THE CHINESE UNIVERSITY OF HONG KONG

Video Streaming over 802.11g Wireless Network

Fong Yu Hang

香港中文大學電子工程學系

DEPARTMENT OF ELECTRONIC ENGINEERING

Video Streaming over 802.11g Wireless Network

Author: Fong Yu Hang

Student I.D.: 01635501

Supervisor: Professor K. L. Wu

Associate Examiner: Professor K. N. Ngan

A project report presented to the
Chinese University of Hong Kong
in partial fulfilment of the
Degree of Bachelor of Engineering

Department of Electronic Engineering
The Chinese University of Hong Kong

April, 2005

Abstract

This thesis aims to develop a product using video streaming technology over 802.11g wireless network, this product uses ARM microcontroller as the core part of the transmitter and receiver. Before implementing the final product, we first simulate it with a prototype. The prototype contains two computers with Linux OS as the transmitter and receiver, a webcam as video source, FFMpeg as the transmitter program, MPlayer as the receiver program, two 802.11g wireless network card as the interface.

From our prototype setup, it is proved that video streaming over 802.11g wireless network is possible. And theoretically, the quality of the video can be improved by using RTSP protocol rather than HTTP protocol.

In our prototype, the quality of the video still needs improvement, and the security issue should also be considered. We will work on modifying FFMpeg to fit the requirement, and learn to use ARM microcontroller as the core part of the final product.

Acknowledgements

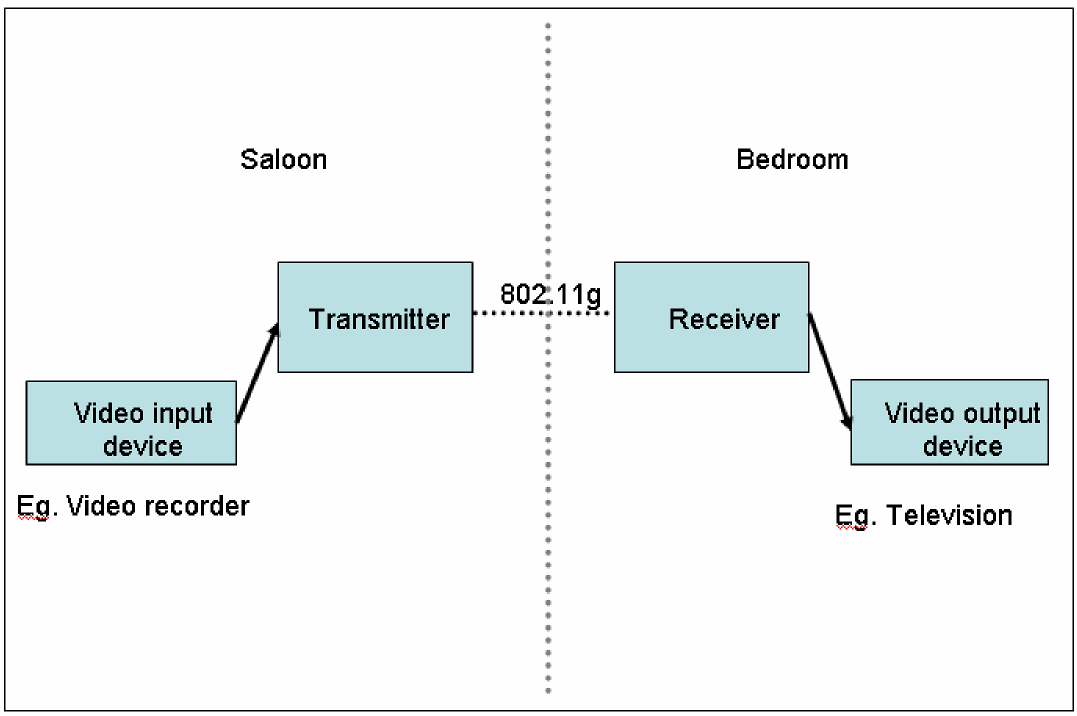
This project would not have been possible without the support of many people. Many thanks to my supervisor, Professor K. L. Wu, who gave me the direction of the project and helped to provide suggestions when I was getting lost. Also thanks to Jacky C. K. Chak, chief scientific officer of AvantWave Limited, who gave me professional consulting on the project and provides me some necessary equipment. Thanks to my partner, Jenny C. M. Ho, her hardworking and solid problem solving skill contributed to the success of the project. And finally, thanks to Electronic Engineering Department, who provided me equipment and financial support.

Contents

Introduction

With more and more researches have done on wireless network, the wireless technology is now becoming a reliable data transmission method. And the transmission speed is getting faster and faster. The IEEE 802.11g standard is now widely used in private network.

My project is to work with Avantwave Limited to design a system for video streaming over 802.11g network. Here is the schematic diagram of our final product.

