Scalable video streaming in wireless mesh networks.

Liu, Yan

Scalable Video Streaming in Wireless Mesh Networks

Yan Liu

465839

Submitted to the University of Wales in fulfilment of the requirements

for the Degree of Master of Philosophy

Swansea University
Prifysgol Abertawe

School of Engineering

Swansea University

Under supervision of Dr. Xinheng Wang

June 2009

ABSTRACT

Wireless mesh network provides efficient and reliable services for large scale communications. Video streaming in wireless networks enhances the services by delivering multimedia information to end users. However, because of the dynamic conditions of networks and variety of users, how to smoothly deliver the multimedia data to users without wasting precious network resources is still a challenge.

This thesis addressed this challenge by investigating several key issues in video streaming in wireless mesh networks. Firstly, a video streaming system, Swan Video Streaming system (SVS), over wireless mesh networks was designed and developed. Secondly, a scalable video coding scheme was adopted in SVS. Video bit streams were split into two layers, base layer and enhancement layer. These two layers of video streams were packed into two multicast groups to allow users to get access them separately based on their processing ability and network conditions. This prevents the waste of network bandwidth by eliminating the delivery of videos to all the users regardless of their conditions.

Thirdly, to improve the video robustness and reduce the overhead of the network for real-time video streaming, the important parameter messages of scale coded videos are transmitted in a reliable manner. SDP (Session Description Protocol) and RTCP (Real-time Transport Control Protocol) were improved to transmit the control messages at the beginning of video transmission and during video transmission stages, respectively. A new rearrangement method in RTCP of received packets was also proposed to improve the efficiency of algorithm and reduce network overhead. In addition, based on the feedback from video server and receivers, server and receivers can adjust their output bit rate and receiving rate according to different conditions of network to reduce the congestion.

The above approaches have been evaluated in the developed SVS testbed. Tests results show the approaches are effective and feasible in real application scenarios.

DECLARATION

This work has not been previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree

Signed ……………………… (candidate)
Date ……14/07/2010……….

STATEMENT 1

This thesis is the result of my own investigation, except otherwise stated. Other sources acknowledged by footnotes explicit references. A bibliography is appended.

Signed ………………………… (candidate)
Date ………14/07/2010……..

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ………………………… (candidate)
Date ………14/07/2010…………..

PUBLICATIONS

1. **Yan Liu**, Xinheng Wang, and Caixing Liu, "Scalable Video Streaming in Wireless Mesh Networks for Education", **invited** by *2009 IEEE International Symposium on IT in Medicine & Education (ITME 2009)*, 14-16 August 2009, Jinan, China.

2. Xu Zhou, Xinheng Wang, Muddesar, Iqbal, and **Yan Liu**, "A Handheld Mobile Device for Wireless Mesh Networks in Healthcare", *2009 IEEE International Symposium on IT in Medicine & Education (ITME 2009)*, 14-16 August 2009, Jinan, China.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I would like to thank the following for all the help and support provided throughout the research and thesis writing period.

Firstly, I would like to thank my supervisor Dr. Henry Wang for his unrelenting help and support throughout all the stages of the research. Henry has always been more than willing to take time out in order to assist me during crucial time in my project. During my research and thesis-writing period, Dr. Wang provided encouragement, sound advice, good teaching, good company and lots of good ideas.

Thanks must also be given to Dr. Shancang Li and Mr Muddesar Iqbal for all their support and guidance they provided throughout the research period.

I would also like to express my heartfelt gratitude to Prof. Jinho Choi, Mrs. Anjana Choudhuri, Prof. Nick Doran, and all staffs of IAT for their vital help and guidance during studying in Swansea University.

Finally I would like to thank my friends, housemates and my family. Their support and patience has been endless. They encouraged me when I have difficulty on the research. Without them, I would not complete my research.

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Background

Wireless networks widely spread the application of video streaming technology. The wireless communications, including satellite, mobile communications and WLAN, provide mobility to users when they transmit videos. WLAN technology, as a widely deployed last-mile wireless solution of the networks, supports low cost, high bandwidth broadband Internet access. It makes people to communicate with friends easily and freely. However, there are some inherent disadvantages of WLAN technology, such as the access points are fixed, the coverage range is short, and the mobility between access points is limited. This hinders the applications of WLAN in some situations where a network is needed temporary and mobile.

Recently, a new networking technology has been developed to break the barriers of WLAN. This is wireless mesh network (WMN) [1]. WMN is a self-organising, self-managing and reliable wireless communication network that provides dynamic topology allowing users to join and leave automatically. It enlarges the coverage of WLAN and provides full mobility for end users. In addition, it can be deployed wherever there is a need by using battery power or solar power for some special occasions, such as emergency communication, outdoor wildlife observations and sport activities. For these applications, video streaming plays an important role.

Video streaming technique, which booms in recent years, significantly extends the application of networks. Compared with text and audio contents, video can provide a more intuitive experience. So deploying real time video streaming can give the video receivers a clear sense about the situation. It suits very well to be used in wireless mesh networks. As a technique with applying prospect, real time video streaming deployed in wireless mesh networks has attracted more and more attention. However, how to transmit videos in the unreliable mesh networks is still a big challenge, especially to multiple receivers simultaneously.

1.2 Motivation

Transmitting real-time video to multiple receivers simultaneously is a hot research topic in recent years, which can be deployed in video conference and so on. To reduce the bandwidth usage, generally, multicast is used to deliver the same video content to different end users. But in reality, simply use multicast to transfer video can not fulfil the requirement of every receiver and the network requirement. The main factor is that the diversity of receivers and network condition. Nowadays, the video receivers are not limited to computers. Many small devices also have the ability to process and display the real-time video, such as PDA, mobile phones and MP4. These devices have different processing abilities and screen sizes, which result in multiple requirement of videos. Another problem is the instability characteristic of networks, especially the wireless networks such as Ad-Hoc and mesh networks. The bandwidth between the video server and every receiver is different as well as changing frequently because of the mobility of the nodes. In addition, compared with the wired networks, interference is more serious in wireless networks, which results in congestion and high error rate. For the time sensitivity real-time video streaming, this problem degrades the video quality seriously.

The three issues which mentioned above lead to simply deploying multicast to transmit real-time video to different receivers in WMN is not a good choice. To tackle the problem, a new real-time video streaming system in wireless mesh network, Swan Video Streaming (SVS) system, has been designed, which improved the scalable video coding to meet the requirement of different clients and network condition. In addition, to address the issue that video quality degraded in the unreliable mesh networks, we improve SDP protocol to adapt the characteristic of H.264/MPEG4

SVC and designed a video quality control scheme.

1.3 Contributions

The objective of my work is to design and implement a real-time video streaming system in unreliable wireless mesh networks which provides video to different hierarchical clients. It also improves the efficiency and reliability of the video transmissions in wireless mesh networks.

Firstly, a real-time video streaming system, SVS, which stands for Swan Video Streaming, was designed. This system commits itself to transmit video to multiple clients with different abilities in case of an emergency situation. It has significance to post-disaster restoration and reconstruction. With the help of wireless mesh network technique, the system can be constructed quickly and automatically configured. It is suitable for the situation that the communication system was destroyed and needed for an emergency replacement. Based on the real-time video system, the manager could grasp the overall situation and issue an order as soon as possible. In addition, for the purpose of widely using the existing devices as well as reducing the data flow on the network, the scalable video coding (SVC) scheme was deployed in the system. Mobile phones, PDAs or computers can be used to receive videos at the same time, significantly extending the scope of application. Obviously, this system can supply video in a more flexible and freely manner, which will play an important role in various areas.

Secondly, to improve the video robustness and reduce the overhead of the network for real-time video streaming, the important video parameters messages of H.264/MPEG4 SVC, mainly including the picture parameters and sequence parameters, are transmitted in a reliable manner. SDP (session description protocol) is improved to transmit the control message at the beginning of video transmission. Before the video content data is transmitted, the most important parameter information is integrated into SDP and conveyed to the end users reliably, which saves time and improves the video robustness. In addition, a new type of RTCP message (new designed RTCP Application packets) plays the same role at the video transferring phase. All the key parameters information of the video is separate from video data and then transmitted to the end users by RTCP. Compare to the traditional method which transfers the key information by TCP, this solution reduces the complexity of the system.

Thirdly, WMN, like other wireless networks, also has problem that the bandwidth is much lower than the wired networks. This system addresses the issue by classifying the end users into different communication groups based on the processing ability of the devices and the conditions of network to achieve optimal transmission results. Comparing to the standard method that transferring all the layers to the clients whatever they need, in SVS, different layers of the video are transmitted by different multicast groups based on the screen size of the clients and the network conditions. In addition, from network's feedback information conveyed by RTCP (real-time transport control protocol), the server can adjust the encoder to control the output video flow as well as the clients can choose whether to receive the high bit rate video layers or not. The clients are not only responsible for sending back the feedback, but also taking part in the video and network quality control initiative.

Fourthly, to reduce the complexity and improve the efficiency of the client, a new packets rearrange method was deployed. Different from the traditional method that using the time information of RTCP, this method just uses the RTP header itself to analyse the sequence of the packets and rearrange them, which slightly reduce the calculation time.

1.4 Thesis organization

The rest of the thesis is organised as following.

In chapter 2, video streaming technique in wireless mesh networks and related works are presented. Firstly, technology of wireless mesh network is introduced, and then the current development of video streaming is indicated. Furthermore, the key techniques in SVS system are also introduced.

In chapter 3, the design of Swan Video Streaming System (SVS) is described, including the system initialization, the server part of the system and the client pat of the system. The improvement of SVC transmission model is the highlight point of the server part as well as the packets rearrangement method is emphasized in the client part.

Chapter 4 introduces the QoS design of the system. In 4.1, H.264/MPEG4 SVC parameters information transfer scheme is proposed. Firstly, the problems of transmitting the most important information of H.264/MPEG4 SVC are introduced. Then the improvement of SDP is proposed in the beginning of video transmission. At last, the design of RTCP scheme is introduced. In 4.2, a video quality control scheme is introduced. Based on the feedback collected by RTCP, the server and clients can adjust the video flows on the networks and control the video quality.

In chapter 5, a testbed is designed and implemented. The proposed methods are evaluated on this testbed. The thesis is concluded in chapter 6 and future research direction is also given in this chapter.

Chapter 2 Wireless Mesh Network based Video Streaming System

This chapter is an overview of the wireless mesh networks and the video streaming technique. In addition, the key technique used in my video streaming system is also introduced. Firstly, in 2.1, the wireless mesh networks in introduced, including the characteristics, the architecture and other researchers work about the wireless mesh network technique. Then in 2.2 the video streaming technique is proposed. Especially, the multicast technique, which is used to send same contents to multiple clients, is described. At last, the key technique deployed in the swan video streaming system (SVS) is presented in 2.3.

2.1 Wireless Mesh Networks

A wireless mesh network (WMN) is a communication network made up of radio nodes organized in a mesh topology, which was appeared firstly in 1990s [2]. Comparing with WLAN, WMN extends the communication coverage with the help of multi-hop routing.

A typical WMN consists of three types of nodes: mesh routers, mesh clients, and gateway. Mesh routers, which form the backbone of the wireless mesh network, have the routing function to support the mesh networking and some of them work as gateway to connect to outside network. Generally, mesh routers have minimal mobility. With the help of this characteristic, a mesh router usually works in dual or more channels with equipping multiple wireless interfaces, which can be built on either the same or different wireless access technologies.

Mesh clients are end users of the network. All the devices, such as laptops, PDAs and cell phones, which have the ability to connect to the traditional wireless networks, can be seen as the clients of WMN. In addition, to further improve the flexibility and robustness of WMN, mesh clients usually have the function for mesh networking, and thus, can also work as the routes if needed. However, for the purpose of reducing the power consumption and dedicating to supply service to the end users, mesh clients only have simple routing abilities without gateway function and always have only one interface.

In WMN mesh nodes and mesh routers can access outside networks, by gateway, for example, Internet and Intranet. Actually, in some WMN, the difference between router nodes and gateway are not very clear. When the mesh routers can offer Internet accessing, they can also be seen as gateway.

The general architecture of WMN is illustrated in Fig 2.1.

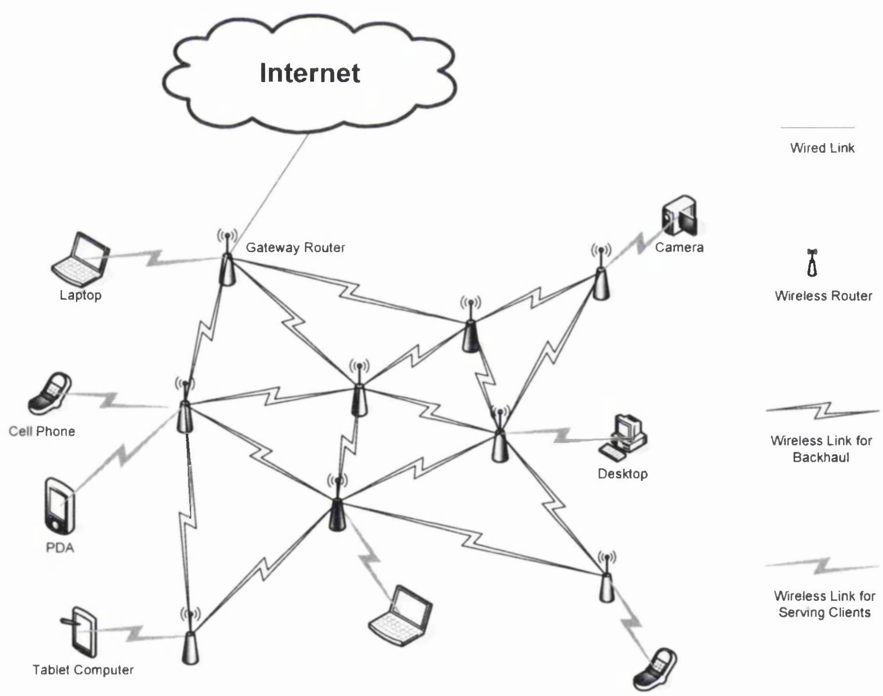

Figure 2.1 Architecture of WMN

In general, WMN has several key characteristics [3]:

1. Multi-hop. With this characteristic, node can communicate with other nodes out of sight without enhancing the radio strength, but via forwarding by neighbour nodes.

2. Self-organisation, self-management and self-healing. This characteristic makes the mesh network more robust. When any node in the network fails, other nodes could remove routes to it and establish new routes to maintain the network.

3. Clients consume less energy. With the help of the mesh router, clients do not need to consume energy to routing. It's an important characteristic when the end users use battery powered devices, such as PDA and smart phones.

As a hotspot both in the academic research and industry area, lots of works focus on how to improve the network performance. In [4], John Bicket evaluated the performance of a temporarily constructed wireless mesh networks in the rooftop of urban area, which proved that the WMN was a good method to supply communications occasionally. In [5], the authors improved the AODV protocol with a new bandwidth aware routing metric as well as route refresh and maintain method. It also proposed a method to quick response to the network dynamics. Marius Portmann and Asad Amir Pirzada [6] proposed a public safety application based on WMN which can transmit the emergency information to related people. In [7], N. Bayer designed a testbed for VoIP in WMN. In [5] and [6], authors proposed good network architectures, but unfortunately, their application was lack of video support. In [8], Danjue Li proposed a multi-source and multi-path video streaming system in WMN and designed a route selection scheme to help the system select concurrent paths. The advantage of this method is the improvement of the robustness of video stream. However, because of the multi-path characteristic, it will waste the precious bandwidth. In [9], Hsien-Po Shiang described a dynamic routing algorithm which decided how to transmit every scalable coded video packet by the intermediate node in a distributed manner. Based on these works, a WMN testbed was designed by the previous work [10], which improved the MAODV protocol and used dual channel technique. This testbed will be deployed for my video streaming system (SVS).

2.2 Real-time video streaming technique

Recently, more and more researchers pay their attentions on real-time video streaming, which can be widely deployed in lots of areas, such as telemedicine, distance learning, live broadcast, and so forth. It helps people to obtain the live information whenever they need it.

David Austerberry [11] reviewed the history of the video streaming technique and also explained real time did not mean live. Pre-recorded video also can be delivered in real-time. But in this thesis, real time only strands for the live video transferring to the clients at a rate that videos can be played back correctly in the normal speed without interruption.

The traditional way to watch a video in the networks, such as browsing the HTML which contains multimedia content, the whole file, including the HTML file and auxiliary video, need to be downloaded first. User can do nothing but wait until the download process done. This method is not suitable for viewing real time or continuous video as well as the receive device does not have enough space to save the file temporarily. Therefore, streaming [12] is introduced to address this issue. With real-time video streaming, when a little part of the video is received, commonly several frames, the received video is displayed at the beginning and the rest video is transferred simultaneously. Due to its real-time nature, real-time video streaming require low delay, low bit loss rate and high-bandwidth.

2.2.1 Video streaming model in WMN

The video streaming technology is mainly divided into two models: one is client pull model and the other is server push model [13].

Fig 2.2 shows the client pull model. Client plays a predominant role in this model, who controls the whole video service process. Now lots of applications use this mechanism. One of the most famous examples is You Tube.

Figure 2.2 Client pull model

Figure 2.3 depicts the server push model in the video streaming, in which the server plays a predominant role. Sometimes the client also needs to send out a request, but the server is responsible for all the control function. This model is also widely used, especially sending stream to multiple end users simultaneously, such as in [14], [15].

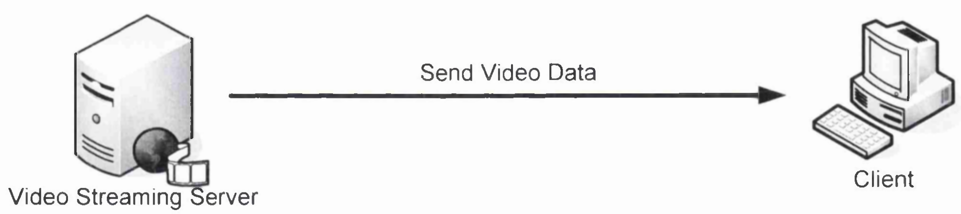

Figure 2.3 Server push model

2.2.2 Multicast technique and its applications in video streaming system

The main challenges of transmitting real time video stream are bandwidth, end-to-end delay, jitters and packet loss rate [16]. In wireless networks, the bandwidth is lower. In addition, because of the multi-hop property of wireless mesh networks, transferring videos in the WMN confronts heavier delay, higher jitter and packet loss than in wired networks. To reduce the bandwidth usage when transferring one video to multiple users, IP multicast technique is deployed and improved. Multicast technology is an efficient mechanism of forwarding data to a group of interested receivers but not need to replicate data in the server, which can dramatically reduce the load of network and video server. There are two types of multicast, network layer multicast and application layer multicast. The application layer multicast, as shown in Fig 2.4, deploys multicast function in application layer. It replicates the packet in the end user. The main disadvantage is that this method has long delay in some end points, so it is not sufficient for real-time applications.

