Data might be seen and controlled by clients in full access mode, possibly with root permissions as well. Therefore this data could be modified and even removed.

Imagine a situation where a presentation that involves different files is portrayed on several clients. It could be configured in 2 ways:

- One display shared by all clients: every client will be able to see this only one display on its own screen. One of these clients could interfere with the presentation by prematurely closing the presentation files, moving them, or even deleting the files.
- Different displays, one for each client – in this scenario the screens will be independent, so we avoid files being easily deleted and the files not accessible for other users.

The basic solution to prevent the above problem with the use of remote servers was to set up the servers to permit read-only access. This configuration could be used in scenarios where interaction by the user with the application is not necessary (as there is someone else controlling the application), for example, during a demonstration of a new mobile phone, it could be interesting to show its screen via a projected display or on the screens of the audience members. However, only the presenter should be operating this phone, while menus and applications from the phone would be shown on the large screen or the many smaller screens.

4.1.2 Sharing the same desktop

Another was to use VNC is to share a desktop. Thus a number of users can access, see, and interact from different screens with the **same** desktop. This desktop is that of the VNC server. This is useful and interesting for applications where the aim is to guide the user to use an application that is running on the server machine. Thus is the person receiving the training hesitate for too long, their guide/teacher/mentor can perform the required operation. This has significant pedagogic advantages as it allows the student to gain experience under the guidance of a tutor. Additional scenarios include remotely controlling a device, such as a phone, using as an input device a device which would not normally be able to control this device; for example, using a Handykey Twiddler connected to a computer to control a cellular phone. It is this last scenario which is especially interesting to use as it returns us to our goal of performing device aggregation.

Our initial testing showed that it was possible to use a VNC client to remotely control a VNC server for a variety of devices equipped with different interface devices and different resolution screen. The next step is to evaluate the performance when using a number of devices connected to a VNC server.

4.1.3 Performance

VNC performance can be affected by a variety of factors ranging from network problems to issues related to the VNC protocol itself. Since sufficient visual

performance is essential for the successful use of VNC, it is really important to configure the client and server with a suitable set of parameters to achieve this. If the performance is not sufficient, then the user will become unhappy and reject this solution. For example, Figure shows the case of a VNC client accessing a VNC server running on a Nokia E70 phone – in this case the screen has only partially been updated.

Figure 20: Screen Update. Main menu and "Messages" menu overlapped

By changing the choice of VNC encoding depending on the properties of the connection used between the devices – we can optimize the end user's perception of the performance of the system. In our case we take advantage of a high-speed WLAN connction, and choose an encoding that provides better performance in this scenario. As we can see in the next figures, Hextile provides better performance than Raw encoding.

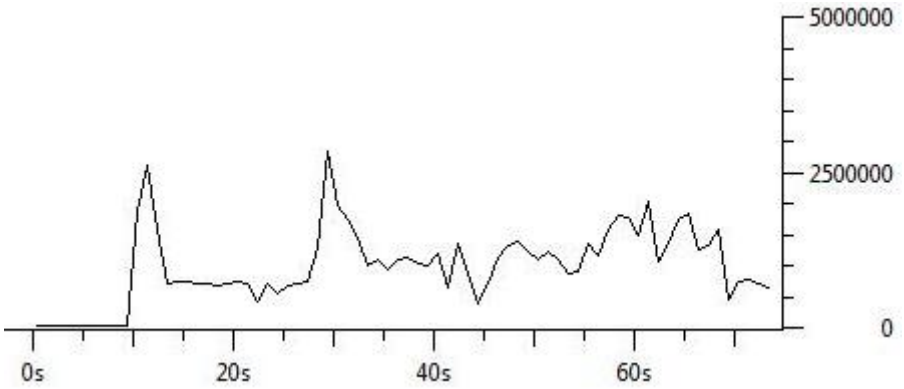

Figure 219: Raw encoding (bytes)

