Development of Multimedia Telemedicine System Based on IPv6 Multicast

Hye Jung Chun

The Graduate School

Yonsei University

Department of Biomedical Engineering

Development of Multimedia Telemedicine System Based on IPv6 Multicast

A Master's Thesis

Submitted to the Department of Biomedical Engineering

and the Graduate School of Yonsei University

in partial fulfillment of the

Requirements for the degree of

Master of Science

Hye Jung Chun

January 2004

감사의 글

세월이 유수 같다는 말이 지난 2년처럼 실감 났던 적이 없었던 것 같습니다. 굉장한 각오와 다짐을 갖고 대학원에 입학한 것이 불과 얼마 전 일인 듯 한데 벌써 마쳐야 할 때라고 하니, 지난 시간 더 열심히 치열하게 생활하지 못한 아쉬움이 못내 남습니다. 부족함이 많은 저의 곁에서 도움을 주신 많은 분들 덕분에 이렇게 석사과정을 마칠 수 있었던 것에 감사하며 짧게 나마 글로라도 마음을 전하려 합니다.

본 논문이 결실을 맺기까지 여러 모로 부족한 저를 많은 배려와 가르침으로 이끌어주신 유선국 교수님께 깊은 감사를 드립니다. 또한 곁에서 항상 마음 써주시고 가르침을 주신 김남현 교수님, 김덕원 교수님, 서활 교수님, 박종철 교수님께도 감사 드립니다. 임상 실험할 때 많은 도움을 주신 응급의학과 박인철 선생님과 신경외과 김선호 선생님께도 감사의 마음을 전합니다.

처음에 연구실 생활을 하면서 이것저것 친절하게 가르쳐 주셨던 광민 선배, 정훈 선배와 얌전하지 않던 저를 귀엽게 봐주시고 마음 써주셨던 석명 선배, 동근 선배, 상우 선배에게도 너무 감사 드립니다. 특히 제가 학업에 어려움이 있을 때마다 함께 고민해주시고 이야기 들어주신 석명 선배에게 진심으로 감사 드립니다. 나중에 합류하신 공학원 멤버 경하 선배, 계동 오빠와 충기에게도 고마운 마음을 전하고, 선영이 언니, 준이 오빠, 김성림 박사님과 유영일 석사과정에게도 감사 드립니다. 후배로 들어와서 이래저래 잔소리 많이 들은 뺀질이 진호와 하영이에게도 미안함과 고마운 마음을 전하고, 지난 일년 저 때문에 돈과 몸이 많이 축난 호현이 오빠와 병수 오빠에게도 이루 말할 수 없는 고마운 마음을 전합니다. 소식에 어두운 저에게 항상 훌륭한 소식통이 되어준 기창 선배와 재성이 오빠, 선희에게도 고마운 마음을 전합니다.

어느덧 6년지기 친구들이 되어버린 징글징글한 기전 2&10반 98학번 동기들과, 회사 가서 자주 만나지는 못했지만 메신저로 출석체크 해주던 삼성맨들, 동옥이와 경화, 밤새고 실험하면서 울고 웃고 같이 했던 올해 최고의 친한척 한나에게도 사랑한다는 말과 고맙다는 말을 전하고 싶습니다. 논문 막바지에 밤샘하느라 정신 없는 후배에게 세상에 둘도 없이 맛있는 커피와 빵을 배달해준 곰 언선이 오빠, 한 학기 더 다니라는 말로 끝까지 채찍질 해 준 대학생활의 정신적 지주 지웅이 오빠에게도 고마운 마음을 전합니다. 가끔이라도 만나면 항상 내가

대단한 일을 하고 있는 양 위로해준 중·고등학교 동창들, 지혜, 희정이, 연진이, 혜정이에게도 고맙다는 말을 전합니다.

마지막으로 추운데 훈련소에서 고생하고 있을 동생 관용이와 공부하는 딸 항상 믿고 후원해주시는 사랑하는 엄마께 이 논문을 바치며, 이 곳에서 여러 분에게 배운 지식과 경험을 바탕으로 사회에서도 바르고 열심히 생활하도록 노력하겠습니다.

2004년 1월
천혜정 올림

Table of Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
ARP	Address Resolution Protocol
BSD	Berkeley Standard Distribution
CBT	Core Based Tree
CT	Computerized Tomography
DVMRP	Distance Vector Multicast Routing Protocol
ICMPv6	Internet Control Message Protocol Version 6
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IPSec	IP Security
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISP	Internet Service Provider
MBONE	Multicast Backbone

MLD	Multicast Listener Done
MOSPF	Multicast Extensions to Open Shortest Path
MPEG	Motion Picture Experts Group
MTU	Maximum Transfer Unit
PIM-DM	Protocol Independent Multicast – Dense Mode
PIM-SM	Protocol Independent Multicast – Sparse Mode
POTS	Plain Old Telephony System
QoS	Quality of Service
RFC	Request for Comments
RP	Rendezvous Point
RPF	Reverse Path Forwarding
TCP	Transmission Control Protocol
TTL	Time-To-Live
UDP	User Datagram Protocol
WAN	Wide Area Network

Abstract

Development of Multimedia Telemedicine System Based on IPv6 Multicast

Chun, Hye Jung
Dept. of Biomedical Engineering
The Graduate School
Yonsei University

The lack of IP address is a serious problem on the current IPv4 (Internet Protocol Version 4) based Internet. The current problem of insufficient address space can be solved by adopting IPv6 (Internet Protocol Version 6) which provides 3.4×10^{38} addresses by using a 128-bit address structure. Moreover IPv6 has many advanced features such as mobility, QoS (Quality of Service), built-in security, address auto-configuration, and plug-n-play.

Based on these new features of IPv6, different kinds of applications are being developed in various fields, and the development of the IPv6 has reached the level where it can be used in practical applications. Therefore the development of a telemedicine system supporting IPv6 is expected for the preparation of the next generation Internet. In this paper we introduce a telemedicine system based on IPv6 multicast. The system is composed of a high-quality video and a bio-signal transmission module. The patient data that are acquired from these two modules is transmitted to more than two receivers using IPv6 multicast. The operability of the system was tested in the native IPv6 network and the local IPv4 network of Yonsei Medical Center using tunneling. IPv6 multicast packets were successfully bypassed through theIPv6 network and the tunnel built in the IPv4 network. Using this multicast telemedicine system, a patient can receive a more prompt and exact treatment from the multi-consultation of more than two specialists. Also by using multicasting, high-bandwidth medical data can be efficiently exchanged in a limited amount of bandwidth.

Showing the possibility of telemedicine system for future network environment, this system can be the first step to the development of a new telemedicine system supporting other advanced features of IPv6.

Keywords: IPv6, Multicast, Multimedia, Telemedicine, Tunneling, High-quality Video, Real-time

Chapter 1

Introduction

Internet has been rapidly expanding its size since early 1990's with the advance of computers and communication technology. Numerous kinds of communication devices are developed, expecting to be connected to the Internet using their own allocated IP addresses. Such rapid growth of Internet has lead to an increased requirement for IP numbers causing the address shortage problem in today's IPv4 (Internet Protocol Version 4)-based Internet. To overcome the address space limitation of IPv4, IPv6 (Internet Protocol Version 6) is introduced, which provides 3.4×10^{38} addresses by using a 128-bit address structure. Other than the abundance in IP addresses, IPv6 provides an advanced packet-forwarding scheme called multicast. Using multicast, a host can send packetized data to a group of receivers without additional bandwidth consumption caused by the packet replication.

Moreover IPv6 has many advanced features such as mobility, QoS (Quality of Service), built-in security, and address auto-configuration. Based on these new features of IPv6, different kinds of applications are being developed in various fields such as an aeronautical telecommunications network using IPv6 [1], a video conferencing over IPv6 on the Linux platform [2], and the cellular mobile IPv6 [3]. Likewise the development of the IPv6 has reached the level where it can be used in practical applications [4], and therefore the development of a telemedicine system supporting IPv6 is expected for the preparation of the next generation Internet.