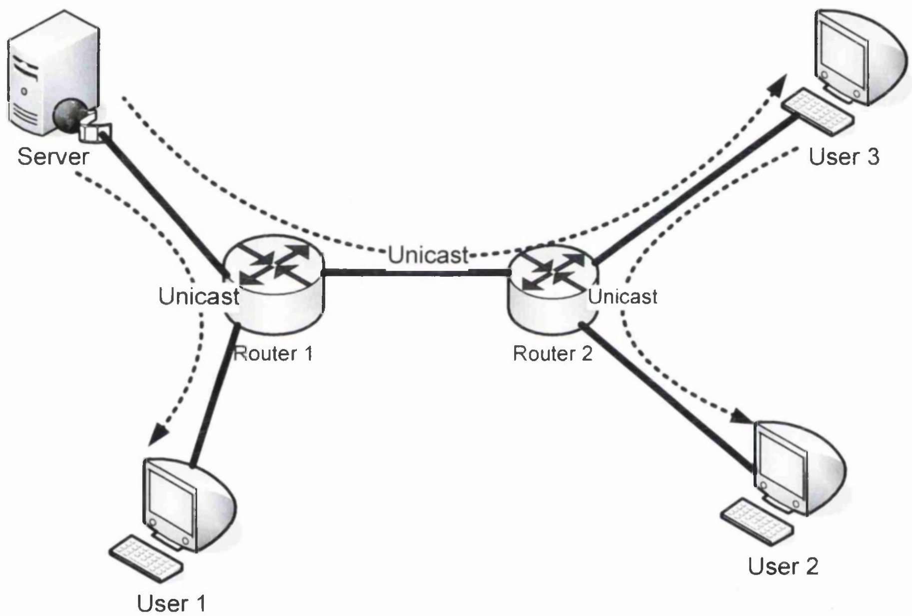

Figure 2.4 Application layer multicast

Figure 2.5 depicts the network layer multicast, which is also called IP multicast. In this method packets are duplicated by the routers then delivered to different receivers. The main advantage is that only one video stream in a line. The IP multicast is a kind of UDP traffic similar to broadcast, but only the hosts that have requested to receive the data will get it. So a host must join a multicast group first. The multicast group IP addresses range from 224.0.0.0 to 239.255.255.255, usually called Class D addresses. Especially, 224.0.0.1 is the all-hosts group, which means when pinging this group, all multicast capable hosts on the network should answer, as every multicast capable host must join that group at start-up on all its multicast capable interfaces. 224.0.0.2

is the all-routers group. All multicast routers must join that group on all its multicast capable interfaces. In any case, the IP address range 224.0.0.0 through 224.0.0.255 is reserved for local purpose and datagram destined to them are never forwarded by multicast routers.

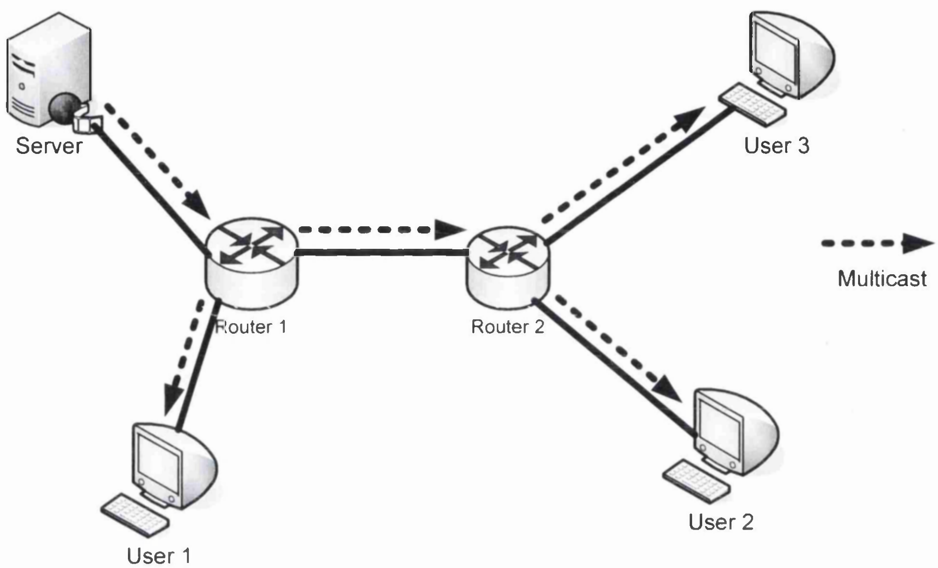

Figure 2.5 Network layer multicast

Comparing the IP multicast and application layer multicast, it is obvious that IP multicast is more suitable for transmitting real-time video in the low bandwidth wireless networks. Lots of researchers focus on how to improve the performance of the video streaming in wireless networks, especially wireless mesh networks. With improved routers, in [17], Ralph Keller presented a method to adjust the multicast video data on the router to control the data flows. In addition, some adaptive approaches are also developed to improve the video qualities played in multiple receivers, which have been classified into two distinct properties [18]: firstly, according to the video rate delivered to the end users, it can be classified into multi-rate and single rate; Secondly, according to the place where adaptation is performed, they can be classified into end-to-end adaptation and active adaptation. According to these properties, two multi-rate, end-to-end adaptation methods are most popular in video streaming area, simulcast and layered coding. The simulcast method sends multiple same video content streams with same or different bit rates.

Multiple description coding (MDC) is one of the most popular methods in simulcast. Although the simulcast (mainly MDC) and layered coding can be used in unicast area, their main application area is in multicast.

Multiple description coding (MDC) [19] [20] generates several independent substreams called description to the end user. The end user receives any single description can decode the video successfully. Generally, different description is transmitted onto different paths. In [21], Ali C. Begena finds that totally link-disjoint paths are rarely available in today's Internet. When deploying MDC, avoiding joint links does not guarantee the best quality video, so in their paper, they develop a framework to model MDC video streaming over multiple paths and then use this framework to design optimal paths selection method. In [22], authors compared the performance between MDC and single description coding (SDC) in single-path and multi-path situation. The conclusion of it showed that the achieved video quality heavily depended on the path(s) over which the video packets were transmitted, so it is critical to select a appropriate route path. In addition, in a work by Chakareski, S. Han and B.Girod [23], the MDC and layered coding (LC, SVC belongs to LC) are compared. Although MDC can significantly improve the robustness of the real-time video and most researches believe that this method is better than LC in worse condition networks. Vu Thanh Nguyen, Ee Chien Chang prove that LC can outperform MDC [24]. Actually, it is difficult to send the descriptions in multiple paths absolutely. Furthermore, LC also can design every layer based on the end users ability, so, LC method will be chosen as a basic method to distribute videos in SVS. Now, few works focus on LC video streaming in WMN. In [25], Xiaoqing Zhu deployed SVC as a video rate adaptation method for WMN.

The above approaches only focuses on one or more issues of the video streaming system and does not consider the end users' action in the real-time video streaming system. Then end users just receive the video, they can not take part in adjusting the video bit rates and so on. To solve theses problems, at first, we develop a whole real-time video streaming system in WMN, known as SVS; then to reduce the delay and improve the system efficiency, a new control message transmission scheme is designed. In addition, a new video quality control method is proposed.

2.3 Key techniques in the system

Generally, to reduce the delay of video streaming, UDP is deployed in the network layer. But UDP is an unreliable method, which can not support reliable end-to-end service. So Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP) were developed to supply time information and quality situation.

The Session Description Protocol (SDP) can be used to format the initialization information between server and client. In SVS, it is also improved to combine with video codec, which can improve the system performance.

H.264/SVC, which is the newest LC coding standard, is deployed in SVS to generate the layered videos. In addition, the standard transfer model is changed in the system to save the bandwidth.

The rest of this chapter will introduce above key techniques in detail. Furthermore, the video capture and video player software is introduced finally.

2.3.1 RTP and RTCP protocol

Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP) [26] are the key protocols for multimedia transport in IP networks. The purpose of designing RTP is to provide the timing information, source identification, loss detection, payload type for the multimedia. In addition, RTCP supplies a mechanism to monitor the multimedia delivery and provide useful information to the application, such as the packets loss, jitter and delay of the data transmitting. Furthermore, it also supplies the mechanism of synchronizing different data in a session, such as the video and audio in the video conference. RTP and RTCP do not guarantee the quality of service for real-time. Generally, applications run RTP and RTCP on top of UDP, and collect the useful information from RTP and RTCP to improve the quality of multimedia services.

When receiving video data, RTP adds its packet header to the data, and envelops into a packet. The header is shown in Figure 2.6.

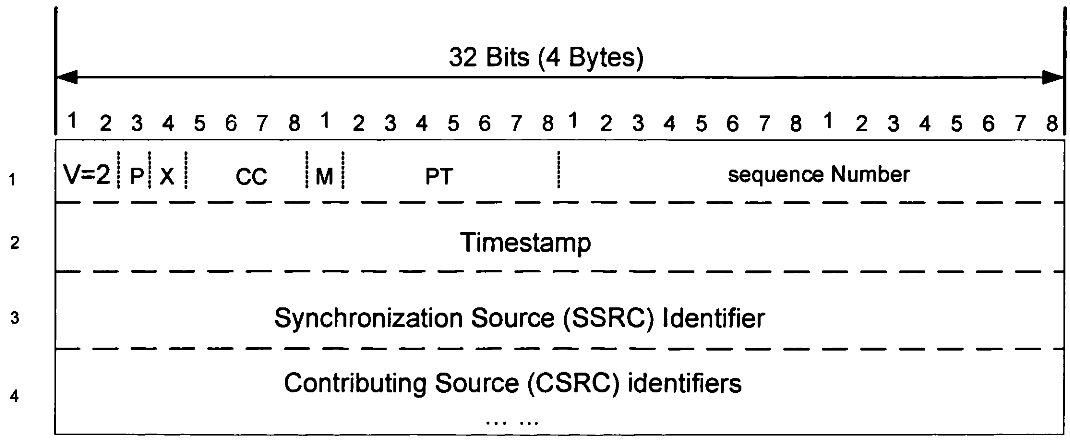

Figure 2.6 RTP Packet header

It can be seen from Fig 2.6 that in the RTP header, V, P, and X stand for version, padding, and extension, respectively. If the padding bit is set, the packet contains one or more padding octets not included in the payload after the regular header, which is used for encryption algorithms with fixed size or carrying several packets in a low layer packet. The extension means at the end of the fixed header, one header extension must be followed. The PT field means Payload Type. RTP protocol defines several video and audio types, which can be used to quickly distinguish the payload. If the payload type has not been defined, the dynamic payload type should be used. SSRC is used to identify the source. All the payload types are defined in RFC3551 [27].

The most important fields in the RTP header are sequence number, timestamp and SSRC identifier. The sequence number, which is similar to the sequence number in

TCP protocol, can be used to determine packets order in the stream. When a new packet is going to be sent, the sequence number in the packet RTP header will increase one from the previous packet. Notice that the sequence number is initialized randomly for the security reason. The timestamp field in RTP also plays an important role as the sequence number, which reflects the sampling instant of multimedia. It is useful to synchronize different session media with the help of timestamps in RTCP. The SSRC identifier, supplies a simple way to distinguish different media, especially the same type media send out by the same server.

RTP protocol can not supply any information for the quality control purpose. The quality control is done by RTCP. RTCP gathers the sender, receiver and transport information (such as the packet loss, the number of packet quantity, delay) and then sends this information to the multicast members periodically. The application can use information to diagnose the situation of the network and control the quality of the service. For multicast, since group members join and leave during the transmission, it is useful to know who is participating at any moment and how well they are receiving the video data. For that purpose, every participant periodically multicasts a reception report plus the name of its user on the RTCP. The reception report indicates how well the video is being received and may be used to control adaptive encodings. RTCP also carries a source sender identifier called the canonical name or CNAME, which can help the receiver keep track of the source sender if it loses the sender information when a conflict happens or the receiver restarts. The CNAME also helps the new participant get the information of the real-time data and the sender. To carry different control information, RTCP defines several different packet types, includes SR (sender report), RR (Receiver Report), SDES (Source description items, including CNAME), BYE (Indicates end of participation), APP (Application-specific functions, not be used generally).

Generally, All the RTCP packets are sent out as a compound packet format, which at least contains two types of packets, the SR and SDES, or RR and SDES. If the server is also a receiver, the compound packet may consist of SR, RR and SDES packets.

The most important types of packets are SR and RR reports. The SR report, send by the server, includes the timestamps and data count information. The timestamps in the SR packets consist of NTP (Network Time Protocol) timestamp and RTP timestamp (corresponds to the time of RTP data stream). The NTP timestamp provides the wall clock in order to offer a unified time to different RTP streams from different machines. The data count fields, also contains two parts, packet count and octet count, are useful for the receiver to calculate the bit rate, packets loss and so on. The syntax of SR is illustrated in Figure 2.7.

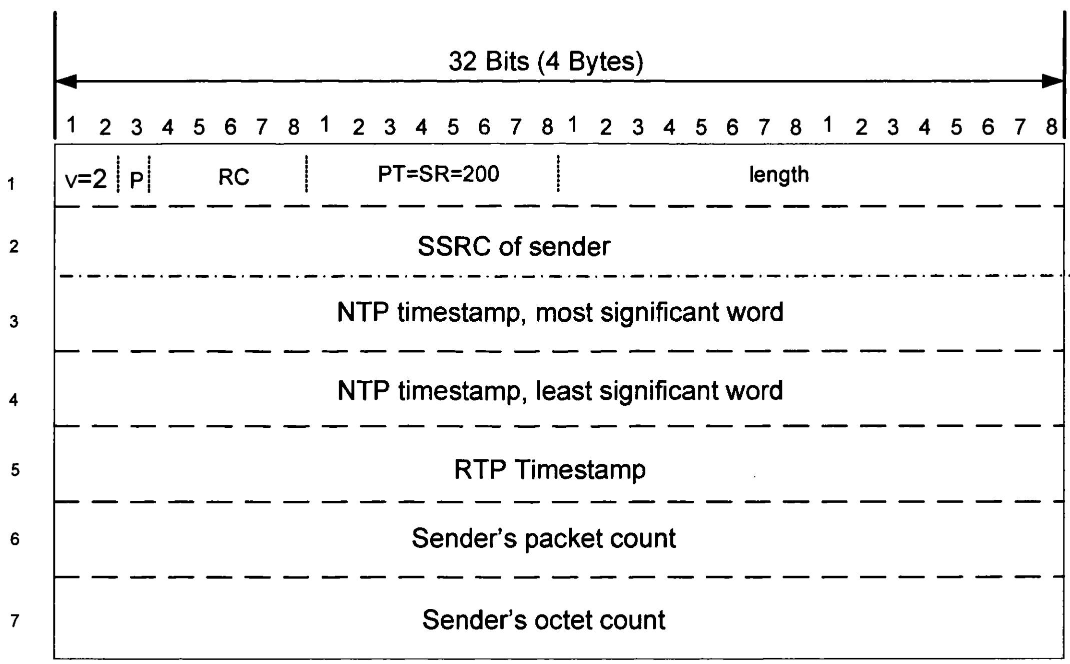

Figure 2.7 SR packets

The RR report, which is created by the clients, contains some important information, such as the fraction lost, number of packets lost, jitter and delay. Based on the information, the application could estimate the condition of network and receivers. The syntax of RR report is illustrated in Figure 2.8.

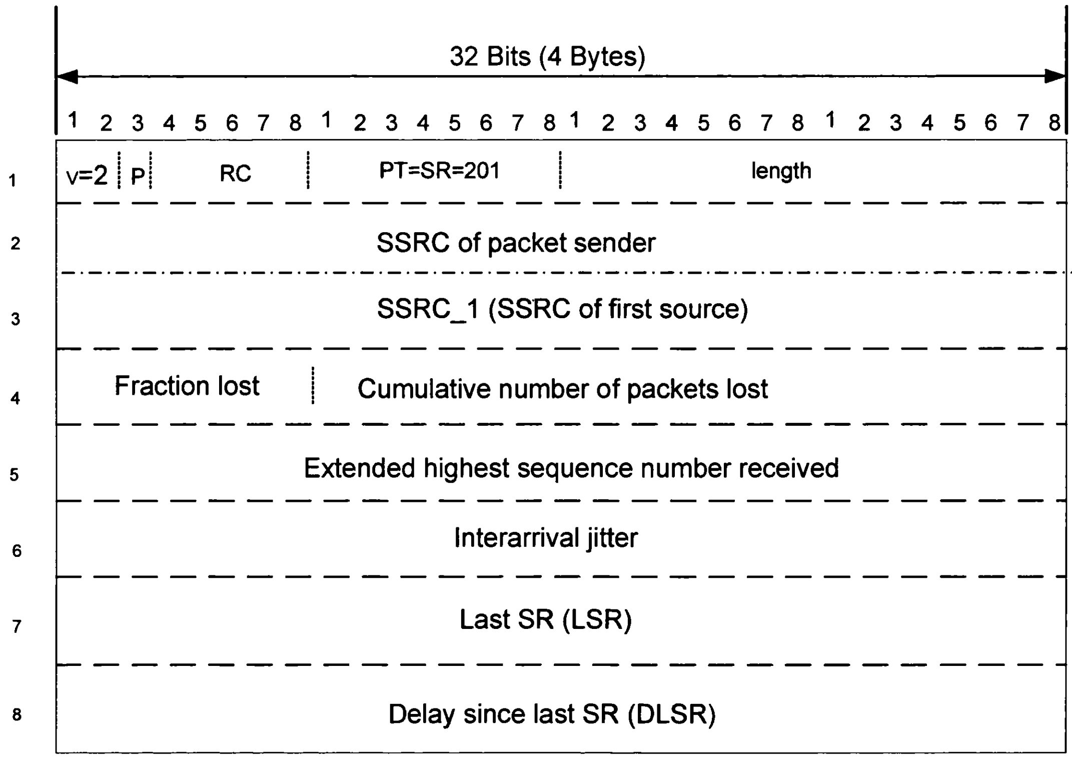

Figure 2.8 RR packets syntax

To reduce the network congestion and every participant can send out the report in time, all the RTCP packets are sent out periodically. This period is called RTCP transmission interval. To allow extending the session scale automatically, the transmission interval is changed based on the number of participant. But in default, the minimal interval is set to 5 seconds.

In a RTP session, RTP and RTCP use two ports separately. The RTP uses an even number port and RTCP uses the even number + 1 (odd) port. When too many clients join the multicast group, all of them will send out the packets, which result in consuming huge bandwidth. To avoid this situation happened, all the members of the session share the fixed RTCP bandwidth that is recommended 5% of the session bandwidth. And one fourth of the RTCP bandwidth is allocated to the senders, in order to make the new participants to receive the senders' information as soon as possible.

2.3.2 SDP protocol

SDP [28], which stands for session description protocol, is designed for negotiation between end points in multimedia communications. It does not deliver the media data itself, but supplies a standard format to exchange the initialization parameters in an ASCII string. With the help of SDP, the end users can have the necessary information before they really transmit the videos. Generally, the SDP protocol includes three parts: session description, time description and media description, which describe the session, time and media information, respectively, as shown in Figure 2.9.

Session description v= (protocol version) o= (originator and session identifier) s= (session name) i=* (session information) u=* (URI of description) e=* (email address) p=* (phone number) c=* (connection information -- not required if included in all media) b=* (zero or more bandwidth information lines) One or more time descriptions ("t=" and "r=" lines; see below) z=* (time zone adjustments) k=* (encryption key) a=* (zero or more session attribute lines) Zero or more media descriptions
Time description t= (time the session is active) r=* (zero or more repeat times)
Media description, if present m= (media name and transport address) i=* (media title) c=* (connection information -- optional if included at session level) b=* (zero or more bandwidth information lines) k=* (encryption key) a=* (zero or more media attribute lines)

Figure 2.9 SDP syntax

As shown in Figure 2.9, SDP session description consists of a number of lines. Every line has the same format, <type>=<value>. Here, <type> is always one character and case-sensitive, which indicates what the line stands for; <value> is a structured text string whose format highly depends on the <type>. As same as <type> part, <value> is also case-sensitive. Not all the lines which are depicted in Figure 2.9 are mandatory required. The line with * means this line is optional. In addition, some parts of <value> are also optional. But all the parts must arrange in the right order which is illustrated in the figure.

The session description part starts with a "v=<value>" line and continues with time description part and then with one or more media description part. Every media description part starts with the "m=<value>" line. In general, session description part values are the default for the whole SDP session unless overridden by the media description with the same <type>.