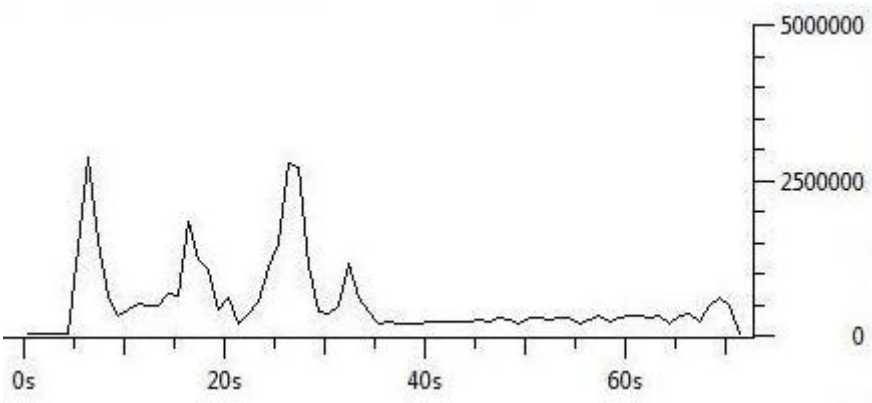

Figure 22: Hextile encoding (bytes)

As we can see, Hextile encoding uses less bandwidth than Raw. With Hextile encoding requiring generally less than 10kBytes per second in throughput. In fact, the Hextile encoding traffic is close to 25 Kbits per second (on average).

Additionally, Hextile encoding leads to the server sending fewer packets than for Raw encoding. However, the number of small packets sent is quite significant, regardless of what the encoding is. This issue is improved on by the new protocol NX.

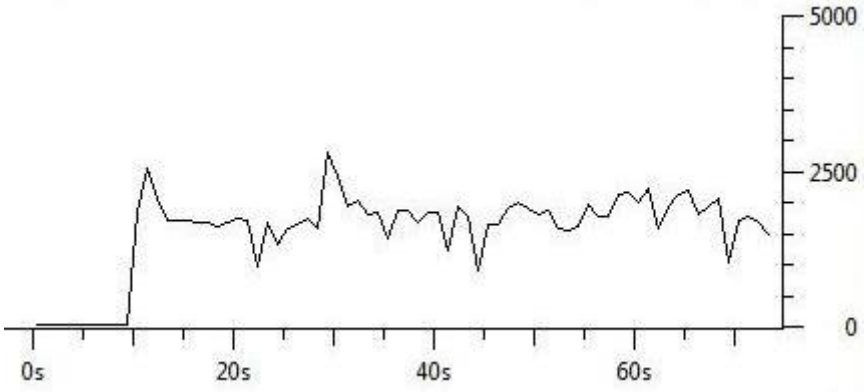

Figure 23: Raw encoding (packets)

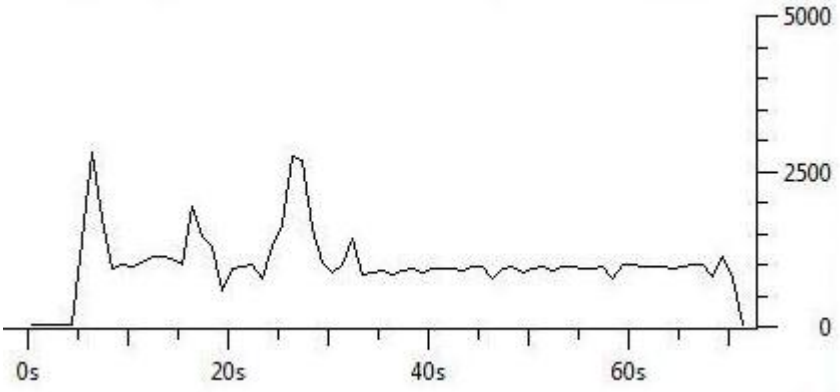

Figure 24: Hextile encoding (packets)

As the following two figures show, NX sends about than 10 times fewer packets, and these packets also smaller. Unfortunately, NX server applications are not yet available for Windows[11] or Symbian systems.