Many kinds of telemedicine system supporting multimedia data are developed with the advance of computers and communication technology. Some of the examples of multimedia telemedicine system are: 1) a web-based telemedicine system supporting the transmission and display of video and still image data of a patient [5] and 2) a multipurpose health care telemedicine system supporting those of bio-signals and still images [6]. The former system was tested on the Internet and the latter system was tested through GSM, satellite links, and POTS (Plain Old Telephony System). Likewise current researches on multimedia telemedicine systems focus on a one-to-one communication between a doctor and a patient in different kinds of network environment. However, due to the subdivision of medical specialties, an expert consultation of more than one medical doctor is required for a proper treatment of the patient. Especially, when under a serious emergency situation, the emergency medical technician has difficulty in taking care of the patient alone. The advice from corresponding subspecialists is needed for a pertinent treatment. Since all the needed subspecialists cannot be available in the emergency room all the time, a multimedia telemedicine system based on a multiple connection can be used effectively to realize the multi-consultation between the medical specialists.

In this paper, we present a multimedia telemedicine system using IPv6 multicast to provide the mutual consultation of the subspecialists. The system is implemented to enable the simultaneous sharing of the patient's multimedia data. Therefore the synchronization of the multimedia data is considered most important when designing the system. For the synchronization of the data, we minimized the data transmission delay by letting the user to choose appropriate data sources, compression methods, video frame rate, and video resolution. We designed experiments in the local test-bed network with three IPv6 end-hosts and a multicast router. Moreover, a clinical experiment was performed to show the efficacy of the system in the real medical environment. We set up the system in the emergency room of Yonsei Medical Center, the office of radiologist and surgeon to experiment the efficient communication of the three specialists in diagnosing the emergency patient.

Chapter 2

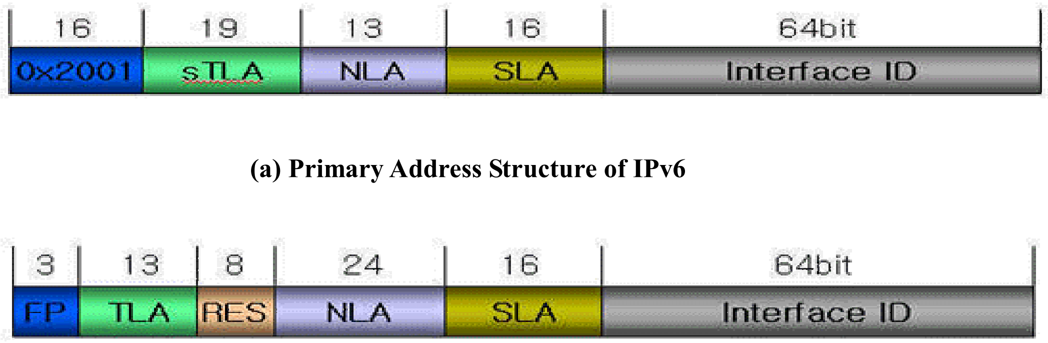
IPv6 Multicast

2.1 Features of IPv6

The lack of IP address is a serious problem in current IPv4 (Internet Protocol Version 4) based Internet. IPv4, which uses a 32-bit address structure, is capable of providing about 4.2 billion address spaces ideally, but it is estimated that the number of available addresses will be much smaller than expected due to an indiscreet assignment of classes (A, B, and C) in the early days of the Internet. However, with the development of network devices, which can be used for home networking, smart home appliances, and mobile communication, it is obvious that the supply of the IP addresses available in current Internet protocol will not be able to meet the demand. The current problem of insufficient address space can be solved by adopting IPv6 (Internet Protocol Version 6) which provides 3.4×10^{38} addresses by using a 128-bit address structure. Other than the abundance in IP address, IPv6 has several advanced features that can supplement the technical limitation of IPv4 when used in multimedia real-time application.

2.1.1 Address Structure

An address structure of IPv6 is shown in Fig 2.1.



(a) Primary Address Structure of IPv6

(b) Final Address Structure of IPv6

Figure 2.1 IPv6 Address Structure

- 0x2001: Fixed   - FP: Format Prefix (001)

- sTLA ID: Top Level Aggregator Identifier

- NLA ID: Next Level Aggregator Identifier

- SLA ID: Site level Aggregator Identifier

- RES: Reserved Interface

- ID: Host Address

At first, only the sub TLA IDs are going to be allocated to users with the first sixteen bits fixed to 0×2001 as shown in Figure 2.1(a). When the TLA registers take over 90% of the NLA ID space, TLA IDs are going to be allocated as shown in Figure 2.1(b). Currently, we are using 2001::/16 for the primary address within the country, and ETRI is holding the upper most identifier section, sTLA. Table 2.1 shows the number of available hosts per sections. As shown below even with the 16-bit end section SLA, it can provide more IP addresses than IPv4.

ID Section	TLA	NLA1	NLA2	SLA
Accommodating Number of Host Addresses	$2^{6+7+16+64}$	$2^{7+16+64}$	2^{16+64}	2^{64}

Table 2.1 The Number of Available Host Addresses in IPv6

The IPv6 addressing architecture document, RFC 2373[7], defines three different type of IPv6 addresses:

• Unicast Address - An identifier to a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.

• Anycast Address – An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the nearest one, according to the routing protocol's measure of distance).

• Multicast Address – An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

The multicast address identifies a group of nodes, and each of these nodes may belong to multiple multicast groups. The multicast address, as shown in Figure 2.2, begins with the Format Prefix 11111111 and includes three additional fields. The flag field contains four one-bit flags. The three most significant flag bits are reserved for future use and are initialized to zero. The fourth flag is called the T, or transient, bit. When T=0, the multicast address is a permanently assigned multicast address, assigned by the global Internet numbering authority. When T=1, a transient multicast address is indicated. The scope field is a four-bit field that is used to limit the scope of the multicast group. The group ID field identifies the multicast group, either permanent or transient, within the given scope.

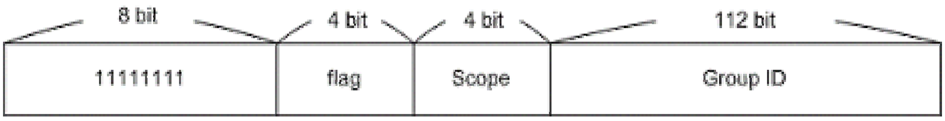


Figure 2.2 Multicast Address Structure

2.1.2 Advanced Features of IPv6

- New header format
- Efficient and hierarchical addressing and routing infrastructure
- Auto-configuration: Stateless and stateful address configuration
- Built-in security
- Better support for QoS
- New protocol for neighboring node interaction
- Extensibility

The IPv6 header reduced the packet-handling overhead by eliminating and making optional some of the IPv4 header fields (Figure 2.3). This is achieved by moving both non-essential fields and optional fields to extension headers that are placed after the IPv6 header. The streamlined IPv6 header is more efficiently processed at intermediate routers. IPv4 headers and IPv6 headers are not interoperable. Therefore a host or router must use an implementation of both IPv4 and IPv6 in order to recognize and process both of the header formats. The new IPv6 header is only twice as large as the IPv4 header, even though IPv6 addresses are four times as large as IPv4 addresses. New fields in the IPv6 header define how traffic is handled and identified. Traffic identification using a Flow Label field in the IPv6 header allows routers to identify and provide special handling for packets belonging to a flow, a series of packets between a source and destination. Because the traffic is identified in the IPv6 header, support for QoS can be achieved.