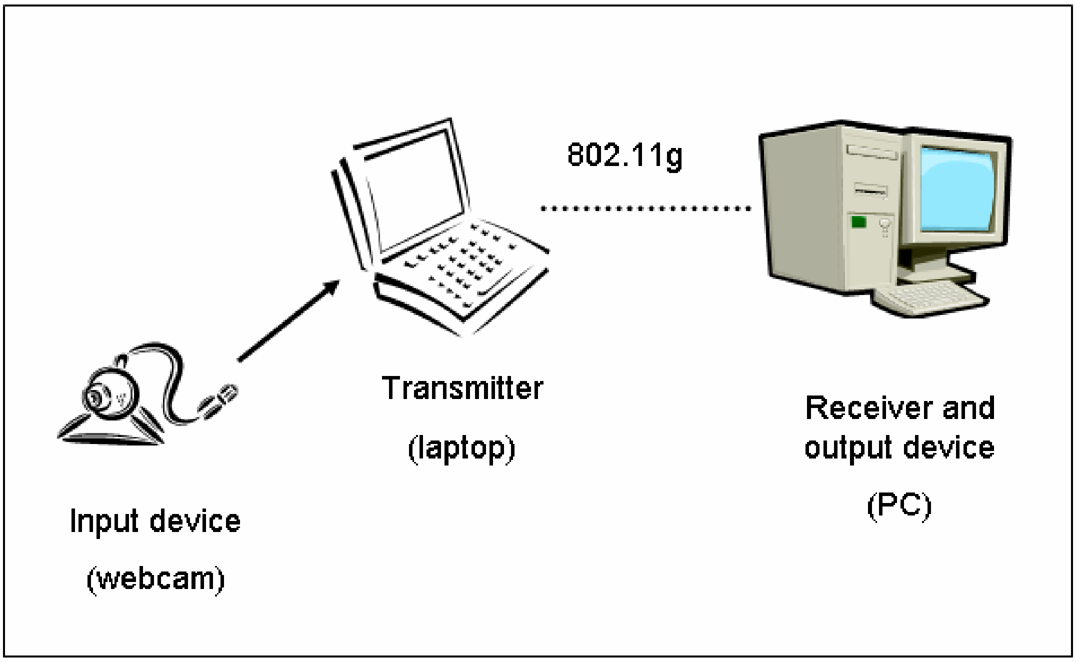


Schematic diagram of the final system

The target market of our product is the domestic market. The selling point of the product is let the user to watch television or movie without connecting a signaling cable. For the traditional system, if you have a video recorder, VCD player or DVD player install at the saloon, you should place your television at the saloon near the player to watch movie. Or if you want to watch movie at bedroom, you should connect a long cable from the television at your bedroom to the player at saloon. This is very ugly and inconvenience. However, with our product, you just need to connect your television with the receiver at the bedroom and you can enjoy the high quality movie. With the wireless technology, no more signaling cable is needed. It is very convenience and can make your home look tidier.

Our task is mainly concentrated on the data transmission part of the system. As I am working with Jenny on this project, she is assigned to work on the transmitter part and I am assigned to work on the receiver part. We are designed to use ARM microcontroller as the core part of the transmitter and receiver, and the operation system of the microcontroller is embedded Linux.

Before working on the final product, we first designed a prototype to simulate the system. The aim of the use of prototype is to familiarize us on the architecture of the system and the platform we are using. Also to test the performance of video streaming over 802.11g wireless network. In the prototype, we used a webcam as the video input device, a laptop as the transmitter, and a PC as the receiver with its monitor as the video output device. Here is the schematic diagram of the prototype setup:



Schematic diagram of the prototype setup

In our prototype, we use two 802.11g network cards as the wireless transmission media. And the operation mode of the cards is Ad-hoc mode.

Theory

In this section, I will explain the detailed theory of each part of our product, which includes the wireless networking standard, the operation system, the video coding standard and video streaming protocol.

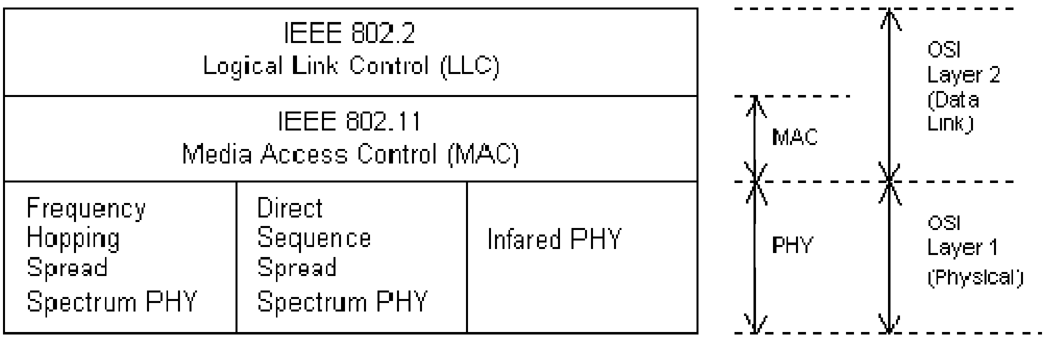
Wireless Networking standard

As the project requires a wireless media for video streaming, it is important to choose a standard with high and stable speed. After some research, I have decided to use IEEE 802.11g as the connection standard.

802.11 standard architecture

802.11 standard was introduced by IEEE in 1997, which was the first wireless local area network (WLAN) in the world. 802.11 standard defines the media access control (MAC) and physical (PHY) layers for a LAN with wireless connectivity. It addresses local area networking where the connected devices communicate over the air to other devices that are within close proximity to each other.[1] the 802.11 standard addresses:

- Functions required for an 802.11 compliant device to operate either in a peer-to-peer fashion or integrated with an existing wired LAN
- Operation of the 802.11 device within possibly overlapping 802.11 wireless LANs and the mobility of this device between multiple wireless LANs
- MAC level access control and data delivery services to allow upper layers of the 802.11 network
- Several physical layer signaling techniques and interfaces
- Privacy and security of user data being transferred over the wireless media



IEEE 802.11 standards mapped to the OSI reference model

With 802.11 standard, a connection between the NIC (Network Interface Card) and the AP (Access Point) can be set up easily.

802.11 standard Security Issue

Security is one of the first concerns of user, as 802.11 standard is a wireless standard, it is easy for someone to capture the data being transmitted between the client and the access point. Base on this security concern, 802.11 committee has provided a security ensure algorithm called WEP (Wired Equivalent Privacy).

WEP algorithm requires a shared key authentication between the NIC and the AP. WEP specifies a shared secret 40 or 64-bit key to encrypt and decrypt the data. Some vendors also include 128 bit keys (know as "WEP2") in their products. With WEP, the receiving station must use the same key for decryption. Each radio NIC and access point, therefore, must be manually configured with the same key.[2]

802.11g Feature

IEEE 802.11g standard was developed base on 802.11b standard. 802.11b standard was created in July, 1999. 802.11b supports bandwidth up to 11 Mbps, comparable to traditional Ethernet.

802.11b uses the same radio signaling frequency - 2.4 GHz - as the original 802.11 standard. Being an unregulated frequency, 802.11b gear can incur interference from microwave ovens, cordless phones, and other appliances using the same 2.4 GHz range. However, by installing 802.11b gear a reasonable distance from other appliances, interference can easily be avoided. Vendors often prefer using unregulated frequencies to lower their production costs.

When 802.11b was developed, IEEE created a second extension to the original

802.11 standard called 802.11a**.** Because 802.11b gained in popularity much faster than did 802.11a, some folks believe that 802.11a was created after 802.11b. In fact, 802.11a was created at the same time. Due to its higher cost, 802.11a fits predominately in the business market, whereas 802.11b better serves the home market.

802.11a supports bandwidth up to 54 Mbps and signals in a regulated 5 GHz range. Compared to 802.11b, this higher frequency limits the range of 802.11a. The higher frequency also means 802.11a signals have more difficulty penetrating walls and other obstructions. Because 802.11a and 802.11b utilize different frequencies, the two technologies are incompatible with each other. Some vendors offer hybrid 802.11a/b network gear, but these products simply implement the two standards side by side. [3]

At November, 2001, the IEEE 802.11g draft standard extends data rates for 2.4-GHz wireless-LAN (WLAN) systems to 54 Mbits/second and provides backward compatibility with existing 802.11b (Wi-Fi) equipment. Both mandatory and optional aspects are included. [4] With its high speed and compatibility with 802.11b standard, 802.11g is now becoming the most popular WLAN standard in the world.