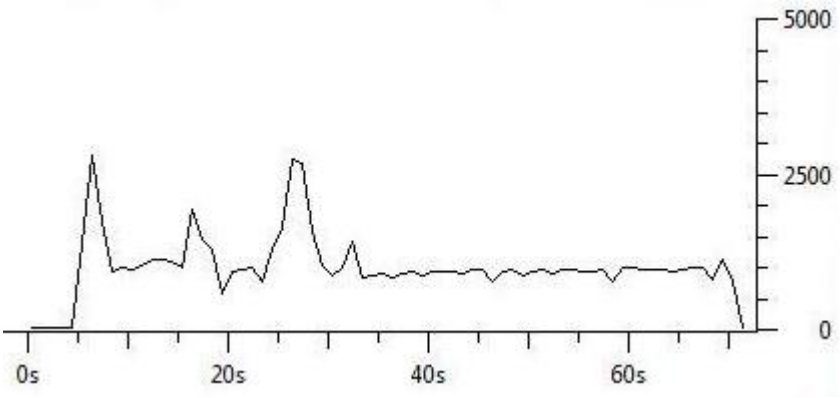

Figure 25: VNC Hextile graph (Packets/Tick)

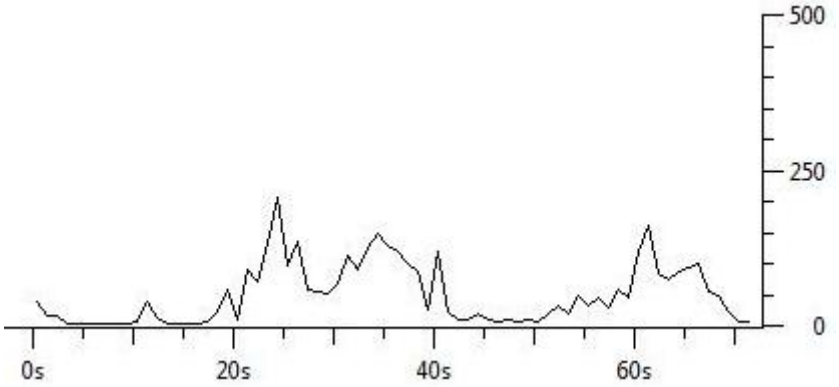

Figure 26: NX graph (Packets/Tick

[11] Nomachine claims Windows NX server applications will be available on future versions of Windows Mobile (5.x onwards)[38]

5 Conclusions and Future work

5.1 Conclusions

Remote desktop applications offer a reliable solution for internetworking devices through the Internet and especially via WLANs, even though other physical channels, such as Bluetooth might also be used. Remote desktop applications offer the user the ability to control a very large number of different applications. The most important uses of this technology are troubleshooting, showing presentations, and as presented in this thesis - device aggregation.

In terms of speed, this kind of applications is able to offer great performance, sometimes even giving the feeling of being in front of the server computer, but without all of the limitations of the input devices of this server computer (see for example the example shown in Figure - where the user of a computer with a physical keyboard is able to enter a new contact address using their full keyboard . rather than being limited to poking at a picture of a keyboard with a stick).