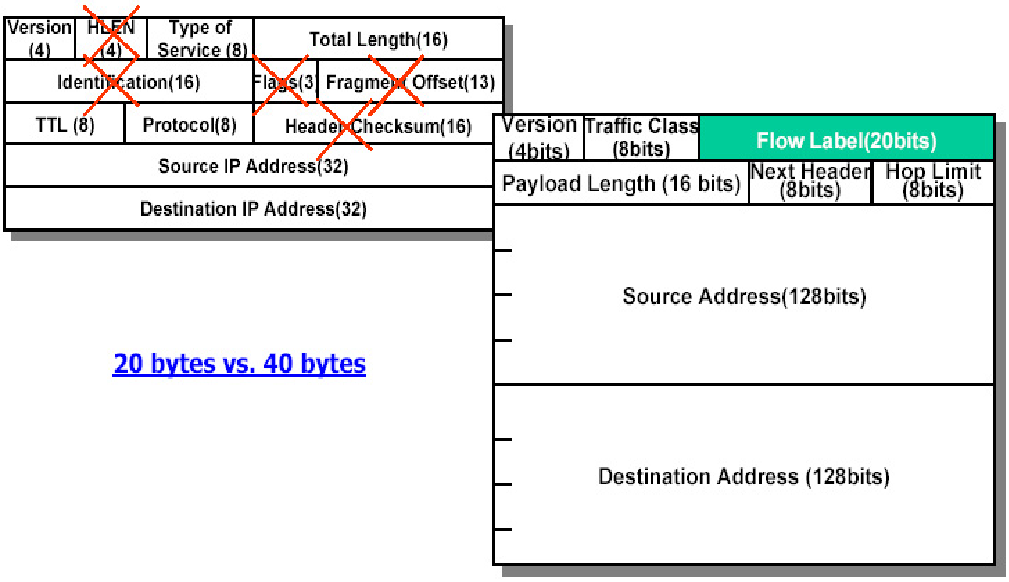
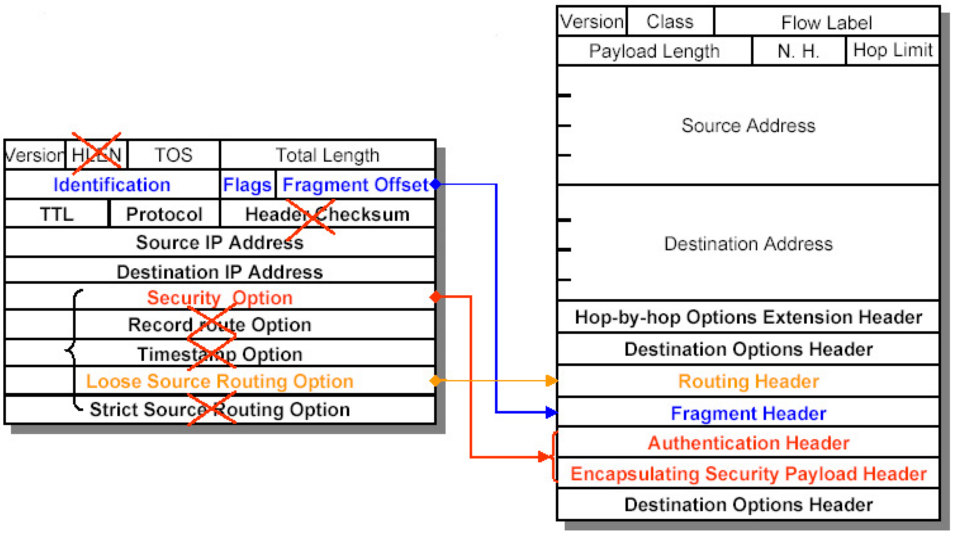


Figure 2.3 The Comparison between IPv4 and IPv6 Headers

IPv6 global addresses used on the IPv6 portion of the Internet are designed to create an efficient, hierarchical, and summarizable routing infrastructure that is based on the common occurrence of multiple levels of Internet service providers. On the IPv6 Internet, backbone routers have much smaller routing tables, corresponding to the routing infrastructure of global ISPs (Internet Service Provider). To simplify host configuration, IPv6 supports both stateful address configuration, such as address configuration in the presence of a DHCP server, and stateless address configuration (address configuration in the absence of a DHCP server). With stateless address configuration, hosts on a link automatically configure themselves with IPv6 addresses for the link (called link-local addresses) and with addresses derived from prefixes advertised by local routers. Even in the absence of a router, hosts on the same link can automatically configure themselves with link-local addresses and communicate without manual configuration. This specific feature can be efficient when considering that there will be a very large amount of IP addresses available in the future network.

Support for IPSec (IP Security) is an IPv6 protocol suite requirement. This requirement provides a standards-based solution for network security needs and promotes interoperability between different IPv6 implementations. The Neighbor Discovery protocol for IPv6 is a series of Internet Control Message Protocol for IPv6 (ICMPv6) messages that manage the interaction of neighboring nodes (nodes on the same link). Neighbor Discovery replaces the broadcast-based Address Resolution Protocol (ARP), ICMPv4 Router Discovery, and ICMPv4 Redirect messages with efficient multicast and unicast Neighbor Discovery messages.

IPv6 can easily be extended for new features by adding extension headers after the IPv6 header. Unlike options in the IPv4 header, which can only support 40 bytes of options, the size of IPv6 extension headers is only constrained by the size of the IPv6 packet (Figure 2.4).



(a) (b)

Figure 2.4 Extension Headers of IPv6

(a) Options in the IPv4 Header (b) Extensions in the IPv6 Header

2.2 Multicast

2.2.1 IP Multicast

IP multicast is a data communication mechanism that sends out the same kind of data to a group of hosts simultaneously without any packet replication. The group of receivers is identified by a single IP multicast address, and the multicast datagram is delivered to all of the group members using its destination multicast address. A host can join and leave the group at any time, and there is no restriction on the location or the number of members in a group. Also, a host may be a member of more than one group at a time. Only one copy of a multicast message will be sent over a link in the network, and the message will be duplicated only where the path is diverged at a multicast router (Figure 2.5). Therefore IP multicast can be performed under reduced end-to-end bandwidth consumption. For applications that need to deliver data to a large number of receivers, IP multicast can use its scalability to transmit the data efficiently.

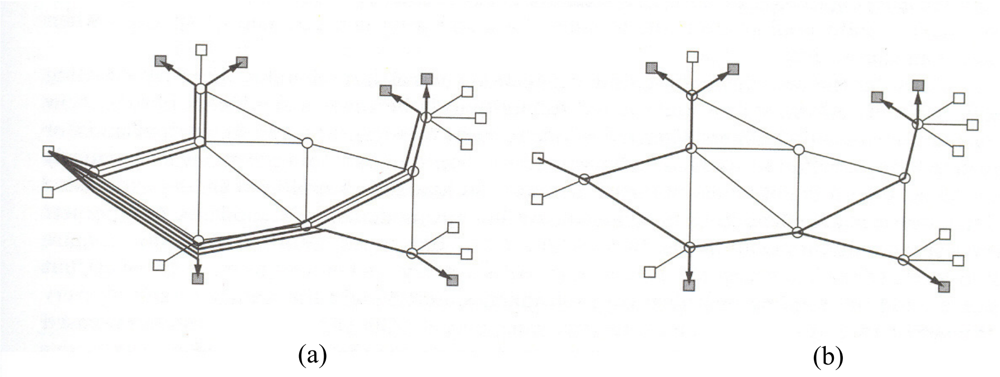


Figure 2.5 The Scalability of Multicast (b) over Unicast (a)

Likewise, multicast scalability plays an important role in transmitting data to multiple users, but there is a trade off of complexity. The complexity comes from the fact that it is only a delivery mechanism. At the transport layer, multicast only works in UDP (User Datagram Protocol) which does not have congestion control or reliable delivery mechanism. Therefore if you are to transmit and receive the data without any errors, you have to design a system considering many different kinds of congestions and errors that can come across. Sending data in UDP help reducing the processing and transmitting time, but on the other hand it does not guarantee reliability in transmission. Nevertheless since multicast has its strong point in scalability as shown in Figure 2.5, it can be used for high-bandwidth consuming applications. IP multicast has following general features:

1) Group address

Each multicast group is a unique class-D address. A few IP multicast addresses are permanently assigned by the Internet authority, and correspond to groups that always exist even if they have no current members. Other addresses are temporary and available for private use.