Here is the summary of 802.11g standard:

Issued time:	November. 2001
Max data rate:	54Mbps
Other possible data rate:	6, 9, 12, 18, 24, 36, 48Mbps
Carrier Frequency:	2.4GHz
Modulation scheme:	OFDM (Orthogonal Frequency-Division Multiplexing)
No. of Channels:	11 Channel - USA, Canada
	13 Channel – European
	14 Channel – Japan
Indoor Operation range:	30 - 100 meters
Outdoor Operation range:	200 - 250 meters

Operation System

In the prototype, we use two private computers to simulate the transmitter and the receiver. As the operation system in the final is an embedded Linux system, so we choose to use Linux as the operation system of the two PC such that it is closer to the real environment. And the version of Linux we use is Mandrake Linux 10.0.

Linux OS

Linux is an operating system that evolved from a kernel created by Linus Torvalds when he was a student at the University of Helsinki. Generally, it is obvious to most people what Linux is, however, both for political and practical reasons, it needs to be explained further. To say that Linux is an operating system means that it's meant to be used as an alternative to other operating systems like MS-DOS, the various versions of MS Windows, Mac OS, Solaris and others. Linux is not a program like a word processor and is not a set of programs like an office suite. [5]

The reason why we choose Linux is mainly because of its open source nature. As it is open source, it is no need for us to pay for a license of Linux, this can reduce the cost of our project in comparing with using other OS such as Windows and Unix. Secondly, as the source code is fully released, it is easier for us to have a fine tune on the OS, and it is also easier for us to do troubleshooting. Thirdly, as Linux can be an embed system, which means that we can only install the kernel with some basic functions only. This is very suitable for the low performance computer or device with less memory.

However, there are also some disadvantages of using Linux as the operation system. Firstly, unlike Windows, Linux is a command line based operation system, you should know each function and its parameters clearly, otherwise, you can’t have a efficient work. This is the reason why Linux is difficult for a beginner to use. Secondly, Linux is much poorer than Windows in terms of hardware support. For a wide range of computer hardware with different brands, Windows (2000 and XP) has the drivers already installed. You just need to connect the hardware with your computer and Windows will automatically detect it and make it workable. But for Linux, it is not the case. There is not many hardware officially support Linux. For example, in our case, there is no 802.11g network card developer officially support Linux. As a result, we need to find a third party driver for that network card to be used in Linux. This is time consuming for us to find the driver and install it. And we should do a deep research before buying a particular network card.

Media Coding and Decoding Standard

There are many coding methods developed for video display, with different video quality, compression rate and coding complexity. The common video coding standards are ITU H.261, ITU H.263, MPEG 1, MPEG 2 and MPEG 4.

ITU H.261

H.261 is video coding standard published by the ITU (International Telecom Union) in 1990. It was designed for datarates which are multiples of 64Kbit/s, and is sometimes called p x 64Kbit/s (p is in the range 1-30). These datarates suit ISDN lines, for which this video codec was designed for.

The coding algorithm is a hybrid of inter-picture prediction, transform coding, and motion compensation. The datarate of the coding algorithm was designed to be able to be set to between 40 Kbits/s and 2 Mbits/s. The inter-picture prediction removes temporal redundancy. The transform coding removes the spatial redundancy. Motion vectors are used to help the codec compensate for motion. To remove any further redundancy in the transmitted bitstream, variable length coding is used.

ITU H.263

H.263 is a provisional ITU-T standard, it is due to be published sometime in 1995/1996. It was designed for low bitrate communication, early drafts specified datarates less than 64 Kbits/s, however this limitation has now been removed. It is expected that the standard will be used for a wide range of bitrates, not just low bitrate applications. It is expected that H.263 will replace H.261 in many applications.
The coding algorithm of H.263 is similar to that used by H.261, however with some improvements and changes to improve performance and error recovery. The differences between the H.261 and H.263 coding algorithms are listed below. Half pixel precision is used for motion compensation whereas H.261 used full pixel precision and a loop filter. Some parts of the hierarchical structure of the datastream are now optional, so the codec can be configured for a lower datarate or better error

recovery. There are now four optional negotiable options included to improve performance: Unrestricted Motion Vectors, Syntax-based arithmetic coding, Advance prediction, and forward and backward frame prediction similar to MPEG called P-B frames. [6]

MPEG1

MPEG1 (Moving Picture Experts Group standard 11172) is a standard that sets the requirements for compressing various motion video streams in order to reduce the amount of memory required to store the video and/or the bandwidth required to transmit. This standard specifies guidelines for the basic compression algorithms such as: motion compensation, discrete cosine transform (DCT), quantization, and run-length encoding. [7] The quality of MPEG1 is not so good in comparing with MPEG2 and MPEG4, and it is suitable for playing movie with VCD quality.

MPEG2

The MPEG2 standards include significant improvements over MPEG1. Like the predecessor, MPEG2 standards have several parts, nine in all. The most important of these are video, audio, and systems. The significant improvements in MPEG2 video over MPEG1 video are: support for interlaced and progressive coding, 4:2:2

and 4:4:4 chrominance modes, higher frame sizes, scalability, and many additional prediction modes. Though not in the original goals, MPEG2 video was shown to be of sufficiently high quality to satisfy the HDTV requirements. The original MPEG3 effort of specifying HDTV technologies was hence abandoned.

While MPEG1 is optimized for storage-based applications, MPEG2 is more generic and intended for a variety of applications. The part 1 of MPEG2 standards, the MPEG2 System specification, defines program streams suitable for application over a reliable medium, and transport streams, with fixed length packets, suitable for networked delivery.

Because of its flexibility and support for a wide range of applications, MPEG2 has been highly successful. MPEG2 has been universally adopted for high quality audio-visual applications including digital broadcast TV, DVD, HDTV, and digital cinema. [8]

MPEG4

MPEG-4 is an ISO/IEC standard being developed by MPEG (Moving Picture Experts Group), the committee which also developed the Emmy Award winning standards known as MPEG-1 and MPEG-2. These standards made interactive video on CD-ROM and Digital Television possible. MPEG-4 will be the result of another international effort involving hundreds of researchers and engineers from all over the world. MPEG-4, whose formal ISO/IEC designation will be ISO/IEC 14496, is to be released in November 1998 and will be an International Standard in January 1999.

MPEG-4 is building on the proven success of three fields: digital television, interactive graphics applications (synthetic content) and the World Wide Web (distribution of and access to content) and will provide the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the three fields. [9]

Here listed the performance of MPEG1, MPEG2 and MPEG4, and also the desire coding method for different video format.

