**A K A D E M I A**
**G Ó R N I C Z O - H U T N I C Z A**
K R A K Ó W
**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,**
**INFORMATYKI i ELEKTRONIKI**
Katedra Informatyki

# *Komunikacja multimedialna w systemach wykorzystujących technologię Bluetooth*

**Marcin Michalak**
***mmich@iris.ics.agh.edu.pl***

**Praca Magisterska**
**kierunek studiów: Informatyka**

***Promotor*: prof. Krzysztof Zieliński**

**Kraków, czerwiec 2001**

U N I V E R S I T Y   O F   M I N I N G
A N D   M E T A L L U R G Y
C R A C O W
FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS, COMPUTER SCIENCE AND ELECTRONICS
Institute of Computer Science

# *Multimedia transmission in Bluetooth-based networks*

**Marcin Michalak**
***mmich@iris.ics.agh.edu.pl***

**Master of Science Thesis**
**Computer Science**

*Supervisor*: **prof. Krzysztof Zieliński**

**Kraków, June 2001**

## Table of contents

# 1 Introduction

Computer networks have been evolving very rapidly during past 20 years. At the same time people have increased demands towards them: they want fast, high quality access in every place. This is why wireless users shall make a substantial share (62.1 %) of the total number of Internet users (1.2 billion) forecast for 2005.

On the other hand demands for services available in wireless networks are same as for wired ones: everybody wants web browsing, e-mail check, chat but also multimedia services.

Multimedia services, which are the topic of this thesis, audio-video conferencing or video streaming have gained much popularity in last years. This has been enabled by the high increase of computing power in home PCs, which allows usage of very efficient video compression technologies. MPEG-1, 2 and 4, RealVideo can reduce the size of video clip by a factor of 200-1000 thus making it possible to put such transmission into regular computer networks, including wireless. This thesis focuses on testing the possibilities of multimedia transfer given by one of the most promising wireless technologies, which is Bluetooth.

Bluetooth has been designed to replace cable connections between devices like PCs, notebooks, PDAs, phones, headsets, cameras etc. Main goals of the technology include:

- low power consumption,
- flexibility,
- conformance to the existing standards,
- low cost ($5-$10/unit),
- short range: 10 – 100 m,
- speeds up to 721 kbps.

Speeds up to 721 kbps allow transmission of the web pages, e-mail, chat but also conferencing with video and audio or streaming of live shows like sports matches or movies.

This thesis discusses possibilities of audio and video transmission over Bluetooth links: which scenarios are possible, what kind of software may be used and where is the actual limit of the Bluetooth speed, i.e. how much of the bandwidth is actually available for the transmission.

The contents are organised as follows: Chapter 2 describes the principles of the Bluetooth technology: how the transmission is realised, tasks performed by each layer and how is IP laid over Bluetooth, since this technology shall be used for testing purposes.

Chapter 3 describes current video and audio compression technologies, as well as ways to request and transmit audiovisual streams. Technologies like MPEG-1, 2 and 4, RealVideo, H.261 and H.263 for video, G.711, G.723, GSM, MP2 and MP3 for audio are presented. From streaming protocols RTSP and PNA has been chosen. In the end, Real-Time Transmission protocol (RTP) is described as the basic way of sending video and audio streams.

Chapter 4 contains installation instructions for creating a testing environment. This includes installation of the Bluetooth cards and software: video servers and clients. A concept of the Bluetooth access point is also described: how to create and configure a node that will provide access from Bluetooth devices to other networks, i.e. Internet.

Testbed described in Chapter 4 is used to measure actual bandwidth offered by Bluetooth links and present possibilities given by this technology regarding to multimedia transfer.

Chapter 5 presents several scenarios: starting from file transfer via http/ftp, through conferencing applications like Microsoft Netmeeting and MBONE tools, to video transfer using publicly available streaming software. This includes Darwin Streaming Server with QuickTime Player from Apple and RealServer with RealPlayer from RealNetworks. Each scenario is tested with streams of various bandwidth and compression techniques. Achieved results are discussed and conclusions are drawn suggesting which technologies are most suitable for multimedia transmission over Bluetooth.

The thesis is ended with summary and references, which contain documents related to this work.

## 2 *Bluetooth Technology Overview*

Bluetooth came up as the idea in the Ericsson laboratories [1] in 1994. Engineers saw a need for the wireless transmission technology that would be cheap, robust and flexible and consume little power. Technology that from one side could replace cables, but also offer new possibilities and cross the boundaries set by the existing solutions. This is why it has been called after the legendary Scandinavian king Harald Bluetooth, who christianised Denmark and united Denmark and Norway and in the Xth century.


**Figure 1. Bluetooth as a wireless link between devices**

Ericsson Mobile Communications, Intel, IBM, Toshiba and Nokia Mobile Phones formed a Bluetooth Special Interest Group (SIG) [2] in 1998. This group's intention is to create new technology and promote it on the market. Up till now, almost 2500 companies joined the SIG and work on the development and promotion of Bluetooth products.

The principles of the Bluetooth technology are:

- low cost: $5-10 per unit,
- small size,
- low power consumption,
- speeds up to 721 kbps,

- range of 10 m (extended 100 m),
- open standard.

Currently Bluetooth specifications release 1.1 is available. It is divided into two files: Bluetooth 1.1 Specifications Book [3] and Bluetooth 1.1 Profiles Book [4].

## 2.1 Radio interface

One of the most important advantages of Bluetooth is the usage of the unlicensed Industrial, Scientific & Medical (ISM) band at 2.45 Ghz, occupying frequency from 2400 to 2.4835 MHz. The same band is used by WLAN technology (IEEE 802.11 [5]) and microwave oven.

| **Location** | **Bandwidth** | **RF Channels** |
|---|---|---|
| USA, Europe and most other Countries | 2400 – 2.4835 GHz | f=2402+k MHz, k=0,…,78 |
| France | 2.4465 - 2.4835 | f = 2454 + k MHz, k= 0,...,22 |

**Table 1. Bluetooth frequencies and channels.**

This band is divided into 79, 1 MHz wide, channels, which are used for both transmission and reception. The frequency is shared basing on Time Division Duplex (TDD) technology.

Due to the strict regulations existing in the ISM band and to the principle of power consumption there are limitations in the power level. Basic transmission power is set to 0 dBm and is regulated by the Link Manager Protocol (LMP) layer.

| **Power class** | **Maximum Output Power (Pmax)** | **Power control** |
|---|---|---|
| 1 | 100mW (20dBm) | Pmin<+4 dBm to Pmax<br>Optional: <-30dBm to Pmax |
| 2 | 2.5mW (4dBm) | Optional: <-30dBm to Pmax |
| 3 | 1mW (0dBm) | Optional: <-30dBm to Pmax |

**Table 2. Power classes.**

In order to make the transceiver as simple as possible, Gaussian Frequency Shift Keying (GFSK) has been chosen as the modulation technique. Binary one is represented by the positive frequency deviation, binary zero by negative frequency deviation.

## 2.2 Division duplex and multiple access technology

Bluetooth uses Time Division Duplex (TDD) principle for changing between transmission and reception. Each time slot occupies 625 μsec, and each packet occupies up to 5 time slots (1, 3 or 5). This concept allows for reduction in the overhead when transmitting lots of data.

For multiple access technology, distinguishing between the users, Frequency-Hopping Code Division Multiple Access (FH-CDMA) has been chosen [6]. The principle of FH-CDMA technology is changing the transmission/reception channels very frequently. In this case, hopping between the available 79 (23) channels occurs 1600 times/second.

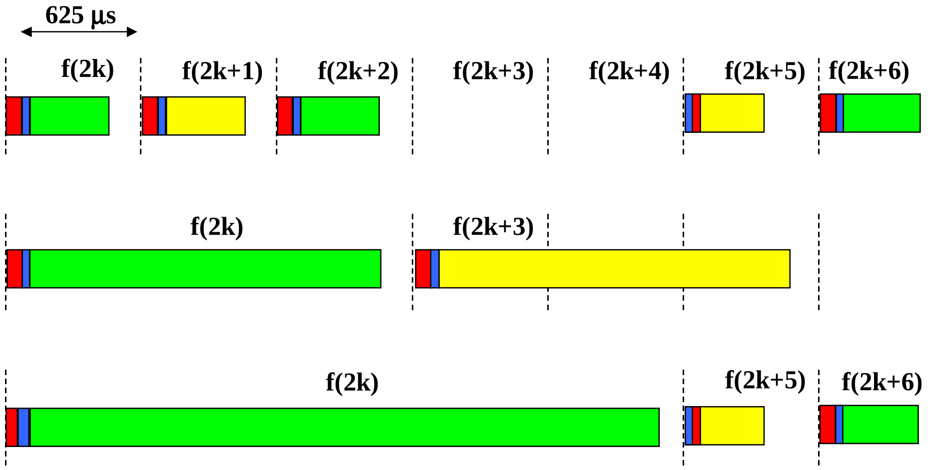


**Figure 2. Division duplex and multi-slot packets.**

Figure 2 illustrates TDD and FH-CDMA concept. Packets sent by master device are marked in green and slave transmission is yellow. Devices share the hopping scheme, i.e. only one may transmit on the channel at a time. 3 rows in the figure represent 3 independent hopping schemes, without overlapping frequencies.

## 2.3 Piconet and scatternet

The network in Bluetooth has a topology of star. It is organised in so-called piconets. Piconets team up to 8 active devices, one of them selected as the master. Master is the device that controls the network: it chooses the hopping scheme, decides who may transmit and is used as a synchronisation reference point.

Apart from active state devices may be 'parked' in the piconet. Number of parked devices is only limited by the length of so-called Parked Member Address (PM_ADDR), which is 8 bits. This allows keeping track of 255 parked devices.

'Parked' state means that device is synchronised with the master and knows the hopping sequence, but cannot transmit or receive data until it becomes 'active'. This allows for example parked device to enter low power consumption mode and simultaneously let other devices use the bandwidth. Figure 3 presents the concept of piconet with active, parked and standby devices.

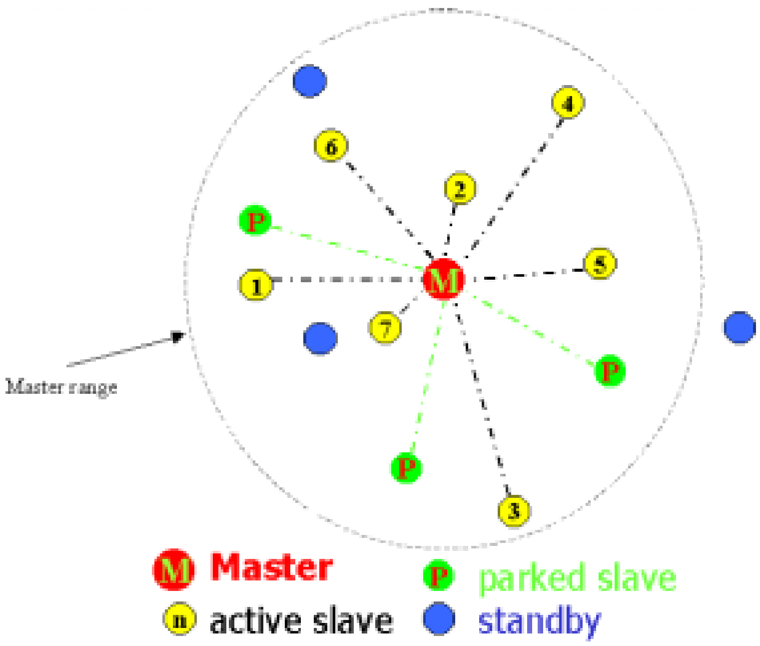


**Figure 3. Piconet with master, active, parked and standby devices.**

Several piconets with overlapping coverage areas form a scatternet. Scatternet architecture allows for transmission between devices that are not directly connected, e.g. due to the long distance. They can communicate via another device, which is in the range of both of them. All the transmission across piconets has to pass through master device. It is important to note that master is only piconet-local, i.e. device being master in one piconet may be slave in another piconet. This concept is illustrated in Figure 4

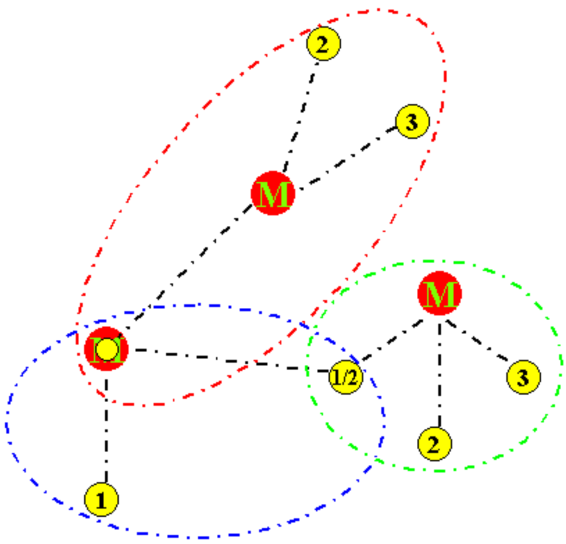


**Figure 4. Three piconets forming a scatternet.**

## *2.4 Layered architecture*

Bluetooth consists of several protocol layers taking care of the issues related to data transmission. Following sections describe briefly protocols, which are of interest for multimedia streaming.

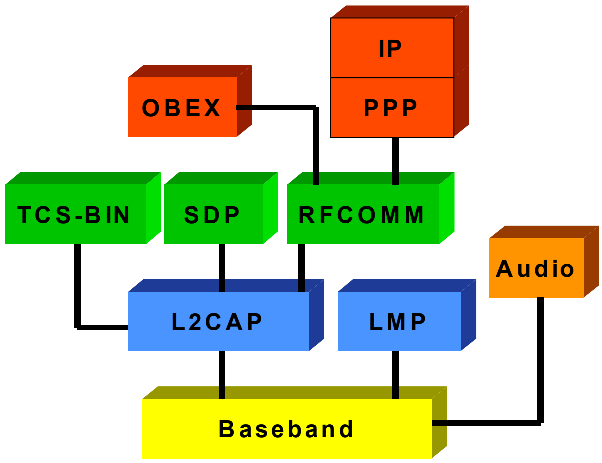


**Figure 5. Bluetooth protocol layers.**

### 2.4.1 Baseband

Baseband protocol forms the lowest layer in Bluetooth architecture. It is responsible for the functionality contained in the physical layer of the OSI/ISO model, but also performs some tasks from higher layers. Its main tasks are:

- synchronization,
- transmission of the information,
- error correction (FEC & CRC),
- logical channels division,
- data whitening (scrambling).

Bluetooth supports both synchronous and asynchronous channels. The possible configurations are:

- 1 asynchronous data channel
- up to 3 simultaneous synchronous voice channels each 64kbps
- channel which simultaneously supports asynchronous data and synchronous voice.

The asynchronous channel may support up to 721kbps downlink and 57.6kbps uplink or symmetrically 433.9 kbps in both directions.

**Physical links**

There are 2 types of physical links: Synchronous, Connection-Oriented (SCO) and Asynchronous, Connection-Less (ACL).

SCO link is a symmetric, point-to-point link between the master and a specific slave. The SCO link reserves time slots and can be considered as a circuit-switched connection between the master and the slave. It provides 64 kbps channels which are initially designed for voice transmission. For voice encoding, either pulse-code modulation (PCM) or Continuous Variable Slope Delta modulation (CVSD) is used. What is important is that SCO packets are never retransmitted, because this link is oriented for real-time transmission.

Second type of link is ACL, which may utilise slots not reserved for the SCO transmission. ACL link is packet-switched and thus is intended for packet transfer, both asynchronous and isochronous (i.e. time-sensitive).

By definition, there can only be one ACL link between a master and a slave. After establishment it carries all the traffic between those nodes. In most cases, retransmission is used to assure data integrity. This is a very important difference to other technologies, like Ethernet or ATM, where retransmission is not applied at all, i.e. higher layers (for example TCP) have to take care of data integrity.

In the ACL link, slave is allowed to send the packet (i.e. occupy the slot(s)) if, and only if, it has been addressed by the master in the preceding slot. This kind of solution is used to control medium access – piconet master decides who shall access radio link next.

To get a more detailed picture of the baseband layer properties, please refer to Appendix B, which contains connection setup procedure.

### **2.4.2** Link Management Protocol (LMP)

LMP is generally used for link set-up, control and maintaining the security. LMP messages have higher priority than user data, and thus pass immediately through the link even if it is congested.

The tasks of this layer include:

- power control: measuring of the RSSI, sending and receiving of the power control messages,
- authentication procedures,
- pairing of devices,
- encryption negotiation and changing of the link key,
- clock synchronisation between master and slave,
- switching between the modes: hold, sniff and park,
- quality of service negotiation,
- transmissions from the master to the particular slave on the ACL link,
- paging,
- connection establishment.

### 2.4.3 Link Layer Control and Adaptation layer Protocol (L2CAP)

L2CAP has many important tasks in the Bluetooth protocol stack. It is the base of all higher layer protocols like RFCOMM, TCS and SDP.

For this reason L2CAP is responsible for protocol multiplexing, packet segmentation and reassembly (SAR), and conveying of Quality of Service (QoS) information. It is defined only for ACL links, since voice in SCO links is transferred directly over Baseband layer.

Protocols are multiplexed basing on the channel concept – each is assigned a separate logical channel. For this work most important is RFCOMM protocol, which is used to carry IP packets.

### 2.4.4 RFCOMM

RFCOMM, being parts of the ETSI TS 07.10 [7] specification is a protocol used for emulation of the serial ports over L2CAP. It supports up to 60 simultaneous connections between 2 Bluetooth devices. For compatibility issues, emulation of the 9-circuit RS-232 port is supported.

ETSI TS 07.10 specification has been chosen because of its flexibility and large number of applications. Among them we have modems, object exchange (OBEx [8]) and TCP/IP stack via PPP [9] protocol. The last possibility is in the direct scope of this document, since all the performance tests and discussions shall be held in relation to TCP/IP over Bluetooth.