Although this protocol is mature and well developed, it only supplies a universal function. In proposed SVS, this protocol is improved to increase the efficiency and reliability of the system, more details can be found in chapter 4.

2.3.3 H.264 scalable video coding standard progress and application

In SVS, H.264 scalable video coding (SVC), which is the most currently LC video coding method, is improved to generate layered videos to multiple video clients. H.264 SVC [29] is the scalable video coding (SVC) extension of the H.264/AVC standard. H.264/AVC [30], which stands for ITU-T H.264 / MPEG-4 (Part 10) Advanced Video Coding, is the most recent international video coding standard. Comparing with prior video coding standards - chronologically, H.261 [31], MPEG-1 [32], MPEG-2 / H.262 [33], H.263 [34], and MPEG-4 (Part 2) [35], H.264/AVC provides significantly improvement for coding efficiency.

Now the modern video streaming system is typically used in IP networks for real-time services. Compared with the traditional broadcast video streaming system, the IP networks are characterized by a wide range of user devices and connection qualities. Work station, computer, laptop, PDA or mobile phones with Wi-Fi function all can be the receiving devices. It is obviously that they have different processing quality and screen size. The varying network connection quality is resulting from the variety amount of data flow in different part of the network and types of networks, such as wireless and wired networks. To address these issues, SVC was developed to generate layered structure videos to heterogeneous environment.

A SVC video stream can be split into one base layer (BL) and multiple enhancement layers (EL). Base layer provides the base quality video while the enhancement layers are used to refine the video quality. In addition, three basic types of scalability are introduced in SVC, temporal scalability, spatial scalability and quality scalability. Temporal scalability can increase the frame rate by enhancement layer; spatial scalability extends the picture size; with quality scalability, more layers contained in a video stream mean a higher fidelity.

As an extension to H.264/AVC, H.264/SVC also inherits the advantages of new features of H.264/AVC. For example, it separates the key video coding information, such as picture parameter set (PPS) and sequence parameter set (SPS), which can be transmitted by a more reliable channel. It also designs network abstraction layer (NAL) to enable simple and effective conveyance by a variety of transport layer or storage media [36].

To simplify the deployment of H.264/SVC in multiple application environments, as the previous coding standard, Profile/Level is defined. There are three Profiles in H.264/SVC, scalable baseline profile, scalable high profile and scalable high intra profile.

1. Scalable baseline profile: Primarily designed for conversational, mobile and

surveillance applications.

2. Scalable high profile: Designed for broadcast, video streaming and video conference applications.

3. Scalable high intra profile: Mainly targeted for high definition video applications.

The standard reference software of H.264/SVC is JSVM [37].Mathias Wien, Heiko Schwarz and Tobias Oelbaum tested the SVC performance based on the JSVM [38]. JSVM is also deployed as the base software to develop the coding part of the SVS.

2.3.4 Simple DirectMedia Layer (SDL)

Simple DirectMedia Layer (SDL) [39] is an open source library for writing computer games or other multimedia applications that can run on many operating systems. The library is divided into several modules including Video, Audio, CD-ROM, Joystick and Timer. The video module, which is widely used in industry and research, is responsible for displaying the video frames on the screen of the video streaming system, such as [40] [41] [42], which are the examples of using SDL for displaying videos.

2.3.5 Video for Linux (V4L)

Video for Linux (V4L) is a video capture API for Linux, which provides a common programming interface for the TV and capture cards on the market, as well as parallel port and USB video cameras.

2.4 Summary

This chapter reviewed the technology of wireless mesh network and video streaming techniques. Special attentions are placed on the techniques to improve the network performance while videos are transmitted, such as multicast, scalable video coding, and QoS. Key protocols and programs for scalable video streaming are presented in details because they are going to be used in the research. Next few chapters will present the details of the design of scalable video streaming system, QoS scheme and improvements of protocols for better network performance.

Chapter 3 Design of Swan Video Streaming System

This chapter describes the design of Swan Video Streaming (SVS) system. Firstly, the architecture of SVS system is introduced. This system consists of three modules, namely system initialisation, server and client. The initialisation module, which is introduced in Section 3.2, defines how to transmit information between server and client. Then in Section 3.3, design of video server is presented. The design of client functions is described in Section 3.4.

3.1 System overview

3.1.1 Architecture

The multicast real-time video streaming system in WMN was designed and developed as shown in Figure 3.1. In this system, a camera serves as a video streaming server, who is responsible for video capturing, encoding and transmitting real-time video streams. The end user devices, which connect to the server by WMN, to receive the videos can be desktop, laptops, PDAs or mobile phones. They can move freely in the coverage range of any mesh router. The mesh routers are the backbone of the video streaming system. Generally they are fixed or placed in a prepared location, but also can be moved locally.

Basically, this system works according to the push model of the video streaming technology. After receiving the request from end users, video server multicasts the real-time SVC video streams to the clients through mesh routers. In addition, to improve the robustness of system, unlike the traditional push model, every end user sends feedback to the server and partly controls the server to adjust the bit rates of the video streams. Figure 3.2 depicts the whole process of the video streaming system. It can be seen that if any client expects to receive real-time videos, it needs to send request with its attributes that give instructions to the server what videos it needs, as shown in step 1 of Figure 3.2. Then the server delivers the necessary information to the client and sends out the real-time videos. For the client, after receiving the necessary information, it will decide which layers of videos are going to be received and at what bit rate. When a client does not want to receive the video anymore, it sends a "BYE" message immediately which can be used by the server to estimate how many users receive the video. If the server detects that no one receives video, it stops automatically. Obviously, the server also can be stopped manually at anytime.

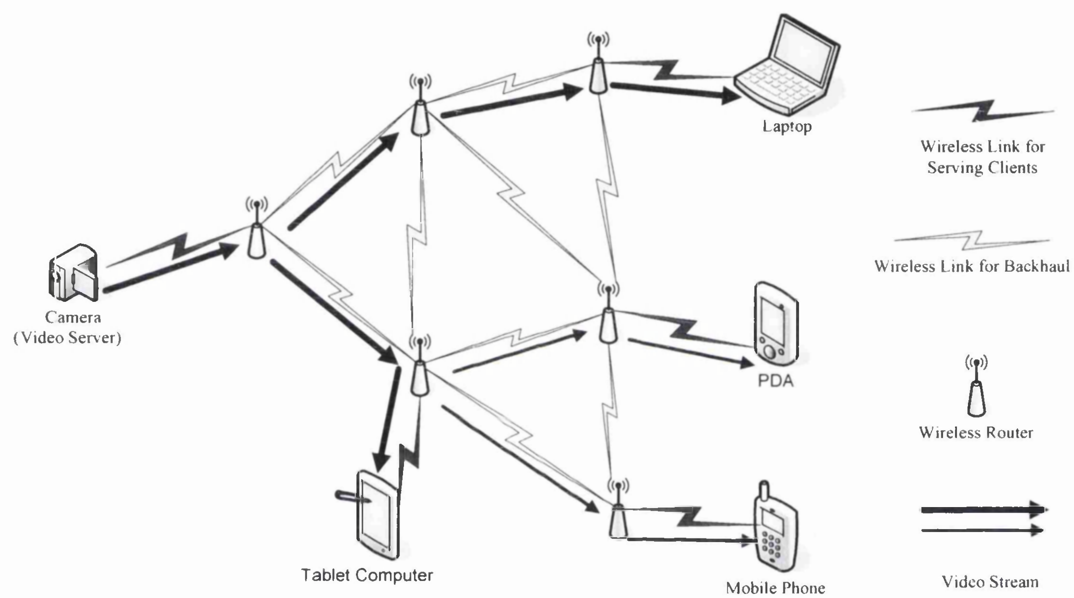

Figure 3.1 Video streaming system in WMN

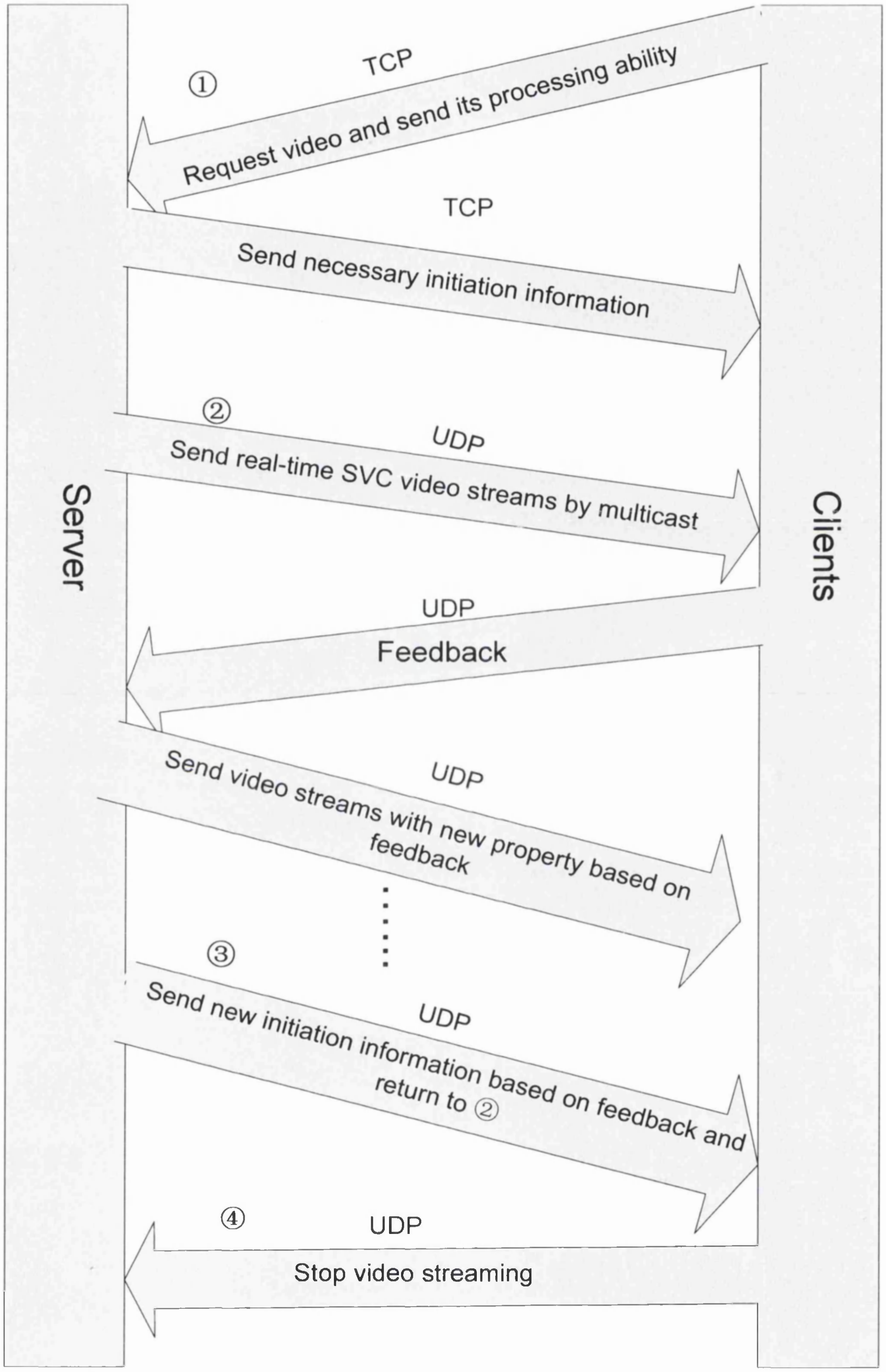

Figure 3.2 Process of video streaming system

Figure 3.3 illustrates the layered architecture of developed video streaming system. In application layer, SDP is used to negotiate between the clients and server at the beginning as in step one of Figure 3.2. Because the information conveyed by SDP is crucial to the video streaming, TCP protocol is used on the transport layer to make sure it is error free. On server side, real-time videos are captured by V4L. Then the captured videos are encoded by H.264/SVC. Before sending out by multicast, the encoded videos are encapsulated by RTP to improve the reliability with the help of RTCP. RTCP is also responsible for sending the feedback and crucial information when transmitting video between server and clients. On client side, after decoding videos, SDL multimedia library is used to replay the videos. In transport layer, UDP and TCP are two main protocols used in this system. Multicast AODV (Ad hoc On-Demand Distance Vector Routing, MAODV) protocol is used for packet forwarding in the network layer [43].

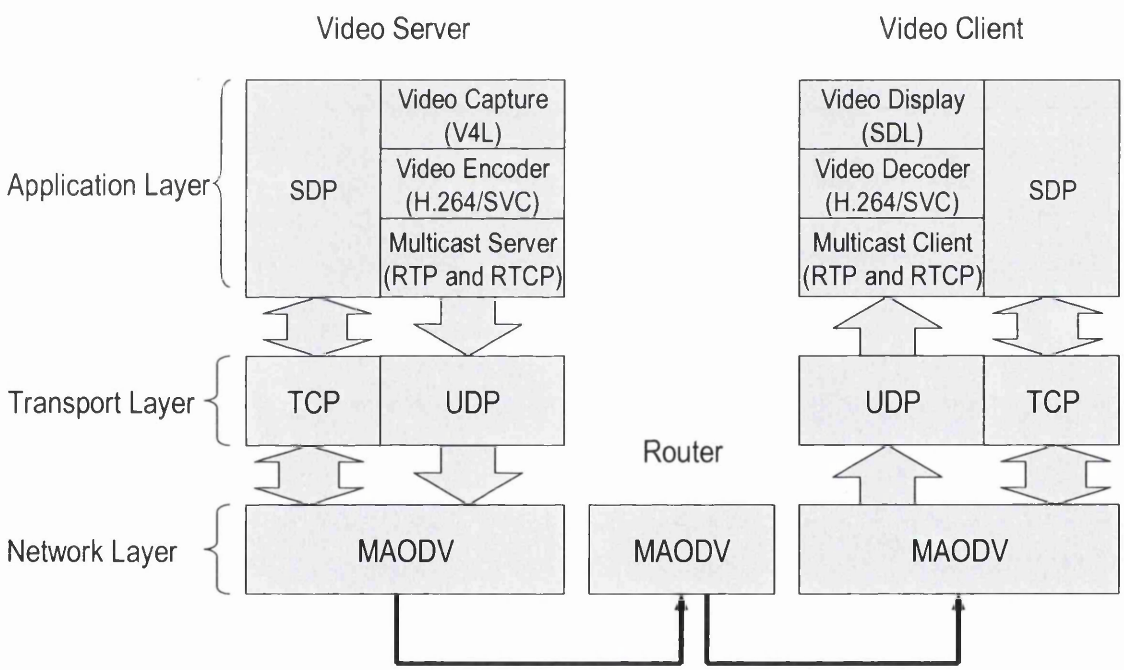

Figure 3.3 Layered architecture of video streaming system

3.1.2 Platform overview

The operating system for SVS is Ubuntu Linux distribution. The server end and clients end applications run on Ubuntu Version 8.04. MAODV routing protocol is improved to run in Linux user space with kernel version 2.6 support. Two channels are used for network communications; one is for the communications among the routers, the other one is used to connect the clients to mesh routers. In addition, the gateway router bridges the network to the outside networks. The architecture of the router is depicted in Figs. 3.4 and 3.5.

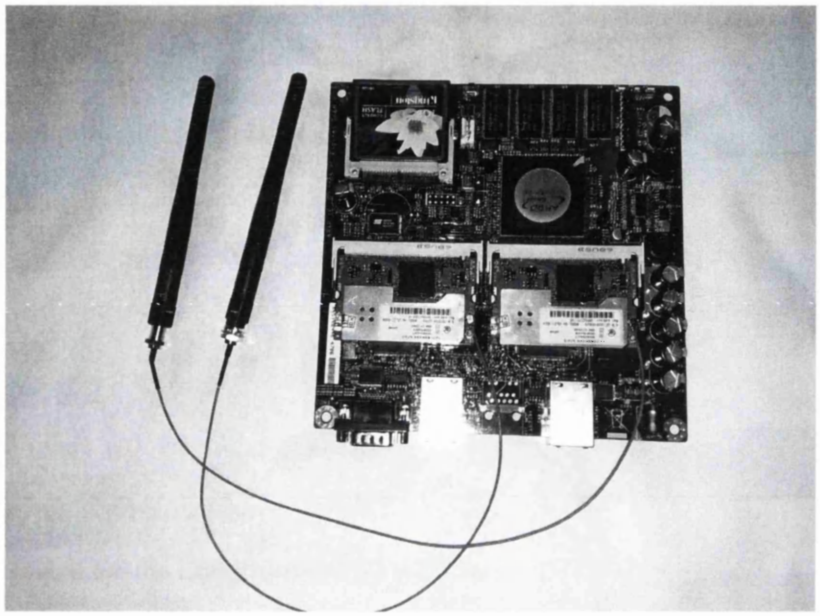

Figure 3.4 Mesh router

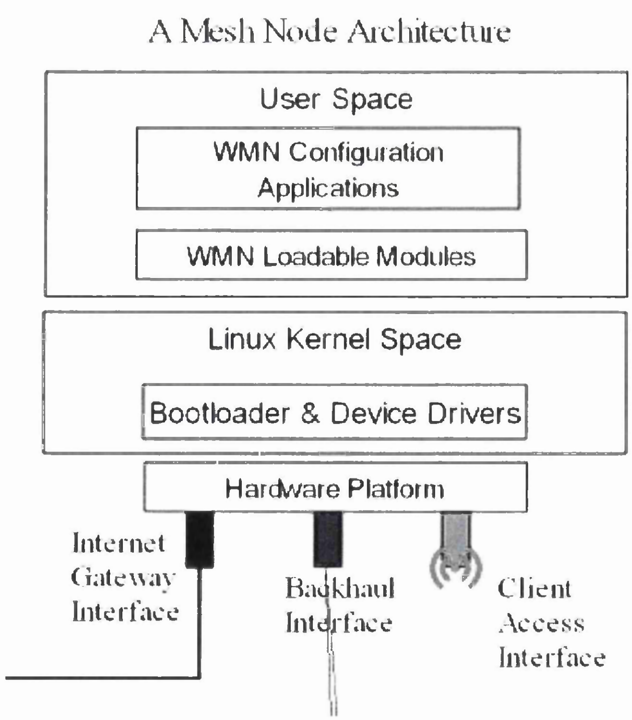

Figure 3.5 Mesh route architecture

3.1.3 System modules

The SVS consists of three modules, namely the initialization module, server and clients modules. The initialization module is responsible for initializing the system, negotiating between the server and clients. More details about initialization module can be found in Section 3.2. SDP is the key protocol used in this system.

Server module is responsible for capturing, encoding and sending the real-time videos to the networks. In addition, it also has the function of receiving the feedback and adjusting the output video streams. Server module will be discussed in Section 3.3.

All the end users will have the client applications to receive the videos. Furthermore, the client applications also have the ability to decide whether to receive part or all of the videos based on the conditions of network and itself.

3.2 System initialization

Video streaming can be started by the server manually or any of the clients automatically. The first method, started by the server, does not care if there is any client receiving the real-time videos or not. The easiest way in this method is just to open the camera with default setting values. But the setting values also can be reset after the start. The main values can be set are depicted following:

A Transmission part

1) Multicast IP addresses.

The multicast IP address can be set manually. It is useful when the default multicast IP address is used by other applications.

2) The server's port number.

The server's port number is used for distinguishing the applications that use the same multicast IP address in the server. Generally in the video streaming system, two ports will be used for one multicast IP address. One is for RTP and the other is for RTCP. The RTP port should be even number and RTCP should be the next odd port number by default. So only the RTP port number needs to be set.

3) The destination's ports number.

The function of ports number on destination is the same as the server's port number. With IP address and destination's ports number, the server knows to send the exact application data to the clients.