2) Number of groups

IP can provide addresses up to 2^{28} simultaneous multicast groups. Thus the number of groups is limited by practical constraints on routing table size rather than addressing capacity.

3) Dynamic group membership

A host can join and leave an IP multicast group at any time, and a host may be a member of an arbitrary number of multicast groups.

4) Use of hardware

If the underlying network hardware supports multicast, IP uses hardware-supported multicast to send data. If not it uses broadcast or unicast to deliver the data.

5) Inter-network forwarding

Since the member of an IP multicast group can be attached to multiple physical networks, certain kind of routers supporting multicast are required in order to forward the multicast datagram.

6) Delivery semantics

Since IP multicast uses the same best-effort delivery semantics, the multicast datagrams can be lost, delayed, or delivered without order.

7) Membership and transmission

An arbitrary host may send datagrams to any multicast group. The group membership is only used to determine whether the host received datagrams sent to the group.

2.2.2 Multicast in IPv4

Multicasting can be implemented in IPv4 satisfying the characteristics that are mentioned in the previous section. In the following, we will introduce the C sockets application programming interface (API) for multicast in IPv4. In addition, a concept called Time-To-Live (TTL), which is used in setting the multicast scope, will be introduced.

1) UDP Socket

Sockets are application programming interfaces (API) that provide basic function calls to send data across a network [8]. Originally developed as part of the Berkeley Standard Distribution (BSD) of Unix, sockets have been expanded to be included in almost all Unix variants, Microsoft Windows, MS-DOS, Apple MacOS, OS2, and newer high-level languages such as Java and C#. The most common use of the sockets API is to send a TCP data stream from one host to another. TCP has important features like connections, multiplexing/demultiplexing, in-order delivery, reliability, and congestion control on top of the IP layer. However, multicast transmission is integrated using the User Datagram Protocol (UDP). UDP only provides the multiplexing/demultiplexing feature of the TCP. Therefore no in-order delivery, reliability, and congestion control is guaranteed in a UDP packet transmission. Unlike TCP, no former establishment of connection is required in UDP. To receive UDP packets, the socket of the receiver's terminal must bind to a certain port number that receives the transmitted packets. Then the receiver's socket will listen to the port number waiting for the transmitted data to arrive. If a packet is lost, the receiver may not be able to detect it or be able to alert the sender to resend the missing data. Any detection or retransmission of lost packets is strictly up to the application level to deal with. In spite of these flaws UDP is more appropriate than TCP in some kinds of applications that requires real-time multimedia streaming. These applications require a constant stream of packets, and the lost packets do not affect the proper operation of the application. Moreover, since TCP has all the overhead of establishing and closing connections, UDP can be significantly faster. This is even more important for multicast saving the time for establishing a connection to every receiver. In order to send multicast packets using UDP socket, following steps must be done in the application level.

Step1	Create a socket.
Step2	Optionally set the scope for the packets.
Step3	Send the data on the socket.
Step4	Close the socket.

Table 2.2 Sending Multicast Packets

The following steps are for receiving multicast packets using UDP socket.

Step1	Create a socket.
Step2	Optionally set the port reuse socket option.
Step3	Bind to the socket.
Step4	Join the multicast group.
Step5	Receive multicast data.
Step6	Drop the multicast group.
Step7	Close the socket.

Table 2.3 Receiving Multicast Packets

Some of the options needed for multicasting in IPv4 will be described in the next subsections: scoping, specifying a multicast address, joining/leaving the multicast group.

2) Specifying a Multicast Address in IPv4

After creating a UDP socket, a destination address structure needs to be prepared in order to send the multicast packets. In IPv4 multicast addresses are called Class D IP addresses and are in the specific range of 224.0.0.0 to 239.255.255.255. Some of the addresses in this range are reserved for special use [8]. Choosing the right address depends on several factors, including the expected scope of the group, the type of multicast service to use, and whether the application is just an experiment or a production tool. Setting a destination address on a sender's UDP socket is done using a sockadder data structure consisting of an address family field, a port field, and an address field.

The multicast address space is reserved, assigned for specific applications, or statistically assigned to autonomous systems and subnets. The Internet Engineering Task Force (IETF) developed a request for comments (RFC) to provide guidance how to divide multicast address space. Table 2.4 shows the basic groups of multicast addresses.

Address Range	**CIDR Block**	**Description**
224.0.0.0 ~ 224.0.0.255	(224.0.0.0/24)	Local Network Control Block
224.0.1.0 ~ 224.0.1.255	(224.0.1.0/24)	Internetwork Control Block
224.0.2.0 ~ 224.0.255.0	·	Ad-hoc Block
224.1.0.0 ~ 224.1.255.255	(224.1.0.0/16)	ST Multicast Groups
224.2.0.0 ~ 224.2.255.255	(224.2.0.0/16)	SDP/SAP Block

224.3.0.0 ~ 231.255.255.255	·	RESERVED
232.0.0.0 ~ 232.255.255.255	(232.0.0.0/8)	Source Specific Multicast Block
233.0.0.0 ~ 233.255.255.255	(233.0.0.0/8)	GLOP Block
234.0.0.0 ~ 238.255.255.255	·	RESERVED
239.0.0.0 ~ 239.255.255.255	(239.0.0.0/8)	Administratively Scoped Block

Table 2.4 Multicast Address Space Categories

3) TTL: Setting Multicast Scope

The Time-To-Live (TTL) field of a transmitted multicast traffic is used when controlling the scope of the multicast packets. When a multicast-enabled router receives a multicast packet, it examines the TTL of the packet, which can be understood as a hop count of the delivery. If the TTL has a value of 1 it will not be forwarded, and if it is greater than 1 the router will decrement the TTL field in the packet by 1 and forward it to the next router. Therefore if a TTL value is set to 1 in the beginning, the multicast packets will never be router off the local subnet. By setting the TTL value in sender's UDP socket we can prevent the packet traffic from reaching unintended areas. TTL valid values range from 0 to 255 (a value 0 restricts to the local host). Other than setting the TTL value of a packet, we can control the scope of a multicast using different types of Class D addresses.

4) Joining/Leaving Multicast Membership

In order to receive multicast packets that are sent to a specific multicast group, a host needs to join the group using C sockets API called setsockopt(). An add membership is requested with the socket option IP_ADD_MEMBERSHIP. When a host makes an add membership request, the IP stack begins to pass packets heard on that multicast group up to the transport layer (UDP in this case) and on the application. Then, an Internet Group Management Protocol (IGMP) message is sent to the routers on the local subnet indicating the host wants to receive packets on that particular multicast address. If the router on the local subnet is multicast enabled and are connected to other networks are also multicast enabled, the IGMP join message will be transformed into a multicast routing protocol join and will be propagated throughout the network. As a result, a tree is formed from the source application to the receiving application assuming that TTL is sufficient to cover the router hops from the source to the receiver. The same socket can listen on multiple multicast addresses at the same time. Simply adding additional setsockopt() calls to IP_ADD_MEMBERSHIP with different multicast addresses will work cumulatively, allowing the socket to receive packets intended for any of the specified addresses.

When an application want to stop receiving packets from a particular multicast address, a call to setsockopt() with IP_DROP_MEMBERSHIP will cause the IP stack to stop forwarding packets received on that multicast address to the application. If that application is the only application on the host to be requesting the traffic, this will also generate an IGMP leave message from the host to the subnet router. If the host is the last host requesting the membership of that multicast group on the subnet, the Internet forwarding tree will be modified to "prune" the connected branches. Closing a socket or terminating a program with multicast membership will also generate a drop request automatically.