### 2.4.5 Service Discovery Protocol (SDP)

SDP is used to locate the services available in the neighbouring Bluetooth devices. It can be used to discover a printer, a camera, wireless headset or any other device meeting the possessing the required properties.

## *2.5 Compatibility with existing standards*

Bluetooth is a relatively new technology, but it allows fast deployment of applications due to its compatibility with existing standards and protocols. Those standards include Object Exchange (OBEX) - used primarily for IrDA interface, TCS-BIN – based on ITU signalling protocol Q.931 [10] and PPP used for Internet Protocol transmission [11].

## *2.6 Future extensions*

Although Bluetooth has many advantages, there are some bottlenecks included in the technology concept. These shall be removed in the second release of specifications. The improvements include:

- increasing the transmission speed to 12 Mbps,
- putting IP directly over L2CAP layer [12], thus reducing the overhead,
- allowing slave-to-slave communication, optimising the connections.

Those features shall make Bluetooth a strong competitor against technologies like Wireless LAN 802.11(b) [5] and even 10BaseT Ethernet.

## 2.7 Summary

Bluetooth is a very flexible technology that may be used in many applications. Those include cordless telephony, emulation of modems, wireless headsets, and data transmission, including multimedia applications, which are in the scope of this document.

The most important features that predispose this technology for multimedia applications are: relatively high transmission speed – up to 432kbps symmetrical, 721/57.6kbps asymmetrical,

- low (~10 ms RTT for IP), predictable delays,
- built-in Quality of Service mechanisms (flowSpec [13]),
- low size and power consumption (important for handheld devices),
- low cost.

# 3 Multimedia streams: format and transmission

Audio-video transmission became very popular since the invention of television in 1940s. Initially, picture has been transmitted in an analogous form. Basically two standards exist for such transmission: PAL, used mostly in Europe and NTSC used in US and Japan. They differ in resolution (704x576 for PAL and 640x480 for NTSC) and frequency of presenting frames (50Hz and 60Hz respectively).

For the transmission in computer networks sound and picture first have to be digitised to the stream of bits. Table 3 shows bandwidth required to transmit digitised video streams.

| Frame size PAL | Required bandwidth at 25fps [Mbps] |
|---|---|
| QSIF: 176x144 | 14.85 |
| SIF: 352x288 | 59.4 |
| Basic: 704x576 | 237.6 |

| Frame size NTSC | Required bandwidth at 30fps [Mbps] |
|---|---|
| QSIF:160x120 | 13.5 |
| SIF: 320x240 | 54 |
| Basic: 640x480 | 216 |

**Table 3. Required bandwidth [Mbps] for various frame sizes at 24 bits color depth (PAL and NTSC).**

As the calculations show, more than 200Mbps bandwidth is needed for the transmission of one stream only. This is more than a single OC-3 link or 100-BaseT Ethernet. Even size reduction doesn't help much: QSIF streams require 13-14 Mbps. That's why compression technologies are needed to reduce the size down to at least several Mbps.

## 3.1 Compression technologies

Due to their properties, audio and video have to be compressed separately to achieve best result. Following paragraphs describe most popular and widely used compression technologies for both sound and picture.

### 3.1.1 Sound compression

Audio compression bases on the properties of the human ear and properties of sound by itself. Different techniques exist which may be divided into 2 groups: conferencing coders-decoders (codecs) and streaming codecs.

Compression is needed because of the same reasons as in video: we may reduce the size significantly without deteriorating the quality. Most popular and efficient codecs are summarised in Table 4.

| Codec name | Description | Bandwidth | (S)treaming/ (C)onferencing |
|---|---|---|---|
| MP2 | MPEG-1 layer 2: audio track in MPEG-1. | 64-256 kbps | S |
| MP3 | MPEG-2 layer 3: audio track in MPEG-2. | 32-192 kbps | S |
| RealAudio | Commercial standard from RealNetworks | 8.5 – 64 kbps | C |
| G.711 | Basic PCM format, no compression. | 64 kbps (8 kHz * 8 bits) | C |
| G.723 | Commercial GSM codec | 5.3-6.4 kbps | C |
| GSM | Standard for mobile telephony | 7.8 - 13.6 kbps | C |

**Table 4. Basic audio codecs and their properties.**

### 3.1.2 Video compression

There are several video compression technologies available nowadays. Among them most popular and efficient are MPEG 1, 2 and 4 for broadcasting and H.261 and H.263 used for conferencing.

**MPEG1**

MPEG-1 [14] is first of the family of standards created by the Motion Picture Experts Group [15], which started the revolution in video compression technologies.

MPEG-1 is a lossy compression: this means that some of the details like picture size, colour depth or motion speed are reduced.

Following steps are performed to reduce the clip size:

- each frame size is reduced to 320x240 pixels,
- Discrete Cosine Transform (DCT) is applied to each 8x8 block in the frame,
- achieved results are rounded, i.e. accuracy is decreased,
- intra-frame compression is applied basing on 16x16-pixel macroblocks.

This algorithm reduces the stream to about 1.5 Mbps, offering VHS quality. This value is acceptable for the transmission in computer networks based mostly on Ethernet connections, although it is still too much for Bluetooth links offering up to 721kbps. Nevertheless, MPEG-1 system stream shall be used as a basic testing format. Such decision has been made because software available both for sending and reception of the streams supports such standard. It is also possible to create streams with accurately desired bandwidth and measure how Bluetooth links manage to transport them.

**MPEG-2**

MPEG-2 [16] has been created as an improvement to MPEG-1. In basic configuration, it offers quality of TV transmission at rates ranging from 2-20 Mbps, but scales up to 40 Mbps offering HDTV quality. It is also a basic standard for Digital Video Broadcasting (DVB [17]) used nowadays widely in digital transmission of satellite TV. DVB uses streams from 2 to 12 Mbps.

The basic upgrades of MPEG-2 to MPEG-1 are:

- support for interlaced video (TV-like transmission),
- improved coding options,
- scalability,
- several Profiles and Levels used for different types of transmission.

**MPEG-4**

MPEG-4 [18] is a further extension of MPEG family of standards. Its main improvements are object-oriented video coding and high scalability: from 8 kbps up to 40 Mbps. MPEG-4 is thought to be the leading standard for 3rd generation mobile networks such as UMTS or cdma2000. Unfortunately, at the time of writing this thesis no player supporting MPEG-4 and any of the streaming standards has been available for Windows. This is why no tests could be performed.

**RealVideo**

RealVideo is a compression standard which has been developed by RealNetworks [19] company. It bases on the same principles as MPEG-2 compression, but has been designed especially to transmit data in computer networks. What is very interesting, RealVideo uses just 1 codec for video, which scales from 10 to about 450 kbps.

Unfortunately RealVideo is a commercial standard and no further details are available. Nevertheless, both server and player are publicly available and this technology shall be used to test the effectiveness of the Bluetooth link.

## 3.2 Streaming protocols

Streaming protocols may be divided into 2 groups: delivery of the content and acquiring (ordering) the transmission.

Multimedia stream is carried usually by the Real-Time Protocol stack (RTP/RTCP) described in [20]. It is being used by most of the solutions, including QuickTime, vic, rat and is also part of H.323 protocol stack [21] used by Microsoft Netmeeting.

The advantages of this protocol are its simplicity and flexibility. In IP domain, it is carried over simple and effective User Datagram Protocol (UDP), but it may also use ATM AAL5 if needed.

### 3.2.1 Real-Time Streaming Protocol (RTSP)

RTSP is an application-level protocol for control over the delivery of data with real-time properties [22]. It provides a framework for enabling on-demand delivery of real-time data, such as audio and video streams (both live and pre-recorded). It is important to understand that RTSP doesn’t deliver the streams by itself - requested streams may be delivered over UDP, TCP, RTP or any other suitable protocol. Delivery parameters, such as transport protocol, used ports and media types are negotiated by means of RTSP messages. It provides mechanisms to play, pause, stop, and resume a stream, of which only some may be implemented.

Basic methods supported by RTSP include:

- SETUP: allocates the resources and starts a session
- PLAY and RECORD: start playing/recording of a stream
- PAUSE: temporarily halts a stream
- TEARDOWN: frees the resources and closes the session

RTSP is supported by both Darwin Streaming Server from Apple and Real Server from RealNetworks and shall be used for testing purposes.

### 3.2.2 Real-Time Transmission Protocol (RTP)

RTP [20] is a universal, flexible and scalable solution for transporting real-time traffic, ranging from low-rate audio at several kilobits/second to multimedia, audio-video streams at tens of megabits/second. It can be used over both unicast and multicast links and its concept is independent from the transport technology and medium.

The typical implementation, which is of interest for this work, runs over UDP/IP protocol. Nevertheless, other implementations exist, for example over ATM AAL2 or AAL5.

Real-time transport has been split into 2 protocols for convenience: content transport, carried over Real-time Transport Protocol (RTP) protocol and control messages, carried over RTP Control Protocol (RTCP). RTP packets carry user data along with some additional information, like timestamp for synchronisation or endpoints identifiers.

RTCP transmits reports and controls the connection. Reports sent via RTCP have been used for measuring the link quality in Chapter 5.

## 3.3 Conferencing

For conferencing basically 2 standards are used for video compression: H.261 and H.263. They offer decent quality of video and what is very important low delay, which is crucial for duplex transmission.

### 3.3.1 H.261

H.261 [23] is a standard developed in 1990 that uses techniques incorporated later into MPEG. It uses CIF and QCIF file formats, 7, 10, 15 or 20 fps and demands bandwidth of several hundreds kbps, depending on the usage of one of 10 available compression modes.

H.261 is available in MBONE conferencing tool: vic. Usage of this codec and results achieved shall be described in Chapter 5.

### 3.3.2 H.263

H.263 [24] can be seen as the improved version of H.261. It requires lower bandwidth offering the same quality. H.263 may also compress subQCIF resolution, i.e. 128x96. It is so efficient that may be used for videoconferencing over modem links (usually 33.6-56k).

H.263 is the default codec for Microsoft Netmeeting and is supported by vic.

## 3.4 Summary

Audio and video compression is a very important issue nowadays. It allows sending multimedia streams of good quality over very slow links, even at several kilobits/second. MPEG-4 shall certainly play its role in next generation networks, being scalable and efficient solution.

Software available for this work allows testing MPEG-1 and RealVideo streaming, as well as conferencing using H.261 and H.263.

Bluetooth offers speeds up to 721 kbps that should enable both conferencing and video streaming. Next chapters discuss possible scenarios and achieved results.

# 4 Preparation of the testing environment

In order to make tests associated with this work, several hardware and software components are needed. Those components have to be set-up and configured to communicate and work together. The architecture described in this chapter resembles the one that is usually implemented in real-life solutions: devices connected vial LAN with one access point acting as a router/firewall to Internet and caching all the possible resources.

This chapter contains the installation instructions for Bluetooth cards and cameras, as well as describes the concept and configuration of the access point. Later configuration of the streaming servers and players is presented, along with the conferencing software.

## 4.1 Hardware components

To create the testing environment, following hardware components have been used:

Computer acis1:

- desktop PC, 600 Mhz Intel Celeron processor, 128 MB RAM, 10 GB HDD,
- 10/100 Mbps Ethernet network card,
- ISA/PCMCIA adapter,
- sound card,
- Windows 98.

Computer mm:

- Acer 522TX notebook, 700 Mhz Intel Pentium III processor, 128 MB RAM, 12 GB HDD,
- integrated sound card,
- Windows 98.

Two Digianswer/Motorola Bluetooth MK II PCMCIA cards [25].

Computer linws1:

- desktop PC with 350 Mhz Pentium II processor, 128 MB RAM, 10 GB HDD,
- RedHat Linux 7.1 [26].
- 2 Philips Vesta cameras [27], USB slot.

The hardware architecture is presented in Figure 6.

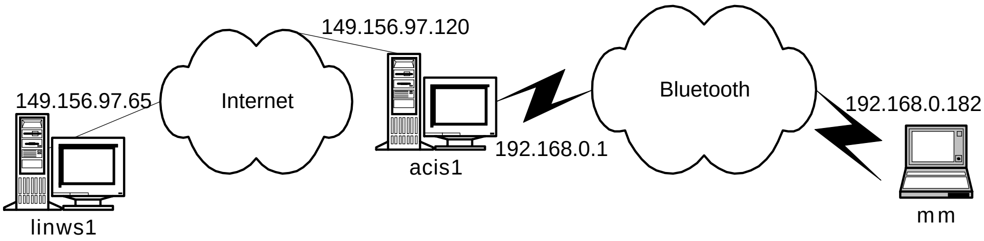


**Figure 6. Testing envoronment hardware architecture**

## 4.2 Bluetooth hardware and software installation

Two Bluetooth PCMCIA cards from Motorola have been used for testing purposes. This allows for creation of a basic piconet with 1 master and 1 slave device.

There are several steps that must be completed to install Bluetooth cards. Those steps are presented in this section.

**1) Hardware drivers installation in Windows 98**

After inserting the card into Type II PCMCIA slot and booting Windows standard driver installation wizard shall appear. Bluetooth card from Digianswer acts like an Ethernet card, so standard settings may be used.

**2) IP address/network settings**

In order to let the Bluetooth devices communicate via TCP/IP protocol stack the cards have to get IP addresses assigned and network settings have to be configured.

For the purpose of this work, private IP addresses have been chosen from the class 192.168.0.X. IP address has to be assigned for both computers equipped with the Bluetooth card.

To set the IP addresses in Windows 98, click with a right mouse button on the ‘Network neighbourhood’ icon and find ‘TCP/IP protocol-> Bluetooth Ethernet Adapter’ in the ‘Configuration’ tab. Click on ‘Properties’ and in the ‘IP Address’ tab choose ‘Specify IP address’. In the ‘IP Address’ field, enter:

```
192.168.0.1 for acis1
```

and

```
192.168.0.182 for mm
```

Network mask is common for both devices and is 255.255.255.0.

Next step is setting the default router on mm. Acis1 shall provide access to the network for mm, so enter:

```
192.168.0.1
```

in the ‘Gateway’ tab.

This ends the basic network configuration.

**3) Bluetooth Software Suite**

Bluetooth Software Suite is a set of programs for configuration, management and operation of Bluetooth cards. The installation files are available on the CD shipped with the cards. Upgrades can be downloaded from Digianswer Support Page [28].

From those powerful tools ‘Bluetooth Neighbourhood’ shall be used. It can be used to discover Bluetooth devices in range and create a wireless link between them.

For the purpose of this work Network profile has been chosen, because it enables TCP/IP connection establishment between 2 devices. Following steps are needed to establish such connection:

a) discovery of the neighbouring devices.

During discovery phase, Bluetooth device must hop through all available channels and send inquiry packets - this is why it lasts about 10 seconds.

To discover neighbouring devices, in ‘Bluetooth Environment’ choose View/Refresh from the menu or press F5 key. The following screen should appear:

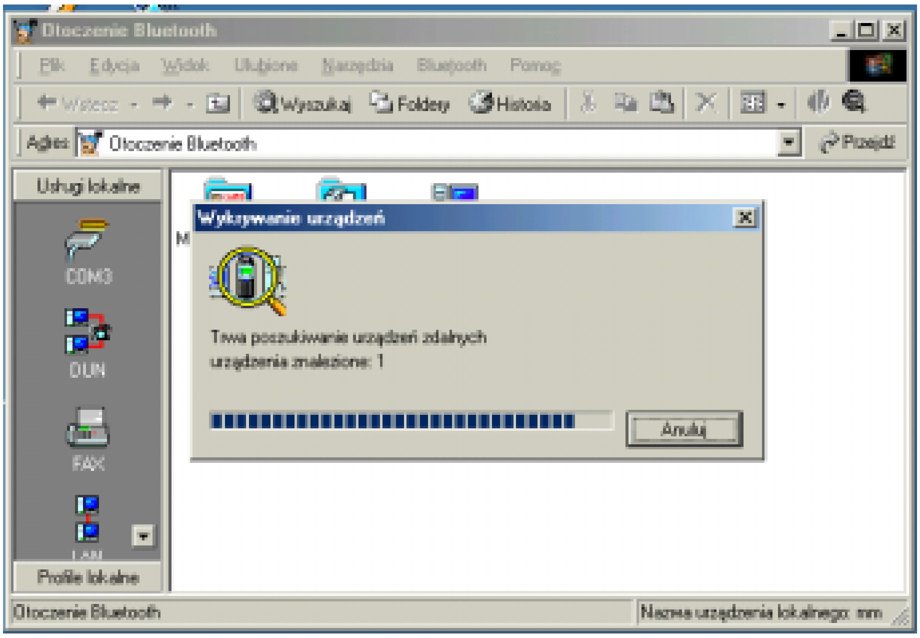


**Figure 7. Discovery of the neighbouring devices.**

When the other device is found (`acis1` in this case), following screen shall appear:

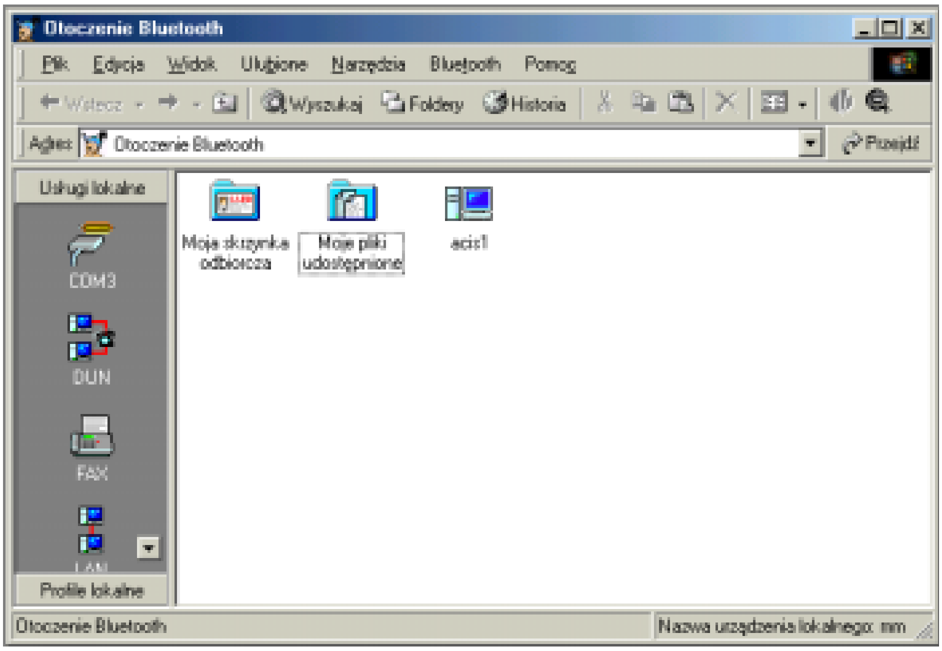


**Figure 8. Discovered device in range.**

After finding the other computer equipped with Bluetooth card connection has to be established. For this, click twice on the icon of the other device (`acis1`). Software will try to detect services available on the other device. Among them there should be 'Network' profile, which has to be used.