B Video coding part

1) Quantisation parameters (Qp).

For a video coder, Qp has a very important impact on the compression rate, which is a key parameter that controls the output bit rate. As a result, Qp can impact the output bit rate. The higher the Qp value is, the lower the output bit rate is. So Qp is a crucial value that can be set to control the initial output bit rate of the video streaming.

2) Group of Pictures (GOP) size

In order to encode the videos with a high compression rate, the frames are always encoded into GOPs, which include I-frame (key frame), P-frame or B-frame. I-frame stands for intra coded frame, which can be decoded independently. P-frame (predictive coded frame) contains the difference information from the preceding frame, so it only can be decoded correctly with the help of preceding I-frame or P-frame. B-frame stands for bidirectional predictive coded picture, which contains difference information from the preceding and following I-frame or P-frame. A GOP always begins with an I-frame and then followed by several P-frames. B-frame is inserted among I-frames or P frames, but it is not mandatory. Figure 3.6 illustrates a GOP with eight frames, where the GOP includes one I frame, one P-frame and six B-frames.

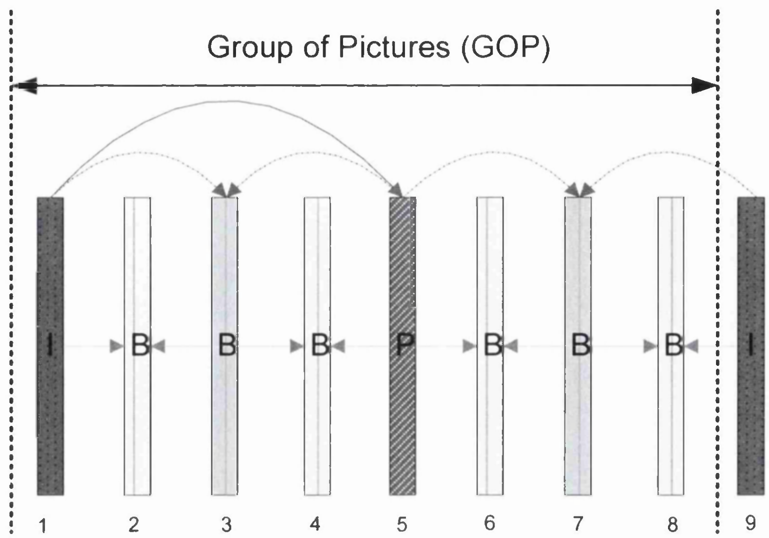

Figure 3.6 Group of Pictures (GOP)

The size of B-frame and P-frame is smaller than the I-frame because only the difference information from other frames is contained. GOP size can be changed for adjusting the output bit rate. GOP size needs to be limited in the real-time video streaming system.

3) Intra period

Intra period indicates the intra coded frame inserted period for the video sequence. Every Intra Period frame is intra coded. When this parameter is set to -1, only the very first frame in the video sequence is I frame. Otherwise the value needs to be equal to GOP size or a multiple of GOP size.

The second method is that the system can be started by the first client connecting to the server. The process is that the first client sends a request to the server by unicast, which includes its screen size. The screen size parameter is used by the server to decide which layer of the videos will be sent to the client. Then the server will send the necessary information to the client by SDP protocol, as shown in Figure 3.5. To transmit this crucial information in a reliable manner, TCP is used in the transport layer. Video is then sent to the client by multicast.

```
v=0
o=swanmesh 88311887031 88311885421 IN IP4 81.96.204.51
s=Swan Video Stream
i=Real time video streaming test
u=http://www. Swanmesh.ac.uk
e=yan6@yahoo.cn
c=IN IP4 226.0.0.11/127/2
t=0 0
m=video 19842 RTP/AVP 96
a=rtpmap:96 h264-svc1
a=fmtp:96 framesize=320*240
m=video 19844 RTP/AVP 97
a=rtpmap:97 h264-svc2
a=fmtp:97 framesize=640*480
```

Figure 3.7 An example of SDP protocol

Figure 3.7 illustrated a SDP example used in the system. In the second line, the "o=" field gives the session a globally unique identifier. We can also get the server's unicast IP address. From "s=" to "e=" fields give more information about the session and supply the method to contact to the manager. "c=IN IP4 226.0.0.11/127/2" indicates this session will use two version four IP addresses, 226.0.0.11 and 226.0.0.12. The TTL is set to 127. Because the server hopes to start as soon as possible when the client is ready and does not know when the session will stop, so "t=<start time > <stop time>" is set to zero, which means this session is permanent. The media description begins with the first "m=video 19842 RTP/AVP 96" field, where 19842 is the port number and RTP/AVP means the video will be conveyed by RTP protocol. In the next line, "h264-svc1" means the encode method is H.264/SVC and this is the base layer of the video streams. "m=fmtp" field indicates the frame size of this layer. With the preceding IP address information, every layer can be sent by one multicast group. From the SDP information, the necessary information will be received by client. Then the client can prepare the RTP channels to receive the video streams, and configure the video codec for decoding. At the same time, with the frame size, the video display application, SDL, also can be started to wait for displaying the videos.

The SDP protocol plays an important role in the second method. In addition, H.264 has a new feature that it separates the video control information into SPS and PPS from the video content, which is crucial to the video decoder. To improve the robustness and reliability of the system, SDP was improved with the H.264 new features. Furthermore, I also introduce a method to transmit the video control information by RTCP after initialization. These improvements will be introduced in next chapter.

Whatever the first or the second method is used, the server needs to know how many clients are ready to receive the videos, which can be used by the server to manage the video streaming system. The server can get the clients' information by RTCP protocol. When a new client sends back the RR report by multicast, the server will record its SSRC to a buffer and increase one to the number of clients, N.

3.3 Server in SVS

3.3.1 The framework of the server

The server consists of three components, video capture, video encoder, and video multicast server, as in Figure 3.8. V4L is used for capturing real-time video frames and then sending video frames to the encoder. After receiving the video frames, H.264/SVC encoder is responsible for compressing the frames. The multicast streaming server packs the encoded video by RTP/UDP, and then sends them to the wireless mesh network. In addition, the multicast streaming server also monitors the receiving statistics of all clients. Then it abstracts and sends the feedback to the encoder. Based on the feedback, the encoder can adjust the output bit rate to improve the quality of service and avoid network congestion.

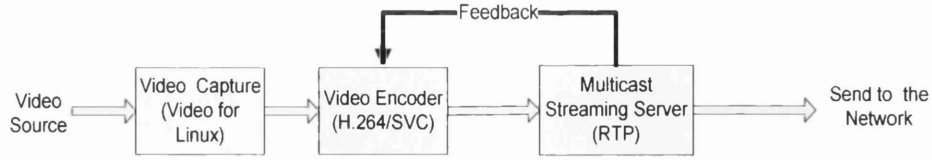

Figure 3.8 Video streaming in the server

Figure 3.9 is the flow chart of the server, which describes the details of the video processing in the server. In this system, B–frame is used to generate low rate video streams, which needs the frame information from preceding and following pictures. To encoder the B-frame, the encoder must obtain and encode the following I-frame or P-frame first, which is commonly the first one or last one of a GOP. In this system, the very first picture of the video stream is captured first and encoded as the I-frame, which is also saved as the reference picture for the B-frames decoding. Then the capturer captures N frames, where N equals the number of frames in a GOP. The N frames include the rest frames of the first GOP and the first key frame (I-frame or P-frame) of the second GOP. So the encoder can get enough key frames to encoder the B-frames of the first GOP. Like this process, every first key frame of a GOP is captured first for the B-frames encoding in the preceding GOP. When all the frames of a GOP are encoded, they are transferred to the multicast server to add the RTP/UDP/IP header and send out to clients.

Since B-frames need other frames as the reference picture, they may increase the delay time. To solve this problem, the size of a GOP is limited to four pictures by default. The architecture of the GOP used in this system is illustrated in Figure 3.10.

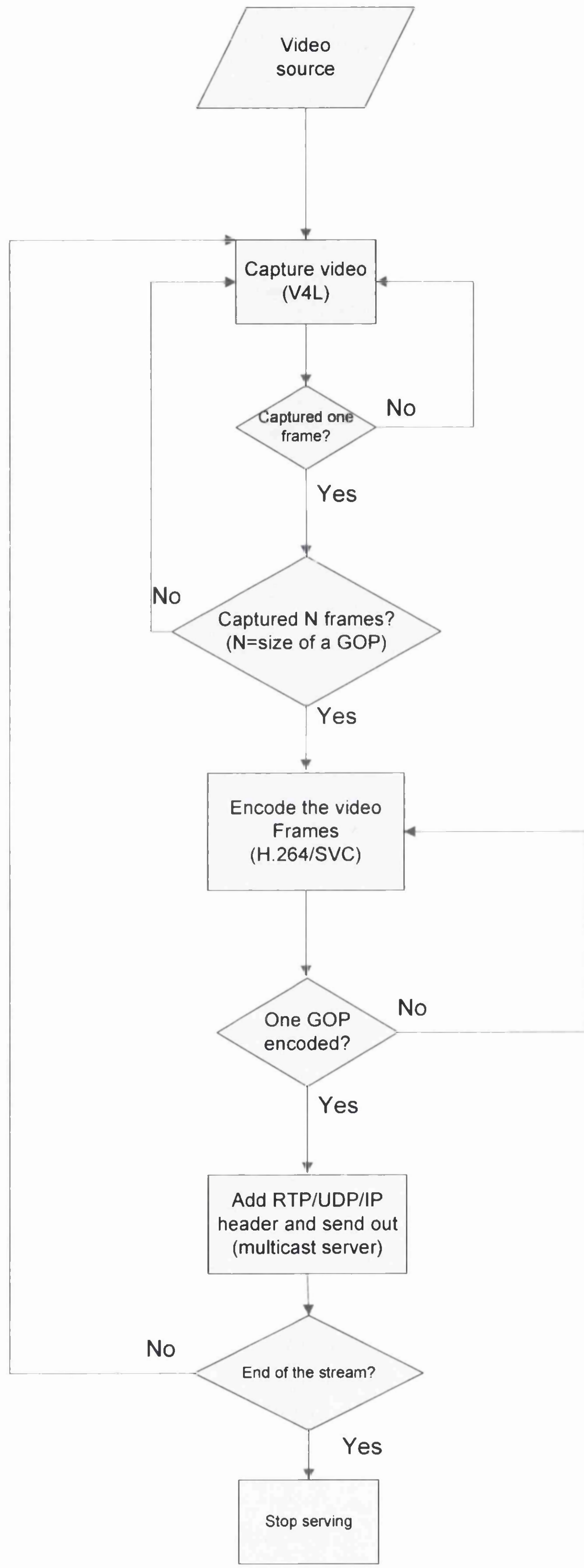

Figure 3.9 Flow chart of the server

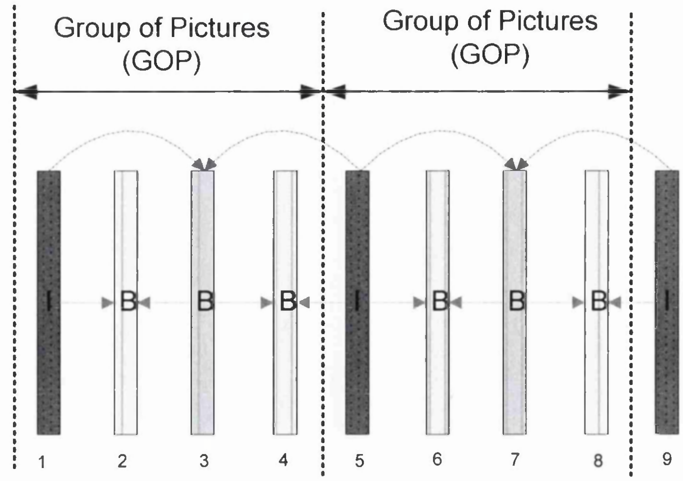

Figure 3.10 Architecture of the GOP

3.3.2 Video Capturing by V4L in Linux

The V4L [44] is responsible for capturing video frames in the system. Generally, there are four steps to capture videos by V4L, which are listed below:

1. Opening the video capture device (webcam)

Firstly, V4L needs to open the video capture device. In Linux, all hardware devices are treated as a file. The device file (also called as a special file) is an interface for device driver and can be used by the end user to access the hardware device. The device file is located at */dev/video*. To distinguish different video capture devices, the numbers from 0 to 63 are used by video capture devices. Generally, 0 is used for the first device. Like opening a usual file, function *open* is used to open the device as following:

```
std::string str="/dev/video0";
device_name=str.c_str();
open (vd->device_name, O_RDWR);    // open the device
```

2. Obtaining the device's capabilities and setting the picture properties

The second step is to check the device's properties and capability. Based on the information and the end user's request, the attribute of the video frames can be set. If the property of the video capture device is already known, it is unnecessary to check it. These processes can be done by the *ioctl(int fd, int request,....)* function. The request argument selects the functions to be performed. The most used request in the V4L is enumerated in table 3.1.

Table 3.1 Argument of the *ioctl* function used by V4L

argument	**description**
VIDIOCGPICT	obtain the capability information for a video device
VIDIOCSPICT	set the capability information for a video device
VIDIOCGMBUF	obtain the frame buffer parameters for a capture card
VIDIOCSFBUF	set the frame buffer parameters for a capture card
VIDIOCGWIN	obtain the setting of capture area
VIDIOCSWIN	set the new value of capture area
VIDIOCGPICT	obtain the default setting of the frame
VIDIOCSPICT	set the frame properties

Two of the most important arguments in Table 3.1 are VIDIOCSWIN and VIDIOCSPICT. The two arguments control the format, size and other most important properties of captured frames.

3. Negotiating a capture method and capturing frames

When collecting a picture from the video capture device, two methods can be used: the first one is using *read*() function, which is an easy and classic method. The second method is memory mapping. Because of not copying data, memory mapping is more efficient. For real-time video, spending less time to deal with the picture will make the video transmission more fluent. So I choose the memory mapping to collect the pictures. The *ioctl* ( ) function is also used by the memory mapping method. VIDIOCMCAPTURE ioctl indicates that the application begins capturing a frame. In addition, the application has to call VIDIOCSYNC ioctl to indicate if the capturing proceeding is finished or not.

When using the memory mapping method, double-buffering is also used to improve the efficiency of video capturing, which needs to be supported by the video capture hardware device. The double-buffering uses two buffers. The frame saved in one buffer is processed while the other buffer is mapping a new frame. The simple way to do double-buffering is listed below:

Prepare to capture;
VIDIOCMCAPTURE(frame 0)
while (whatever)
{
VIDIOCMCAPTURE (frame1);
VIDIOCSYNC (frame 0);
process frame 0 while the hardware captures frame 1;
VIDIOCMCAPTURE (fram e0);
VIDIOCSYNC (frame 1);
process frame 1 while the hardware captures frame 0;
}

4. Output the video

After the video frames are captured, the frames are conveyed to the encoder for encoding. *memcpy* ( ) function is used for this purpose.

5. Closing the device

Like the first step, the device can be closed as a usual file. *close* ( ) function is called for closing the device.

3.3.3 H.264/SVC codec

The original frames which are captured by V4L are too big to be transmitted on the networks. To compress the video streams is essential for video stream. In SVS, H.264/SVC is adopted in the system to compress the video streams; specifically temporal scalability and spatial scalability are applied.

After a video frame is sent to the H.264/SVC encoder, the spatial downsampling method is used to generate small size frames as the lower spatial layer. Generally, the lowest spatial layer is the base layer. Then every layer applies the temporal codec to encode different temporal frames. Generally, encoding high layer frames needs the support from low layer, which can reduce the output bit rate of enhancement layer. The architecture of a standard two spatial layer H.264/SVC codec example is illustrated in Figure 3.11.

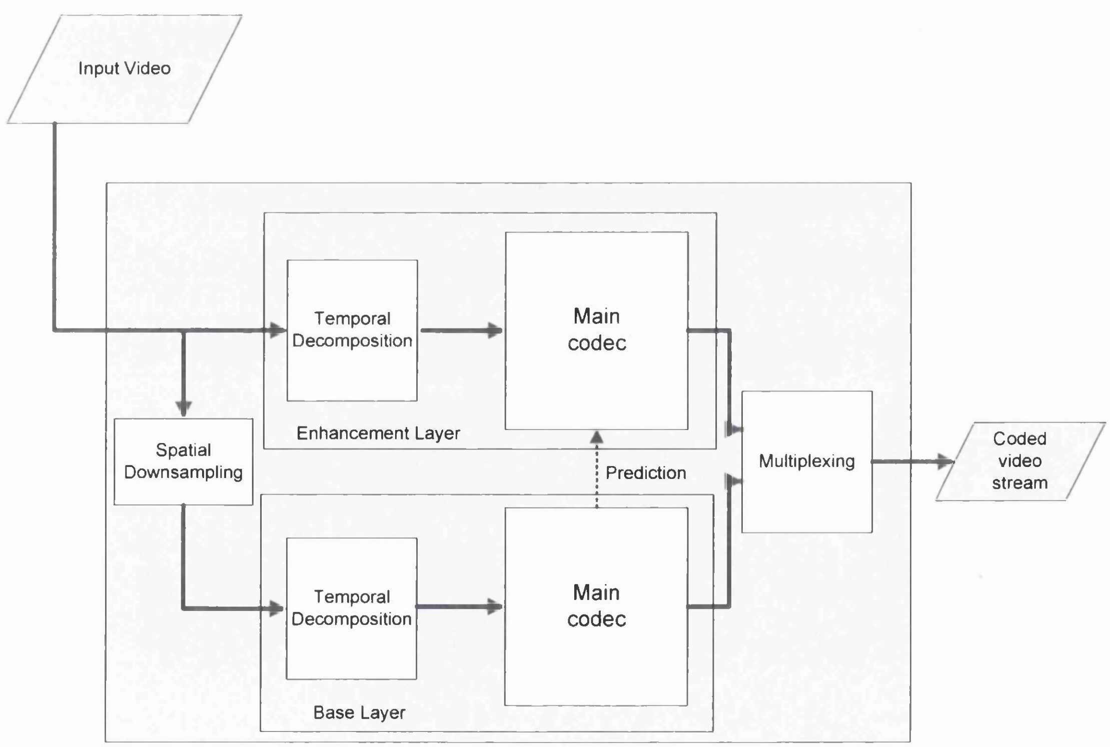

Figure 3.11 Standard two spatial layers H.264/SVC codec

After the base layer and enhancement layers are generated, the encoder marks every layer and then aggregates all of them into one stream before transmitting the videos on the network. The Multiplexing is responsible for this function in Figure 3.11. Receivers will then decode one or more layers in the stream based on their ability, keep the wanted and discard the unwanted, as depicted in Figure 3.12.