2.2.3 IPv6 Multicast

In IPv4, multicasting was introduced as an extension of the basic specification; hence IPv4 nodes do not necessarily support multicasting. On the other hand, specifications of IPv6 require that all IPv6 nodes support multicasting. Differences of multicasting between IPv4 and IPv6 require several original approaches for the implementation including handling of multicast interfaces, using scoped addresses, and lack of multicast tunnel.

Traditional implementations of IPv4 multicasting use unicast address to identify a network interface. However, such an approach is not suitable for IPv6 since an IPv6-capable node may assign multiple addresses on a single interface, which tends to cause a configuration mismatch. Also a link-local address is not always unique within a node. Consequently it may not identify a single interface. A user must specify the interface index as well as the address in such a case. Therefore in IPv6 uses a specified index to identify a single interface.

IPv6 explicitly limits the scope of a multicast address by using a fixed address field, whereas the scope was specified using TTL of a multicast packet in IPv4. In order to improve the routing scalability of IP multicast, an additional field called scope is being used in IPv6 multicast address (Figure 2.2). It is a 4-bit field that specifies the range of a multicast group. The range of multicast group according to the value of the scope field is shown in Table 2.5.

Scope Value	Range of the Group	Scope Value	Range of the Group
0	Reserved	8	Organization Local
1	Interface Local	9	Not Allocated
2	Link Local	A	Not Allocated
3	Subnet Local	B	Community Local
4	Administrator Local	C	Not Allocated
5	Site Local	D	Not Allocated
6	Not Allocated	E	Global
7	Not Allocated	F	Reserved

Table 2.5 Range of a Multicast Group according to the Scope Field

For the multicast group management, IP multicast uses IGMP (Internet Group Management Protocol) while IPv6 multicast uses newly introduced MLD (Multicast Listener Discovery). Multicast Listener Discovery (MLD) is the IPv6 equivalent of Internet Group Management Protocol version 2 (IGMPv2) for IPv4. MLD is a set of messages exchanged by routers and nodes, enabling routers to discover the set of multicast addresses for which there are listening nodes for each attached interface. Like IGMPv2, MLD only discovers the list of multicast addresses for which there is at least one listener, not the list of individual multicast listeners for each multicast address. Unlike IGMPv2, MLD uses ICMPv6 messages instead of defining its own message structure. All MLD messages are ICMPv6 messages types 130, 131, and 132. The three types of MLD messages are:

1) Multicast Listener Query

Multicast Listener Query is used by a router to query a link for multicast listeners. There are two types of Multicast Listener Query messages: The General Query and the Multicast-Address-Specific Query. The General Query is used to query for multicast listeners of all multicast addresses. The Multicast-Address-Specific Query is used to query for multicast listeners of a specific multicast address. The two message types are distinguished by the multicast destination address in the IPv6 header and a multicast address within the Multicast Listener Query message.

2) Multicast Listener Report

Multicast Listener Report is used by a multicast listener to either report interest in receiving multicast traffic for a specific multicast address or to respond to a Multicast Listener Query.

3) Multicast Listener Done

Multicast Listener Done is used by a multicast listener to report that it is no longer interested in receiving multicast traffic for a specific multicast address.

2.3 Multicast Routing Protocol

The multicast traffic is transmitted from the source to the group of receivers via a spanning tree that connects all the hosts in a group. To construct these multicast-spanning trees, different kinds of multicast routing protocols are being used, and once a tree is constructed, all multicast traffics are distributed over it. The multicast routing protocols generally follow one of the two basic approaches depending on the expected number of multicast group members. The first approach is based on the assumption that the multicast group members are densely distributed throughout the network and that the bandwidth is plentiful. So-called ‘dense-mode’ multicast routing protocols rely on a technique called flooding to propagate the information to all network routers. The dense-mode routing protocols include Distance Vector Multicast Routing Protocol (DVMRP) [9], Multicast Extensions to Open Shortest Path First (MOSPF) [10], and Protocol-Independent Multicast – Dense Mode (PIM-DM) [11].

The second approach of a multicast routing protocol assumes that the multicast group members are sparsely distributed throughout the network and that the bandwidth is not widely available enough. In this case, flooding would unnecessarily waste network bandwidth and hence could cause serious performance degradation. Therefore, the sparse-mode protocols are required to rely on more selective techniques to set up and maintain multicast tress. The sparse-mode routing protocols include Core Based Trees (CBT) [12] and Protocol-Independent Multicast – Sparse Mode (PIM-SM) [13][14].

2.3.1 Distance Vector Multicast Routing Protocol (DVMRP)

The original multicast routing protocol, DVMRP [9], creates multicast trees using a technique called broadcast-and-prune. Because of the way the tree is constructed by DVMRP, it is called a reverse shortest path tree.

First, the source broadcasts each packet on its local network. An attached router receives the packet and sends it on all outgoing interfaces. Second, each router that receives a packet performs a reverse path forwarding (RPF) check. That is, each router checks to see if the incoming interface on which a multicast packet is received is the interface the router would use as an outgoing interface to reach the source. In this way, all packets received on the proper interface are only forwarded on all outgoing interfaces. All others are discarded. Third, a packet will reach a router eventually with some number of attached hosts. This leaf router will check to see if it knows of any group members on any of its attached subnets. A router discovers the existence of group members by periodically issuing Internet Group Management Protocol [15][16] queries. If there are members, the leaf router forwards the multicast packet on the subnet. Otherwise, the leaf router will send a prune message toward the source on the RPF interface. If prune messages are received on all interfaces except the RPF interface, the router will send a prune message of its own toward the source. DVMRP has been widely used on the MBONE (Multicast Backbone).

2.3.2 Multicast Extensions to Open Shortest Path First (MOSPF)

MOSPF [9] uses the Open Shortest Path First (OSPF) protocol to provide multicast.

Basically, MOSPF routers flood an OSPF area with information about group receivers. This allows all MOSPF routers in an area have the same view if group membership. This "link-state" information is used to construct multicast distribution trees. In this same way that each OSPF router for independently construct the shortest-path tree or each source and group. MOSPF uses the Dijkstra algorithm to compute the shortest-path tree. While group membership reports are flooded throughout the OSPF area, data is not. To reduce the number of calculations and to spread the calculation out somewhat, a router only makes this calculation when it receives the first datagram in a stream.

2.3.3 Core Based Tree (CBT)

CBT [12] uses the basic sparse mode paradigm to create a single shared tree used by all sources. The tree us rooted at a core. All sources send their data to the core, and all receivers send explicit join messages to the core. There are two differences between CBT and PIM-SM. First, CBT uses only a shared tree, and is not designed to use shortest path trees. Second, CBT uses bi-directional shared trees, but PIM-SM uses unidirectional shared-trees. Bi-directional shared trees involve slightly more complexity, but are more efficient when packets traveling from a source to the core cross branches of the multicast tree. In this case, instead of only sending traffic "up" to the core, packets can also be sent "down" the tree.

2.3.4 Protocol Independent Multicast – Dense Mode (PIM-DM)

PIM-DM [11] is very similar to DVMRP. There are only tow major differences. The first is that PIM (both dense mod and sparse mode) uses the unicast routing table to perform RPF checks. While DVMRP maintains its own routing table, PIM uses whatever unicast routing table is available. PIM simply requires the unicast routing table to exist, and thus is independent of the algorithm used to build it. The second difference between PIM-DM and DVMRP is that DVMRP tries to avoid sending unnecessary packets to neighbors who will then generate prune messages based on a failed RPF check. The set of outgoing interfaces built by a given DVMRP router will include only those downstream routers that use the given router to reach the source. PIM-DM avoids this complexity, but the trade-off is that packets are forwarded on all outgoing interfaces. Unnecessary packets are often forwarded to routers, which must then generate prune messages because of the resulting RPF failure.