After clicking twice on the 'Network' icon and confirming that we want to establish the connection following screen shall appear:

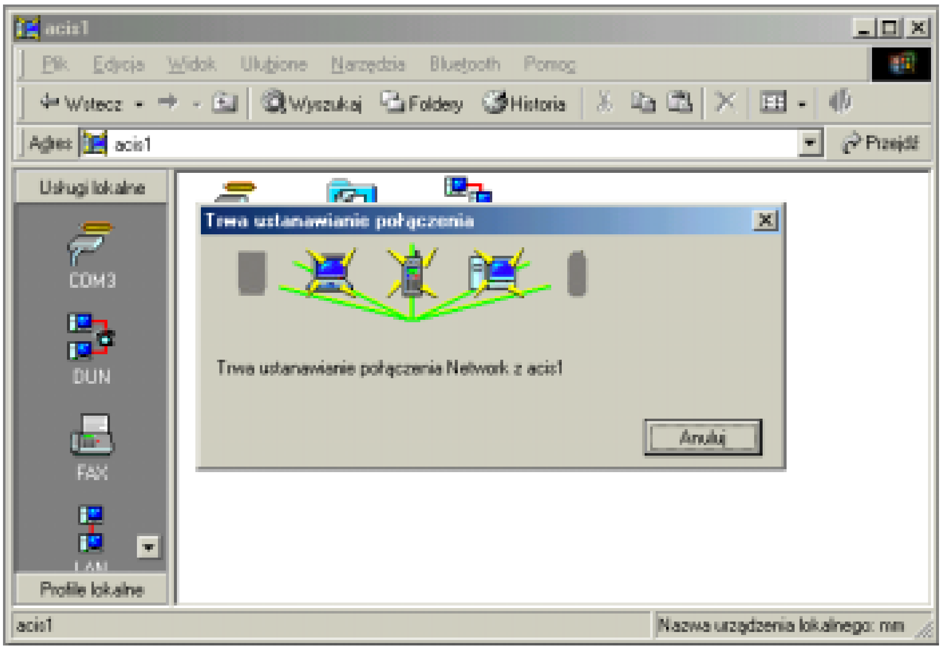


**Figure 9. Network connection establishment.**

If everything goes all right, we should see the 'Network' icon with yellow cross, meaning that we have the connection:

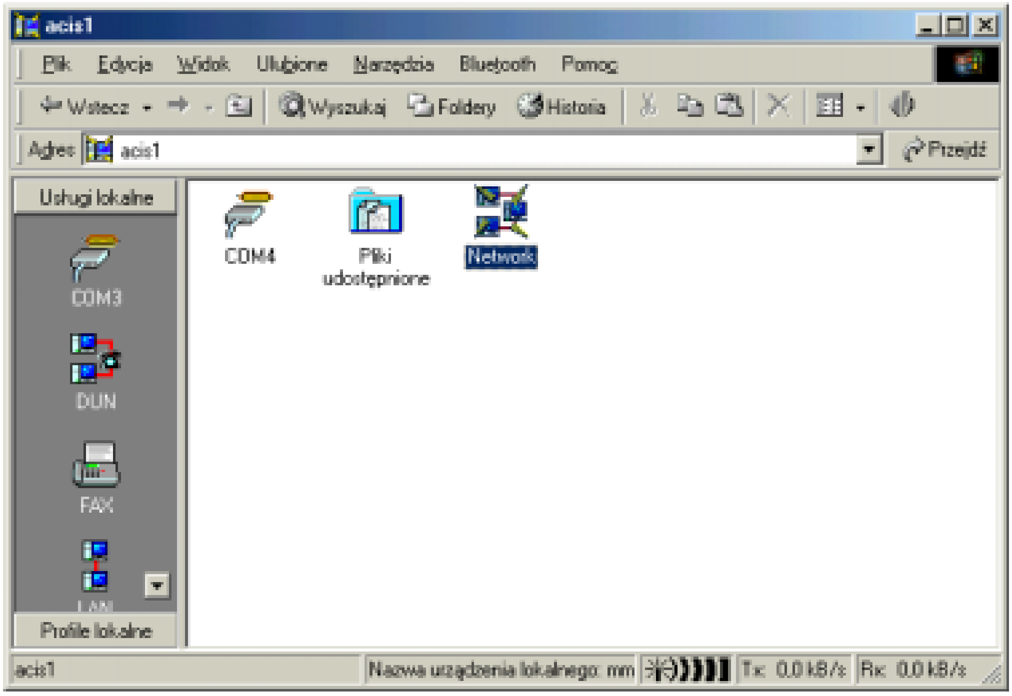


**Figure 10. Connected to acis1.**

To test if there actually is a IP connection `ping` command may be used. To do it, click Start/Run and enter:

```
ping [IP_of_the_other_device]
```

In this case on mm:

```
ping 192.168.0.1
```

or on acis1:

```
ping 192.168.0.182
```

If this test is successful it means that the IP connection has been established.

## 4.3 Access point installation and configuration

In order to provide access to the resources and services in the external networks (i.e. Internet) from Bluetooth piconet, access point is needed. Such device has 2 interfaces: Bluetooth and usually Ethernet which connects to Internet. The concept of the access point is presented in Figure 11.

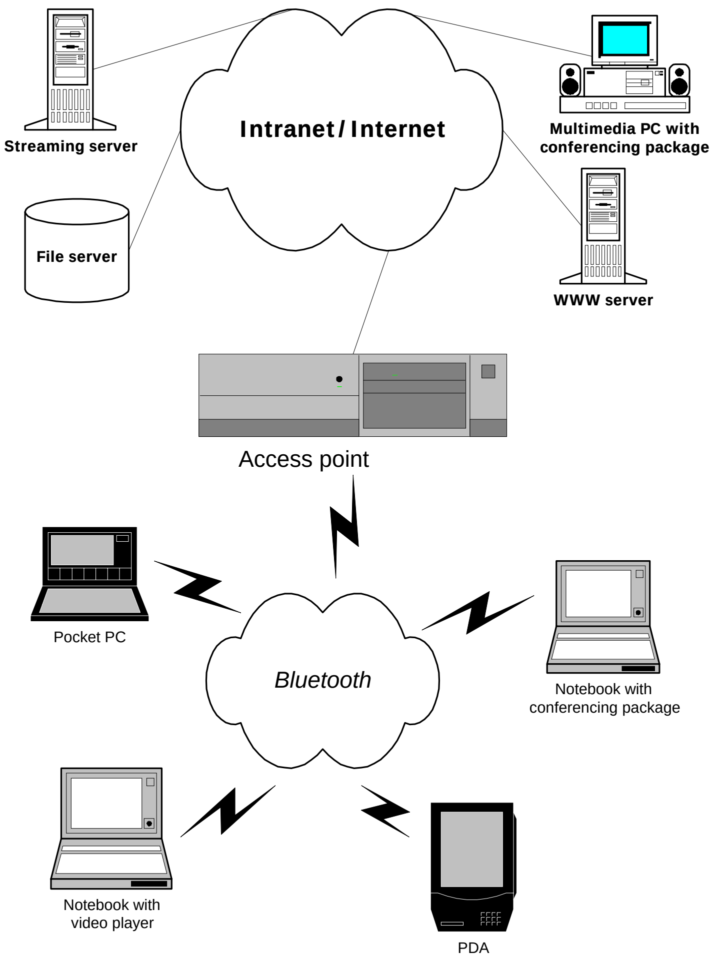


**Figure 11. Concept of the access point**

Basic functionality of the access point includes just routing of the packets, but may extend to a very powerful node, including:

- HTTP/HTTPS proxy,
- FTP proxy,
- Network Address Translation (NAT),
- DNS forwarder,
- RTSP proxy.

Such functions enable easy configuration of the devices and minimise the usage of the external network. Although Internet links are becoming faster and faster, network congestion is a serious problem. Proxies help to decrease the usage of external connections, caching most often used files or DNS entries.

Those functions are usually realised by a specialised hardware or by workstations with linux. Unfortunately, drivers for Digianswer Bluetooth cards are available only for Windows, so a similar solution under this system had to be found.

From several examined packages, WinRoute from TinySoft [29] has been chosen, since it has al the necessary properties.

### 4.3.1 Installation and configuration of WinRoute

After downloading the installation file and executing it the installation wizard shall appear. This setup program installs all the files and creates a new group in the Programs menu.

When all the necessary files are downloaded, WinRoute has to be configured to provide access from wireless, i.e. Bluetooth network to Internet services and resources.

After installing the package and restarting the system WinRoute engine monitor icon  shall appear in the system tray. Click twice on the icon and press ‘OK’ – the password is initially blank.

Several steps need to be completed:

**1) Enabling Network Address Translation (NAT)**

To enable NAT, choose ‘Interface table’ from the ‘Settings’ menu. In the appearing Window, choose the interface providing Internet connection (in this case, Ethernet). Click on ‘Properties’ and check the box: ‘Perform NAT with the IP address of this interface on all communication passing through’.

**2) HTTP proxy settings**

HTTP proxy is enabled by default, listening on port 3128. To change the default port, choose Settings/Proxy Server.

**3) DNS forwarding**

DNS forwarding is also enabled by default. To change specific properties, choose Settings/DNS forwarder.

**4) Port mapping**

Unfortunately, WinRoute doesn’t interpret RTSP packets and thus a special solution must be used to enable multimedia streaming. Certain range of UDP ports, used by RTSP sever to address a client has to be sent directly to one, chosen device connected via Bluetooth interface. This means that if WinRoute sees a packet with destination UDP port 6790-7110 it shall send it to the specified host.

To enable this, from the Settings menu Advanced/Port mapping has to be chosen. Then 'Add', and enter information as shown in Figure 12.

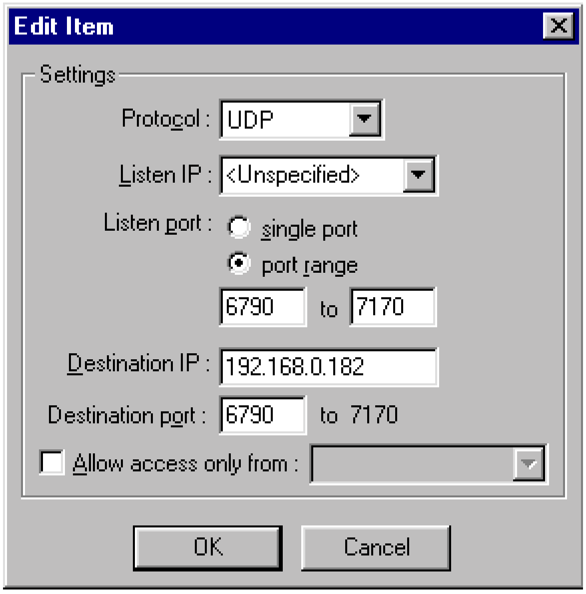


**Figure 12. Port mapping for streaming**

**IMPORTANT NOTE**: RTSP Proxy and Port mapping cannot operate together. Either of the solutions may be used. Instructions how to enable/disable those solutions are contained in the description of testing scenarios in Chapter 5.

## *4.4 Conferencing software and hardware*

Conferencing packages became very popular in the last years. They offer new possibilities for the users, such as international voice or video calls across the globe using low-cost Internet connections. They also enable remote work, application and file sharing, distributed whiteboards or text chat.

There are many packages offering such services, but for this thesis two of them have been tested: Microsoft Netmeeting 3.01 and MBONE tools vic/rat [30]. Netmeeting is shipped with every Windows installation and this is its strongest advantage: it can be found everywhere.

On the other hand, `vic` and `rat` are available for many platforms like Linux, Solaris, Windows, FreeBSD and are widely used in the University environment.

### 4.4.1 Hardware installation

Conferencing packages used for audio and video communication require devices for playing and recording of sound and picture.

While nowadays virtually all computers are equipped with a sound card video capture devices are still rare. For the purpose of this thesis 2 Philips Vesta webcams have been used. Those devices are connected to the USB port and offer resolutions up to 640x480 pixels with 25 fps.

**Figure 13. Philips Vesta webcam**

Such cameras make a perfect solution for the video conference, since they produce enough output for the H.261 and H.263 codecs.

USB interface makes the installation very easy. It is enough to plug the camera cable into the right slot and insert the CD with drivers provided by the producer.

### 4.4.2 Software installation

As mentioned above, Microsoft Netmeeting comes with every Windows installation and is installed by default. If the installation has been disabled, it can be found in every Tucows Mirror [31]. The initial screen from Netmeeting is shown in Figure 14.


**Figure 14. Microsoft Netmeeting**

Vic and rat may be found at the University College London, Multimedia Group software page [30].

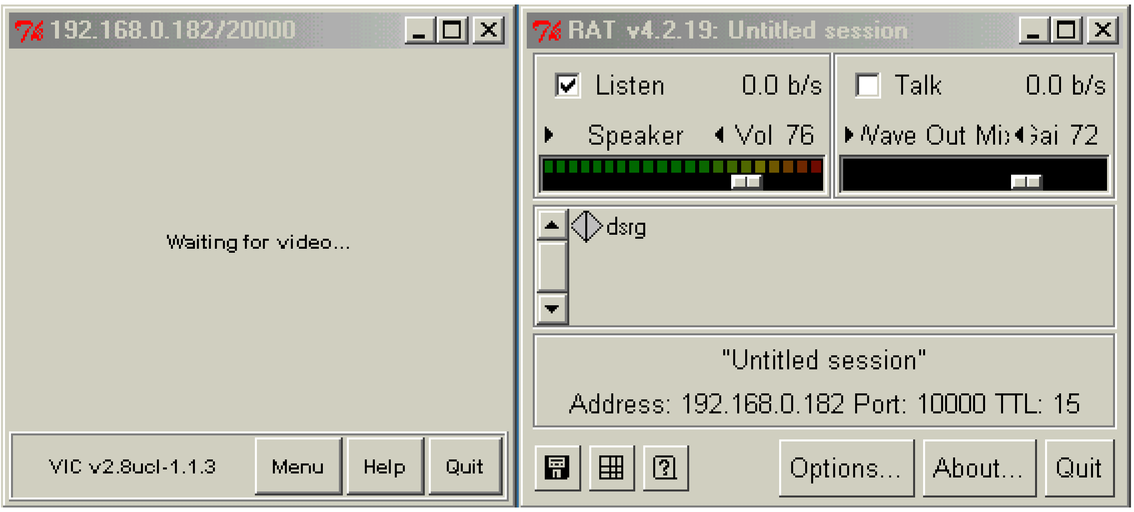


**Figure 15. Vic and rat**

They both include installation wizards, which guide the user through the setup process, asking for the properties like destination directory and user information. After filling in all the information the software is ready to use. Execution commands shall be presented with testing scenarios, present in Chapter 5.

## 4.5 Video servers and proxies

In order to get access to the video clips basically two technologies may be used: file download and file streaming. While file download is available virtually on any Internet server through FTP or HTTP server, it has a very strong disadvantage comparing to streaming: whole video clip has to be downloaded before it starting playback. The advantage of streaming is especially visible in the case of long clips, lasting several minutes or even hours. Viewer may start to watch the video almost instantly, after downloading just a small initial part used to keep the video playing smooth (buffering).

Two products available on the market have been chosen for testing: Real Server 8.01 [32] from RealNetworks and Darwin Streaming Server 2.0 [33] from Apple Computers. The decision has been made basing on the features offered by those products and their availability. Both products are very widely used in professional applications, have Linux releases and are available for evaluation.

### 4.5.1 Darwin Streaming Server

Darwin Streaming Server is a free distribution of the QuickTime Streaming Server [34]. It is a universal architecture for streaming multimedia using RTSP/RTP protocol stack. The package comes in form of the source code in C++ packed into the tar.gz file. Those sources are prepared for compilation on Linux (both Intel and PowerPC), SunOS, and FreeBSD. In this case, Linux (RadHat 7.1) has been used as the compilation and execution platform.

**Installation and running**

Several steps must be completed to make the server running. They are presented in this section.

**1) Downloading the sources.**

Darwin Streaming Server sources are available for download from Apple Public Sources page [33].

At the time of writing this thesis, file is named `StreamingServer-2.0-4-1.tar.gz`. This name shall be used as a reference in the following steps.

**2) Unpacking the sources**

The packed sources should be copied to the desired destination directory, where

```
> tar –zxvf StreamingServer-2.0-4-1.tar.gz
```

command should be executed. This command shall unpack the sources and create new subdirectory 'DarwinStreamingSource'.

**3) Building**

After entering the DarwinStreamingSource directory, the sources may be compiled. Special script is prepared to build the sources. It should be enough to run the command:

```
> ./Buildit
```

If everything goes all right, server executable should be built in the form of the `DarwinStreamingServer` file.

**4) Configuration**

The last step of the installation process is editing the configuration files to adapt the defaults to the needs.

Darwin Streaming Server has 2 configuration files: `streamingserver.conf` and `streamingrelay.conf`. Both of them should be placed in the `/etc` directory and edited, if needed. For the purpose of the tests default values shall be used, including the directory for placing movies, which will be `/usr/movies`.