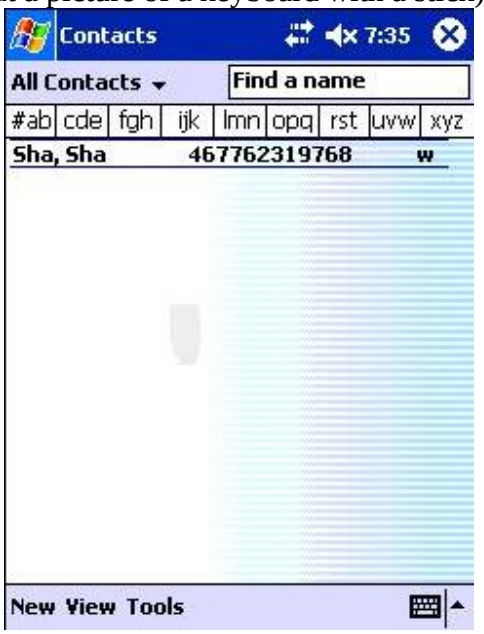

Figure 27: Contact added on a PDA agenda application from a laptop using VNC

VNC is one of the best solutions when it comes to remote desktop solutions, besides fast connections, it is available on a wide range of systems and is highly portability (hence a version is likely to be available for nearly any platform the user would want to use). Furthermore it is an open-source technology, thus enabling improvements made by one developer to quickly be propagated to other implementations. On the other hand, NX achieved faster updates than VNC in our tests, but it is still a new technology and it has not yet been deployed on operating systems as Symbian or Microsoft's Pocket PC. Thus despite NX's very good performance it does not satisfy our requirements for device aggregation today.

A weak point of remote desktop applications (particularly for VNC) is that security is not well enforced. For example, brute-force attacks could allow an attacker to access and control our systems. While security options exist, depending on the user's needs, the layer chosen to apply this solution to, and the need for compatibility with

other protocols – the task of defining and managing the security of a remote desktop application is not a simple one and there are many possible mistakes that can be made.

Moreover, the potential security configurations are reduced when we try to apply this technology on handheld devices as many of the requisite protocols do not exist or are crippled in either performance or features. For example, the WLAN infrastructure mode worked properly between the PDA and a desktop/laptop computer. On the other hand, the Nokia E70 mobile phone was not accessible from any of the other devices on this network. It seems that Nokia E series have problems related to the use of DHCP.[40] Other difficulties were also found when trying to run a VNC server on Symbian, specifically when trying to connect to the WLAN access points. Similar problems also happened sometimes with the PDA. Furthermore, the mobile phone sometimes became disconnected from Wi-Fi LAN after several minutes (an example error message is shown in Figure).

```
shasha@ccsright:~> vncviewer 130.237.239.199
vncviewer: ConnectToTcpAddr: connect: Connection refused
Unable to connect to VNC server
```

Figure 28: Error trying to connect to VNC server on PDA

Note that one of the problems of the access control method implemented by the university's WLAN network is that to ensure that if a device leaves a cell that other nodes can not pretend to be the departed node, the access control infrastructure sends ICMP echo request packets to the device and if the device fails to response to this request the MAC address of the device is removed from the list of devices that are permitted access through this access point. A problem with many mobile devices is that they utilize power saving techniques, such as turning of the network interface or stopping the processor to save power – however, this may cause the device to fail to hear and respond to an ICMP echo request – thus the device will be disconnected from the network.

5.2 Future work

Still today there are few remote desktop server applications available for handheld devices. Client applications for Symbian mobile phones are difficult to find and NX applications are yet to come. This represents an opportunity for developing client/server applications for such devices. Future work should mainly focus on improving performance, modifying existing remote desktop server applications or creating new ones, for example a porting the NX server to more platforms. In order to enforce security, it would be interesting to develop a SSH server application for Symbian.

Providing discovery and multicasting for our devices would enable users to quickly set up & automatically configuration a wide range of devices. For example, creating a "services application" to take advantage of the device aggregation by utilizing PDA, mobile phone, laptop, Twiddler, GPS adapter, etc. The goal would be to aggregate devices that offer different services in order to offer the best service - by making use of the best characteristics of each device. Ideally, these services should be available for any collection of devices, regardless the physical/MAC layer protocols used. Thus, all these devices could communicate with the badge exploiting the same protocol. Some of the obviously desirable service include: VOIP, file sharing, Internet access, contact synchronization, geographical location, printing, and of course remote desktop access.