2.3.5 Protocol Independent Multicast – Sparse Mod (PIM-SM)

PIM-SM [13][14], is much more widely used than CBT. It is similar to PIM-DM in that routing decisions are based on whatever underlying unicast routing table exists, but the tree construction mechanism is quite different. PIM-SM's tree construction algorithm is actually more similar to that used by CBT than to that used by PIM-DM. PIM-SM constructs a multicast distribution tree around a router called a rendezvous point (RP). This rendezvous point plays the same role as the core in the CVT protocol; receivers "meet" new sources at this rendezvous point. However, PIM-SM is a more flexible protocol than CBT. While CBT with trees are always group-shared tree, with PIM-SM an individual receiver may choose to construct either a group-shared tree or a shortest-path tree. RP discovery is done using a bootstrap protocol. The basic function of the bootstrap protocol, in addition to RP discovery, is to provide robustness in case of RP failure. The bootstrap protocol includes mechanisms to select an alternate RP if the primary RP goes down. Receivers send explicit join messages to the RP. Forwarding state is created in each router along the path from the receiver to the RP. A singled shared tree, rooted at the RP, is formed for each group. As with other multicast protocols, the tree is a reverse shortest path tree – join message follow a reverse oath from receivers to the RP. Each source sends multicast data packets, encapsulated in unicast packets to the RP. When an RP receives one of these register packets, a number of actions are possible. First, if the RP has forwarding state for the group, the encapsulation is stripped off the packet, and it is sent on the shared tree. However, if the RP does not have forwarding state for the group, it sends a register-stop message to the RP. This avoids wasting bandwidth between the source and the RP. Second, the RP may wish to send a join message toward the source. By establishing multicast forwarding state between the source and the RP, the RP can receive the source's traffic as multicast and avoid the overhead of encapsulation.

Chapter 3

Construction of IPv6 Multicast Network

In order to form a multicast test network based on IPv6, we construct an IPv6 multicast router to run MLD (Multicast Listener Done) and a multicast routing protocol such as PIM-DM or PIM-SM. Since CISCO commercial network routers do not support IPv6 multicast routing protocols, we used PC based routers equipped with FreeBSD. In this paper we constructed two different types of IPv6 multicast test networks: a native IPv6 multicast network with PC-based routers and ATM network, a tunnel-based IPv6 multicast network with configured tunnels and PC-based routers.

3.1 Native IPv6 Multicast Network

A native IPv6 network is a network where there is only IPv6 traffic with IPv6 routing protocols. The most common way to construct a native IPv6 network is to use Ethernet for LAN (Local Area Network) and to use ATM as WAN (Wide Area Network). We constructed a native IPv6 multicast test network according to this basic scheme as shown in Figure 3.1.

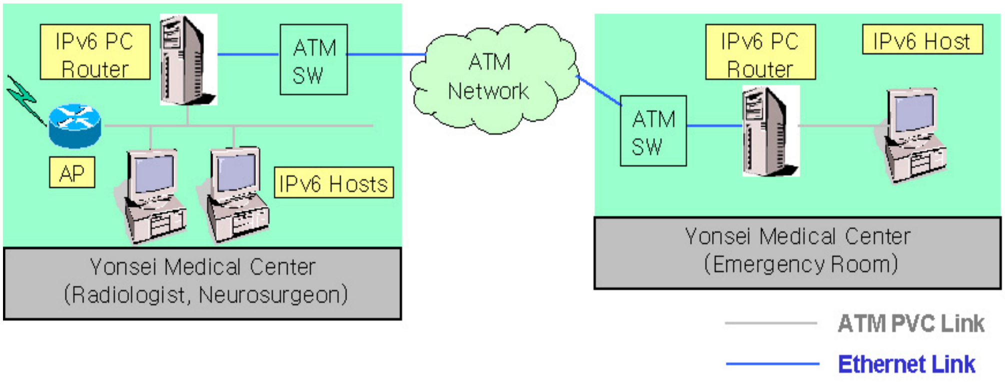


Figure 3.1 Native IPv6 Multicast Network using ATM Network

As shown in Figure 3.1, the native IPv6 multicast network consists of two subnets each having a PC-based IPv6 router equipped with FreeBSD OS (Operating System) to support multicast routing protocols. Each router is connected to the ATM switch using ATM PVC link, and IPv6 hosts are connected to the router using Ethernet link. To enable such link connections we installed an ATM interface an Efficient Networks ENI-155Mbps ATM Card (ENI) in FreeBSD routers, and formed a PVC link between the router and the ATM switch [17]. Finally if only the ATM cells with IPv6 datagrams are passing through the ATM network, it is called native IPv6 multicast network. An IPv6 host which acts as a multicast packet source is located in the emergency room of Yonsei Medical Center, and other two hosts in the radiologist's office and in the neurosurgeon's office are receiving multicast packets transmitted through the PC-based multicast router.

3.2 IPv4/IPv6 Heterogeneous Network

In spite of the benefits derived from the new Internet protocol, neither corporate internetworks nor the global Internet will immediately move from IPv4 to IPv6. Instead of being upgraded from IPv4 to IPv6 in the immediate future, most internetworks will become heterogeneous, with various routers and hosts (Figure 3.2). Therefore it is necessary to develop strategies for IPv4 and IPv6 to coexist until such time as IPv6 becomes the preferred option. We can consider three transition mechanisms that enable packet exchange between IPv4 and IPv6: IPv4/IPv6 dual stack for hosts or routers, IPv4/IPv6 translation technique for gateways, and IPv6-in-IPv4 tunneling through the network [18].

In this paper, we used IPv6-in-IPv4 tunneling method as the transition mechanism in an IPv4/IPv6 heterogeneous network. Tunneling is a process whereby information from one protocol is encapsulated inside the frame or packet of other architecture, thus enabling the original data to be carried over the second architecture. The tunneling method for IPv4/IPv6 is suggested to enable an existing IPv4 infrastructure to carry IPv6 packets by encapsulating the IPv6 information inside IPv4 datagrams.

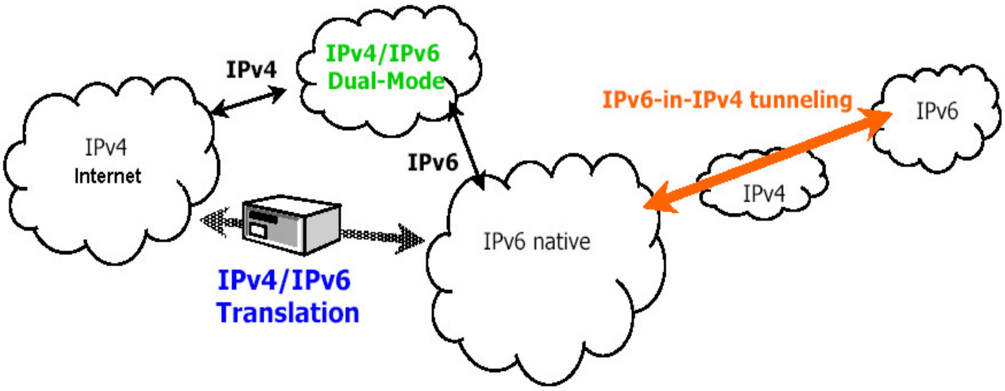


Figure 3.2 IPv4/IPv6 Heterogeneous Networks

The encapsulation process is illustrated in Figure 3.3. As shown in the figure, the resulting IPv4 datagram contains both an IPv4 header and an IPv6 header with all of the upper layer information such as the TCP/UDP header, application data, and etc. The reverse process, decapsulation, is illustrated in Figure 3.4. In this case, the IPv6 header is removed, leaving only the IPv6 packet. The tunneling process involves three steps: encapsulation, decapsulation, and tunnel management. At the encapsulation node (tunnel entry point), the IPv4 header is created and the encapsulated packets are transmitted. At the decapsulation node (tunnel exit point), the IPv4 header is removed and the IPv6 packet is processed. In addition, the encapsulation node may maintain configuration information regarding the tunnels that are established, such as the maximum transfer unit (MTU) size that is supported in the tunnel.