**5) Starting the server**

RTSP server, using default port 554 needs to be run with root privileges. It is best to put the server in the background, executing:

```
> ./DarwinStreamingServer &
```

After completing all the steps, the server is ready for operation, i.e. multimedia streaming using the combination of RTSP for content ordering and RTP/RTCP for delivery. This scenario shall be tested in Chapter 5.

DarwinStreamingServer puts all the information necessary for debugging in the logs located in the /var/streaming/logs/ directory. To keep trace of the server operation, enter:

```
> tail –f /var/streaming/logs
```

### 4.5.2 Real Server

Real Server, available at the time of writing this thesis in release 8.01 is a product from RealNetworks [19] – producer of the RealPlayer. RealServer is used to provide content for the RealPlayer, but may be used as an RTSP server for QuickTime Player as well.

The installation file, which may be downloaded from the RealNetworks Products download page [32] comes in the form of binary executable. Currently, the file is called
‘`rs-8-01-linux-libc6.bin`’ and this name shall be used as a reference.

Since Linux has been chosen as the platform for the installation of streaming servers, all instructions are related to this system. Nevertheless, installation process on the other Unix-like systems should look very similar.

After downloading the installation file it has to be executed, i.e. command:

```
> ./rs-8-01-linux-libc6.bin
```

must be issued.

The installer unpacks the files guides the user through the setup process, asking for properties like destination directory, ports to listen on etc.

After the installation, license file obtained from RealNetworks has to be placed in the license directory, usually 'License' in the RealServer tree.

To start the server, enter the command:

```
> ./Bin/rmserver rmserver.cfg
```

in the RealServer directory. This shall load the executable, all available modules and configuration from the rmserver.cfg file.

Note that RealServer and DarwinStreamingServer cannot run at the same time with default settings, since they use the same port (554) for RTSP server. There are 2 solutions: RTSP server port has to be changed in either of the configuration files or servers can be run on separate machines or in separate time. The last solution has been chosen in work.

### 4.5.3 RTSP Proxy

RTSP proxy is a solution allowing audio-video transmission through firewalls. It has to be installed on the computer, which plays a role of a firewall. RTSP proxy receives the request from the client and asks RTSP server for the requested stream on behalf of the client.

For testing purposes, RTSP Proxy from RealNetworks is used.

What is important to note is that RTSP proxy receives UDP packets from the server on one network interface and sends them to the client on another interface. Thus media stream reaches the client indirectly. This improves the security of the network. RTSP proxy may be configured for example to allow communication with certain hosts only or to filter some content.

The RTSP Proxy installation files are available from the Sourceforge RTSP Proxy project page [35]. They come in the form of the tar.gz file, which has to be unpacked and compiled on the desired operating system (in this case Windows).

To start the proxy it is enough to run the executable. To put the proxy in the debug mode, enter:

```
> rtspproxy –d
```

from the command line.

Sample output from the proxy working in the debug mode is presented below:

```
Listening on port 554
CRtspProxyApp::OnConnection: new client
Setup request for url: rtsp://149.156.97.65/ads600.mov/trackID=2
Setup response: server session = '1191389088191670667', transport =
'RTP/AVP;unicast;client_port=6978-6979;server_port=2000-2001'
We are playing!!!
Teardown request
```

In order to use the proxy, its settings have to be entered into the client application. Those settings are presented together with the installation tips for video streaming players (see Section 4.6)

**IMPORTANT NOTE:** WinRoute with port mapping enabled and RTSP Proxy cannot operate simultaneously, since their functionality is the same and WinRoute overrides RTSP Proxy.

To present a more complete view of RTSP, Appendix A contains the transcript of one of the sessions, including pause in the middle of transmission.

## *4.6 Video streaming players*

In Chapter 3 two video streaming standards have been described: RTP/RTSP and PNA. Those standards make it possible to send the video stream to a certain client. This section describes installation of the chosen clients: Real Player 8 and QuickTime 5.01.

### 4.6.1 Real Player 8

Real Player [36] is a product from RealNetworks which has initially been designed for audio streaming. Due to large demand and increasing capacity of Internet connections video streaming has been added. It is available for Microsoft Windows and various Unix platforms, including Linux.


**Figure 16. RealPlayer 8**

Real Player supports a wide range of codecs and RTSP/RTP and PNA as the streaming methods. The package also provides some statistical information about the bandwidth used, displayed/lost frames, audio quality etc. This data is very useful for the purpose of this thesis.

The product can be downloaded from the RealPlayer page [36]. Installation is similar to other Windows software – the installation wizard asks for the destination directory and user information. After entering this information, Real Player is ready to operate.

**RTSP Proxy setup**

In order to use the RTSP Proxy (see Section 4.5.3) its properties have to be entered in the RealPlayer preferences. Those settings can be found in the menu View/Preferences.../Proxy tab. To use the proposed configuration, check ‘Use RTSP Proxy’ field, enter ‘192.168.0.1’ in the IP field and ‘554’ in the ‘Port’ field, as presented in Figure 17.

To disable the proxy when using WinRoute port mapping, un-check the ‘RTSP Proxy Server’ in the settings shown below.

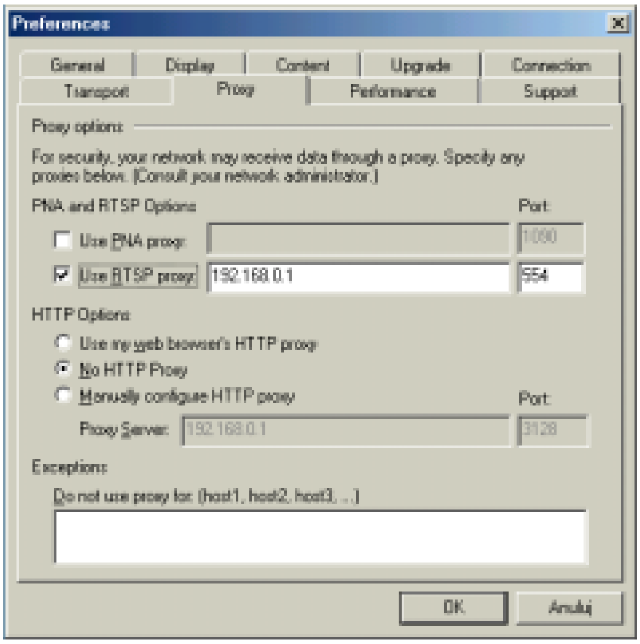


**Figure 17. Setting RTSP Proxy in RealPlayer**

### 4.6.2 QuickTime 5.01

QuickTime Player 5.01 [37] is a product available from Apple Computers. It supports RTSP together with RTP as a streaming technology. A wide set of codecs is available, among which MPEG-1 can be found. Additionally, MPEG-4-compatible codec from 3ivx company [38] may be installed.


**Figure 18. QuickTime Player**

QuickTime Player 5 offers statistics like average bandwidth usage and error rate, which will be discussed in Chapter 5.

The installation file may be downloaded from the QuickTime download page [37]. It contains standard Windows installation wizard, which helps with the package installation and configuration.

**Installation of the 3ivx MPEG-4 codec**

3ivx company prepared a MPEG-4-compatible codec, which can be used to create and play movies in this format.

After downloading the archive from 3ivx page [38] it needs to be installed in the QuickTime codecs directory. It is enough to unpack the provided file (`currently thriv1_d3.qtx`) into the `Windows\System\QuickTime` directory, where all the QuickTime codecs reside.

**RTSP Proxy setup**

If RTSP Proxy running on acis1 is to be used it has to be entered into the QuickTime configuration. It can be found in the Edit/Preferences/QuickTime preferences menu. Then 'Streaming Proxy' has to be chosen from the available configuration sections. Check the 'RTSP Proxy Server' box and enter '192.168.0.1' into IP field and '554' into the 'Port ID' field, as in Figure 19.

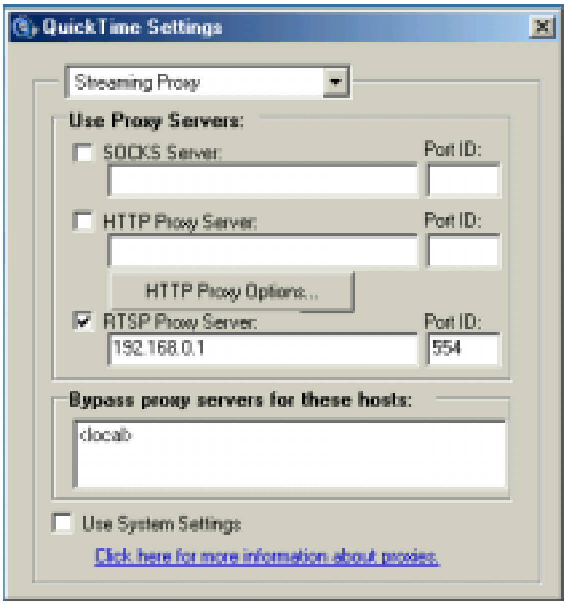


**Figure 19. Setting RTSP proxy in QuickTime Player**

To disable the proxy when using WinRoute port mapping, un-check the 'RTSP Proxy Server' in the settings shown above.

## 4.7 Network monitors

This work focuses on testing the multimedia transmission over Bluetooth links. In order to get the complete picture of the bandwidth usage special software is needed. This package should collect the link utilisation and preferable present the traffic on the chart.

Etherpeek from WildPackets [39] meets those requirements. It has been installed on the client computer (mm) and used for testing purposes in Chapter 5.

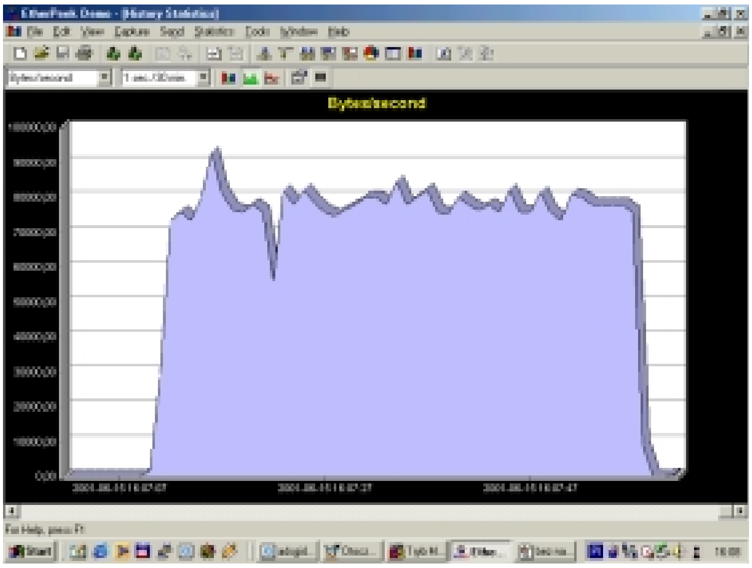


**Figure 20. Etherpeek in operation.**

The evaluation version of the package may be obtained at the Etherpeek page at WildPackets. It allows to capture packets up to 5 minutes, which has been enough to test a single stream.

Apart from Etherpeek, ‘Bluetooth static viewer’ which is part of the Bluetooth Software Suite from Digianswer [40] has been used for testing purposes. It provides detailed information about the link utilisation, packets sent/received, link quality etc. Unfortunately, it doesn’t provide any mechanisms to export this data and reported values can only be observed as presented.

This application can be found in the Start menu, then ‘Programs\Digianswer Bluetooth Demo Card MK II v1.00\Digianswer Bluetooth Democard\Bluetooth Static Viewer’. Sample output from this package is presented in Figure 21.

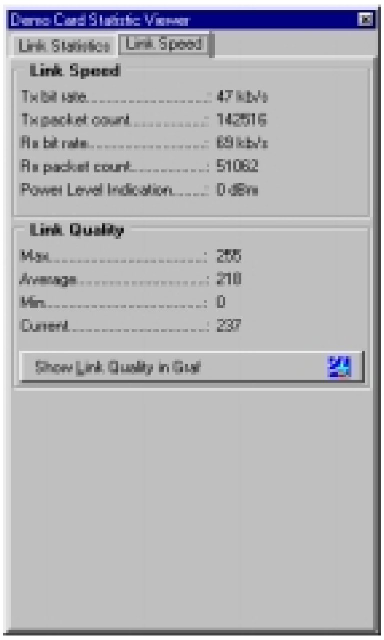


**Figure 21. Bluetooth Static Viewer**

Most important for the purpose of this work are ‘Tx bit rate’ and ‘Rx bit rate’ (top of the window).

Packet count and distribution, presented in the ‘Link Statistics’ tab has been used to examine how IP packets are put into Bluetooth packets. This information has also been necessary to calculate the overhead introduced by the Bluetooth layers.

## 4.8 Video clips: preparation, coding, hinting

Several video clips have been prepared to test the usability of Bluetooth IP link as the multimedia carrier. Following requirements have been put on the clips:

- lots of colours,
- vivid motion,
- many plans, quickly changing.

Such properties are demanded to make the potential errors visible, since intra-frame compression like MPEG or RealVideo bases on the reduction of details.

For those purposes the advertisement recorded from TV has been chosen as meeting all criteria. Basic properties of the recorded video are summarised in Table 5.

| | |
|---|---|
| **Filename** | Ads.avi |
| **File size** | 111.8 Mbytes |
| **Clip length** | 50:09 sec |
| **Video** | |
| **Size** | 110.7 Mbytes |
| **Rate** | 18 Mbps |
| **Frame size** | 320 x 240 pixels |
| **Color depth** | 24 bit |
| **Frame rate** | 10 fps |
| **Audio** | |
| **Size** | 1.1 Mbytes |
| **Rate** | 172 kbps |
| **Sampling rate** | 11.025 Hz |
| **Sampling resolution** | 16 bits |
| **Channels** | 1 (mono) |

**Table 5. Properties of the video clip.**

This clip has been later encoded using `avi2mpg` utility [41]. `Avi2mpg` allows encoding a video clip in Windows AVI format to MPEG-1 stream of the desired bandwidth. It is a command-line utility, but GUI interface is also available (see Figure 22). GUI utility allows entering all the desired parameters, which are converted to the execution command.

For testing purposes, initial stream has been encoded with various speeds ranging from 600 kbps to 725 kbps, with a step of 25 kbps. 32 kbps has been devoted for audio in each case, leaving the rest for video.

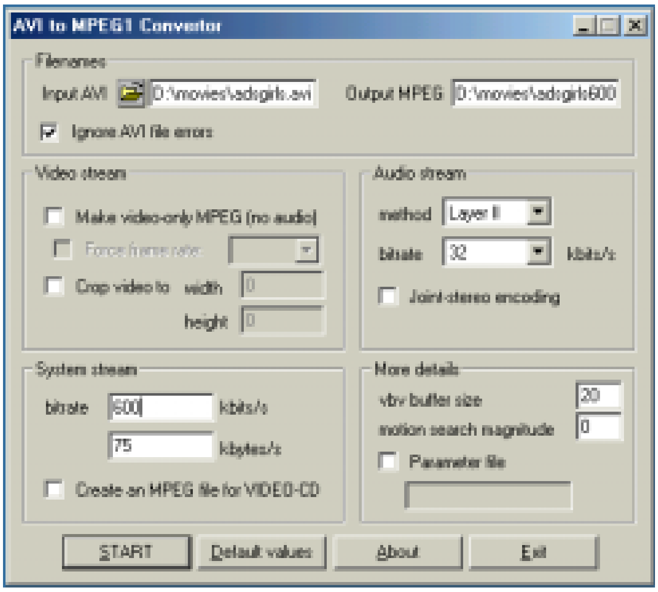


**Figure 22. AVI2MPG GUI utility**

Next, MPEG-1 streams were loaded into QuickTime Player and exported to ‘hinted mov’ format. Hinting movies can be done by choosing Export from the ‘File’ menu, and then Setting ‘Movie to Hinted movie’ in the ‘Export’ field.

‘Hinting’ is the process of augmenting bare video format with information needed for streaming. Table 6 lists encoded files’ sizes, both for MPEG and hinted MOV.

| Format | File size [bytes] | Hinted MOV file size [bytes] |
|---|---|---|
| Uncompressed AVI (~19 Mbps) | 117.501.896 | - |
| MPEG 600 kbps | 3.421.206 | 6.960.863 |
| MPEG 625 kbps | 3.584.189 | 7.285.076 |
| MPEG 650 kbps | 3.684.475 | 7.501.596 |
| MPEG 675 kbps | 3.906.773 | 7.947.368 |
| MPEG 700 kbps | 4.033.244 | 8.204.066 |
| MPEG 725 kbps | 4.163.738 | 8.464.070 |

**Table 6. Encoded MPEG streams with sizes.**

### 4.8.1 Preparing MPEG-4 (3ivx) MOVs

MPEG-4 is a very promising format, designed to give higher quality than MPEG-2 at the same bandwidth. After the installation, 3ivx codec may be used to prepare MPEG-4-compressed QuickTime MOV files.

In order to create such streams, complete the following steps:

1) Launch QuickTime player

2) Load the AVI file into QuickTime (File/Open movie..),

3) Choose File/Export...

4) In the Export field, choose ‘Movie to QuickTime Movie’

5) Press ‘Options...’ button,

6) In the ‘Video’ tab, choose ‘Settings’. Dialog ‘Compressor’ should appear.

7) In the topmost field, choose ‘3ivx Delta 3’.

8) The quality slider should be set to the desired value. In this work 81 and 91 have been used.

9) For sound, Qualcomm PureVoice compressor with 11.025 Hz has been chosen.

10) Choose the desired name and click ‘Save’.

Three files have been created with this codec: two with high quality (81 and 91) and medium quality (53) (see Table 7).

| **Format** | **File size [bytes]** | **Hinted MOV file size** |
|---|---|---|
| Uncompressed AVI (~19 Mbps) | 117.501.896 | - |
| 3ivx Medium Quality (53) | 2.017.829 | 2.017.841 |
| 3ivx High Quality (81) | 3.168.074 | 3.374.342 |
| 3ivx High Quality (91) | 3.736.438 | 3.959.942 |

**Table 7. 3ivx-encoded files**

Transmission quality of those files shall be examined in Chapter 5.