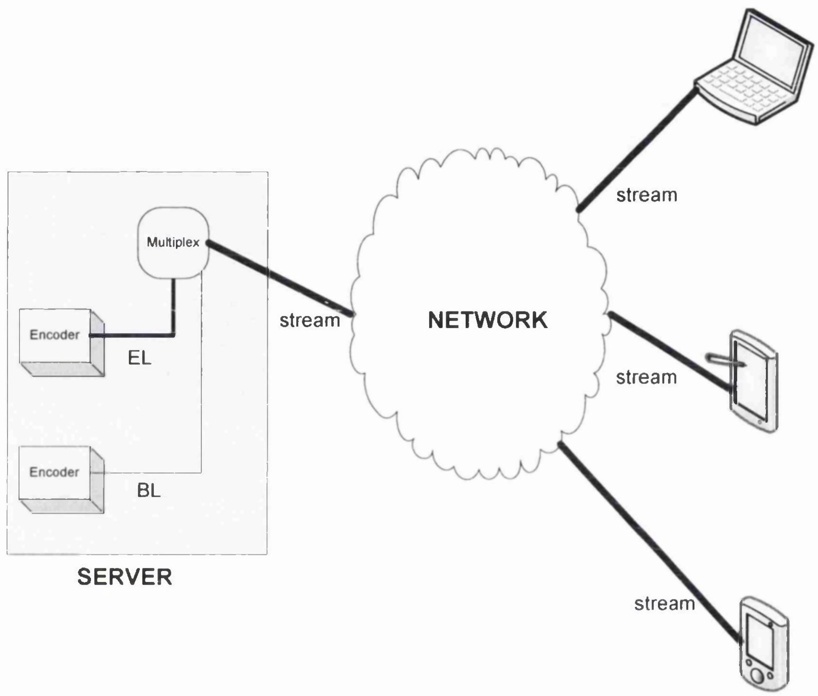

Figure 3.12 Standard SVC transmission

The advantage of this method is that it just needs to open one door on the firewall on receiver side. It is more secure in complex environments, such as Internet. System administrators are reluctant to open pinhole on their firewall. More pinholes imply more opportunities for the hackers to invade the system. However, this method has a big disadvantage that it could waste lots of bandwidth, which is the most important concern in wireless networks. Every router in the network needs to transmit the whole video streams, including the base layer and enhancement layers although some enhancement layers will be discarded by the end users.

In private wireless networks, security is not the crucial point especially the network is temporarily constructed. Actually, the disadvantage of this standard model is fatal to the wireless mesh network. Most of the end users in the WMN usually are smart devices such as PDAs and Mobile Phones with Wi-Fi, which always do not need the enhancement layer. So to transmit all the enhancement layers in the wireless networks is very inefficient and wastes the precious bandwidth resource, which easily leads to network congestion. To tackle this problem, another transmission method is designed in this system.

The goal of developing SVC is to meet different end users' requests as well as reduce the data bits flow on the network. So the crucial issue is to classify the end user devices. Generally, the criteria to distinguish different devices are processing ability and screen size. Nowadays with the development of powerful CPUs, the small device also has powerful processing ability. Obviously, screen size becomes the most important criterion. Although various screen size can be found in the mobile devices, they can simply be classified into two types: big screen and small screen devices. In this system, the screen size which is big enough to display 640*480 pixels (VGA) is classified as big screen device. Otherwise, it is small screen device.

Considering the screen size and the complexity of deploying multicast group, the encoder generates only two spatial layers of video streams, base layer and one enhancement layer. They are packed into two multicast groups separately, as shown in Figure 3.13. The reason two multicast group are used is that too many multicast groups deployed in the network may increase the overhead of the video streaming. In addition, it is difficult to manage too many multicast groups by the server, which will increase power energy consumption. For wireless networks, there is an additional problem that different routes in the network may interfere each other, which will significantly reduce the network performance. So to reduce the overhead and complexity must be considered in the design.

The base layer is designed to replay on small screen devices, so the video size is set to 320*240 pixels (QVGA). The enhancement layer is packed into another multicast group and sent to big screen devices. VGA is set as the enhancement layer's video size. The video capturer captures the VGA size videos and sends them to the encoder. The encoder firstly uses the downsampling tool to generate the base layer with QVGA size, and then encode the two layers.

Compared to the standard transmission model (Figure 3.12), it does not need to aggregate the two layers into one stream. The aggregation in the encoders is removed. Devices with big screen need to join two multicast groups. However, the device with small screen only joins one multicast group. The enhancement layer does not need to transmit to small screen device, which could save huge bandwidth.

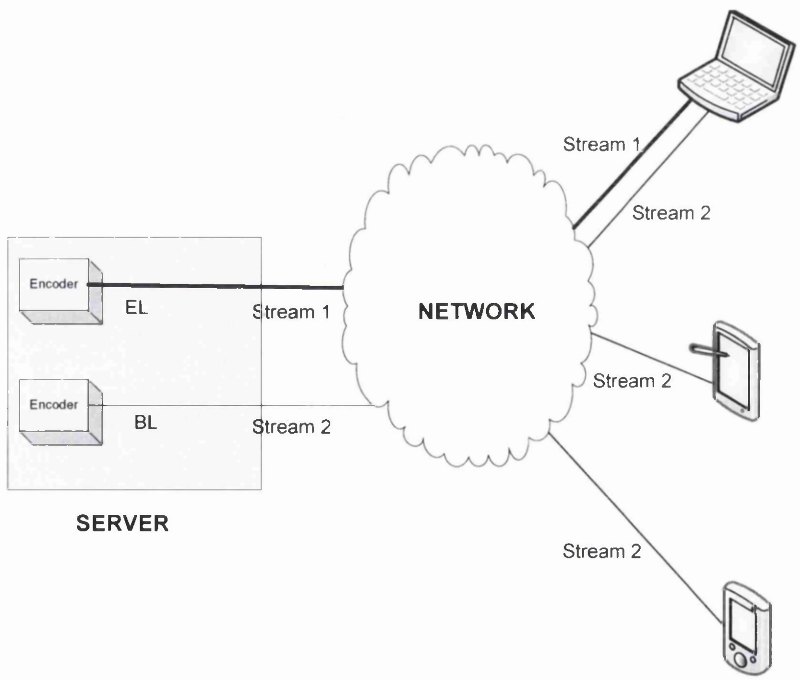

Figure 3.13 Scalable video streaming

3.3.4 Multicast server

The multicast server is responsible for sending out the videos via two multicast groups. To distinguish the video data in different layers, the server is mixed with the H.264/SVC encoder. Two RTP instances application are used. One connects to base layer, and the other connects to enhancement layer. The two instances work as parallel. They add the RTP/UDP/IP header to the packed NAL unit data and send them to the wireless mesh network.

The open source RTP application JRTPLIB [45], which is written by C++, is deployed as the multicast server. Basically, JRTPLIB can automatically process RTCP packets in the background. The end users just need to consider the RTP part. Before sending the first packet, the library needs to be initialised. Firstly, RTP session have to be created. In the system, because of using two multicast groups, two sessions are created as following:

```
RTPSession      sess1;
RTPSession      sess2;
```

Secondly, to actually create the session, the general options and parameters for the transmission component must be set first. Two classes are used for this purpose, *RTPSessionParams* and *RTPUDPv4TransmissionParams* (*RTPUDPv6TransmissionParams* for IPv6). Every RTP session must be set separately. So the two classes are created and set as following in the system:

```
/*for sess1*/
RTPUDPv4TransmissionParams    transparams1;
RTPSessionParams              sessparams1;
sessparams1.SetOwnTimestampUnit(1.0/25.0);
//set the timestamp unit, the frame rate is 1/25 sec
transparams1.SetPortbase(portbase1);
// set the rtp port of the server, default portbase=7070

/*for sess2*/
RTPUDPv4TransmissionParams    transparams2;
```

```
RTPSessionParams                sessparams2;
Sessparams2.SetOwnTimestampUnit(1.0/25.0);
Transparams2.SetPortbase(portbase1); //default port is 8070
```

Then the session can be created and set the destination multicast address as well as port number.

```
/*for sess1*/
sess1.Create(sessparams1,&transparams1);
RTPIPv4Address addr1(destip1,destport1); //the multicast address and port number
sess1.AddDestination(addr1);

/*for sess2*/
sess2.Create(sessparams2,&transparams2);
RTPIPv4Address addr2(destip2,destport2);
sess2.AddDestination(addr2);
```

The main steps to initialize the RTP session have been done till now. To send out the data by RTP, just obtain the data from encoder and call the *SendPacket*() function as following:

```
sess1.SendPacket( data , size );//data is the packed video need to be sent
sess2.SendPacket( data , size );//data is the packed video need to be sent
```

After the sessions were initialized, *SendPacket*() can be invoked continuously until the server stops working. RTPLIB also has the ability to process the RTCP packet, which can be abstracted by the end user. It can be used as the basic to design video quality control part, which will be discussed in the fifth chapter.

3.4 Client module design

3.4.1 Framework of client module

Similar as the server in SVS, the application that runs on the client also consists of three parts, multicast client, H.264/SVC decoder and video displayer. The video processing flow chart is illustrated in Figure 3.14. The multicast client receives video packets from the networks, and then removes IP and UDP header. It analyses the information contained in the RTP header, such as sequence number and timestamp. Based on the information, the multicast client rearranges the packets and sends them to the H.264/SVC decoder. For the same reason as in the encoder B-frames need to be processed. The decoder firstly needs to collect enough frames (at least two I-frames in this system), and then decode them. If the decoder finds the sign which indicates the end of the stream, the application will stop and exit. Otherwise, the decoded frames will be rearranged and replayed in sequence. At the same time, new packets will also be received by the multicast clients.

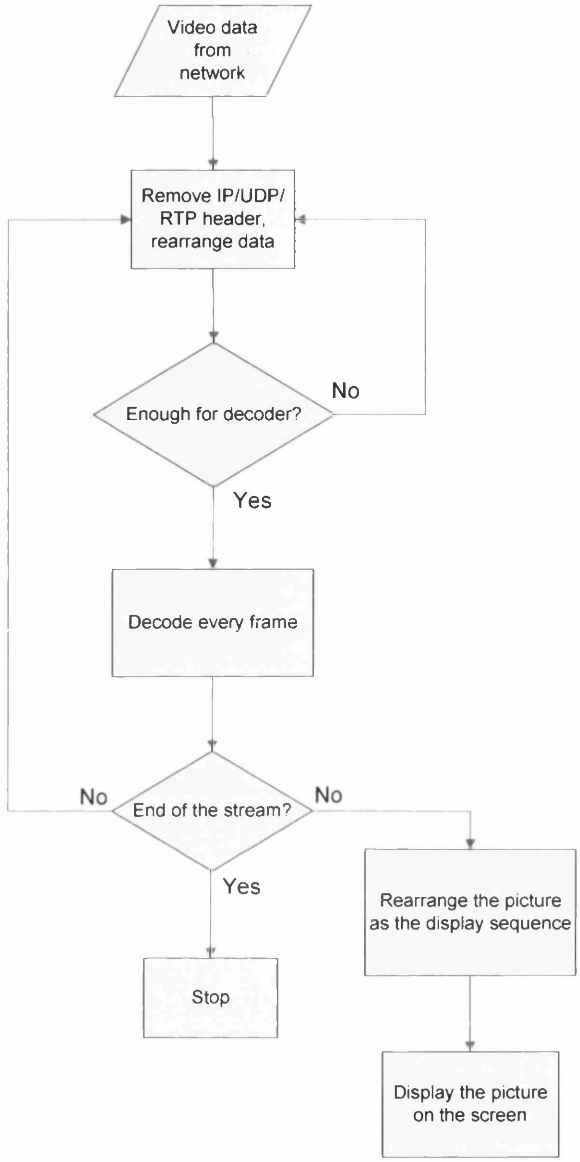

Figure 3.14 Video processing flow chart in client

3.4.2 New design in arranging the received packet

In this system, two multicast groups are used to transfer the layered videos. For the devices with big screens, the two layer videos need to be synchronized. Commonly, the RTP timestamp in RTP protocol and NTP (Network Time Protocol) timestamp in RTCP protocol are combined to synchronise the different multicast groups. NTP supplies a unity time standard. Every RTCP sender report (SR) in one multicast group contains the NTP and RTP timestamp. The RTCP uses the same RTP timestamp as in the RTP of one multicast group. Comparing with different group's RTP timestamps by conveying them to the NTP time could synchronise the layers. But this method has a disadvantage that the packets in the RTP can not synchronize by themselves but need the help of RTCP. The problem is that RTCP is not transmitted in the same channel with RTP, RTCP has independent transmission scheme. In addition, to calculate and convert the time to the same standard is complex, which will cost extra time and power energy. For the real-time video streaming devices with battery energy, their processing ability is worse than the devices with AC powers. This method can lead to slightly delay. To address this issue, we designed a new rearrangement and synchronisation method for multi layers based on the RTP protocol.

In the initialisation phase of the system, the clients have been told the layers transmitted in which multicast groups by SDP or by default setting. In the two layered video streaming system, for every frame, the first multicast group contains the prefixed packets (can be seen as the extension of NAL header in SVC) and base layer frames. The second multicast group contains the enhancement layer frames. The server always sends out the prefixed packets first, then the base layer packets, and finally the enhancement layer packets, as shown in Figure 3.13. So sending the packets using the sequence as in Figure 3.15 to the decoder, the video can be decoded correctly.

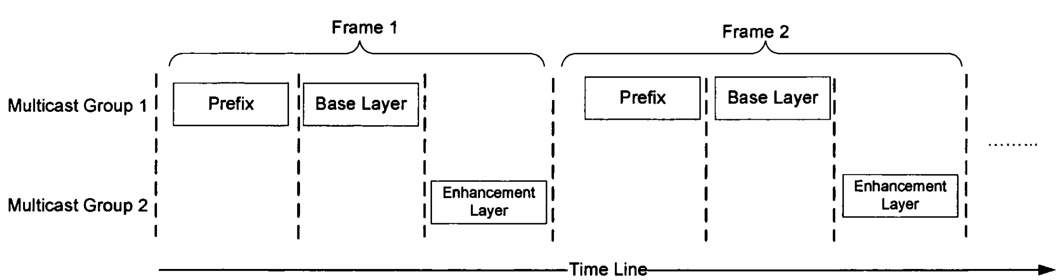

Figure 3.15 Sequence of the packets

When the packets arrive at the video client device, in transport layer the application checks the UDP port number to decide which SVC layer these packets belong to and then removes the UDP header. In application layer, RTP header is checked first. The sequence number is used to rearrange the packets in the SVC layers. In addition, UDP service is unreliable. The packet loss can not be avoided. So the sequence number is also used to check if packets are lost or not. If lost, a sign will be inserted into location where the packet is lost. Then based on the order as shown in Figure 3.15, the decoder rearranges all the packets into a timeline. When lost sign was found, the decoder will remove the sign and put next packet in order. The process is illustrated in Figure 3.16. With this approach, the decoder doesn't need to calculate the timestamp, which is significant to the end users with low processing ability and energy power.

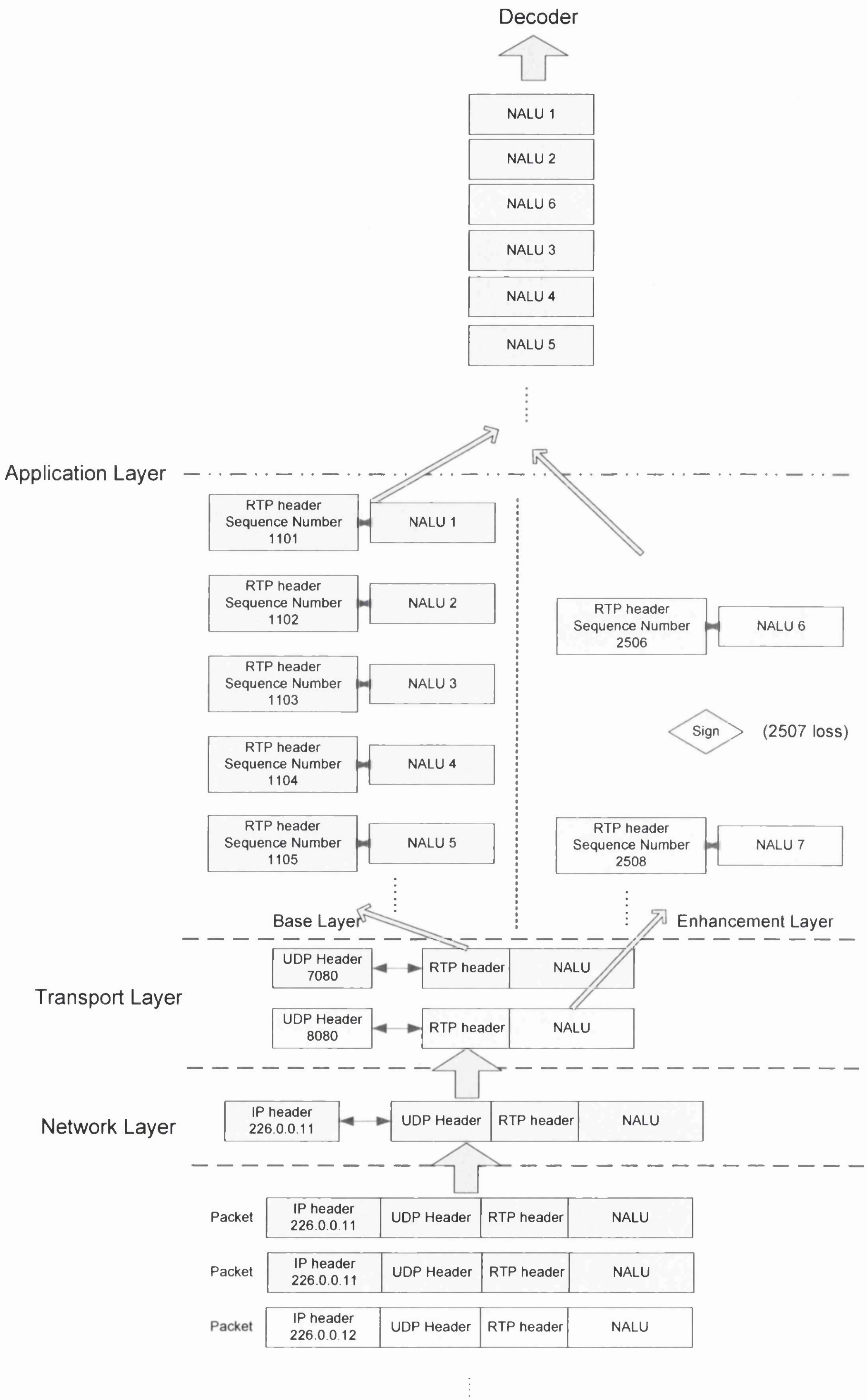

Figure 3.16 Packets rearrange method

3.4.3 H.264/SVC decoder in the client

To make sure the decoder can decode the videos as soon as possible, the frame sequence transmitted on the networks is not the same as the play sequence, as shown in Figure 3.17. With the encoded sequence transmitting to the client, the decoder can continue to decode when the first frame data is received completely. However, to play the video on the screen, the decoder needs to rearrange the frame as the play sequence. So a buffer is created at the end of the decoder to make sure the continuous video output.

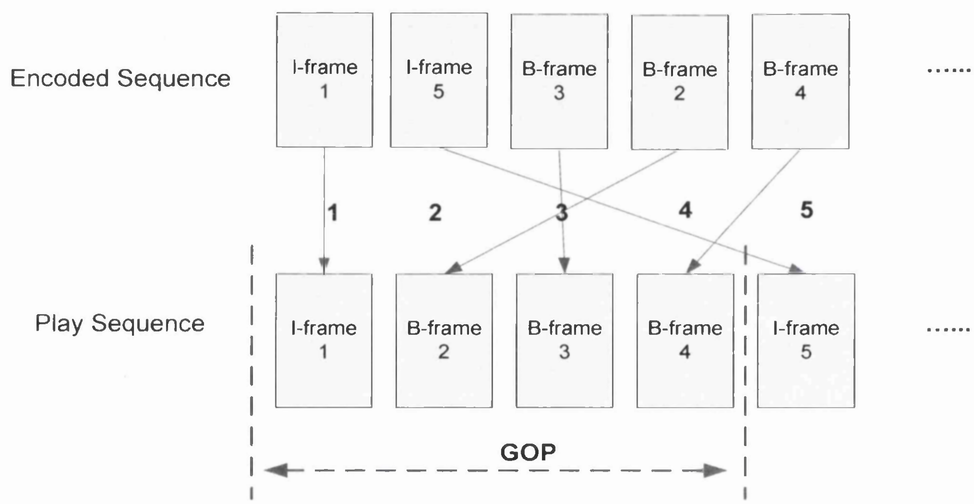

Figure 3.17 Encoded frame sequence vs. play sequence

3.4.4 SDL used in the system

In this system, SDL is used to display the videos on the screen. In addition, SDL also supports some simple operations by event, such as closing the video display screen. In the server end, from capture to encoder, the YUV420P is deployed as the default colour space, which is widely supported by most video capture devices. Unfortunately, in the clients end, SDL can not support it. To tackle this issue, *SDL_YV12_OVERLAY* is chosen as the replacer. The difference between the YUV420P and SDL_YV12_OVERLAY is that the U and V exchange their location, where U and V stand for two chrominance components. If *SDL_YV12_OVERLAY* is used directly in

SDL, the colour will become the contract colour of the real one. So when using the SDL to display YUV420P pictures, the chroma signal, U and V, must exchange their locations.