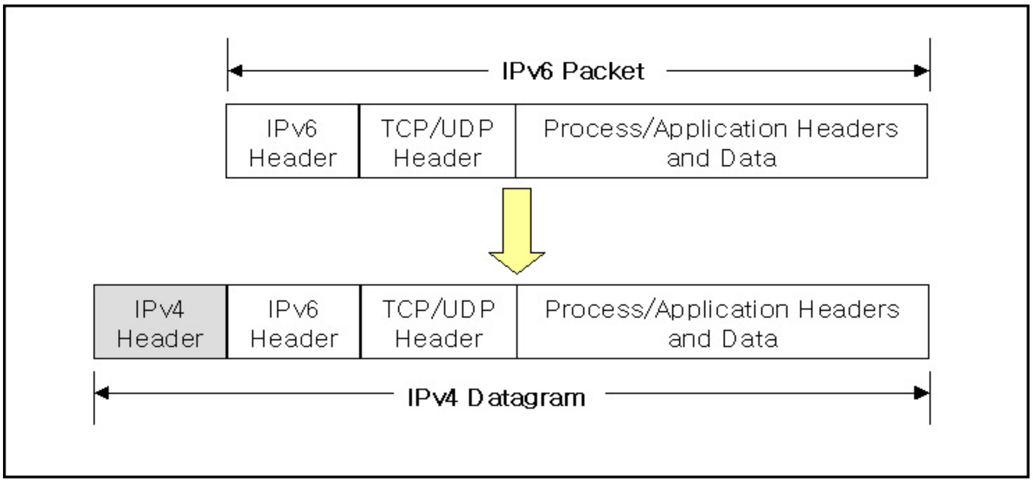


Figure 3.3 Encapsulating IPv6 in IPv4

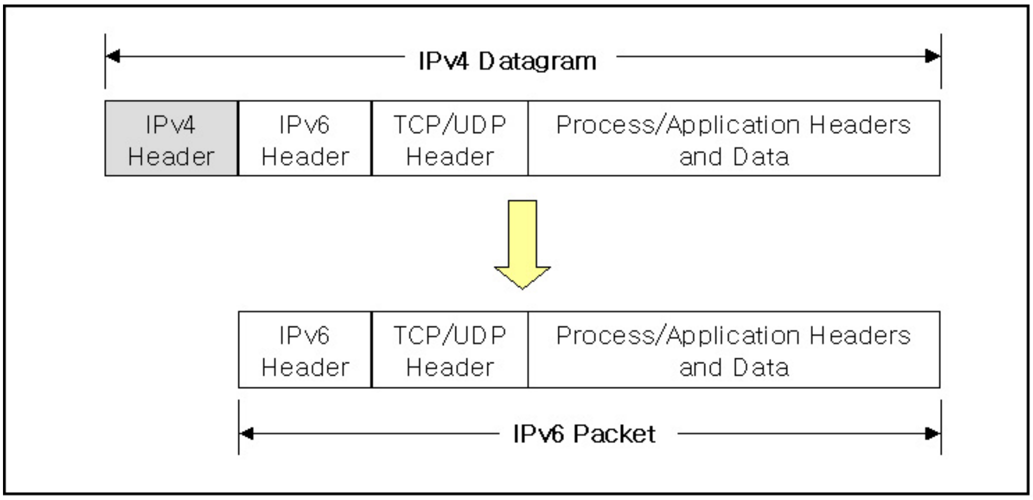
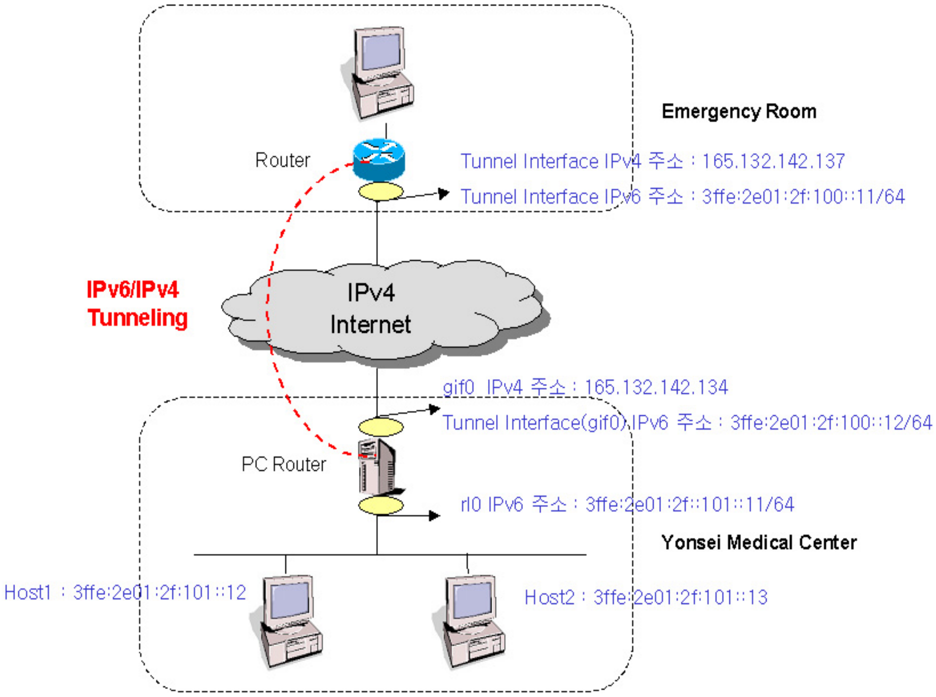


Figure 3.4 Decapsulating IPv6 from IPv4

For a tunnel to operate properly, addresses of both the tunnel endpoint and the packet's destination must be known, not necessarily the same. When IPv6-over-IPv4 tunneling is used in multicasting, the IPv4 tunnel endpoint address must be determined using Neighbor Discovery. It does not require any address configuration or the use of IPv4-compatible addresses. However, the existing IPv4 infrastructure must support multicasting. The motivation for this multicast tunneling method is to allow isolated IPv6 hosts located on a physical link that has not directly connected to IPv6 router, to become fully functional IPv6 hosts by using an IPv4 multicast domain as their virtual local link.



Emergency Room
Router
Tunnel Interface IPv4 주소 : 165.132.142.137
Tunnel Interface IPv6 주소 : 3ffe:2e01:2f:100::11/64
IPv6/IPv4
Tunneling
IPv4
Internet
gif0 IPv4 주소 : 165.132.142.134
Tunnel Interface(gif0) IPv6 주소 : 3ffe:2e01:2f:100::12/64
PC Router
rl0 IPv6 주소 : 3ffe:2e01:2f::101::11/64
Yonsei Medical Center
Host1 : 3ffe:2e01:2f:101::12
Host2 : 3ffe:2e01:2f:101::13

Figure 3.5 Multicast Test Network using IPv4/IPv6 Tunneling

Chapter 4

System Design

4.1 Hardware Configuration

The system consists of three end terminals: one patient's terminal and two specialist's terminal. Pentium-IV with 512 Mbytes RAM and 2.4 GHz clock is used as the end terminal computers. In the patient's terminal (Figure 4.1) the dedicated device (KTMED Co.), specially designed to acquire bio-signals from the patient monitor, transmits measured patient data in RF signal and the terminal computer receives the data using an RF receiver and interface with the PC using RS-232 serial interface. It samples bio-signals (ECG, BP, respiration, and SpO_2) with 300 Hz and 12 bits resolution, and acquires the text string of other bio-data every 30 seconds: SpO_2 value, temperature, systolic pressure, diastolic pressure, and heart rate. The high-quality video of the patient is acquired by the Canon VCC-4 high quality video camera, and they were compressed using MPEG-2 and MPEG-4 to be transmitted to the other side of the system. Also we prepared a multicast enabled PC router equipped with FreeBSD 4.3 (Figure 4.2). In the three end systems, Microsoft IPv6 Technology Preview for Windows 2000 is loaded to be able to use both of IPv4 and IPv6. The LAN (Local Area Network) card (100 Mbps Ethernet) through PCI interface made connections between the three end terminals.