Format	VCD	SVCD	DVD	HDDVD HDTV (WMVHD)	DivX XviD WMV
Resolution NTSC/PAL	352x240 352x288	480x480 480x576	720x480[2] 720x576[2]	1440x1080[2] 1280x720[2]	640x480[2]
Video Compression	MPEG1	MPEG2	MPEG2, MPEG1	MPEG2 (WMV-MPEG4)	MPEG4
Video bitrate	1150kbps	~2000kbps	~5000kbps	~20Mbps (~8Mbps)	~1000kbps
Audio Compression	MP1	MP1	MP1, MP2, AC3, DTS, PCM	MP1, MP2, AC3, DTS, PCM	MP3, WMA, OGG, AAC, AC3
Audio bitrate	224kbps	~224kbps	~448kbps	~448kbps	~128kbps
Size/min	10 MB/min	10-20 MB/min	30-70 MB/min	~150MB/min (~60MB/min)	4-10 MB/min
Min/74min CD	74min	35-60min	10-20min	~4min (~10min)	60-180min
Hours/DVD	N/A	N/A	1-2hrs (2-5hrs[a])	~30min (~1hrs)	7-18hrs
Hours/ DualLayerDVD	N/A	N/A	2-4hrs (5-9hrs[a])	~55min (~2hrs)	13-30hrs
DVD Player Compatibility	Great	Good	Excellent	None	Few

Computer CPU Usage	Low	High	Very High	Super high	Very High
Quality	Good	Great*	Excellent*	Superb*	Great*

Performance of MPEG1, MPEG2 and MPEG4 (source: www.VideoHelp.com)

The advantage of MPEG4 is that it has a higher compression rate than MPEG2, but the quality of the video can be remained. As our product use wireless network for data transmission, bandwidth is limited to 54Mbps, and it is decided for transmit video with DVD quality, MPEG4 is the most suitable video coding standard for our project.

Video Streaming Protocol

There are various kind of protocols developed for video streaming over the local area network. And the commonly used protocols are HTTP, RTSP and RTP.

HTTP

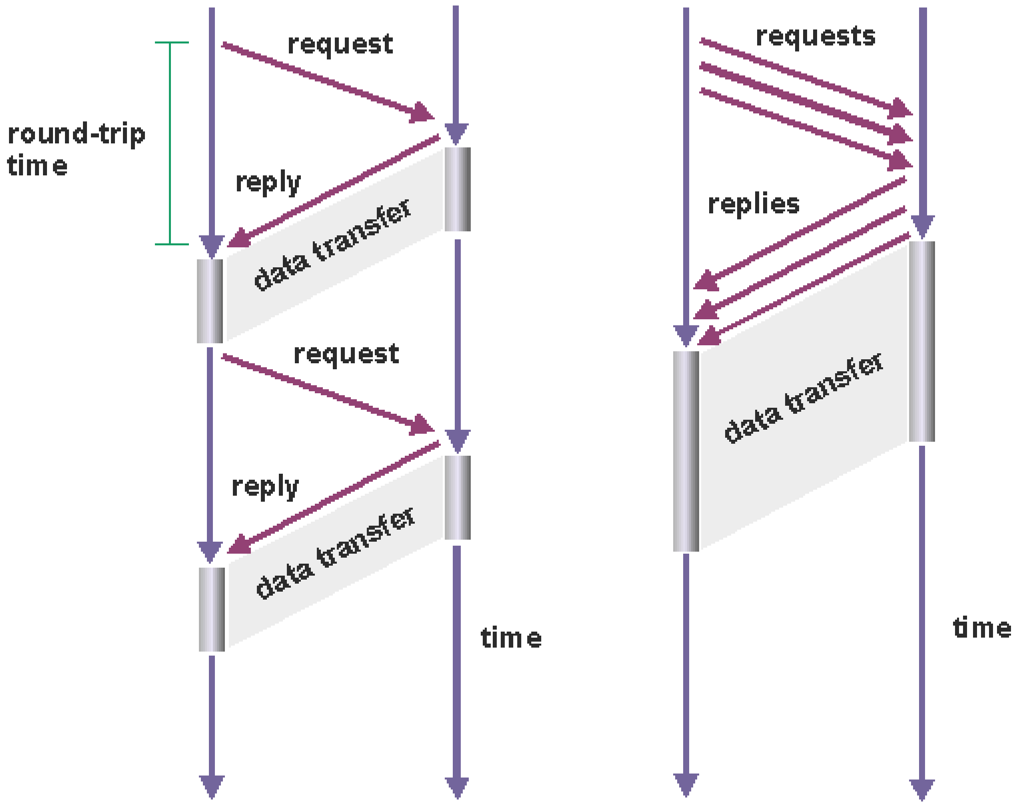
HTTP (Hypertext Transfer Protocol) is a protocol used mainly to access data on the World Wide Web. The protocol transfers data in the form of plain text, hypertext, audio, video, and so on. However, it is called the Hypertext Transfer Protocol because its efficiency allows its use in a hypertext environment where there are rapid jumps from one document to another.

HTTP functions like a combination of FTP (File Transfer Protocol) and SMTP

(Simple Message Transfer Protocol). It is similar to FTP because it transfers files and uses the services of TCP (Transmission Control Protocol). However, it is much simpler than FTP because it uses only one TCP connection (well-known port 80). There is no separate control connection; only data are transferred between the client and the server.

HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like (Multipurpose Internet Mail Extension) headers. However, HTTP differs from SMTP in the way the messages are sent from the client to the server and from the server to the client. Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client. SMTP messages are stored and forwarded, but the HTTP messages are delivered immediately [10].

Here is the request and respond action between the client and the server once a HTTP section is created.



Request and respond action of HTTP
(Source: http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol2/pcg1)

RTSP

The Real-Time Streaming Protocol (RTSP) establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not typically deliver the continuous streams itself, although interleaving of the continuous media stream with the control stream is possible. In other words, RTSP acts as a "network remote control" for multimedia servers.

The set of streams to be controlled is defined by a presentation description. This memorandum does not define a format for a presentation description.

There is no notion of an RTSP connection; instead, a server maintains a session labeled by an identifier. An RTSP session is in no way tied to a transport-level connection such as a TCP connection. During an RTSP session, an RTSP client may open and close many reliable transport connections to the server to issue RTSP requests. Alternatively, it may use a connectionless transport protocol such as UDP.