References

[1] Sun Yu, Context aware applications for a Pocket PC. Master of Science thesis, School of Microelectronics and Information Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, COS/CCS 2007-28, 17 December 2007. Available from: http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/071220-Sun_Yu-with-cover.pdf

[2] The Official Bluetooth Technology Info Site. Get Technical. Available at: http://www.bluetooth.com/Bluetooth/Technology/

[3] Osmosis Latina. Sistemas de Información con enfoque de software libre y de código abierto. “LANs Inhalámbricas y “Bluetooth”. Available at: http://www.osmosislatina.com/conectividad/bluetooth.htm

[4] Webopedia. What is Bluetooth? Available at: http://www.webopedia.com/TERM/b/bluetooth.html

[5] Wikipedia. Bluetooth. Available at: http://en.wikipedia.org/wiki/Bluetooth

[6] Kioskea.net. Knowledge kiosk. “Cómo funciona Bluetooth”. Available at: http://es.kioskea.net/bluetooth/bluetooth-fonctionnement.php3

[7] Wikipedia. Bluetooth. Personal Area Network. Available at: http://en.wikipedia.org/wiki/Personal_area_network

[8] The Official Bluetooth Technology Info Site. Communication Topology. Available at: http://www.bluetooth.com/Bluetooth/Technology/Works/Communications_Topology.htm

[9] The Official Bluetooth Technology Info Site. Bluetooth Wireless Technology Profiles. Available at: http://www.bluetooth.com/Bluetooth/Technology/Works/Profiles_Overview.htm

[10] Allan Beaufour Larsen, Secure Access Control Using Mobile Bluetooth Devices, Master of Science Thesis, University of Copenhagen, July 2003. Available at: http://www.diku.dk/forskning/distlab/Publication/Master's_Theses/2003/beaufour03secure.pdf

[11] The Official Bluetooth Technology Info Site. Security. Available at: http://www.bluetooth.com/Bluetooth/Technology/Works/Security/

[12] Manoj Nair, A Paper In Security Concerns In Bluetooth Technology. Available at: http://www.datastronghold.com/security-articles/general-security-articles/a-paper-on-security-concerns-in-bluetooth-technology.html

[13] Wi-Fi. Knowledge Center. Glossary. Available at: http://www.wi-fi.com/knowledge_center_overview.php?type=3 - 3802

[14] Wi-Fi. Wireless Technology. Wireless Fidelity. Wi-Fi, What is Wi-Fi? Available at: http://www.wifinotes.com/what-is-wifi.html

[15] Wi-Fi. Knowledge Center. The How and Why of Wi-Fi. Available at: http://www.wi-fi.org/knowledge_center/kc-howandwhyofwi-fi

[16] Harold Davis, Anywhere computing with laptops. Making mobile easier, September 2005

[17] H: Schulzrinne, S. Casner, V. Jacobson, R. Frederick , RTP: A transport Protocol for Real-Time Applications. IETF. RFC 3550. July 2003. Available at: http://www.ietf.org/rfc/rfc3550.txt

[18] Wikipedia. MAC Address. Availabe at: http://en.wikipedia.org/wiki/MAC_address

[19] Wikipedia. Transport layer security. Available at: http://en.wikipedia.org/wiki/Secure_Sockets_Layer

[20] Wikipedia. Remote desktop software. Available at: http://en.wikipedia.org/wiki/Remote_desktop_software

[21] Real VNC. Applications of VNC. Available at: http://www.realvnc.com/vnc/how.html

[22] Tristan Richardson, "The RFB Protocol", RealVNC Ltd., Version 3. 8, 28 August 2008. Available at: http://www.realvnc.com/docs/rfbproto.pdf

[23] Real VNC. The original open source cross-platform remote control solution. Available at: http://www.realvnc.com/vnc/why.html

[24] Wikipedia. Terminal Services. Availabe at: http://en.wikipedia.org/wiki/Terminal_Services

[25] Microsoft Developer Network. Remote Desktop Protocol (Windows). Available at: http://msdn.microsoft.com/en-us/library/aa383015.aspx