3.5 Summary

This chapter presents the design of SVS system. SVS uses two multicast groups to deliver base layer and enhancement layer videos to clients. Base layer videos are delivered to the small screen devices only, but enhancement layer videos are delivered to big screen devices as well as base layer videos. The details of the implementing the system are presented. In addition, a new packets rearrangement method on client side was proposed. By using this new method, it reduced the processing time of client.

Chapter 4 QoS Design of the System

In this Chapter, The QoS design of the system is proposed, which has two parts. In 4.1, a H.264/SVC parameters information transferring scheme is described, which is based on SDP and a new designed RTCP APP protocol. In 4.2, a video quality control scheme is proposed based on the feedback collected by RTCP. In addition, not only the server can control the video output bit rate, but also the clients can help to improve the video quality by choosing the video data.

4.1 H.264/SVC Parameters Information Transferring Scheme

Generally, the video data, which is encoded by the encoder, contains encoding parameters that are used to instruct the decoder to decode the videos. Without these parameters, the decoder can not decode the video. To improve the robustness and reliability of the video, SDP and RTCP are improved to transmit the most important information in this part.

4.1.1 Introduction

Generally, the video control information includes the most important information in H.264 standard, which indicates how to decode the video. Different from loss-tolerant video content, the video quality will be degraded drastically if the video control information is lost. For this reason, H.264 separates the common but most important parameters and transmits them first. Comparing with the previous codec, such as MPEG-4, which transmits the information with the video content every time they are used, this new feature saves the bandwidth and increase the reliability of the video. In addition, it is recommended to transfer the information in a reliable manner.

The crucial information is packed into two types of packets, SPS and PPS in H.264/AVC. In the SVC, SPSE (sequence parameters set extension) was added to support the extension of scalable video parameters. To distinguish different type of packets, *nal_unit_type* field is indicated in the header of every NAL (Network Abstraction Layer) packet of H.264. The NAL header syntax is illustrated in Figure 4.1.

bit	1	2	3	4	5	6	7	8
Description	forbidden _zero_bit	nal_ref_idc		nal_unit_type				

Figure 4.1 NAL header syntax

forbidden_zero_bit shall be equal to zero (defined by the standard). *nal_ref_idc* indicates the priority of this NAL packet which is based on the *nal_unit_type*. If the NAL packet contains the important information, such as PPS and SPS, this field must be greater than 0. *nal_unit_type* uses five bits to show the type of packet. The frequently used packet types are depicted in Table 4.1.

Table 4.1 H.264 packet types

Type description	***nal_unit_type***
Coded slice of a non-IDR picture	1
Coded slice data partition C	4
Coded slice of an IDR picture	5
Supplemental enhancement information (SEI)	6
Sequence parameter set (SPS)	7
Picture parameter set (PPS)	8
End of sequence	10
End of stream	11
Sequence parameter set extension (SPSE)	13
Prefix NAL unit in scalable extension	14
Subset sequence parameter set	15
Coded slice in scalable extension	20

In Table 4.1, *the nal_unit_type for* PPS is 8, SPS is 7 as well as SPSE is 13. All these packets can be easily distinguished just by parsing the NAL header by the decoder.

PPS applies to the decoding of one or more individual pictures within a coded video sequence. Every PPS has a *pic_parameter_set_id* field that indicates the PPS's number, which can be used to identify whether this PPS is active or not. In addition, the *seq_parameter_set_id* is contained in the PPS to indicate which SPS is activated currently. PPS packet contains the infrequently changed parameters of the picture, such as Qp value and how many slices are contained in one picture. SPS includes parameters that can be referred to by one or more PPSs or some SEIs for a serial of pictures. PPS, SPS contains identification field, profile, level information, and the number of reference pictures. The SPSE is an extension packet for SPS, so it must have the same identification number as the SPS and always follows the SPS packet.

There are two methods for transmitting the PPS, SPS and SPSE packets: in band and out of band. The in band method includes this crucial information into the RTP packets and transmits them with other video content packets, which uses the unreliable UDP protocol in the transport layer. Because UDP based transmission is not reliable manner, generally these control information will be sent more than once to ensure at least one copy will be received by the clients. Unlike the in band method, out of band method transmits the important information by a reliable channel before sending the first video slice. TCP is deployed commonly in this situation. Comparing with the in band method, it supplies a more reliable way to transmit the important information, which improves the video robustness. But it has an obvious disadvantage that maintaining another channel increases the complexity of the system and it adds more burden to the server and client. In addition, for multicast video streaming, deploying TCP to send the video information to every client will significantly increase the overhead on the networks. To address this issue, a hybrid method is used to transmit the crucial information. Out of band method is used in the beginning phase of video streaming with the improved SDP protocol without creating a new reliable channel. When the video is streaming to the clients, an improved in band method is used with RTCP protocol.

4.1.2 Out of Band Method and Improvement of SDP Protocol

The three video parameters, PPS, SPS and SPSE, are added to SDP during the server client negotiation stage. They are added to the "a=" field in session description part of SDP. The format is described as following:

$$a = PPS \quad x/y: \quad (Value)$$

$$a = SPS \quad x/y: \quad (Value)$$

$$a = SPSE \quad x/y: \quad (Value)$$

Here, *x/y* denotes the *x*th packet out of *y* packets. *Value* is the content of the packet in decimal format. For example, there are three PPS packets. If the first one has 4 bytes, the format is show in Figure 4.2.

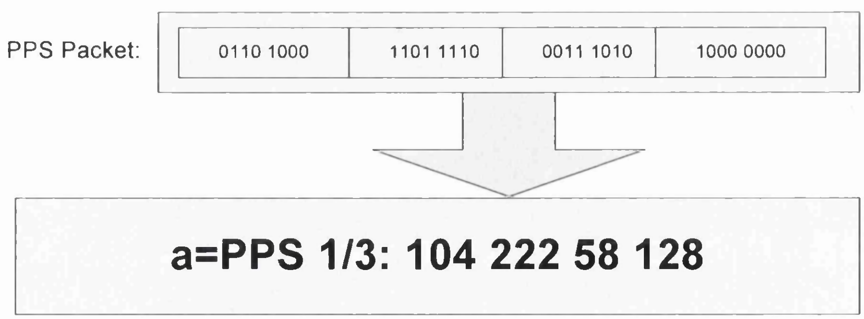

Figure 4.2 Structure of video information

After receiving the request from the client, the server firstly adds the session information to SDP as shown in Figure 4.3. Then the server will invoke the camera and encoder. The PPS, SPS and SPSE packets are abstracted from the encoder and added to the SDP message.

```
v=0
o=swanmesh 88311887031 88311885421 IN IP4 81.96.204.51
s=Swan Video Streaming
i=SDP improvement
u=http://www. Swanmesh.ac.uk
e=abc@swansea. ac.uk
c=IN IP4 227.0.0.11/127/2
a=SPS 1/1: 103 77 96 13 154 203 10 15 200
a=SPSE 1/1: 111 86 64 30 172 53 150 10 3 216 8 64
a=PPS 1/3: 104 222 58 128
a=PPS 2/3: 104 87 129 26 128
a=PPS 3/3: 104 119 129 24 128
t=0 0
m=video 19842 RTP/AVP 96
a=rtpmap:96 h264-svc1
a=fmtp:96 screensize=320*240
m=video 19844 RTP/AVP 97
a=rtpmap:97 h264-svc2
a=fmtp:97 screensize=640*480
```

Figure 4.3 An improved SDP example

With the help of this scheme, when the receiver receives the SDP reply from the sender, the video parameters information could be sent to the decoder immediately. So, once the decoder receives the streaming data, the videos will be decoded as soon as possible.

4.1.3 Transmitting Video Control Information by RTCP

Since the video parameters may be changed during video transmission, the SPS, SPSE, PPS must be sent again. As said above, traditional out of band method increases the network data flow and complexity of devices. An efficient method is required.

To supply a more reliable and low energy consumption method, an improved in band information convey method is used in the system when transmitting the real-time video data. The RTCP protocol is used to convey the crucial information by RTCP APP packets. In addition, the clients also use a new designed APP message to feed back to the server, which will reduce the unnecessary data flow on the network.

In this method, the RTCP APP packets are designed to convey the crucial information. Two new types of APP packets were designed. One is for server, sending out the SPS, SPSE and PPS information; another one is for clients, sending back the feedback to the server. The APP for server mainly contains session identification, time information and SPS, SPSE and PPS messages. The syntax of the server APP packet is illustrated in Figure 4.4.

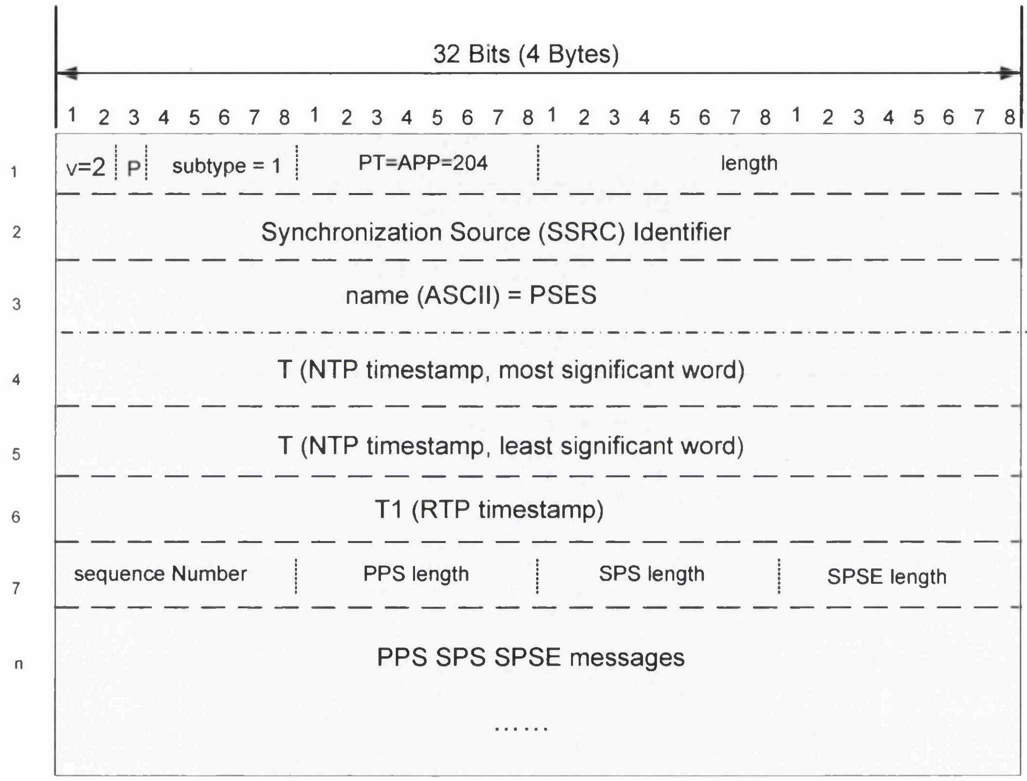

Figure 4.4 RTCP APP packet for server

The server APP RTCP packet consists of two sections, as shown in Figure 4.4. The first section, which is defined by the RTCP protocol, is 12 octets long. The fields have the following meaning:

version (V): 2 bits

It identifies the version of RTCP, which is the same as other types of RTCP packets and RTP data packets. The version defined by this specification is 2.

subtype: 5 bits

A number can be used to identify the packets with the same name (the third line, 9 to 12 octets). In general, some different type of RTCP APP packets can be defined by a unique name when they are designed for the same purpose. In this situation, the subtype field can be used to distinguish these packets. For the SPS, SPSE, and PPS transmissions, two types of APP packets are deployed. One is for server and the other one is for client. So in this server APP packet, the value of subtype is set to 1.

packet type (PT): 8 bits

Packet type contains the constant 204 to identify this as an RTCP APP packet.

length: 16 bits

Length indicates how many bits are contained in this APP packet, which includes the APP header.

name: 4 octets

A name is chosen by the application creator which defines the set of APP packets to be unique with respect to other APP packets this application might receive. The name is interpreted as a sequence of four ASCII characters, with uppercase and lowercase characters being treated as distinct. In this packet, the name is defined as PSES, which stands for a set of PPS, SPS and SPSE.

The second section is the details of the packet. The fields have the following meaning:

T: 64 bits

T is the exact time when PPS, SPS and SPSE are used. To ensure the successful delivery of RTCP APP packet, RTCP APP packet is sent twice within certain time, *T*. *T* is calculated as:

$$T = \begin{cases} T_i \times 2 & (if\ T_i > T_s) \\ T_s \times 2 & (if\ T_i < T_s) \end{cases} \tag{4.1}$$

where T_i is the time interval between first and second RTCP APP packets sending time, and T_s is the round trip time of packet travelling from server to receiver.

T_i can be calculated by

$$T_i = [S_a \times N_r + (S_a + S_s) \times N_s] / B \tag{4.2}$$

here, S_a is the average size of the RTCP packet, S_s is the size of RTCP APP packet, N_r and N_s stand for the number of the receivers and senders, respectively, and B is the bandwidth for RTCP transmission.

The time T's format is NTP (Network Time Protocol), where the first 32 bits are the most significant word, and the rest 32 bits are the least significant word. The difference between the most significant word and least significant word is that in a short time session, only least significant word is enough. Dividing NTP into two parts can reduce the complexity when it is deployed in the short time session. NTP is a protocol for synchronizing the clocks of different computers over network, which use UDP on port 123 on its transport layer. The NTP timestamp is 64 bits long, unsigned fixed-point number. The most significant in RTP is the integer part of the NTP timestamp and the least significant is the fraction part. The NTP time will overflow some time in 2036.

T1: 32 bits

T1 is the RTP timestamp that corresponds to the same time as the NTP timestamp (above), but in the same units and with the same random offset as the RTP timestamps in data packets.

Sequence number: 8 bits

Sequence number increases by one if a new server APP packet which contains new

PPS, SPS and SPSE information. This field is used to distinguish different RTCP APP packets.

PPS length: 8 bits

PPS length indicates the length of PPS information which is in the last part of this packet (PPS SPS SPSE messages).

SPS length: 8 bits

SPS length stands for length of SPS message.

SPSE length: 8 bits

SPSE length stands for the length of SPSE message.

PPS SPS SPSE messages: n bits

This field contains the data of PPS, SPS and SPSE. For the purpose of distinguishing different messages of the same type, the number of messages and length of every message is indicated first. The format of this part is depicted in Figure 4.5.

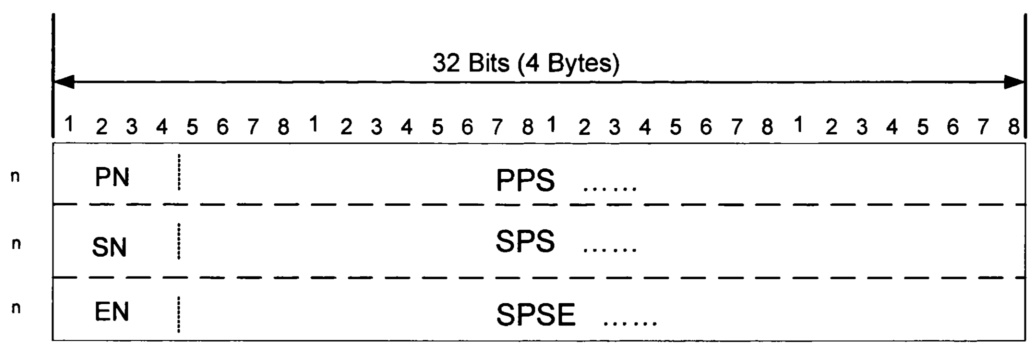

Figure 4.5 Format of PPS, SPS and SPSE in server RTCP APP packet

The PPS SPS SPSE messages part consists of three fields. The fields have the following meaning:

PN/SN/EN: 3 bits

PN indicates how many PPS message is contained in the packet. Generally, the number of PPS packets is no more than eight. 3 bits are used in this field. SN and EN have the same function as PN but stands for the number of SPS and SPSE.

PPS: n bits

This field consists of two parts, one for every message's length of the whole PPSs, and the other one for the PPS data. For the first part, the length is PN*8 bits, which means every message uses 8 bits to indicate its size. For example, if there are 4 PPS messages, the PN is 4, and 4*8=32 bits are used to indicate the size of PPS. The first 8 bits indicate the size of the first PPS message and so on.

SPS/SPSE: *n* bits

As described in the PPS field, but stand for SPS and SPSE, respectively.

The RTCP APP for clients is designed to send back the feedback information. The format of this type APP packet is depicted in Figure 4.6.

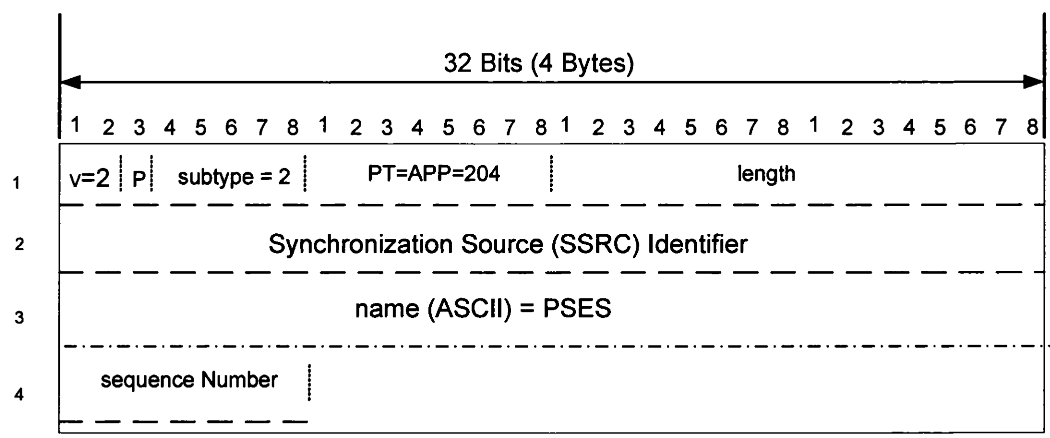

Figure 4.6 RTCP APP packet for client

The format of the client RTCP APP packet which is illustrated in Figure 4.6 is similar as the format of RTCP APP for server (Figure 4.4). The difference between them is that the subtype in the client APP is set to 2. Some fields which can not be used in this packet are removed.

When the server needs to send a new server RTCP APP packet, at first the server needs to acquire each PPS, SPS and SPSE message from the encoder and the size of each message. The total size of the three types will be calculated. All these messages will be added to the APP packet. At the same time, the time T will be calculated. The last step is to add the rest fields of the packet and send it out immediately by multicast. Before time T, two server RTCP APP packets had been sent out. The server then changes the mode to wait for the reply from the clients. In addition, the new PPS, SPS and SPSE messages will be used at time T.