RTSP differs in a number of important aspects from HTTP:

- RTSP introduces a number of new methods and has a different protocol identifier.
- An RTSP server needs to maintain state by default in almost all cases, as opposed to the stateless nature of HTTP.
- Both an RTSP server and client can issue requests.
- Data is carried out-of-band by a different protocol. RTSP is defined to use ISO 10646 (UTF-8) rather than ISO 8859-1, consistent with current HTML internationalization efforts.
- The Request-URI always contains the absolute URI. Because of backward compatibility with a historical blunder, HTTP/1.1 carries only the absolute path in the request and puts the host name in a separate header field [11].

As there is more support for HTTP over Linux OS on the internet, we have chosen to use HTTP as the video streaming protocol of our project rather than RTSP. However, HTTP is based on TCP while RTSP is based on UDP. TCP keeps trying to resend the lost packet before sending anything further, causing greater delays and breakups in the audio broadcast [12].

Experiment Results

In this section, I will explain the choice of wireless communication protocol, operation system, network card, the networking setup and the programs used in our prototype.

Choice of wireless communication protocol

There are so many wireless communication protocols developed, such as infrared, 802,11a/b/g, Bluetooth or WiMAX. In our project, we have chosen to use IEEE 802.11g as our wireless communication protocol, it is mainly because of its high speed, high compatibility and high operation range.

First of all, IEEE has defined the maximum bandwidth of 802.11g is 54Mbps, which is much higher than 802.11b (11Mbps), infrared (115.2Kbps) and Bluetooth (721Kbps), but the same as 802.11a. For our project, the system is required to stream video with high quality, such as DVD or even HDTV quality, so 54Mbps is high enough for such kind of video quality.

Secondly, the frequency used in 802.11g is 2.4 GHz, which is the same as 802.11b and Bluetooth, which means that our system has the possibility to compatible with other devices using 802.11b or Bluetooth as the wireless communication protocol. And actually, some researches are doing on making these three protocols to communicate. However, the operation frequency of 802.11a is 5 GHz, which is difficult to communicate with protocols using other operation frequency. This is the reason why we prefer to use 802.11g rather than 802.11a.

Thirdly, the operation range of 802.11g is about 30 – 100 meters at indoor environment, and up to 250 meters at outdoor environment. As the target user of our system is the domestic users in Hong Kong, 30 meters operation range is high enough for the housing environment in Hong Kong.

Here is the summary of comparison of different wireless communication protocol:

802.11 Wireless LAN Standards Comparison				
	802.11a	802.11b	802.11g	Bluetooth
Data Rate (Mbps)	54	11	54	721Kbps 56Kbps
Operating Frequency (GHz)	5	2.4	2.4	2.4
Typical power output (mw)	40-800	100	100	100
Compatibility	Not compatible with 802.11b or 802.11g	Not compatible with 802.11a or 802.11g	Compatible with 802.11b	Not compatible with 802.11a/b.
Range	150feet	150feet	150feet	30feet
Interference risk	Low	High	High	High
Price	Expensive	Cheap	Moderate	Moderate
Hot-spot access	Poor	Good	Good	Poor

Choice of operation system

In our project, we have chosen to use Linux as our operation system rather than Windows, Unix or other operation systems, it is mainly because of its free, open source nature.

Linux is created at 1991 base on the Unix-type command line interface. It is a free OS under the protection of GNU. GNU is the first software-sharing community in the world. The mission of GNU is to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software users [13].

Under the protection of GUN, the users of GNU software have the following right:

- You have the freedom to run the program, for any purpose.
- You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
- You have the freedom to redistribute copies, either gratis or for a fee.
- You have the freedom to distribute modified versions of the program, so that the community can benefit from your improvements.

With the above statements, we can download and use Linux for free, no need to pay for the licensing fee, which can reduce the budget of our project. And with its open source nature, the source code of Linux is fully released, which can let us fine tuning the OS easily. It is also helpful to us when doing troubleshooting, by inserting some debugging code inside the original source code.

Secondly, Linux is an embedded system. Unlike Microsoft Windows, we don’t need to fully install Linux before running it. We only need to install the Linux kernel with some necessary functions, which is Suitable for computer with low performance or system with less memory. And for our project, we are designed to implement the system in ARM microprocessor, the embedded nature of Linux can reduce the hardware cost of the product.

Furthermore, there are so many experienced Linux users and related communities, it is easy for us to find solutions or ask for help if we encountered some problems during the project.

Choice of network card

In the setup of the prototype, we have used two 802.11g network card as the interface of wireless connection. For the transmitter, we used 3Com

OfficeConnect Wireless 802.11g PCMCIA Card as the interface. As it is connected to a laptop, PCMCIA interface is the most suitable choice of interface.

For the receiver, we used SMC 54Mbps Wireless g PCI card (SMC2802W) as the networking interface. As it is connected to a PC, PCI interface card is the most suitable choice.

This two network cards are not official support Linux OS, they are expected to be used under Windows environment. As a result, there is no driver for Linux contained in their packages and there is also no installation guide on using the cards on Linux.

The two cards have a similar point: they both used prismGT as their chipset. PrismGT is one of the common chipsets used in 802.11g network card. A noncommercial group of people called Prism54 is working on using 802.11g on Linux. They have successfully developed a Linux firmware for 802.11g network card with chipset prismGT. According to www.whatis.com,

> "Firmware is programming that is inserted into programmable read-only memory (programmable ROM), thus becoming a permanent part of a computing device. Firmware is created and tested like software (using microcode simulation). When ready, it can be distributed like other software and, using a special user interface, installed in the programmable read-only memory by the user. Firmware is sometimes distributed for printers, modems, and other computer devices."

The firmware developed is called isl3890, which is a freeware. After placed it in /usr/lib/hotplug/firmware/ under Linux filesystem, the network cards work properly.

Network setup

For the network setup, an IP should be assigned to each network card such that they can use this IP to communicate. As it is a Local Area Network, the IP used should be a private IP. The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets (local networks):

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

In our setting, we used 192.168.0.1 as the IP of the transmitter, and 192.168.0.2 as the IP of the receiver. For two network cards to be able to communicate with each other, other than IP, their subnet mask and gateway should also be defined.

A subnet mask is the mask used to define which subnet the IP address belongs to. For two computers in the same LAN, they should have the same subnet mask. And the subnet mask used in our prototype is 255.255.255.0.

Gateway is a network point that acts as an entrance to another network, such as the server through which people on a company's local area network access the internet. Often a gateway conceals the IP address of the specific user sending out information, and outsiders can only see the IP address of the gateway itself. In our case, the gateway is the other IP address. For the transmitter, its gateway is 192.168.1.2. And for the receiver, its gateway is 192.168.1.1.