### 4.8.2 Preparation of the RealAudio/RealVideo streams

To create streams for RealServer, Real Networks created a tool called Real Producer, currently release 8.5 [42]. This product can convert movies in AVI format to RealAudio/Video files, which can later be uploaded to the Real Server.

RealProducer installation may be downloaded from the package page [42].

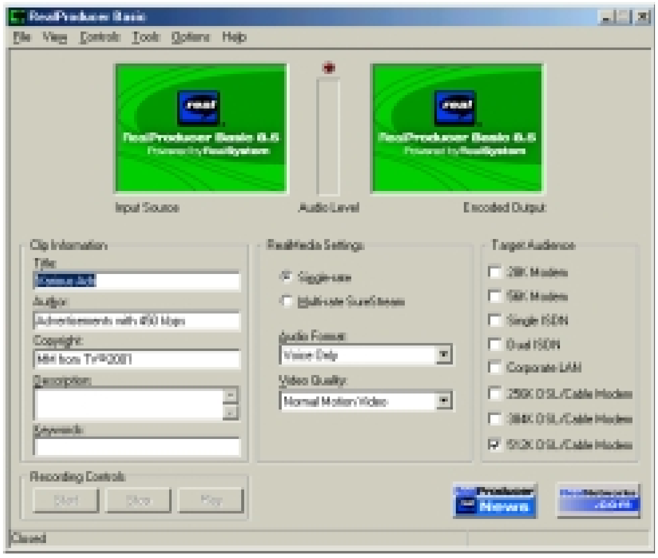


**Figure 23. Real Producer.**

Same videoclip as in section 4.8 has been used to create a streaming content. RealProducer creates streams ranging from 20 to 450 kbps. Those streams are destined for links starting from 28.8 kbps modem to broadband DSL connection (Table 8).

Bandwidth available for video is the difference between the value in the second column (Speed) and selected audio bandwidth from one of the audio columns. Thus, the highest bandwidth available for the video is 450 – 32 = 418 kbps. It is worth to note that bandwidth for Corporate LAN is only 150 kbps. This is because standard LAN connections (usually 10 Mbps Ethernet) are shared by multiple users, and one user shouldn't create constant bandwidth occupation of capacity exceeding this value.

<table>
<tr><th>Target Audience</th><th>Top Speed</th><th>Voice only</th><th>Voice and Music</th><th>Mono Music</th><th>Stereo Music</th></tr>
<tr><td>28.8 kbps modem</td><td>20 kbps</td><td>8.5 kbps</td><td>8.5 kbps</td><td>8 kbps</td><td>8 kbps</td></tr>
<tr><td>56 kbps modem</td><td>34 kbps</td><td rowspan="2">16 kbps</td><td>16 kbps</td><td>20 kbps</td><td>20 kbps</td></tr>
<tr><td>64 kbps single ISDN</td><td>45 kbps</td><td rowspan="3">32 kbps</td><td rowspan="3">32 kbps</td><td rowspan="3">32 kbps</td></tr>
<tr><td>112kbps dual ISDN</td><td>80 kbps</td><td rowspan="2">32 kbps</td></tr>
<tr><td>Corporate LAN</td><td>150 kbps</td></tr>
</table>

<table>
<tr><td>256 kbps DSL/cable</td><td>225 kbps</td><td>44 kbps</td><td>44 kbps</td><td rowspan="3">64 kbps</td><td>64 kbps</td></tr>
<tr><td>384 kbps DSL/cable</td><td>350 kbps</td><td rowspan="2">64 kbps</td><td rowspan="2">64 kbps</td><td rowspan="2">96 kbps</td></tr>
<tr><td>512 kbps DSL/cable</td><td>450 kbps</td></tr>
</table>

**Table 8. Real Producer encoding for various link speeds**

RealProducer uses different codecs for audio compression, but only one, universal codec for video (see Chapter 3).

To create the stream, wizard with standard settings has been chosen. As an output, 3 files have been created (see Table 9).

| **Filename** | **Size [bytes]** | **Audio rate [kbps]** | **Video rate [kbps]** |
|---|---|---|---|
| Ads256.rm | 1,454,994 | 44 | 181 |
| Ads384.rm | 2,230,029 | 64 | 286 |
| Ads512.rm | 2,865,034 | 64 | 384 |

**Table 9. Files created with RealProducer**

The top bandwidth available in RealProducer is 450 kbps, which is less then the maximum Bluetooth speed (721 kbps). This is because products from RealNetworks are designed for private use and are not aimed at providing TV quality: their aim is to provide highest quality at the available bandwidth.

## *4.9 Summary*

Configuration presented in this chapter resembles the scenario used in most companies: one computer with one public IP address acting as a router to Internet for the computers in the local network. Those computers have private IP numbers, both for security reasons and because of the lack of the public ones. Acis1 acts as a router (access point) in this scenario and mm represents computers in the local network. The only difference to the common scenario is the connection technology, as wireless link based on Bluetooth is used.

Streaming and conferencing packages presented will be used in the testing scenarios presented in Chapter 5 and will provide information on how Bluetooth link is suitable for sending multimedia.

# 5 *Testing scenarios and results*

Bluetooth link offers speeds up to 721kbps downlink and simultaneously 57.6 kbps uplink for baseband packets. Due to the overhead introduced by the higher layers speed available for the end-user is reduced. This chapter presents several testing scenarios that shall provide information to what extent the speed is reduced. Those scenarios aim at testing how Bluetooth links cope with multimedia transmission. Please note that all installation steps and settings, which are used in this chapter, have been presented in the previous chapter.

This chapter contains several testing scenarios and results. First, some basic tests are presented, like ping, file transfer and netio, which give an overview of the link capabilities. Then two conferencing packages are examined: Netmeeting and MBONE tools. Streaming scenarios are tested in the end.

## *5.1 Basic tests*

In order to test the Bluetooth link and compare it to the other technologies 3 tests have been made: ping showing link delays, large file transfer via http and netio test showing the TCP throughput.

### 5.1.1 Ping test

Ping is a basic network application operating on the IP layer. It can be used to test the presence of the connection between 2 devices, but also to see the delay introduced by the link.

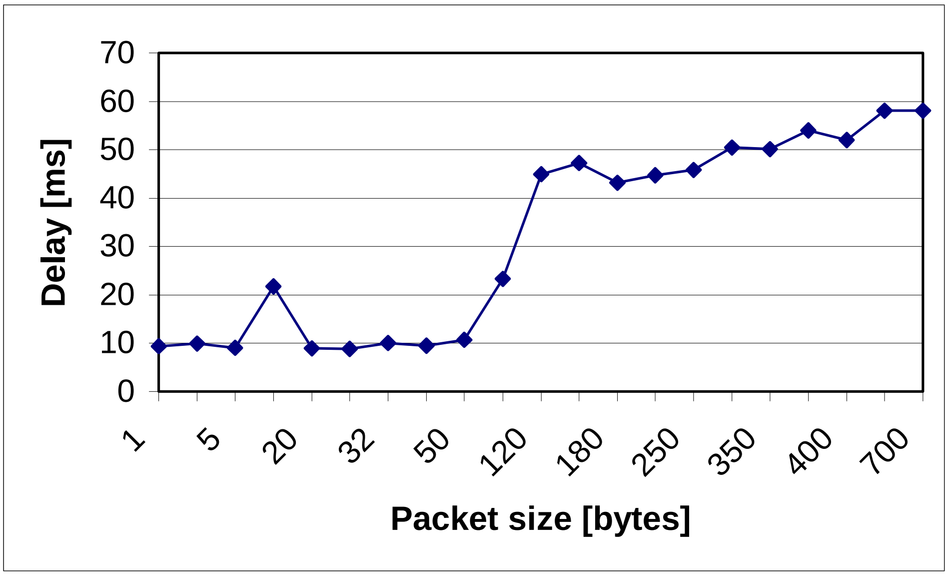


**Figure 24. Ping results**

Figure 24 shows ping results for various packet sizes, ranging from 1 byte to 800 bytes. Those sizes are without IP and ICMP header, which by default take 20 + 8 = 28 bytes. But to get more accurate result, modified ping application had to be used, since standard ping available in Windows gives results in milliseconds. For this purpose, sample ping source from MSDN [43] has been modified to give accuracy in microseconds. The modification extends the message header to 12 bytes, adding timestamp field. This way 32 bytes have to be added to payload to get the actual IP packet size. Listing of this application is available on the enclosed CD.

Packet sizes in this test range from 33 bytes to 832 bytes (1-800 bytes without header). The delays range from 10 ms to 60 ms, depending on packet size. From those delays, a rough estimation of the bandwidth may be calculated. The minimum delay for 832 bytes is 58 ms and this is the time for the packet to travel in both directions. To get the estimation for 1 direction, it has to be divided by 2.

Then, to get the value in kbps the result of division of packet length by time has to be multiplied by 8. The calculation looks as follows:

$$832 \text{ bytes} / (58 \text{ ms} / 2) * 8 = 229{,}5 \text{ kbps}$$

As already mentioned, this is a very rough estimation of the bandwidth because of the delay introduced by the segmentation, reassembly and layer processing. Nevertheless, it gives a perspective for the results possible from other tests.

During the ping test the usage of Bluetooth packets to carry IP packets has been observed. Table 10 lists the packets used to carry user data on ACL links. The sizes indicated show space available for the user data, i.e. layer L2CAP and above.

| **Packet name** | **User payload size [bytes]** | **Description** |
|---|---:|---|
| DM1 | 17 | 1 time slot, no FEC, 16-bit CRC |
| DH1 | 27 | 1 time slot, 2/3 FEC, 16-bit CRC |
| DM3 | 121 | 3 time slots, 2/3 FEC, 16-bit CRC |
| DH3 | 183 | 3 time slots, no FEC, 16-bit CRC |
| DM5 | 224 | 5 time slots, 2/3 FEC, 16-bit CRC |
| DH5 | 339 | 5 time slots, no FEC, 16-bit CRC |

**Table 10. Bluetooth packets carrying user data**

Observations show that if packet size exceeds of the limits, next packet of the minimum size is taken. For example, to transmit 400 bytes in IP packet DH5 + DH1 packets are used, to transmit 500 bytes a pair of DH5 and DH3. This method reduces the number of occupied slots to the minimum.

| IP packet size [bytes] | Payload size (w.o. IP header) [bytes] | Bluetooth packets used | Bluetooth packets size [bytes] |
|---|---|---|---|
| 28-103 | 8-83 | DM3 | 121 |
| 104-165 | 84-145 | DH3 | 183 |
| 166-206 | 146-186 | DM5 | 224 |
| 207-317 | 187-297 | DH5 | 339 |
| 318-334 | 298-314 | DH5 + DM1 | 356 |
| 335-344 | 315-324 | DH5 + DH1 | 366 |
| 345-438 | 325-418 | DH5 + DM3 | 460 |
| 439-500 | 419-480 | DH5 + DH3 | 522 |
| 501-541 | 481-521 | DH5 + DM5 | 563 |
| 542-652 | 522-632 | DH5 + DH5 | 678 |
| 653-669 | 633-649 | DH5 + DH5 + DM1 | 695 |
| 670-679 | 650-659 | DH5 + DH5 + DH1 | 705 |
| 680-773 | 660-753 | DH5 + DH5 + DM3 | 799 |
| 774-835 | 754-815 | DH5 + DH5 + DH3 | 861 |
| 836-876 | 816-856 | DH5 + DH5 + DM5 | 902 |
| 877-987 | 857-967 | DH5 + DH5 + DH5 | 1017 |
| 988-1004 | 968-984 | DH5 + DH5 + DH5 + DM1 | 1034 |

**Table 11. Bluetooth packets carrying IP**

Table 11 shows which Bluetooth packets are used for which ranges of IP packet sizes. It is very significant to note that packets are fragmented on L2CAP layer, not on IP layer. This is a different approach to for example Ethernet, where layer 2 doesn't provide any fragmentation. Such approach allows full utilisation of the link resources, since lower layers are aware of the network properties.

The ping test shows that link speed increases with the slots used and should be at least 230 kbps. It also shows that there's a significant overhead of at least 18 bytes when taking DM3 packet. Those bytes are used for the layers below IP, i.e. Baseband, L2CAP, RFCOMM and PPP.

### 5.1.2 File transfer (http)

File transmission is a basic service in Internet: we download web pages, pictures, video clips, applications. For testing purposes, download of the Netscape Navigator installation from http server on `linws1` has been performed. The file (`tucows_n32e408.exe`) occupies 9.73MB (10 207 782 bytes).

| | Bluetooth | 10BaseT Ethernet |
|---|---|---|
| **Transmission speed (kB/s)** | 81.7 | 773.3 |
| **Transmission speed (kbps)** | 653.6 | 6186.4 |

**Table 12. File transmission speed**

This test shows that effective bandwidth available for file download, basing on TCP is around 650 kbps for Bluetooth. This is due to the overhead introduced by the layers' headers: starting from TCP, ending on Baseband. Thus the overhead is about 11%.

Bluetooth link is about 10 times slower than the most popular 'wired' technology – 10BaseT Ethernet. It is important to note that the results give the effective speed measurements, which are smaller than the theoretical values.

### 5.1.3 Netio

Netio [44] is a simple tool testing the transmission speed between 2 computers basing on several packet sizes sent by TCP. One instance of the program must act as a server, the other one as a client. Netio sends 1- to 32 kB-long packets to the server and calculates the link throughput.

In the test, `acis1` acted as a server and `mm` as a client. The testing procedure is as follows:

a) create the Bluetooth connection between `acis1` and `mm`,

b) close all network applications to keep the link free of traffic,

c) on `acis1`, in the netio `bin\` directory enter the command executing Windows version in server mode for TCP:

```
bin\> nt –s –t
```

d) on mm, in the netio bin\ directory enter the command executing it in client mode, connecting to `acis1` (see IP configuration in Chapter 4):

```
bin\> nt –t 192.168.0.1
```

Below is the sample output from netio:

```
NETIO - Network Throughput Benchmark, Version 1.11
(C) 1997-1999 Kai Uwe Rommel

TCP/IP connection established.
1k packets:     67623 bytes/sec
2k packets:     66384 bytes/sec
4k packets:     69945 bytes/sec
8k packets:     70136 bytes/sec
16k packets:    62333 bytes/sec
32k packets:    47815 bytes/sec
```

For the purpose of this document 10 tests have been made and the results for each packet size have been averaged and converted to kbps (i.e. multiplied by 8).

| Packet length [bytes] | Bluetooth [kbps] | 10BaseT Ethernet [kbps] | WLAN 802.11b [kbps] |
|---|---|---|---|
| 1k | 551.640 | 8653.333 | - |
| 2k | 549.576 | 8763.556 | - |
| 4k | 572.237 | 8788.444 | 3561.455 |
| 8k | 571.362 | 8724.444 | - |
| 16k | 482.048 | 8792.889 | - |
| 32k | 405.557 | 8852.444 | - |

**Table 13. Averaged netio results**

For comparison, averaged results collected on 10BaseT Ethernet link are shown in the 3rd column and average results for 4k packet for 11 Mbps WLAN taken from c't article on WLAN solutions [45] in the 4th column. 10BaseT Ethernet link has been free from any other traffic – 2 computers were connected via single cable and no other network applications were running.

As we can see in Table 13, the throughput achieved for Bluetooth link is about 15 times lower than from the Ethernet connection and about 6 times lower comparing to WLAN. What is also worth stressing is the distribution of results. In case of Bluetooth link, highest results are achieved with packet sizes of 4kB and 8 kB. For packets larger and equal to 16 kB the results are significantly lower, reaching 22% less than average with 32KB. For Ethernet link, the largest difference between the average and achieved result is less than 1.3% in the case of 1k packets, which is least efficient.

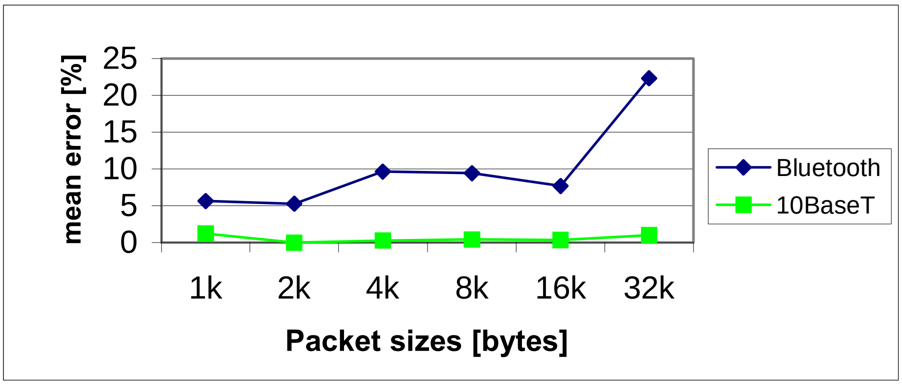


**Figure 25. Mean error for netio results**

Figure 25 shows mean error for the Bluetooth and 10BaseT results. This error is calculated as a ratio between the given value and the average.

## *5.2 Conferencing scenarios*

Audio-video transmission may be used to create a conference with users spread across the globe. It may have both professional applications, like connecting 2 distant company departments or private, like calls to family living on another continent.

As presented in Chapter 4, two products have been selected for evaluation: Microsoft Netmeeting and MBONE tools (vic/rat). Possible configurations are achieved results are presented below.

### 5.2.1 Microsoft Netmeeting

Microsoft Netmeeting 3.01 is a videoconferencing tool supporting H.323 protocol. It offers transmission and reception of audio, video, as well as whiteboard and chat sessions. It is also possible to share applications and send files. From those features only audio and video session shall be used to test the multimedia capabilities.

After creating the test environment (see Chapter 4) we have the following components:

- 2 computers with Bluetooth PCMCIA cards,
- Bluetooth connection between them,
- each computer with installed sound card, speakers and microphone,
- each computer with installed and connected Philips Vesta camera,
- Microsoft Netmeeting running.