On the client side, when a server RTCP APP packet is received by client, firstly, the sequence number is checked to make sure this packet is not an old one. If the sequence number is greater than the last received, this packet will be analyzed; otherwise it will be discarded. A client RTCP APP packet will be sent out by multicast immediately if the packet has been ensured. To make sure the server can receive the feedback information, the client RTCP APP packets are also sent twice. The second packet is added to the compound packet and sent out with the RR report. Then the PPS, SPS and SPSE are conveyed to the decoder and at time T, they will be used to decode the video.

To use the client RTCP APP packet, the server needs to maintain a buffer to contain the received clients' SSRC. In addition, the number of clients, N, will be used. If the server receives a client RTCP APP packet, firstly it checks the sequence number to ensure it is the latest server RTCP APP packet's feedback. Then the SSRC field is abstracted to compare with the buffer. If no same SSRC is found, this SSRC will be added to the buffer, and N will be decrease by one; otherwise this client RTCP APP packet will be discarded. At the T time, the server will check the value of N. If N=0, the server will stop sending the server RTCP APP packet; otherwise, the server continues sending until N decreases to 0. So each client has the opportunity to receive the PPS, SPS and SPSE information although they had been used by the server already.

A special situation is that when the sender sends out the server RTCP APP packet and waits for the response, a client quits the session. Two methods can be used for a client who wants to quit. Firstly, when the client quits, a Bye message will be sent. Secondly, if the server can not receive the RR report from a client at a given time slot, the client is considered quit already, which will happen if the Bye packet was lost or the client can not connect to the server. For the first situation, the server can compare the SSRC contained in the Bye packet to the buffer. If no the same SSRC was found, N will decrease one but not add this client's SSRC to the buffer; otherwise it will be ignored. For the second situation, the SSRC of client who lost connection to the server is abstracted and also compared to the buffer. Then the same process will be taken like the first situation.

With this method, the server can guarantee every crucial information sent to the client without an extra reliable channel, which significantly improves the robustness and efficiency of the system.

4.2 Video Quality Control Scheme

In this part, a video quality control scheme is designed based on the information collected by RTCP, in which the server controls the output bit stream to meet the request of all the clients and the client also takes part in the quality control by adjusting the video it received.

The condition of networks has significantly affected the quality of the video. Actually, the screen size is considered as a factor to decide which layer of videos should be delivered to the receiver. The condition of network is collected by RTCP. Periodically, the video server can receive the Receiver Report (RR) from RTCP, whilst end user can receive the Sender Report (SR). From the SR and RR results, packet loss, jitter and delay can be calculated [46]. In this system, packet loss is chosen as a metric to control the video quality.

For the multicast video streaming, the server sends out the same data to all the multicast clients. However, different end users in the mesh network may have different conditions. It is impossible for server to adjust the output bit rate to make every client comfortable. The server, in the quality control design of this system, is responsible for making sure every client could receive the basic quality video. Furthermore, to fill the gap with server, the client can reduce the data transferred to it by leaving one of the multicast groups it joined. For example, the client joined two multicast groups to receive one base layer and one enhancement layer of the video. When the packets loss goes beyond the threshold, the client could leave the multicast group that transfers the enhancement layer of videos to reduce the network traffic, which can make the network transmission more fluent and reliable.

In RR packets, the *Fraction Lost* field indicates the fraction of RTP data packets lost since previous SR packet was sent. This field is chosen as the metric for controlling the video quality. In the server end, all the clients send back the RTCP RR report by multicast. The *Fraction Lost* fields contain different values in the RR reports. The worst one of them must be picked out. Firstly, the server abstracts the fraction lost from the first received RTCP RR packet and saved it in the buffer and then compares it with the following received fraction lost values. The worst fraction lost is saved instead of the previous saved one. This process lasts T time, where T is the RTCP transmission interval time. Commonly, during the interval, the server could receive the RR reports from all the clients if the packets loss and congestion are not considered. If exceeding T time, the worst value that saved in the buffer will be sent to the encoder as a reference. The whole proceeding is shown in Figure 4.7.

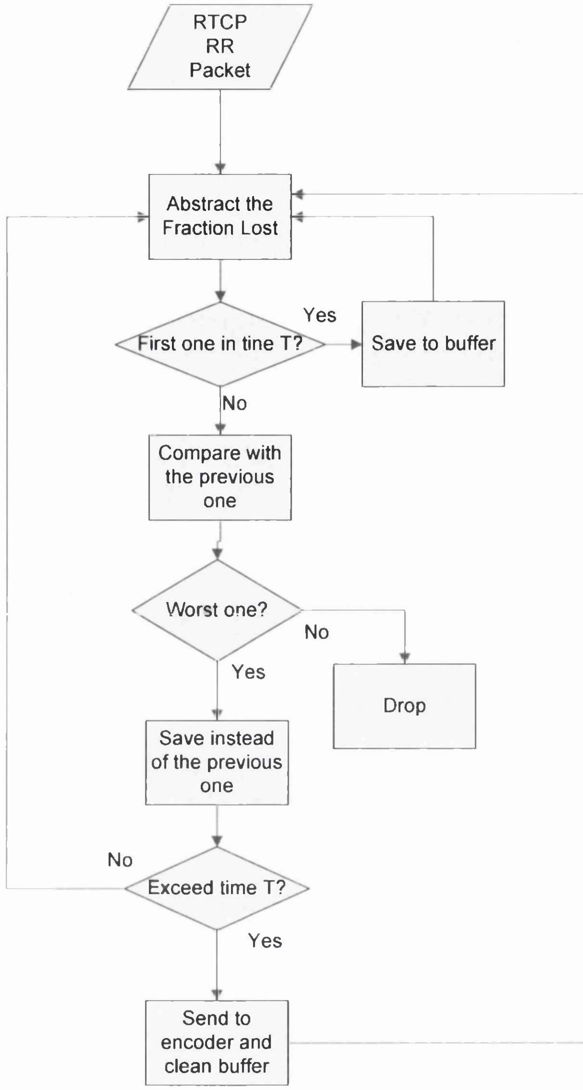

Figure 4.7 Qos design in server

In order to control the output video stream, Quantisation parameter (Qp) [47] in the encoder is examined. When the Qp value is increased, the output bit rate is decreased. Based on the received fraction lost information, the Qp value is changed in the encoder to control the output video stream.

On the client side, almost in every RTCP transmission interval, the client can receive the RTCP SR report. The packets count number can be used to calculate the fraction lost for the RR report. Before sending out the RR report, the calculated fraction lost value has been extracted. Comparing it with the limitation values, the client decides if the multicast group which contains the enhancement layer needs to be removed or not.

By default, the RTCP transmission interval is set to 5 seconds at least. The application will automatically set the transmission interval to 5 seconds if the number of clients is too small. But 5 seconds is too long for the server to wait for the feedback. It is too late to adjust the output video stream based on the out of time information, so the 5 seconds threshold is removed. In this situation, the transmission interval is calculated as usual whatever how many clients in the session.

It is important to set the reference packet limited value to adjust the video streams by server and clients. But to different encoding standard and different captured videos, this value is not unique. Previous experimental results of this system in a static situation showed when the packet loss is below 4%, the video quality is acceptable. If higher than 7%, the decoded video is nonsense to the end users. If lower than 0.5%, the video has little quality degradation. Based on these results, the minimum and maximum packet loss limitations are set in the video server as well as in end users. If the packet loss exceeds the limitation, the Qp will be adjusted. At the same time, every end user will decide to accept or reject to receive the video layer(s). This approach is shown in Figure 4.8.

```
Server:
   If  critical packet loss  >= 4%
      Qp=Qp+1
   If  critical packet loss <= 0.5%
      Qp=Qp-1

End User:
  if packet loss  >= 7%
      enhancement  layers -1
  if packet loss <= 0.5%
      enhancement layers +1
```

Figure 4.8 Quality control algorithm

Two situations must be considered. The first one is that the mesh network condition is very good. This situation commonly happens when only a few devices connect to the server and all of them are not far away from the server. The worst fraction lost value is always lower than 0.5%. As a result, the Qp value in the encoder is decreasing continuously and the output bits are higher and higher. Although the network has the ability to transfer the high bit rate video, but too high bit rates almost have no improvement to the video quality. However, it will affect the network performance because of occupying too many bandwidths. In addition, the lifetime of the devices with battery energy will be reduced seriously when the duty cycle is too high. Another situation is that the network condition is very bad. In fact, the Fraction Lost is always higher than 4%. Under this situation, the server has to reduce the output bit stream continuously. Unfortunately, the video quality is not acceptable if the bit rate is too low because of the small screen size. To address these two issues, the Qp value is limited in a acceptable range. In H.264, the Qp value is ranging from 0 to 51. In the system, the initialization of Qp value is from 20 to 30 (default value is 26). To avoid generating too high or too low bit rate, the limitation is set to ±15 from the test result.

Similarly, the server and end users can control the video quality automatically together, which not only save the bandwidth, but also improve the video robustness. In addition, it can reduce the network congestion, which could significantly improve the performance of the whole network.

4.3 Summary

In this chapter, firstly, we discussed the H.264/SVC parameters information transmitted by the improved SDP and new designed RTCP packets. To ensure the most important parameters information transmitted to the clients, they are added to the SDP before transmitting the video data. In addition, a new RTCP APP packet was designed to contain the parameters information. It can be deployed when the server is sending video data. Secondly, a video control scheme was designed. In this scheme, the RTCP collects the packet loss information and sends the information to the server and clients. Based on the information, the server can adjust the output video bit rates as well as the clients can decide which layer of the video can be received.

In next chapter, a testbed is going to be constructed and the system performance is tested.

Chapter 5 Testbed and Results

To evaluate the performance of SVS, a testbed is developed. In section 5.1 the development environment and devices of the testbed are introduced. In section 5.2, the test results for evaluating SVS are presented.

5.1 Environment and Devices

To evaluate the effectiveness and performance of proposed video streaming system, the new approaches were implemented and tested in a previously developed WMN testbed [48]. Experiments have been done in a home environment as shown in Figure 5.1. It can be seen that a camera and a computer work together as a video server where all the server algorithms are implemented in Linux. Three mesh routers are set in three different rooms. Video server is laid on the ground floor, and then two clients are put on the first floor. Video server sends one base layer (BL) and one enhancement layer (EL) simultaneously by two multicast groups. One end user works as the device with small screen, a laptop, to receive base layer videos. The other one, laptop, with big screen receives two layers of videos. The distance among the devices is no less than 5 meters and can be adjusted to achieve diverse network conditions. Figure 5.2 shows the laptops are receiving videos. Figures 5.3-5.5 show the devices used in the system.

MAODV (Multicast Ad-Hoc On-Demand Distance Vector) protocol is deployed in mesh routers as the multicast algorithm. We are not aware of any implementation of MAODV that support Linux kernel version 2.26 when we test our video streaming system. Therefore we have developed a novel implementation of MAODV in kernel user space which based on the open source software AODV-UU 0.9.5 version. The mesh router is built using a WRAP board.

The Swan Video Streaming (SVS) system supports IEEE 802.11 a/b/g standards. Every mesh router has two antennas. Atypical Omni 5dB antenna at 5.8GHz which is used to connect with other routers and a 3dB antenna at 2.4GHz is used to communicate with the end users.

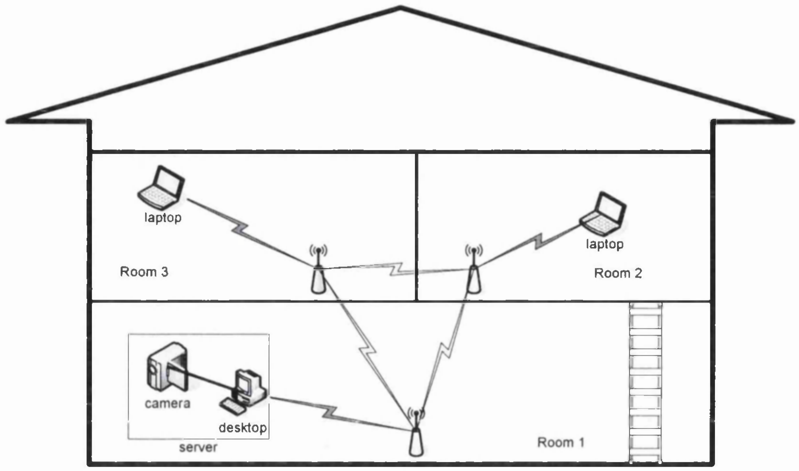

Figure 5.1 Schematic illustration of experiments

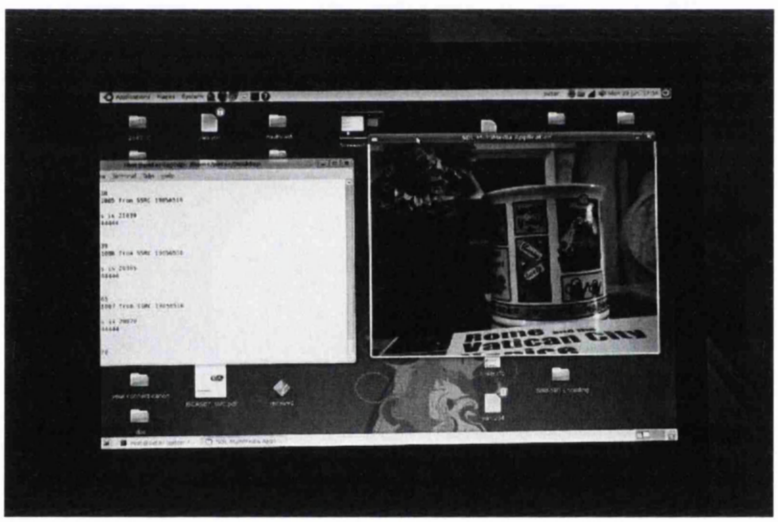

Figure 5.2(a) Video playing on laptop with big screen

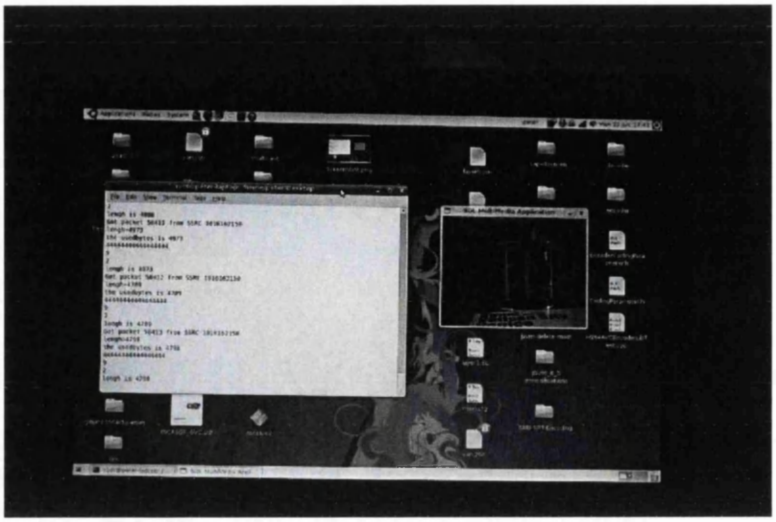

Figure 5.2(b) Video playing on laptop with small screen

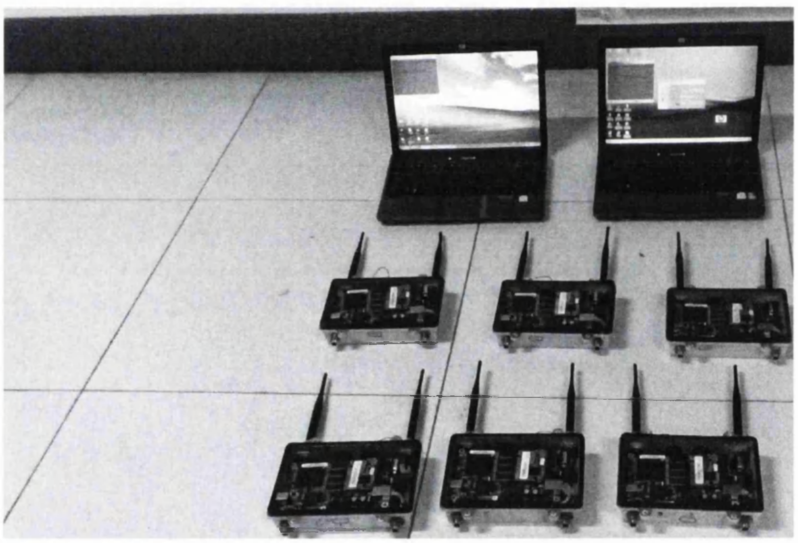

Figure 5.3 Mesh routers and clients

Figure 5.4 Video streaming server

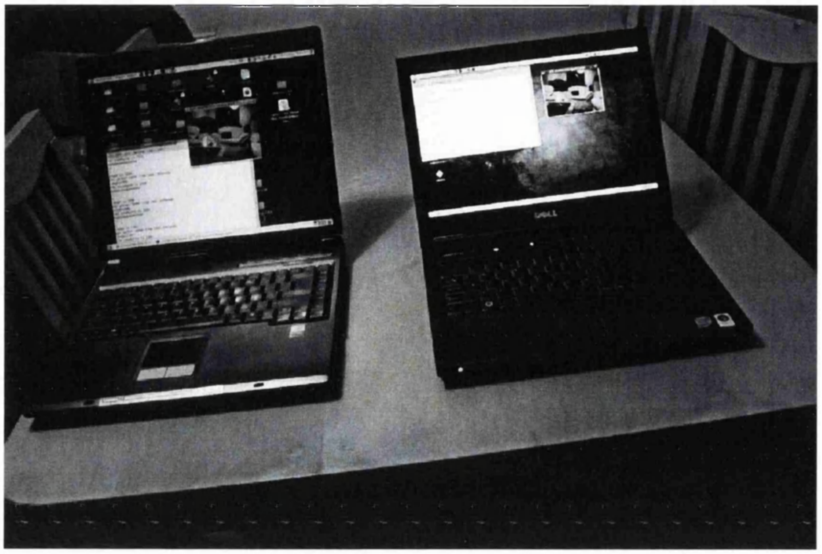

Figure 5.5 Clients of the system

5.2 Results

To evaluate the effectiveness of quality control method, three working scenarios were evaluated, as shown in Figures 5.6-5.8. In these Figures, GOP stands for group of picture, which consists of four frames. After sending one GOP, video server receives the RR report.

In Figure 5.6, since the packets loss is less than 0.5%, the server increases the output bit streams. The trend of two lines goes up. So every receiver can receive the good quality videos.

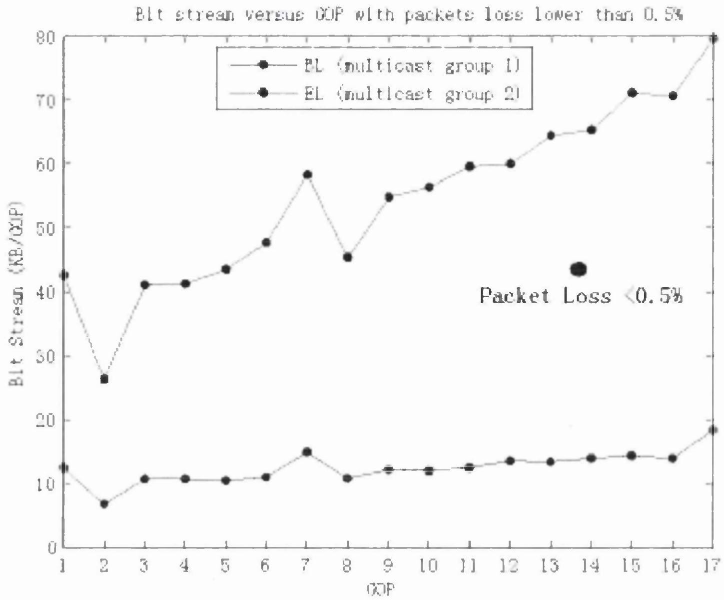

Figure 5.6 Bit stream versus GOP with packet loss lower than 0.5%

When in a usual network condition as shown in Figure 5.7, at the beginning, the packet loss of the EL is more than 4%, so the server reduces the EL output stream until the packet loss is less than 4%. The output bit streams of the two layers change frequently based on the network condition. At GOP 14, because of the huge motion captured in the picture, the bit stream increases unexpected.