The command line setting of IP address, subnet mask and gateway in Linux is as follow:

For transmitter:

```
#ifconfig eth0 addr 192.168.0.1
#ifconfig eth0 mask 255.255.255.0
#route add 192.168.0.1 gw 192.168.0.2
```

For receiver:

```
#ifconfig eth0 addr 192.168.0.2
#ifconfig eth0 mask 255.255.255.0
#route add 192.168.0.2 gw 192.168.0.1
```

After the above setting, the two network cards can communicate with each other. It can be proved by the command “ping”. The command “ping” sends a small packet through the network to a particular IP address. The computer that sent the packet then waits (or 'listens') for a return packet. If the connections are good and the target computer is up, a good return packet will be received. After typing the following command at the transmitter:

#ping 192.168.1.2

The following result can be seen:

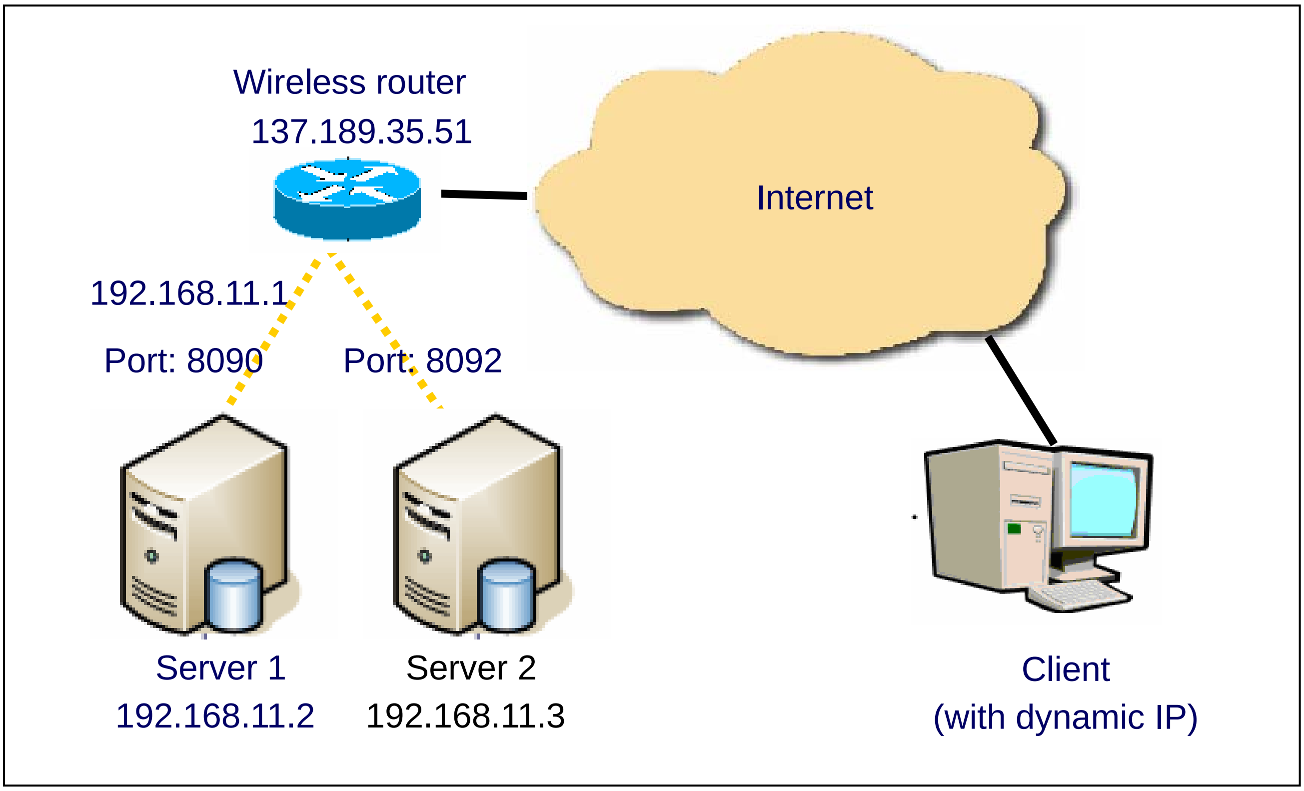
```
PING 192.168.0.2: 56 data bytes
64 bytes from 192.168.0.2: icmp_seq=0 time=16 ms
64 bytes from 192.168.0.2: icmp_seq=1 time=9 ms
64 bytes from 192.168.0.2: icmp_seq=2 time=9 ms
64 bytes from 192.168.0.2: icmp_seq=3 time=8 ms
64 bytes from 192.168.0.2: icmp_seq=4 time=8 ms
^C
----192.168.0.2 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 8/10/16
```

From the above result, we can prove that the network setting is succeeded.

Router setup

To access the video streaming server through the internet, we need to connect our server to a wireless broadband router. The router used in this project is Buffalo WBR-G54.

The function of the wireless router is to connect the server to the internet by providing a private IP to the server, with the router, we can setup more than one server in our system. Here is the network diagram of the setup:



There are three parts in the router that are needed to be set:

1. Setup the static IP, gateway and DNS provided by the department at the WAN port of the router, such that any networked computer can access thte router through the internet.

2. Configure DHCP server to the LAN ports of the router, then the router will assign a private IP to each network device (including wireless devices) connected to it. In this case, IP 192.168.11.2 is assigned to server 1, 192.168.11.3 is assigned to server 2.

3. Configure address translation, that is to map the different LAN IP with a port no. for communication between client and different servers. In this

case, we mapped 192.168.11.2 with port 8090, and mapped 192.168.11.3 with port 8092.

Programs used in the prototype

For the transmitter, we need a programme to encode the video and stream it over the internet. For the receiver, we need a programme to receive the stream, decode the video and display it. There are many open source software developed for Linux, we can use them in our prototype to test the performance of video streaming over 802.11g wireless network. We have used FFMpeg in the transmitter and MPlayer in the receiver.

FFMpeg

FFMpeg is an audio/video conversion tool. It includes libavcodec, the leading open source codec library. An experimental streaming server for live broadcasts is also included. It is a project hosted by SourceForge.org. The latest released version is 0.4.9-pre1.