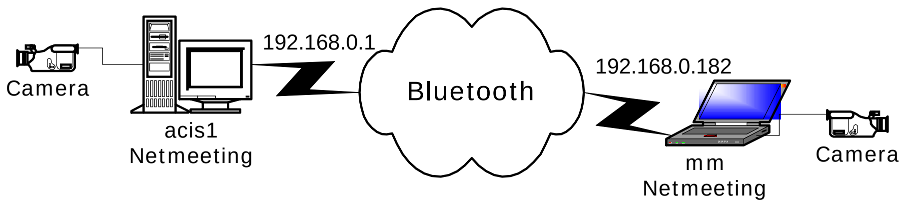


**Figure 26. Hardware/software configuration for Netmeeting**

The hardware and software configuration for this scenario is presented in Figure 26. Access point functionality of acis1 is not needed, since both computers are in the same IP subnet. WinRoute should be disabled on acis1.

To create a session between mm and acis1, follow the instructions in Figure 27.

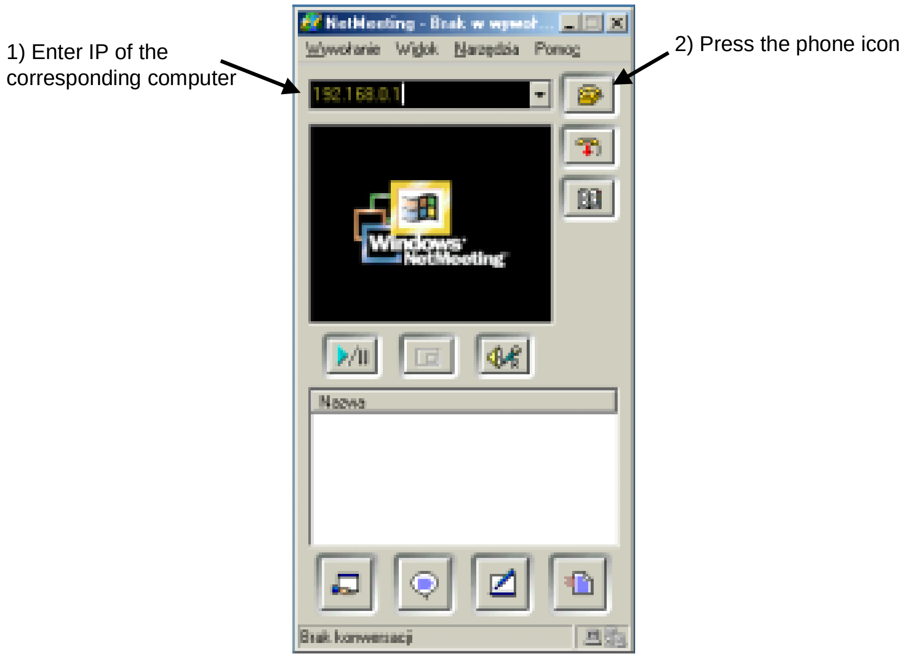


**Figure 27. Creating Netmeeting connection**

For connecting from `mm` to `acis1` IP 192.168.0.1 has to be entered.

Then the connection request has to be accepted on acis1 and the conversation may start. Because Netmeeting uses very efficient codecs: H.263 [24] for video and G.723.1 [43] for audio, bandwidth usage is very limited. During duplex conversation, bandwidth usage of 100-350 kbps has been observed (see Figure 28).

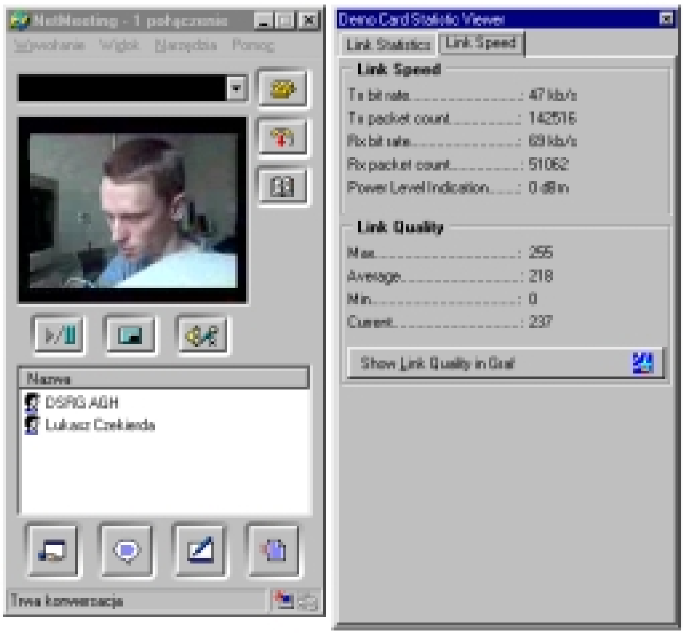


**Figure 28. Netmeeting session with Bluetooth statistics**

Netmeeting doesn't offer any statistics or error rate control, but observations have shown that the quality did not suffer even when the picture contained lots of motion, producing high rate output from the H.263 codec.

To show the bandwidth usage, Etherpeek statistics have been collected. They are shown in Figure 29.

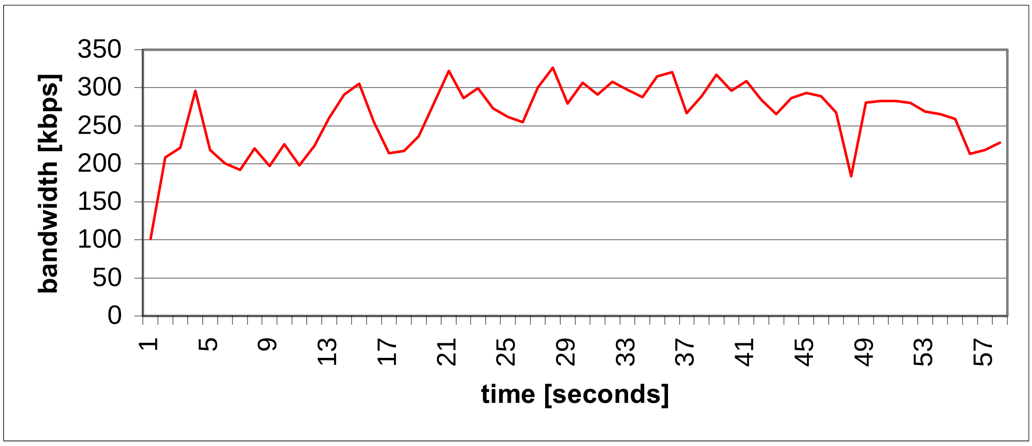


**Figure 29. Bandwidth usage for Netmeeting**

The graph shows that bandwidth usage doesn't exceed 350 kbps, even at the peaks when high motion occurred in front of the camera. This proves that the Bluetooth link should be more than enough for the single user.

On the other hand, when multiple users share the link they might not get the top quality. Then they can try to reduce the size of the captured video or agree to get some errors on video, which may not disturb the conversation at all. Picture in the conference is usually only augmenting the sound and it is the sound that carries most important information. Audio is transmitted with 5.4 kbps by default and its quality shouldn't suffer even at highest link congestion.

### 5.2.2 Vic/rat

Vic and rat are tools widely used in the University environment for conferencing. They support various codecs and work on practically all platforms: Linux, Solaris, FreeBSD, HP/UX and Windows.

For the purpose of this work the following scenario shall be used: 2 PCs with Windows 98, each having Philips Vesta camera, sound card with speakers and microphone. H.261 shall be used to code the video stream in vic and GSM codec in rat for speech.

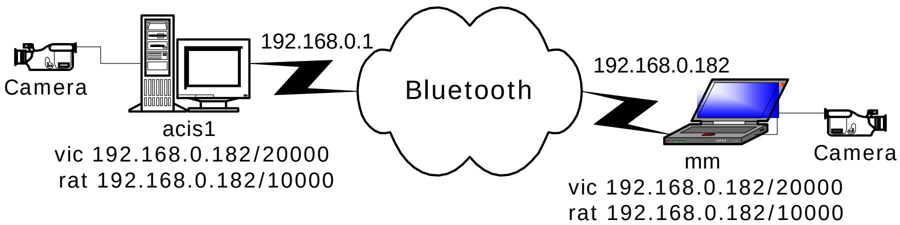


**Figure 30. Hardware/software configuration for vic/rat**

To create a session between mm and acis1, enter:

On acis1:

```
> rat 192.168.0.182/10000
> vic 192.168.0.182/20000
```

On mm:

```
> rat 192.168.0.1/10000
> vic 192.168.0.1/20000
```

Figure 31 presents the link utilisation during Netmeeting conference. The average occupation is about 150 kbps, reaching slightly above 250 kbps at the peak rate. Those values are much smaller than the Bluetooth link capacity, thus leaving space for the other traffic.

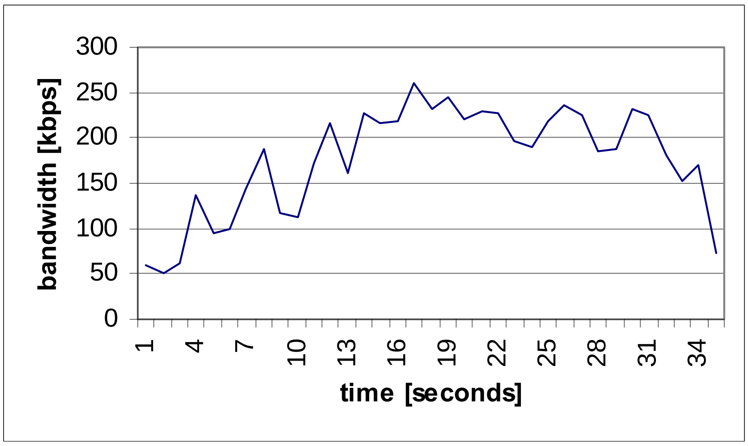


**Figure 31. Bandwidth utilisation with vic/rat**

Even for high frame rate and bitrate the packet loss, taken from RTCP, have been equal to 0, making Bluetooth an extremely reliable connection. This is because Baseband layer supports reliability and retransmits Bluetooth packets if necessary.

The tests have shown that Bluetooth links are more than enough to carry Netmeeting conference. Audio and video quality is the same as achieved on the connections like Ethernet, making Bluetooth an alternative for wired networks.

## *5.3 Video streaming*

Video streaming is the most demanding service tested in this work. Professional video streaming may demand as much bandwidth as 40 Mbps (HDTV MPEG-2 stream), but lower speeds are also possible. Bluetooth offers speeds up to 721 kbps, so the streams tested range from 600 to 725 kbps. Preparation of those streams is presented in Chapter 4.

Prepared streams have been uploaded to the content directories of the available streaming servers: Darwin Streaming Server and RealServer. The following scenarios have been examined (protocols used are indicated in the brackets):

- Darwin Streaming Server with QuickTime Player (RTSP),
- Darwin Streaming Server with Real Player (RTSP),
- RealServer with QuickTime Player (RTSP),
- RealServer with RealPlayer (RTSP & PNM).

### 5.3.1 Real Server + Real Player (PNM and RTSP)

RealServer with RealPlayer from RealNetworks [19] are one of the most popular solutions for streaming multimedia in Internet. By default, they use PNM protocol for transmission, but

RTSP/RTP companion has been implemented in both server and player for compatibility with other solutions (e.g. QuickTime from Apple).

As presented in Chapter 4, several streams have been prepared for examination. For the RealAudio/Video format, 3 are available:

- ads256.rm, 44 kbps audio, 181 kbps video,
- ads384.rm, 64 kbps audio, 286 kbps video,
- ads512.rm, 64 kbps audio, 384 kbps video.

Those streams do not reach the top Bluetooth speed with their bandwidth requirements but when multiple users share the link the available speed decreases. Thus is may be advisable to use clips with smaller bandwidth requirements.

Before streaming, following configuration steps have to be performed:

- Bluetooth connection between acis1 and mm has to be set-up,
- either WinRoute or RTSP Proxy has to run on acis1. Please note that for PNM-based streaming WinRoute has to be enabled,
- corresponding Player configuration on mm,
- RealServer with clips in the Content/ directory has to run on linws1.

The hardware and software configuration for this scenario is shown in Figure 32.

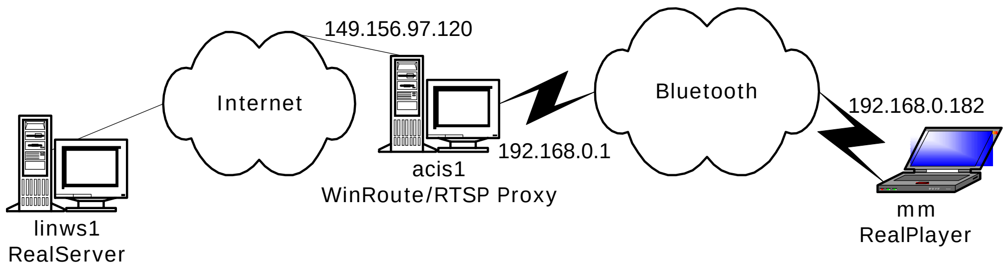


**Figure 32. Hardware/software configuration for RealServer/Player**

Each of the prepared files shall be sent with both available protocol stacks: PNM and RTSP/RTP.

**225 kbps stream (ads256.rm)**

This stream contains 44 kbps audio track and 181 kbps video track, requiring 225 kbps bandwidth. It is the smallest of the RealAudio/Video prepared files.

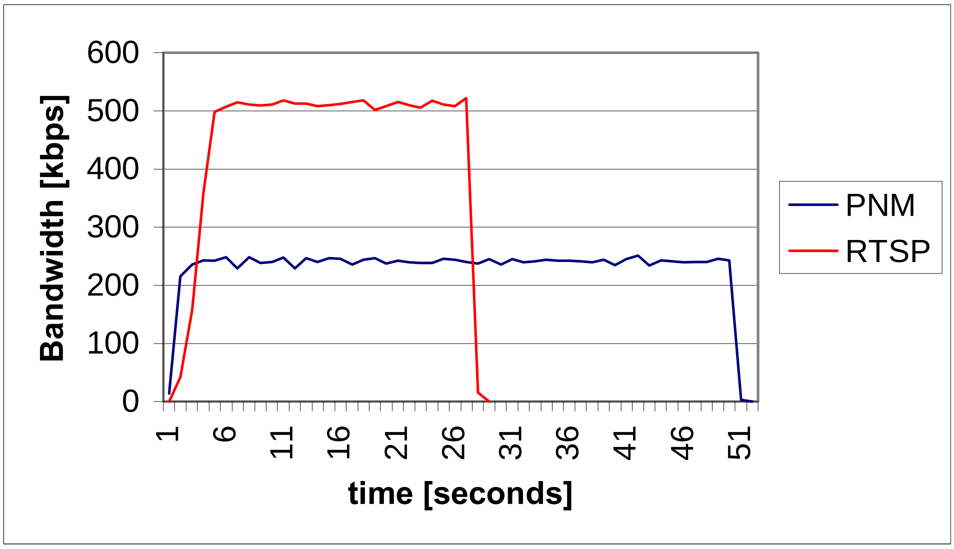


**Figure 33. Streaming 225kbps file**

Figure 33 illustrates bandwidth consumption for streaming with PNM and RTSP protocol. The difference is very significant: PNM tries to keep the bandwidth at the lowest possible level (here around 240 kbps), while RTSP sends the data as fast as possible and lets the application buffer them. When we calculate the number of bytes sent the values for both protocols are very similar. For RTSP the integration gives 1541844 bytes, while for PNM it is 1478130. The difference comes from the inaccuracies in statistical measurements.

The top bandwidth usage for RTSP is limited by the RealPlayer settings, because DSL 512 kbps line has been chosen as the connection link. This type resembles Bluetooth link best. 512 kbps value is passed to the server in the RTSP request and server adapts its output speed accordingly.

**350 kbps stream (ads384.rm)**

350 kbps stream, designed for 384 kbps DSL link contains 64 kbps audio and 286 kbps video. It should take about a half of the fastest Bluetooth link.

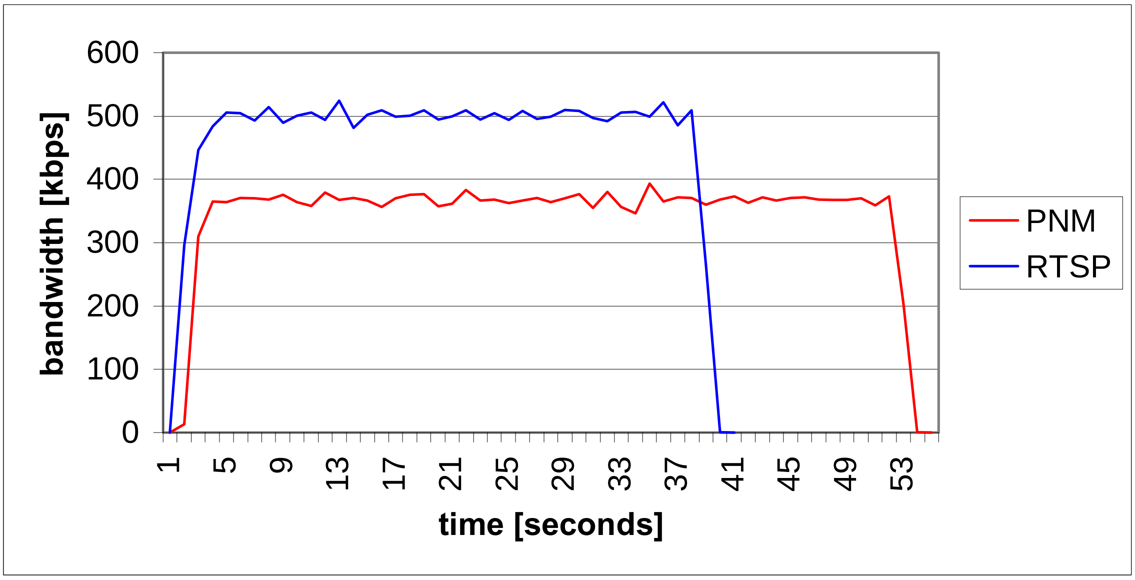


**Figure 34. Streaming 350 kbps file**

Achieved results are similar to the ones for 225 kbps stream. Again, PNM keeps the demand at lowest possible value (this time around 350 kbps), while RTSP uses all the (limited) bandwidth. The packet loss read from RealPlayer statistics is 0, which means that Bluetooth copes with the transmission of this stream perfectly in both cases.