[26] Microsoft Help and Support. Understanding the Remote Desktop Protocol (RDP). Available at: http://support.microsoft.com/default.aspx?scid=kb;EN-US;q186607

[27] X(7). Manual Page. Available at: http://www.xfree86.org/current/X.7.html

[28] NX Technology. Available at: http://www.xfree86.org/current/X.7.html

[29] Tecnología NX. Available at: http://es.wikipedia.org/wiki/Tecnolog%C3%ADa_NX

[30] T. Ylonen and C. Lonvick (Ed.). The Secure Shell (SSH) Protocol Architecture. IETF. RFC 4521. January 2006. Available at: http://www.ietf.org/rfc/rfc4251.txt

[31] Wikipedia. RC4. Available at: http://en.wikipedia.org/wiki/RC4

[32] Wikipedia: Wi-Fi Protected Access. Available at: http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

[33] Wi-Fi Alliance. Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks. April 29, 2003. Available at: http://www.wi-fi.org/files/wp_8_WPA Security_4-29-03.pdf

[34] Wi-Fi Alliance. Knowledge Center. MAC Filtering. Available at: http://www.wi-fi.org/knowledge_center/kc-macfiltering

[35] Wikipedia. Dynamic Host Configuration Protocol. Available at: http://en.wikipedia.org/wiki/Dhcp

[36] Wikipedia. Virtual Network Computing. Available at: http://en.wikipedia.org/wiki/VNC

[37] Nomachine. The Network Computing Company. Articles & FAQs. NX Technology. Available at: http://www.nomachine.com/ar/view.php?ar_id=AR02D00349

[38] Gerald Q. Maguire Jr.'s notes about using the HP iPAQ h5550. Available at: http://web.it.kth.se/~maguire/ipaq-notes.html

[39] Tips and tricks for Nokia E70. Available at: http://www.iptel.org/config/nokia_e70

[40] Handykey Corporation, Twiddler, http://www.handykey.com/, last accessed 2008.12.01

[41] Cécile Ayrault, Service discovery for Personal Area Networks, Master of Science thesis, School of Microelectronics and Information Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, IMIT/LCN 2004-07, 2004. Available from: http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/040826-Cecile_Ayrault-with-covers.pdf

[42] Jia Zhou, Adding bandwidth specification to a AAA Sever, Master of Science thesis, School of Microelectronics and Information Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, COS/CCS 2008-19, September 2008, Available from: http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/080914-zhoujia-with-cover.pdf

[43] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin, A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP), Transactions on Information and System Security (TISSEC), ACM, Volume 7, Number 2, May 2004, pp 319-332 http://doi.acm.org/10.1145/996943.996948

[44] Shasha Zhang, Device aggregation with data networking: Implementing a Personal Area Network, Master of Science thesis, School of Microelectronics and Information Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, COS/CCS 2008-23, October 2008, Available from: http://web.it.kth.se/~maguire/DEGREE-PROJECT-REPORTS/081003-Shasha_Zhang-with-cover.pdf

[45] Microsoft Corporation, LLTD: Link Layer Topology Discovery Protocol, Version 1.0.9 – September 15, 2006. Available from http://download.microsoft.com/download/a/f/7/af7777e5-7dcd-4800-8a0a-b18336565f5b/LLTD-spec.doc

Appendices

A. Using VNC on PC

VNCViewer is a free X viewer client for VNC. It allows the user to easily access a VNC server from a PC (both Windows and Linux versions are available). Common features of VNC clients including tunnelling, full screen mode, data encoding, or image configuration are available with this application.

VNCViewer can be download from:
http://www.realvnc.com/products/download.html

B. Using VNC server for iPAQ (Pocket PC)

B.1. WLAN configuration

On the iPAQ: Firstly we enable WLAN access for the iPAQ

Then we go to Start->iPAQ Wireless if we are not yet connected and enable WLAN (All wireless features on). Note that it is recommended to have this option *not* enabled unless we are using it because of increased energy consumption when the radio is on.