FFMpeg use HTTP as the video streaming protocol, and the RTSP part is under development. It supports a various kinds of coding standard and file format, such as MPEG1, MPEG2, MPEG4, MP3, AVI, ASF, Real Media, etc. This is very suitable for our prototype. One of the advantages of using FFMpeg is that it allows us to have fine tuning on the video encoding and streaming part. For example, we can tune the frame per second, the video size, the transmission rate, the video quality, etc. The parameters can be adjusted by modifying the doc/ffserver.conf file. By using FFMpeg, we can achieve video streaming using the following commands:

```
#./ffserver -f doc/ffserver.conf &
#./ffmpeg http://localhost:8090/feed1.ffm
```

MPlayer

MPlayer is a movie player for Linux (runs on many other Unices, and non-x86 CPUs, see the documentation). It plays most MPEG, VOB, AVI, Ogg/OGM, VIVO, ASF/WMA/WMV, QT/MOV/MP4, FLI, RM, NuppelVideo, YUV4MPEG, FILM, RoQ, PVA files. MPlayer is the wide range of supported output drivers. It works with X11, Xv, DGA, OpenGL, SVGAlib, fbdev, AAlib, DirectFB, but you can use GGI, SDL (and this way all their drivers), VESA (on every VESA compatible card, even without X11!) and some low level card-specific drivers. The latest released version of MPlayer is v1.0pre5.

MPlayer support both the HTTP and RTSP as video streaming protocol. It can let the users tuning the buffer size, buffer time, maximum bandwidth, vide quality, etc.

And it work well with FFMpeg. As a result, it is suitable to our project. The command of using MPlayer is:

```
#./MPlayer http://192.168.0.1:8090/test.mpg
```

Cost summary

In our project, we have used several equipment to setup the prototype. Here I will list all the equipment we have used and we will use, their cost and whose they are provided by.

PC

Model: Dell OptiPlex GX1 450+

CPU: Pentium III 450MHz

Memory: 256MB

Provided by: Electronic Engineering Department

Laptop

Model: Samsung NQ25

Provided by: Jenny Ho Chit Ming

Webcam

Model: Maxell USB PC Web Camera MPCC-1

Provided by: AvantWave Limited

Network card for transmitter

Model: 3Com OfficeConnect Wireless 802.11g PCMCIA Card

Cost: $390

Network card for receiver

Model: SMC 54Mbps Wireless g PCI card (SMC2802W)

Cost: $410

3G Phone

Model: Motorola E1000

Provided by: AvantWave Limited

Thanks to AvantWave Limited, Electronic Engineering Department and Jenny, we only need to buy the two network cards for this project. The budget of this project is around $800.

Discussions and Conclusions

In this section, I will discuss the problem we faced during the setting and their solutions, and the possibility of developing other applications based on this system.

After the setup of the prototype, our system is functionally working. However, the video quality is very poor. Jittering effect is very serious. In this section, I will discuss the factors accounting for the poor video quality and the methods to improve it.

Here is the data obtained for the bandwidth, CPU loading, delay time and video quality for different video encoding standard:

Size Quality	Bandwidth usage	CPU usage	Delay time	Video output quality
160x120 low	630Kbps	7.5%	3-5s	poor
160x120 standard	680Kbps	7.7%	3-5s	poor
320x240 low	1.10Mbps	7.6%	3-5s	acceptable
320x240 standard	1.23Mbps	7.8%	3-5s	acceptable

640x480 low	2.14Mbps	53%	3-5s	Jittering occur
640x480 standard	2.46Mbps	56%	3-5s	Jittering occur

In the above table, we can see that the bandwidth usage of the video quality 640x480 is about 2.5Mbps, which is enough for our system to stream video with higher quality. The CPU usage is depends on the video size, and it is quite high for video with frame size up to 640x480. At this frame size, jittering problem occur. We can also see that the delay time is about 3 – 5 seconds, which is acceptable for video stream. And it does not depend on the the video quality, so we can conclude that the delay is due to video encoding, signal transmission and video decoding.

Jittering Problem

For the jittering problem, we find that after the video is played for some time (around 10 seconds), the smoothness of the video decrease and the video become discrete frames of picture, and the video speed is faster than normal. We first concluded that the jittering problem is due to not enough buffering in the receiver, so we tried to increase the buffering size. However, the problem was still here. We observed that the amount of buffer was decreasing once the video started to play, and the jittering problem occurs when the amount of buffer decreased to zero.

We then concentrate our work on the transmitter. After tracing the source code of FFMpeg, we find that after FFMpeg is run, it get the image from the file "feed1.ffm", at which the image data is provided by the webcam. Then it encodes the image to the appropriate file format according to the file "doc/ffserver.conf". By default, FFMpeg encodes the file to both the mpg, avi and asf format, which use a lot of computer resources. After edited the config file to encode mpg format only, the problem solved. No jittering problem is found, the buffer size remains in a stable level, and the speed of the movie becomes normal.

Video quality degrade as frame size increase

By default, FFMpeg encodes the video with frame size of 160 X 120. However, this small frame size is not acceptable if we want to use it to play movie. We tried to increase the frame size to 800 X 600 (normal video frame size), the quality of the video seriously degraded. If there is a fast moving object captured by the webcam, the display of MPlayer will become unclear, and the video will become less smooth.

After searching some related resources from the internet, we found that there are two reasons accounting for this problem.

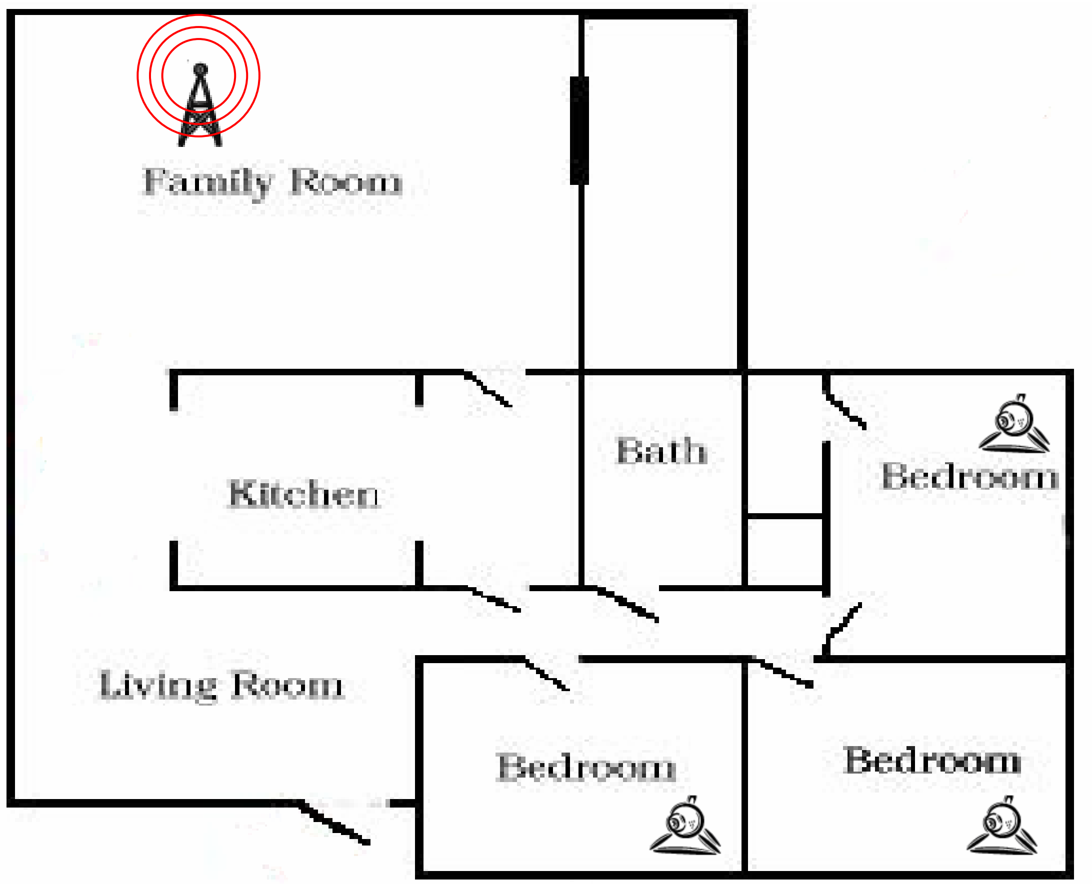
Firstly, according to some expert users and developers of FFMpeg, FFMpeg still has some problems on resources handling. FFMpeg can only use up to 15% of the computer resources, if the job requires heavy loading or the performance of the computer is poor, video quality degrade problem will occur. The solution to this problem is to modify FFMpeg on resources handling, or use more powerful hardware.