**450 kbps stream (ads512.rm)**

450 kbps stream, which is designed for 512 kbps DSL link is the highest bitrate available for streaming RealAudio/Video files. Nevertheless, it is still less than the bandwidth available in Bluetooth connections.

Figure 35 shows bandwidth utilisation for ads512.rm file streamed with PNM and RTSP protocol. The characteristics are similar to the other situations. It is important to note that the Statistics view (View/Statistics) shows 0% packet loss for both audio and video streams. This proves that Bluetooth link capacity is enough to carry even highest possible RealAudio/Video transmissions with top quality.

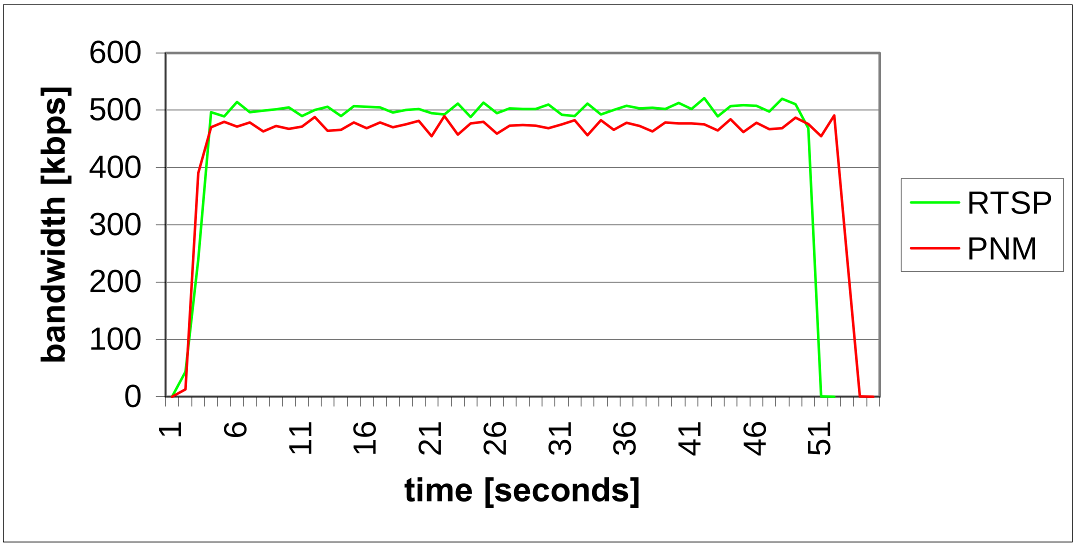


**Figure 35. Streaming 450 kbps clip.**

### 5.3.2 Real Server and QuickTime Player (RTSP)

Real Server and QuickTime Player proved to be the best testing tools in this work. Real Server supports MOV and RealMedia (RM) files, PNM and RTSP streaming, being a universal solution for both QuickTime Player and Real Player.

QuickTime Player supports many compression methods from which MPEG-1 and MPEG-4 (3ivx) have been chosen for evaluation. MPEG-1 creates constant-bitrate streams of the desired bandwidth occupation. This allows testing up to what extent Bluetooth links can carry multimedia transmission: where is the practical limit of the connection capacity.

MPEG-4 codec from 3ivx has been tested because it belongs to the newest and most modern family of codecs. As opposed to MPEG-1, it produces variable-bitrate streams. This allows to test how Bluetooth link is suitable for such transmission.

**Testing environment**

The software and hardware configuration for this scenario is shown in Figure 36.

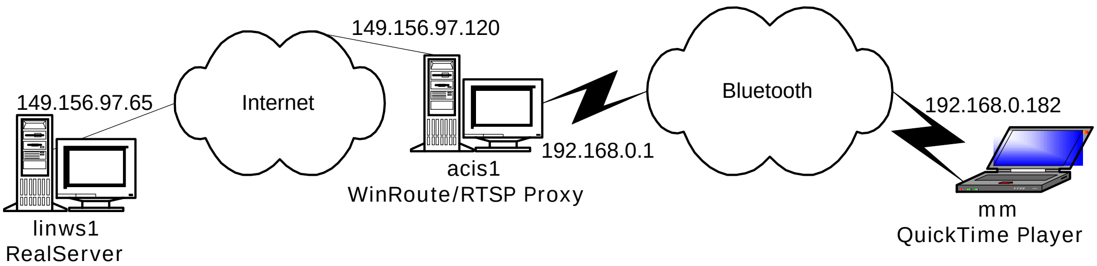


**Figure 36. Software/hardware configuration**

Following steps have to be completed before running the tests:

- Bluetooth/IP connection has to be established between mm and acis1,
- on acis1 either WinRoute or RTSP Proxy has to run,
- if RTSP Proxy is installed on acis1, it has to be enabled in QuickTime Player on mm. Otherwise it has to be disabled.
- RealServer has to run on linws1, with video clips in the Content/ directory.

**Tests performed and results**

To find the limit of error-free multimedia transmission over Bluetooth link MPEG-1 and 3ivx clips have been streamed using RTSP protocol. To start streaming, choose ‘Open URL...’ from the QuickTime Player ‘File’ menu, and enter:

`rtsp://149.156.97.65/<filename.mov>`

in the address field, where `<filename.mov>` is the name of the desired MOV file.

MPEG-1 streams range from 600 kbps to 750 kbps, with 32 kbps devoted to audio in each case. The files are named adsxxx.mov, where xxx is the bitrate in kbps.

For each file, 10 tests have been made to average the results. The results have been collected from QuickTime Player statistics. The statistics view can be enabled by choosing Movie/Get Movie Properties, and then setting option field on the left to ‘Streaming Track’ and right field to ‘Bit rate’. Sample output screen from QuickTime Player with statistical view is shown in Figure 37.


**Figure 37. QuickTime Player with statistics**

The tests have been made on a client machine with a fast processor (Pentium III 700 MHz) to avoid the presence of errors coming from the lack of computing power. It is important to note that QuickTime Player doesn't distinguish between network congestion errors and the lack of computing power – they are indicated on the same graph and named 'packet loss'.

All the streams tested have been transmitted through 10BaseT link first and provided 0% loss. From this fact assumption has been drawn that the client has enough processing power to decode the streams in real-time. Other tests made on a 300 Mhz Pentium machine show that this processor is too slow for MPEG-1 streams of 600 kbps and above, and can't practically cope with 3ivx streams.

It is also worth mentioning that the values shown in the 'Data rate' section haven't been taken into the consideration. This is because they don't provide enough accuracy. Value 'average bit rate' tends to differ by 20-30% for the same stream and same conditions and 'maximum bit rate' usually exceeds all the possible values (i.e. is above 800 kbps).

Nevertheless, 'average packet loss', as well as 'maximum packet loss' seem to give a good indication on the link quality.

Figure 38 shows the average and maximum packet loss, averaged from 10 tests.

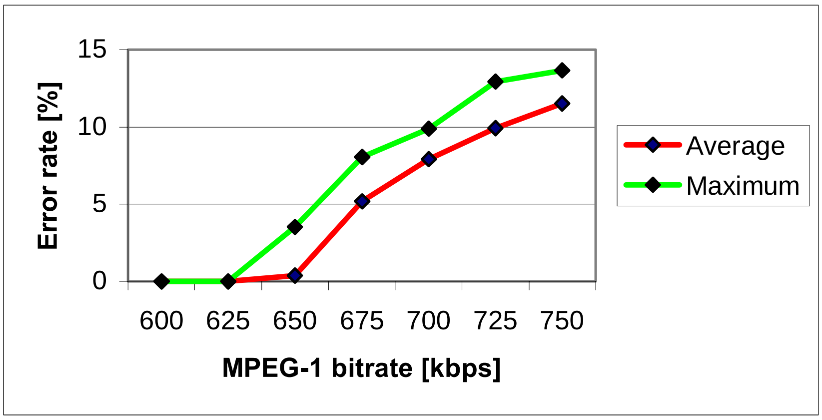


**Figure 38. Error rate with MPEG-1 streaming**

The results show that streams up to 625 kbps may be transmitted without any errors over the Bluetooth link. Small packet loss appears at 650 kbps, and grows with characteristics close to linear as the bitrate increases.

The error at 650 kbps is so small (0.4%) that it is negligible and practically invisible to the viewer. Serious picture quality decrease is observed with packet loss of 5% and above. Large blocks appear on the screen and motion is discontinuous.

The conclusion is that 650 kbps is the absolute limit of the high quality transmission. It is even better to keep the bitrate slightly smaller, since there won't be much initial difference in quality between 625 and 675 kbps clips, but when transmitted over the Bluetooth link 625 kbps stream will look much better because of the 0% packet loss.

Figure 39 shows the average bandwidth requirements for the prepared streams. The average is taken from the middle 40 seconds to cater for irregularities on the borders. Those results show that the actual maximum traffic on the interface is around 680 kbps. Nevertheless, at this value some packets are lost and this results in the low quality of reception.

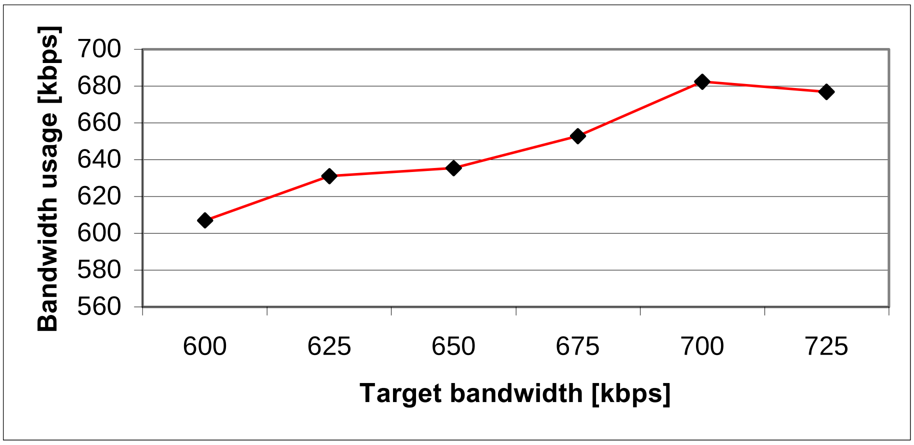


**Figure 39. Average bandwidth usage for MPEG-1 streams**

The situation is significantly different with MPEG-4 streams. The main difference is that 3ivx codec produces stream of variable bandwidth with extreme differences. Even if the average bitrate is far below the limit of 650 kbps, the peak requirements are so high that none of the prepared streams keep the packet loss on the decent level.

It is important to note that bandwidth usage is averaged by Etherpeek to 1 second and peak values may be much higher than indicated on the chart (see Figure 40). By averaging, the characteristics are flattened but Etherpeek doesn't provide accuracy higher than 1 second.

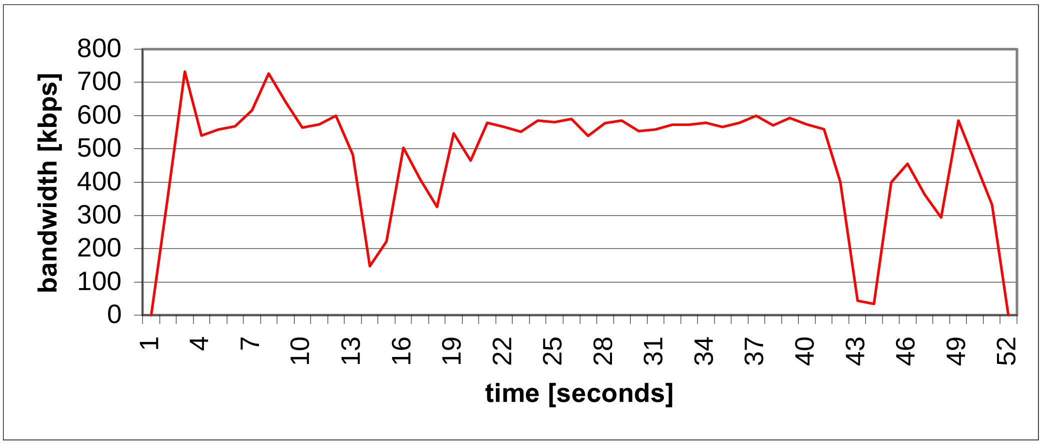


**Figure 40. Bandwidth usage from 3ivx clip (81)**

The characteristics for 3ivx quality 53 and 91 is similar, just shifted down/up.

Average and maximum error rate are interesting for this work, since they provide an overview of the transmission quality. Averaged values for 3ivx streams are shown in Figure 41.

What may seem strange is that the error rate for the stream of higher quality and bandwidth (91) is smaller than for the 81. This probably is caused by the characteristics of the 3ivx codec, which are not publicly accessible.

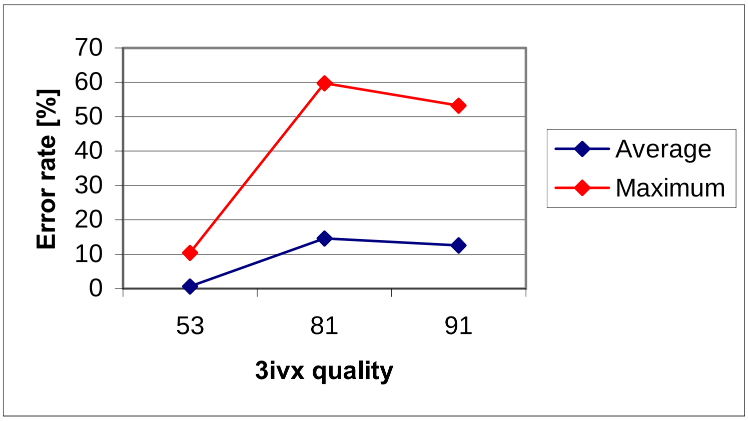


**Figure 41. Average and maximum error rate for 3ivx**

The conclusion from the above tests is that although MPEG-4 codec should provide higher quality at lower bandwidth usage 3ivx doesn't follow this rule. MPEG-1 stream is much more suitable, offering less error rate and thus higher quality.

### 5.3.3 Darwin Streaming Server and QuickTime Player (RTSP)

Darwin Streaming Server and QuickTime Player with RTSP transmission present the same scenario as Real Server with QuickTime Player (see above). This results in the same results.

That’s why only the testing scenario is presented in this section: for results, refer to the previous section.

The hardware and software configuration is presented in Figure 42.

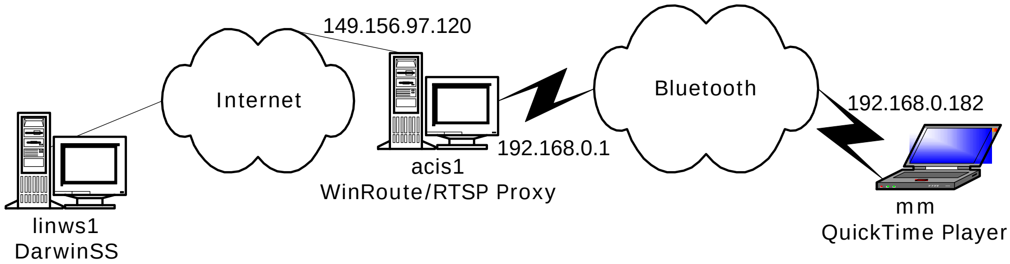


**Figure 42. Hardware and software configuration**

The following configuration steps have to be completed:

- Bluetooth/IP connection set-up between mm and acis1,
- WinRoute/RTSP Proxy running on acis1,
- RTSP Proxy disabled/enabled respectively in QuickTime Player on mm,
- DarwinStreamingServer with video clips running on linws1.

The invocation of streaming is the same as in the previous scenario, i.e. by entering the appropriate RTSP address in the ‘File/Open URL...’ menu.

### 5.3.4 Darwin Streaming Server + Real Player (RTSP)

To test all the possibilities, Real Player with Darwin Streaming Server using RTSP protocol has been tested. Unfortunately, those 2 products don’t work with each other: Darwin Streaming Server doesn’t understand the initial OPTIONS [22] request sent by the player and responds with 400 error code (bad request).

This is because some functionality of RTSP has not been implemented into Darwin project. Nevertheless, this scenario shouldn’t provide any additional results for this work, since changing the server application doesn’t make any difference. Results should be equal to the ones achieved in Section 5.3.1.

## *5.4 Conclusions*

During testing sessions, Bluetooth proved to be a very good carrier of multimedia streams. Giving around 600 kbps for the user data, it allows transmission of the multimedia streams of very high quality. Especially using very efficient MPEG-4 codec, which gives much higher quality than MPEG-1 or MPEG-2 streams at the same bandwidth usage.

On the other hand, MPEG-4 codec puts very high demands for the computing power on the client side. This is especially important for the handheld devices, which will be most often used in conjunction with Bluetooth links. Special DSP for MPEG-4 have to be designed and embedded into small computers like Compaq iPAQ, HP Jornada or Palm series to allow playback of such movies.

Although it may seem that MPEG-1 should belong to the history it is still popular and widely used. MPEG-1 player [46] is already available for Pocket Windows. Bluetooth devices, like PCMCIA cards from Digianswer also work with such devices. This is why this work mostly focuses on MPEG-1 streams transmitted on Bluetooth link. Results presented above may be directly related to handheld device equipped with a Bluetooth card, connected to Internet on the airport or train station.

# 6 Summary

Bluetooth is a very promising technology, which is gaining popularity very quickly. More and more Bluetooth devices appear on the market, ranging from pens to photo cameras, from mobile phones to handheld computers. The properties of this technology, allowing connection between various types of devices with a set of protocols have been described in Chapter 2.

Multimedia transmission has become one of the basic services in Internet, with many conferencing packages and streaming technologies. Most popular products, compression technologies and streaming protocols have been presented in Chapter 3.