If we want to connect to, for example KTH-Open, we will go to the Start->Internet Explorer, the home page will ask us to login, following this we will be logged on and can access the network with other applications that use the WLAN interface.

B.2. VNC Client

VNCviewer 3.3.2 is another VNC client program available - in this case we have used the version for the Pocket PC. It is available for download from:
http://www.cs.utah.edu/~midgley/wince/vnc.html

It works with the version 3 of the VNC protocol, so it is not compatible with VNC servers using the latest protocol versions (4.0).

In contrast to VNCViewer for Linux, fewer features can be managed – these include which type of encoding to use for pixel data (Hextile, CoRRE, RRE, Raw) and image quality.

B. 3. VNC Server

There is also a VNC server program (see) available for Pocket PC, called PocketVNC . A Demonstration version is available from:
http://www.pocketvnc.com/projects/pocketvnc/index.php

We will need to know our IP address, we can learn this either using a local program or by visiting the website www.whatismyip.com – we have to provide this IP address to the VNC viewer.

We will run our VNC Server, then click “Start VNC Server” to start receiving VNC client requests. If we click “ok”, then our program will be minimized. We can access it again by clicking on the white icon in the lower right on the Task Bar.

If we use vncviewer to connect to this server should use the “bgr233” option to get a suitable 8 bit *pseudocolor* mapping.

After 1 minute of VNC interaction the demonstration version will stop working.