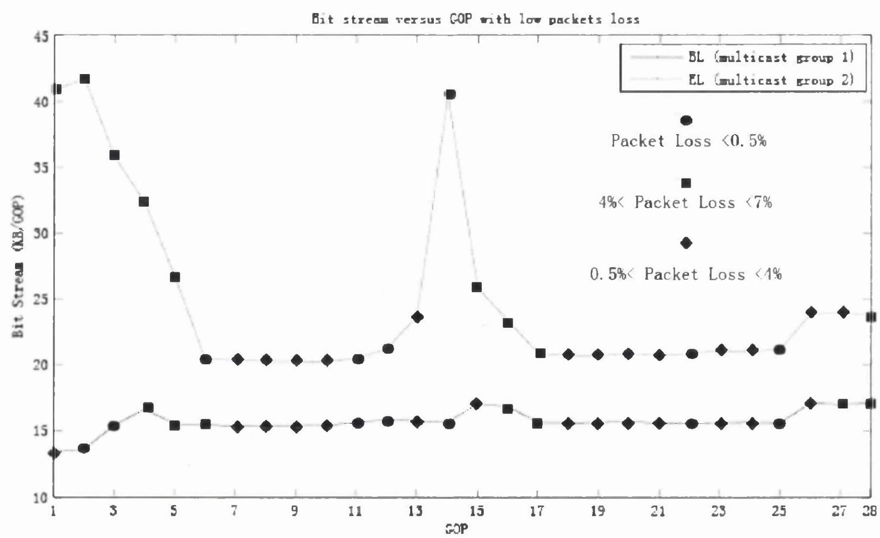

Figure 5.7 Bit stream versus GOP with low packet loss

In Figure 5.8, with the worse network condition, the EL is dropped by the end user to reduce the bit stream in the network so as to ensure BL can be received in a steady and acceptable quality.

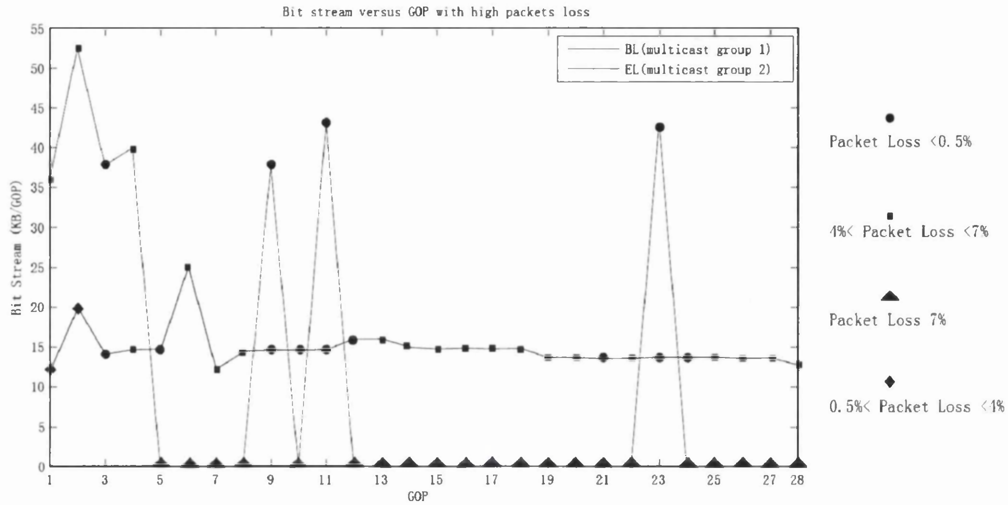

Figure 5.8 Bit stream versus GOP with high packet loss

As we can see, the base layer and enhancement layer have different packet loss rates. There are two reasons Firstly, the packets of the two layers in the same picture are not send out at the same time. Because the network condition changes continuously, the packet loss rates of the two layers are different. Secondly, the server controls the video output rate based on the feedback, so the bit rate of the video transmission is not fixed. Generally, bit rates could affect the packet loss, so the two layers have different packet loss rates.

In Figure 5.9~5.11, three test scenarios are given, in which three classifications of the video frames, good quality, acceptable quality and unacceptable quality are shown. It can be seen that with proposed video quality control mechanism, the unacceptable video frame is reduced.

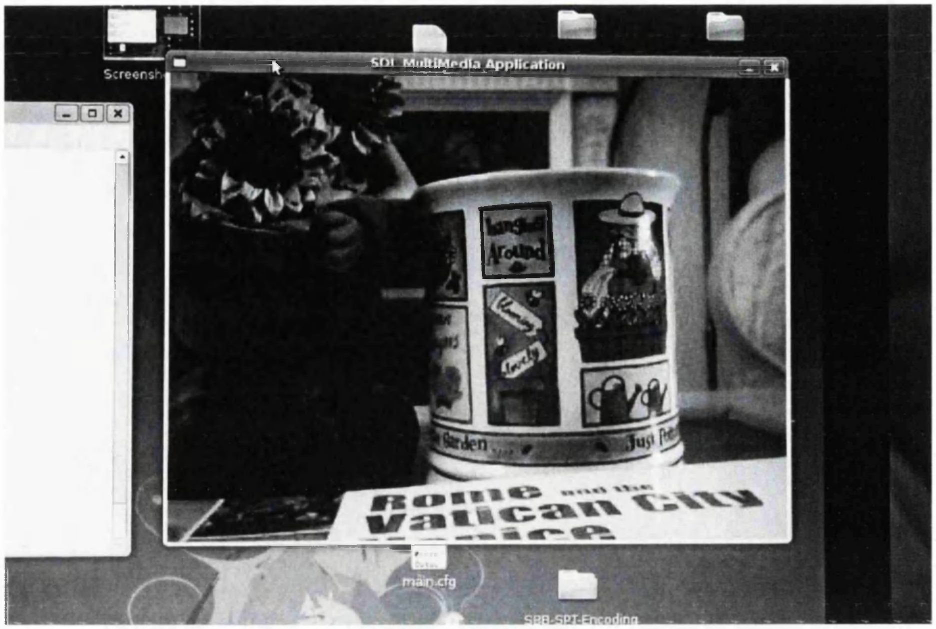

Figure 5.9   Good quality frame

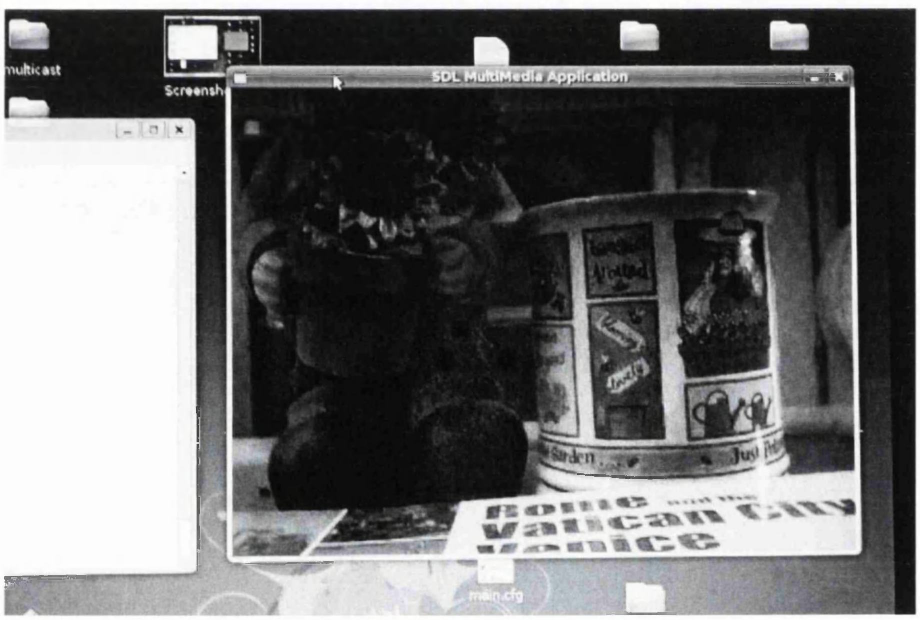

Figure 5.10   Acceptable quality frame

Figure 5.11 Unacceptable quality frame

It is obvious according to the experiment results that the system has a good control of quality of service under any network conditions. With the help of the wireless mesh network technology and proposed scalable video streaming scheme, SVS system can deliver flexible and diverse videos to the end users. In addition, the improvement of quality control in the system overcomes the unsteady problem of wireless networks.

6. Conclusion and Future work

We conclude this thesis with a summary of my contributions and directions for future research, especially about video streaming control and QoS guaranty in WMN.

6.1 Conclusion

At first, a scalable video streaming system (SVS) based on wireless mesh network was designed in this thesis. The testbed consists of a number of self-contained and self-configured nodes, called Swanmesh nodes. These nodes are based on a Linux embedded system with dual radios. In SVS, the base layer and the enhancement layer of the videos are transferred by different multicast groups, which significantly reduce the video data transmitted on the network and, therefore, reduce the possibility of congestion in the mesh network. In order to improve the quality of video in transmission with as low as possible bit rate, a packets rearrangement method in the client application was designed to improve the efficiency.

To improve the robustness of video transmission, the video control messages, PPS, SPS and SPSE, were transferred by improved SDP protocol at the beginning of transmission phase and the updated video information were transmitted by RTCP during video transferring phase. By this method, without maintaining a reliable channel, the important massage can be transmitted to the end user quickly and reliably as well as reducing the server complexity and the data traffic on the network.

In addition, based on the mesh network condition, which measured by the packet loss, the output video bit streams can be adjusted automatically by the server through the Qp value. Furthermore, end users also can accept or reject to receive relevant layers of videos automatically based on the packet loss, which can reduce the network traffic in case of congestion and ensure the video can be received smoothly.

Experiment results demonstrated the system can provide good performance real-time videos in wireless mesh network with low bandwidth.

6.2 Future Work

Clearly, SVS system reported in this thesis needs to be enhanced in several ways:

The most important function of RTCP protocol is to collect the feedback about the network situation. But a big problem is that the transmission interval will increase when the number of clients rises. So a large number of clients subjects to a long transmission interval, which means that the feedback is out of time. How to reduce the transmission interval time is a big challenge in the future works. The proposal is that every client listens to others' feedback, and then decides to send out its feedback based on other clients' feedback or not.

In the thesis, the initial output video bit rate needs to be set manually or use the default value. But the default value may not fit and setting manually is troublesome. So I propose to collect the bandwidth information from the network layer, which can be used to set the initial bit rate generated by the server.

The base layer of the video is the most important one among all the video layers. Losing the video data in base layer will cause to more serious quality degradation than other layers. A method that makes sure the base layer to be transmitted to the end users will increase the video quality significantly.

7. Reference

[1] X. H. Wang, "Wireless Mesh Networks", leader focus, Journal of Telemedicine and Telecare, Vol. 14, No. 8, pp.401-403, 2008.

[2] Ian F.Akyildiz, Xudong Wang. "A Survey on Wireless Mesh Networks", IEEE Radio Communications, September 2005. pp s23-s30.

[3] Ian F. Akyildiz, Xudong Wang, Weilin Wang, "Wireless mesh networks: a survey", computer networks, 2005.

[4] John Bicket, Daniel Aguayo, Sanjit Biswas, Robert Morris, "Architecture and evaluation of an unplanned 802.11b mesh network", International Conference on Mobile Computing and Networking, Proceedings of the 11th annual international conference on Mobile computing and networking,2005. pp 31 – 42.

[5] Yingnan Zhu, Hang Liu, Mingquan Wu, Dekai Li, Saurabh Mathur, "Implementation Experience of a Prototype for Video Streaming over Wireless Mesh Networks", Consumer Communications and Networking Conference, 2007 (CCNC 2007. 4th IEEE). pp.23-28.

[6] Marius Portmann, Asad Amir Pirzada, "Wireless Mesh Networks for Public Safety and Crisis Management Applications", IEEE INTERNET COMPUTING, 2008.

[7] N. Bayer, D. Hock, A. Roos, M. Siebert, B. Xu, V. Rakocevic, J. Habermann, "VoIP performance in 'MeshBed' - a Wireless Mesh Networks Testbed", Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE.

[8] Danjue Li, Qian Zhang, Chen-Nee Chuah, S. J. Ben Yoo, "Error Resilient Concurrent Video Streaming Over Wireless Mesh Networks", Packet Video Workshop, April 2006.

[9] Hsien-Po Shiang, Dilip Krishnaswamy, Mihaela van der Schaar, "Quality-aware Video Streaming over Wireless Mesh Networks with Optimal Dynamic Routing and Time Allocation", Signals, Systems and Computers, 2006 (ACSSC '06). pp 969-973.

[10] Zhou Xu, Wang Xinheng, IQBAL Muddersar, Liu Yan, "A handheld Mobile Device for Wireless Mesh Networks in Healthcare", 2009 IEEE International Symposium on IT in Medicine & Education.

[11] David Austerberry, "The technology of video and audio streaming", second edition, Focal press, 2005, ISBN 0-240-80580-1. pp 3-9.

[12] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, Jon M. Peha, "Streaming Video over the Internet: Approaches and Directions", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, March 2001. pp 282-300.

[13] Jack Y. B. Lee, "Concurrent Push—A Scheduling Algorithm for Push-Based Parallel Video Servers", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 3, April 1999. pp 467-477.

[14] T. C. Chiueh, C. Venkatramani, and M. Vernick, "Design and implementation of the stony brook video server," Computer Science Department, State University of New York at Stony Brook, Tech. Rep. TR-16, Aug. 1995.

[15] H. Taylor, D. Chin, and S. Knight, "The magic video-on-demand server and real-time simulation system," IEEE Parallel Distrib. Syst., vol. 3, no. 2, pp. 40–51, 1995.

[16] JG Apostolopoulos, W Tan, SJ Wee, "Video streaming: Concepts, algorithms, and systems", HP Laboratories, report HPL-2002-260, 2002. PP10.

[17] Ralph Keller, Sumi Choi, Marcel Dasen, Dan Decasper, George Fankhauser, Bernhard Plattner, "An Active Router Architecture for Multicast Video Distribution" IEEE INFOCOM, 2000.

[18] Jiangchuan Liu, Bo Li, Yaqin Zhang, "Adaptive Video Multicast over the Internet", Multimedia, IEEE,2003.pp 22-33.

[19] Reibman A, Jafarkhani H, Wang Y, Orchard M, Puri R, "Multiple description coding for video using motion compensated prediction". Proc. international conference on image processing, Kobe, Japan. 1999, IEEE Press, NewYork, 3:837–841.

[20] WangY, Lin S, "Error resilient video coding using multiple description motion compensation". Proc. workshop on multimedia signal processing, Cannes, France. 2001, IEEE Press, New York, pp 441–447.

[21] Ali C. Begena, Yucel Altunbasaka, Özlem Ergunb, Mostafa H. Ammar, "Multi-path selection for multiple description video streaming over overlay networks", Signal Processing: Image Communication 20, 2005. pp 39-60.

[22] Ali C. Begen, Yucel Altunbasak, M. Reha Civanlar, Gokce Gorbil, "High-Resolution Video Streaming In Mesh-Networked Homes" image Processing, 2005. ICIP 2005. IEEE International Conference. 11-14 Sept. 2005.

[23] J. Chakareski, S. Han, B. Girod, “Layered coding vs. multiple descriptions for video streaming over multiple paths”, Multimedia Systems 10: 275–285, 2005.

[24] Vu Thanh Nguyen, Ee Chien Chang,Wei Tsang Ooi, “Layered Coding with Good Allocation Outperforms Multiple Description Coding Over Multiple Paths”, Proc. international conference on multimedia and exhibition, Taipei, Taiwan. 2004, IEEE Press, New York.

[25] Xiaoqing Zhu, Thomas Schierl, Thomas Wiegand and Bernd Girod, “Video Multicast over Wireless Mesh Networks with Scalable Video Coding (SVC)” Visual Communications and Image Processing 2008.

[26] “RTP: A Transport Protocol for Real-Time Applications” http://tools.ietf.org/html/rfc3550

[27] “RTP Profile for Audio and Video Conferences with Minimal Control” http://www.ietf.org/rfc/rfc3551.txt

[28] Handley, Mark, Van Jacobson, Colin Perkins. "SDP: Session Description Protocol (RFC 4566)". IETF. 2006-07

[29] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on circuits and system for video technology, Vol.13, No.7, 2003, pp.560-576.

[30] Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification", Redmond, WA, July 2004, JVT-L050.

[31] ITU-T, "Video codec for audiovisual services at px64 Kbits/s," ITU-T Rec. H.261 v1: Nov 1990, v2: Mar. 1993.

[32] ISO/IEC JTC 1, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s – Part 2: Video," ISO/IEC 11172 (MPEG-1), Nov. 1993.

[33] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2), Nov. 1994

[34] ITU-T, "Video coding for low bit rate communication," ITU-T Rec. H.263; v1: Nov. 1995, v2: Jan. 1998, v3: Nov. 2000.

[35] ISO/IEC JTC 1, "Coding of audio-visual objects – Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 Part 2), Jan. 1999

[36] Gary J. Sullivan, "Video Compression—From Concepts to the H.264/AVC Standard", Proceedings of the IEEE, Vol. 93, NO. 1, JANUARY 2005.

[37] http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm

[38] Mathias Wien, Heiko Schwarz, Tobias Oelbaum, "Performance Analysis of SVC", IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Scalable Video Coding, Vol. 17, No. 9, September 2007. pp. 1194-1203.

[39] Simple DirectMedia Layer. SDL-Library. http://www.libsdl.org/.

[40] P. Cesar & J. Vierinen, P. Vuorimaa, "Open graphical framework for interactive TV" Springer Netherlands, August 2006. pp 189-203.

[41] A. Hutanu, R. Paruchuri, D. Eiland, M. Liska, P. Holub, S.R. Thorpe, Yufeng Xin, "Uncompressed HD video for collaborative teaching — an experiment", Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference, Nov. 2007. pp 253-261.

[42] Pablo Cesar, Petri Vuorimaa, Juha Vierinen, “A graphics architecture for high-end interactive television terminals”, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), Volume 2, Issue 4,November 2006.

[43] M. Iqbal, X. H. Wang, D. Wertheim, and X. Zhou, “SwanMesh: A Multicast enabled Dual-Radio Wireless Mesh Network for Emergency and disaster recovery Services”, J. of Communications, 2009, Vol.4, Issue 5.

[44] http://www.linuxtv.org/downloads/video4linux/API/V4L1_API.html

[45] http://research.edm.uhasselt.be/~jori/page/index.php?n=CS.Jrtplib

[46] Yoshimura, T. Ohya, T. Kawahara, T. Etoh, M. “Rate and robustness control with RTP monitoring agent for mobile multimedia streaming”, Communications, 2002. ICC 2002. IEEE International Conference, 2002. pp 2513-2517.

[47] Abdul H. Sadka, “Compressed Video Communication”, John Wiley & Sons, Ltd. 2002. Chapter 3.

[48] X. H. Wang, M. Iqbal, and X. Zhou, “Design and Development of a Dual Radio Wireless Mesh Network for Healthcare”, 5th Intern. Conf. Information Tech. and App. in Biomedicine (ITAB 2008), 30-31 May, 2008, Shenzhen, China, pp. 300 – 304.