Secondly, FFMpeg uses HTTP as the video streaming protocol. Since HTTP is based on TCP, which use some bandwidth on flow control and error-recovery. Once a packet is lost during transmission, the transmitter should resend that particular packet to ensure there is no packet lost. This will increase the usage of bandwidth and make the video quality degrade as it needs to wait for the transmitter to resend the packet. However, for video streaming, a minor packet lost problem will not lead to a noticeable video problem, so it is not necessary for the transmitter to resend the packet. As a result, we have planned to modify FFMpeg to support RTSP as video streaming protocol. RTSP is a better protocol on video streaming because it is based on UDP, where UDP headers contain fewer bytes and consume less network overhead than TCP.

Other Applications

Other than entertainment purpose, we have also thought of two possible applications of this system.

First of all, with a proper hardware design, this system can be used as a home supervision device. Here is the proposed installation diagram of this supervision system.



The design includes two parts, camera and receiver. The FFserver and a 802.11g wireless circuit is embedded in the camera, and a 802.11g wireless circuit with routing function is embedded in the receiver. The receiver is connected to a broadband line for remote access. One a user access the router, a signal will send to the camera, and the camera will start to stream video to the router, and the video is transmitted to the remote computer through the internet.

The user only need to install this system with several camera (depends on the user's need), a broadband service provided by the ISP (internet service provider) and an IP or a domain, then he can supervise different corners of his house using a remote computer while he is not at home. He can use our system to see if there is thief get inside your house, or he can use it to supervise how the domestic helper treats his baby.

As in the setup of the prototype, we have tested that our system can support more than one server, by assigning different port number for each server, so we can also use this method to access each camera. Then the user can access the corresponding camera by typing the IP and the port number in the browser of any remote computer connected to the internet.

This is an example of the link:

Eg. http://137.189.35.51:8090/test1.mpg

Secondly, this system can be compatible with 3G mobile network. As 3G phone is becoming more and more popular in the world, it will dominate the mobile network in some years, so it is valuable that if this system can stream video over 3G network. In our project, we have tried to use the 3G phone to access our server. The client programme used in the 3G phone is Real player. Real player use standard HTTP protocol for video streaming, so it is theoretically possible to communicate with the server by tuning the parameters in FFserver to fit 3G standard.

Conclusions

Our design of prototype is concluded to be working functionally. However, due to the limitation of time and budget, we cannot implement this system on ARM microprocessor.

There are two extensions to this project.

Firstly, with the ready of the well tuned operation system, modified FFserver and mplayer, it is recommended to implement this system on ARM microprocessor. With selected Linux functions, this embedded system can be designed to operate as a single hardware, no need to install it in computer. The advantage of doing so is that this can ease the setup procedure of the system, such that it can be operated by people with no computer background. And this can also decrease the size the price of this system to suit the market.

Secondly, this system has the possibility to stream video over 3G network, and 3G network is a growing market in recent years. It is reasonable to develop a video streaming system over 3G network. As the standard 3G network use WAP (Wireless Application Protocol) as the video streaming protocol, it is recommended to modify the server programme to respond with WAP video streaming request. It is also recommended to build a web server using WML (Wireless Markup Language, a mark-up language inherited from HTML using XML) to act as a database to store video for supervision purpose, entertainment purpose or other business purpose.

References

[1] : Intelligraphics Inc. "Introduction to IEEE 802.11", , 2001

[2] : Jim Geier, "802.11 WEP: Concepts and Vulnerability", Wi-Fiplanet, 2002

[3] : Bradley Mitchell, "802.11 Standards - 802.11b 802.11a 802.11g", About, Inc., 2004

[4] : Jim Zyren, "802.11g spec: Covering the basics", EE Times, 2002

[5] : Michael Jordan, "Getting Started with Linux", Linux Online, 2003

[6] : Peter Cherriman, "Mobile Media Networking", Mobile media research group, 1996

[7] : Precision Digital Images Corporation, "MPEG1 Alpha Version Documentation", Precision Digital Images Corporation, 2001

[8] : Hari Kalva, "Delivering MPEG-4 Based Audio-Visual Services", Kluwer Academic Publishers, pp.3, 2001

[9] : Rob Koenen, "Overview of the MPEG-4 Version 1 Standard", Internation organization for Standardization, 1998

[10] : Behrouz A. Forouzan, "TCP/IP Protocol Suite", 2nd edition, McGraw-Hill, pp.649, 2003

[11] : H. Schulzrinne [*Columbia U.*], A. Rao [*Netscape*], R. Lanphier [*RealNetworks*], "RFC 2326, Real Time Streaming Protocol (RTSP)", The Internet Society, 1998

[12] : Josh Beggs & Dylan Thede, "Design Web audio", O'Reilly, 2001

[13] : Richard M. Stallman, Lawrence Lessig (Introduction), Joshua Gay, "Free Software, Free Society: Selected Essays of Richard M. Stallman", O'Reilly, 1999