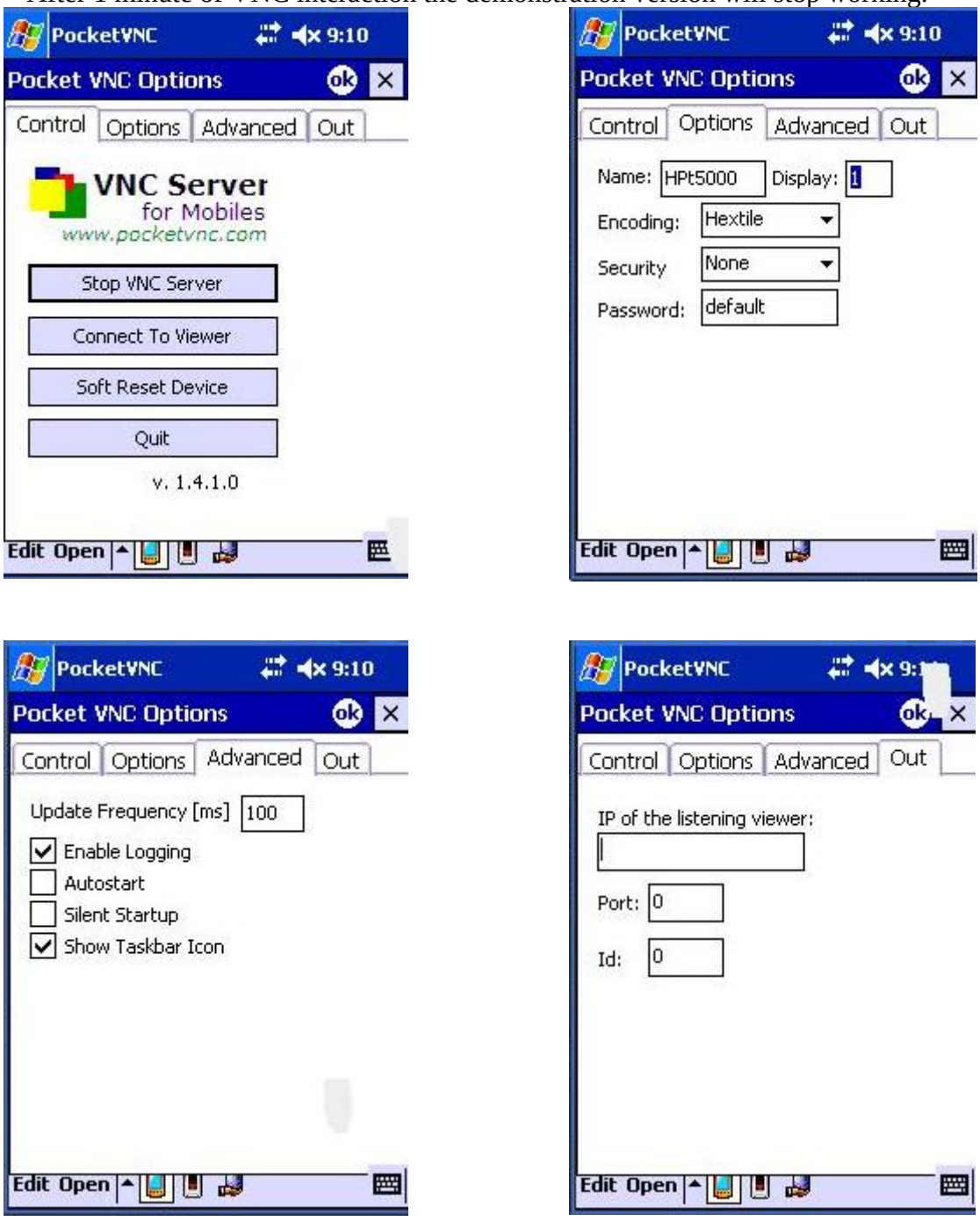

Figure 29: PocketVNC screenshots

C. Using VNC server for Nokia E70 (Symbian)

MVNC demo is a VNC server for Symbian systems available for download at:

http://www.m-shell.net/Download.aspx

It is necessary to send a SMS with the activation code, to get going the demo free program before using it. This program, either the demo or full version, needs a SIM card inserted to work, otherwise it will not work.

First of all, we need to connect our phone to the WLAN (KTH-Open for example). After entering the website, we will go to www.whatismyip.com to get our IP address.

A set of parameters must be configured: name, WLAN IP; this is the IP of the client we want to connect to, so somehow here the client-server paradigm has changed, but we have provided security: just allowed IPs will have access. This feature may be inconvenient when devices we are going to interconnect are not in the "same" place: we must set up our devices before we are going to start using them, whether we use dynamic IPs or static. So we have to know which IP need to communicate data and the port we are using. Other options on are Port (TCP 5900 normally), an optional password (optional), and the image quality.

First the program will look for available Access Points. Note that if we have not connected yet, in the case of KTH-Open, saying our KTH username and our password, we will not be able to connect.

The client will have to run a viewer specifying the IP that wants to connect to. So, the server and client will exchange the message: TCP start, version of protocol, etc and then the client will (optionally) receive a password request. This provides more security although this password is saved on a file on the registry and it could be easily to crack.

D. Connecting Nokia E70 and iPAQ via Bluetooth

Firstly, we have enable authorization, authentication, and encryption for all the services existing on the iPAQ: File transfer, Information Exchange, Dial-Up Networking, Personal Network Server and Audio Gateway. The PDA can be configured to be hidden for discovery and to be connected exclusively just by the devices that it has been paired with before.

After synchronizing we will find the devices' common Bluetooth services: OBEX File Transfer, and Dial-Up Networking. Then, we select the service(s) that we want to use and the pairing procedure starts, giving the option to establish a PIN[12] key that may be used in future connections to authenticate device and to encrypt data. Once the password is set, then the phone will immediately ask the user to enter the same password to establish the pairing and we will be asked to authorize the iPAQ to make connections automatically as well. Now, the devices are ready to use the services.

E. Connecting Nokia E70 and Motorola Bluetooth Headset

On Nokia E70:

Connectivity –> Bluetooth –> Paired Devices -> New Paired Device

Then, Nokia starts looking for devices within range. A list of Bluetooth devices will be shown. After selecting one of these devices, the user will be ask for a password. Finally, the phone will ask the user to automatically authorize connections with that device.

[12] Note this password must be numeric.

COS/CCS 2008-15

www.kth.se