After presenting the overview of the technology this thesis focused on testing how Bluetooth links are suitable for multimedia transmission: transmission of audio and video, both real-time and previously recorded. In order to perform the tests, software and hardware architecture has been created (Chapter 4). Several solutions have been proposed for streaming and conferencing, which are the most popular multimedia applications.

The architecture proposed allows evaluation of several types of testing scenarios. Basic scenarios include file or data transfer (http), ping tests and throughput test using netio utility. Those test have shown that Bluetooth allows file download of about 80 kB/s. This value may seem small if compared to Ethernet, but Bluetooth is not designed to replace standard ‘wired’ solutions. It is rather to provide access in public places like airports or train stations, where the speeds should be compared to mobile phone Internet access, ranging from 9.6 (standard GSM) to 115 kbps (fastest GPRS). In comparison to such technologies, Bluetooth offers significantly higher bandwidth and thus faster transmission.

Next, two conferencing products have been tested: MBONE tools vic/rat supporting plain RTP and H.323 package shipped with every Windows: Microsoft Netmeeting. Audio-video transmission performed with those tools occupied up to 50% of the Bluetooth link, even with highest motion in front of the camera. During the transmission no packets have been lost, because the Baseband layer from Bluetooth provides reliable transmission link.

In the end, client/server architecture from RealNetworks and Apple Computers has been evaluated, providing answer to the question how much bandwidth in the Bluetooth link is actually available for the transmission based on UDP. Tests with MPEG-1 streams performed show that the limit is slightly below bandwidth needed to transmit a file prepared for 650 kbps. The average bandwidth requirement for such file is 640 kbps and this is the practical limit.

Additionally MPEG-4 codec from 3ivx company has been tested, but it didn’t provide satisfactory results. This codec uses very variable compression rate, resulting in extremely changing bandwidth requirements, which Bluetooth link wasn’t able to fulfil.

The conclusion from the test can be drawn that Bluetooth link is very suitable for multimedia transmission, as long as appropriate compression technologies are used. The size of the picture and frame rate has to be somehow limited, but still providing very high quality.

## *Appendix A – RTSP Session*

In order to present the concept of video streaming and client-server interaction a sample RTSP session shall be presented and described. Table below contains RTSP packets exchanged between Client (QuickTime Player) and Server (Darwin Streaming Server) before, during and after transmission of movie file *ads600.mov*.

| **Packets Client->Server (C->S) and Server->Client (S->C)** | **Description** |
|---|---|
| **C->S:**<br>DESCRIBE rtsp://linws1/ads600.mov RTSP/1.0<br>CSeq: 1<br>Accept: application/sdp<br>Bandwidth: 2147483647<br>Accept-Language: en-US<br>User-Agent: QTS (qtver=5.0.1;os=Windows 98 ) | Client introduces itself (User-Agent: ...) and asks for description of the stream in the form of SDP message |
| **S->C:**<br>RTSP/1.0 200 OK<br>Server: QTSS/2.0 [v79]-Linux<br>Cseq: 1<br>Content-length: 237<br>Content-Type: application/sdp<br>Content-Base: rtsp://linws1/ads600.mov/<br><br>v=0<br>s=ads600.mov<br>u=http://mm.ics.agh.edu.pl/<br>e=admin@mm.ics.agh.edu.pl<br>c=IN IP4 192.168.0.84<br>a=control:/<br>b=AS:723<br>a=range:npt=0-  50.36 | Request accepted (200 OK).<br>Server introduces itself (QTSS...) and sends stream description:<br><br>s= : session name (filename)<br>c= : connection information<br>a=range: movie length: 50.36 seconds |

| | |
|---|---|
| **C->S:**<br>SETUP rtsp://linws1/ads600.mov/trackID=2 RTSP/1.0<br>CSeq: 2<br>Transport: RTP/AVP;unicast;client_port=6970-6971<br>x-retransmit: our-retransmit<br>x-transport-options: late-tolerance=1.500000<br>User-Agent: QTS (qtver=5.0.1;os=Windows 98 )<br>Accept-Language: en-US | SETUP request: prepare the stream for transmission.<br>Transmit using RTP ports 6970-6971 |
| **S->C:**<br>RTSP/1.0 200 OK<br>Server: QTSS/2.0 [v79]-Linux<br>Cseq: 2<br>Session: 7199428275302761610<br>Transport: RTP/AVP;unicast;client_port=6970-6971;server_port=2000-2001 | Request accepted.<br>Server ports are 2000-2001.<br>Please note that Cseq numbers in request/response pair are the same: here =2. |
| **C->S:**<br>PLAY rtsp://linws1/ads600.mov RTSP/1.0<br>CSeq: 3<br>Range: npt=0.000000-50.366667<br>x-prebuffer: maxtime=2.000000<br>Session: 7199428275302761610<br>User-Agent: QTS (qtver=5.0.1;os=Windows 98 ) | Start the transmission of the whole clip (Range: ...) using negotiated parameters.<br>Client will keep the buffer of 2 seconds. |
| **S->C:**<br>RTSP/1.0 200 OK<br>Server: QTSS/2.0 [v79]-Linux<br>Cseq: 3<br>Session: 7199428275302761610<br>RTP-Info:<br>url=trackID=2;seq=43078;rtptime=822847505 | Request accepted.<br>rtptime is passed to the transport layer |

| | |
|---|---|
| **C->S:**<br>PAUSE rtsp://linws1/ads600.mov RTSP/1.0<br>CSeq: 4<br>Session: 7199428275302761610<br>User-Agent: QTS (qtver=5.0.1;os=Windows 98 ) | Pause the transmission. |
| **S->C:**<br>RTSP/1.0 200 OK<br>Server: QTSS/2.0 [v79]-Linux<br>Cseq: 4<br>Session: 7199428275302761610 | Accepted. |
| **C->S:**<br>PLAY rtsp://linws1/ads600.mov RTSP/1.0<br>CSeq: 5<br>Range: npt=6.768333-50.366667<br>x-prebuffer: maxtime=2.000000<br>Session: 7199428275302761610<br>User-Agent: QTS (qtver=5.0.1;os=Windows 98 ) | Resume the transmission, starting from 6.768333. second till the end. |
| **S->C:**<br>RTSP/1.0 200 OK<br>Server: QTSS/2.0 [v79]-Linux<br>Cseq: 5<br>Session: 7199428275302761610<br>RTP-Info:<br>url=trackID=2;seq=53213;rtptime=2131227648 | Accepted. |
| **C->S:**<br>PAUSE rtsp://linws1/ads600.mov RTSP/1.0<br>CSeq: 6<br>Session: 7199428275302761610<br>User-Agent: QTS (qtver=5.0.1;os=Windows 98 ) | Stop the transmission. Whole clip has been received, but client tells the server to cease the transmission anyway. |

| | |
|---|---|
| **S->C:**<br>`RTSP/1.0 200 OK`<br>`Server: QTSS/2.0 [v79]-Linux`<br>`Cseq: 6`<br>`Session: 7199428275302761610` | Accepted. |
| **C->S:**<br>`TEARDOWN rtsp://linws1/ads600.mov RTSP/1.0`<br>`CSeq: 7`<br>`Session: 7199428275302761610`<br>`User-Agent: QTS (qtver=5.0.1;os=Windows 98 )` | Finish the session – free all resources. |
| **S->C:**<br>`RTSP/1.0 200 OK`<br>`Server: QTSS/2.0 [v79]-Linux`<br>`Cseq: 7`<br>`Session: 7199428275302761610`<br>`Connection: Close` | Accepted. At this point session no. 7199428275302761610 no longer exists. |

# Appendix B – Bluetooth connection setup

This appendix presents the procedure of the connection setup for Bluetooth devices. For simplicity, both master and a slave start from the 'standby' states. Master also knows the slaves device address, either from the inquiry or explicitly. The procedure presented below is designed for 79-hop systems and is presented on pp. 96-109 in [3].

Figure 43 presents the messages exchanged during connection setup. Each of the steps presented is described below.

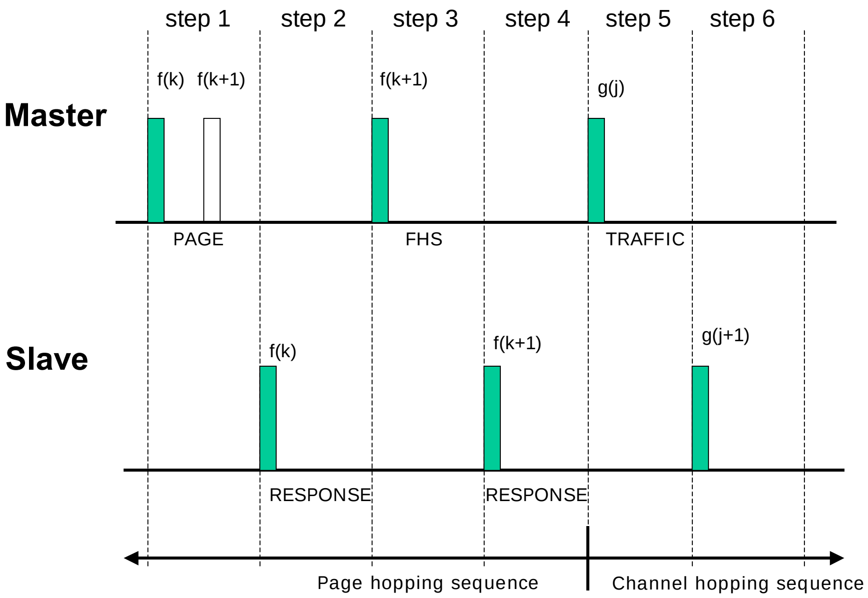


**Figure 43. Exchange of messages during connection setup**

**Step 1**

Step 1 is the most complicated and lasts usually longest, since master has to 'hit' the slave's page scan frequency. This is a very difficult task, since the clocks of the devices are desynchronised and slave 'wakes up' only in certain periods. Master can derive the slave's hopping sequence from its device address, but doesn't know the phase between the clocks. Thus certain actions have to be performed.

First of all, master estimates the clock difference, basing usually on last values (e.g. from the inquiry). Nevertheless, this estimation may be completely wrong. That's why master uses 2 train sequences (see Table 14), each lasting 10ms and containing 16 frequencies when trying to 'hit' the slave.

If we name the estimated frequency f(k), the training sequences shall look as follows:

| | Frequencies related to predicted f(k) | Suitable when phase P |
|---|---|---|
| Train A | f(k-8), f(k-7), ..., f(k), ..., f(k+7) | -8x1.28s < P < +7x1.28s |
| Train B | f(k-16), f(k-15), ..., f(k-9), f(k+8), ..., f(k+15) | -8x1.28s > P > +7x1.28s |

**Table 14. Training frequencies**

Because ID packet with the page message is only 68 bits long, hop rate may be increased twice to 3200 hops/second. This way 16 frequencies may be probed in 16 time slots lasting 10ms, because only half of them (the even-numbered ones) are used for transmission, leaving the rest for the slave to respond.

However, the training sequences have to be repeated many times since the master does not know when the slave enters page scan substate. This value depends on the 'waking-up' interval of the slave and is called $N_{page}$. Table 15 presents $N_{page}$ values referred to the slave scan interval.

| SR mode | Page interval | $N_{page}$ |
|---|---|---|
| R0 | continuous | ≥ 1 |
| R1 | ≤ 1.28s | ≥ 128 |
| R2 | ≤ 2.56s | ≥ 256 |

**Table 15. Scan interval, train repetition and paging modes**

The final scenario is as follows: master repeats Train A Npage times, then tries Train B Npage times, and then again Train A Npage times and so forth, listening for the response on RX slots/frequencies. This continues until the response is received, page timeout is exceeded or the procedure is interrupted by the user. If the response is received, master may enter master response substate and change to Step 3.

**Step 2**

In step 2, the slave responds with ID packet containing its Device Access Code (DAC) exactly 625μs after the start of master transmission, i.e. uses normal hopping rate of 1600 hops/second. Then it awaits the FHS packet from the master.

**Step 3**

In this step, master is in the 'master response' substate and sends the Frequency Hopping Selection (FHS) packet with:

- its real-time Bluetooth clock,
- BD_ADDR,

- BCH parity bits,
- class of device.

**Step 4**

Slave is in the 'slave response' substate and returns a response (ID packet) to acknowledge the reception of the FHS packet. After this, slave switches to the channel (i.e. master's) access code and clock, derived from the FHS contents. Both devices can change to the connection state and proceed to step 5

**Step 5**

Step 5 is the initial phase of the connection. Both devices are in the connection state and use channel hopping sequence, derived from the master's BD_ADDR. Master starts the channel transmission with the POLL packet, which is acknowledged by the slave in step 6.

**Step 6**

In step 6, slave acknowledges the POLL packet with any type of the packet and the connection is established.

Table 16 summarises the steps and messages exchanged during the connection setup. For details, please refer to [3].

| Step | Message | Master state | Slave state | Direction | Hopping Sequence | Access code and clock |
|---|---|---|---|---|---|---|
| 0 | | standby | standby | | | |
| 1 | slave ID | page | page scan | M→S | page | slave |
| 2 | slave ID | | slave response | S→M | page response | slave |
| 3 | FHS | master response | | M→S | page | slave |
| 4 | slave ID | | | S→M | page response | slave |
| 5 | 1st packet master | connection | connection | M→S | channel | master |
| 6 | 1st packet slave | | | S→M | channel | master |

**Table 16. Connection setup steps**

## *References*


[1] Ericsson Bluetooth Website
`http://www.ericsson.com/bluetooth`

[2] The Official Bluetooth SIG Website
`http://www.bluetooth.com`

[3] Bluetooth 1.1 Specifications Book
`http://www.bluetooth.com/developer/specification/Bluetooth_11_Specifications_Book.pdf`

[4] Bluetooth 1.1 Profiles Book
`http://www.bluetooth.com/developer/specification/Bluetooth_11_Profiles_Book.pdf`

[5] Wireless Standards Package (802.11)
`http://shop.ieee.org/store/product.asp?prodno=SH94842`

[6] 3rd Generation benefits (FH-CDMA)
`http://www.pll-dynamics.com/g3.html`

[7] ETSI Technical Specification 07.10
`http://webapp.etsi.org/pda/home.asp?wki_id=9198`

[8] IrOBEX Protocol Specification
`http://www.irda.org/standards/pubs/IrOBEX12.pdf`

[9] W. Simpson, The Point-to-Point Protocol (PPP) - RFC 1548

[10] [Q.931] Recommendation Q.931 (05/98) - ISDN user-network interface layer 3 specification for basic call control
`http://www.itu.int/itudoc/itu-t/rec/q/q500-999/q931.html`

[11] G. McGregor: The PPP Internet Control Protocol (IPCP) - RFC 1332

[12] Kulwinder Atwal, Ron Akers, Transmission of IP Packets over Bluetooth Networks
`http://www.ietf.org/internet-drafts/draft-akers-atwal-btooth-01.txt`

[13] C. Partridge, A Proposed Flow Specification (RFC 1363)

[14] ISO/IEC 11172: Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s (MPEG-1)

[15] Motion Pictures Experts Group Home Page
`http://www.cselt.it/mpeg`

[16] ISO/IEC 13818: Information technology -- Generic coding of moving pictures and associated audio information: Systems (MPEG-2)

[17] Digital Video Broadcasting
http://www.dvb.org

[18] ISO/IEC 14496: Information technology -- Coding of audio-visual objects (MPEG-4)

[19] RealNetworks.com
http://www.realnetworks.com

[20] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: RTP: A Transport Protocol for Real-Time Applications (RFC 1889)

[21] ITU-T Recommendation H.323 (11/00) - Packet-Based Multimedia Communications Systems
http://www.itu.int/itudoc/itu-t/rec/h/h323.html

[22] H. Schulzrinne, A. Rao, R. Lanphier: Real Time Streaming Protocol (RFC 2326)

[23] Recommendation H.261 (03/93) - Video codec for audiovisual services at p x 64 kbit/s
http://www.itu.int/itudoc/itu-t/rec/h/h261.html

[24] Recommendation H.263 (02/98) - Video coding for low bit rate communication
http://www.itu.int/itudoc/itu-t/rec/h/h263.html

[25] Bluetooth Products from Digianswer
http://www.digianswer.com/prod_view.asp?P_ProdID=14

[26] RedHat Linux
http://www.redhat.com

[27] Philips Vesta cameras
http://www.pcstuff.philips.com/cameras

[28] Digianswer Support Page
http://www.digianswer.com/support

[29] Tinysoft Corporation (WinRoute)
http://www.tinysoft.com

[30] MBONE Conferencing Applications
http://ww-mice.cs.ucl.ac.uk/multimedia/software

[31] Tucows
http://www.tucows.com

[32] RealServer 8.01
http://www.realnetworks.com/deliver/index.html

[33] DarwinStreamingServer 2.0
http://www.publicsource.apple.com/projects/streaming/

[34] QuickTime Streaming Server

http://www.apple.com/quicktime/products/qtss

[35] Sourceforge RTSP Proxy project page
http://sourceforge.net/projects/rtsp/

[36] RealPlayer
http://www.real.com/player/index.html

[37] QuickTime Player from Apple Computers
http://www.apple.com/quicktime/download/

[38] 3ivx company: MPEG-4-compliant codec
http://www.3ivx.com/

[39] Etherpeek from WildPackets
http://www.wildpackets.com/products/etherpeek

[40] Bluetooth Software Suite from Digianswer
http://btsws.digianswer.com

[41] AVI2MPG Utility
http://www.vcdhelper.com/avi2mpg.htm

[42] RealProducer from RealNetworks
http://www.realnetworks.com/products/producer/index.html

[43] Microsoft Developer Network
http://msdn.microsoft.com/default.asp

[44] Netio – Network Throughput Measurements
http://www.wintotal.de/softw/netzwerk/netio.htm

[45] Funk-LAN-Systeme, p. 271, c’t 22/2000

[46] Recommendation G.723.1 (03/96) - Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s
http://www.itu.int/itudoc/itu-t/rec/g/g700-799/g723-1.html

[46] MPEG-1 Player for PocketWindows
http://www.mpegtv.com/wince/pockettv/

## *List of figures*

## *List of tables*