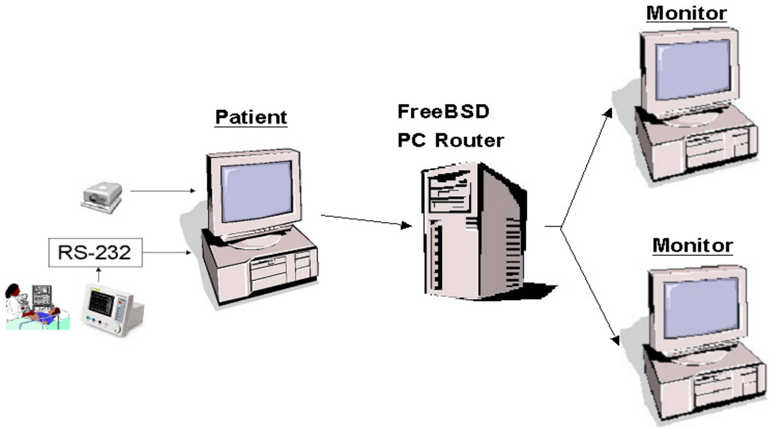
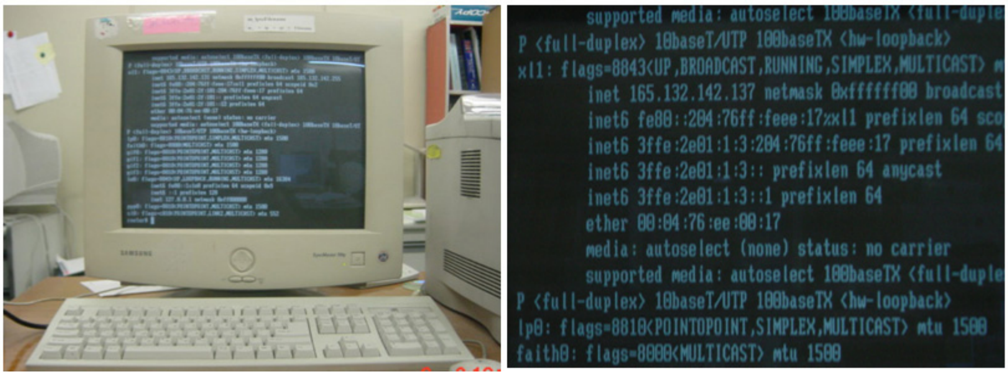


Figure 4.1 Hardware Architecture of the System



```
        supported media: autoselect 100baseTX <full-duple
P <full-duplex> 10baseT/UTP 100baseTX <hw-loopback>
xl1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> m
        inet 165.132.142.137 netmask 0xffffff00 broadcast
        inet6 fe80::204:76ff:feee:17%xl1 prefixlen 64 sco
        inet6 3ffe:2e01:1:3:204:76ff:feee:17 prefixlen 64
        inet6 3ffe:2e01:1:3:: prefixlen 64 anycast
        inet6 3ffe:2e01:1:3::1 prefixlen 64
        ether 00:04:76:ee:00:17
        media: autoselect (none) status: no carrier
        supported media: autoselect 100baseTX <full-duple
P <full-duplex> 10baseT/UTP 100baseTX <hw-loopback>
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
faith0: flags=8000<MULTICAST> mtu 1500
```

Figure 4.2 PC Router Set Up

4.2 Software Architecture

4.2.1 Multimedia Module Design

The user interface of the designed system is as shown in Figure 4.3. It consists of two multimedia data modules: a high-quality video module, a biosignal transmission/reception module. Each module for multimedia-data acquisition uses a unique PC interface in order to obtain patient data from the dedicated devices. For optimum delivery of data packets, we used MPEG2/4 and DPCM for the compression of the video and the biosignal respectively.

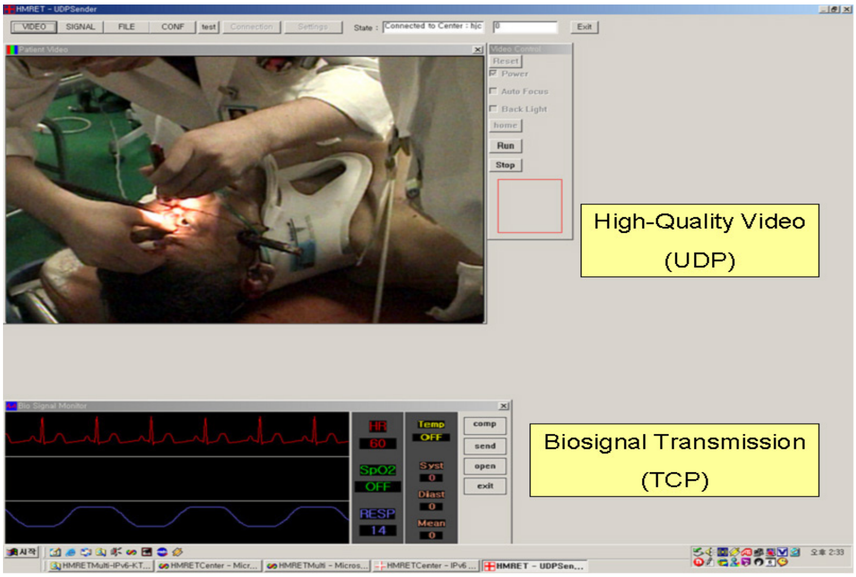


Figure 4.3 User Interface of the System

1) High-Quality Video Module

The block diagram of the data flow in high-quality video module is as shown in Figure 4.3. A video capture board (ATI Radeon 9000, ATI Technologies Co, Canada) with PCI interface acquires the motion video data of a patient. These video data is compressed using MPEG-2 and MEPG-4. Data acquisition/display/compression parts are implemented using Microsoft DirectShow filters.

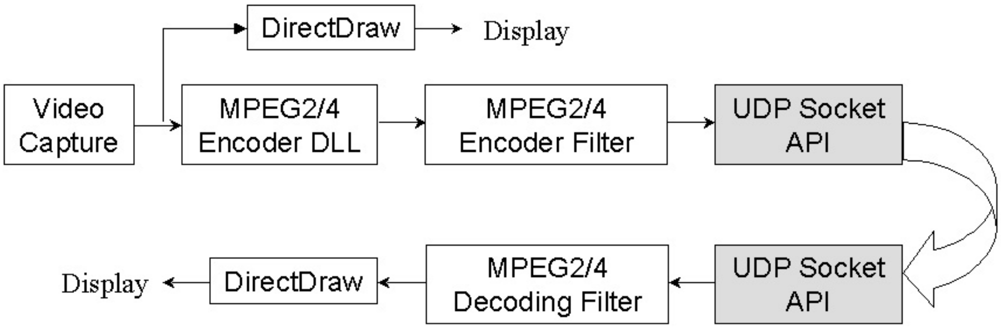


Figure 4.4 Data Flow of High-Quality Video Module

2) Biosignal Module

The dedicated device (KTMED Co.), specially designed to acquire bio-signals from the patient monitor, transmits measured patient data using wireless RF transmitter. The RF receiver converts analog data to digitized data and transfers them to the terminal computer through RS-232 serial interface. Then the Biological Signal Manager (BSM) selects one of the biosignals obtained from the patient monitor and encodes it using DPCM as the compression method. With TCP socket API the biological signal data is transmitted through the network (Figure 4.4). Unlike the high-quality video, we used TCP when transmitting/receiving biosignal since when delivered using UDP the data can have too much error or packet loss. Considering the bandwidth needed to transmit biosignal, which is comparatively small to that of the high-quality video, using TCP for the signal data communication will not affect the system performance.

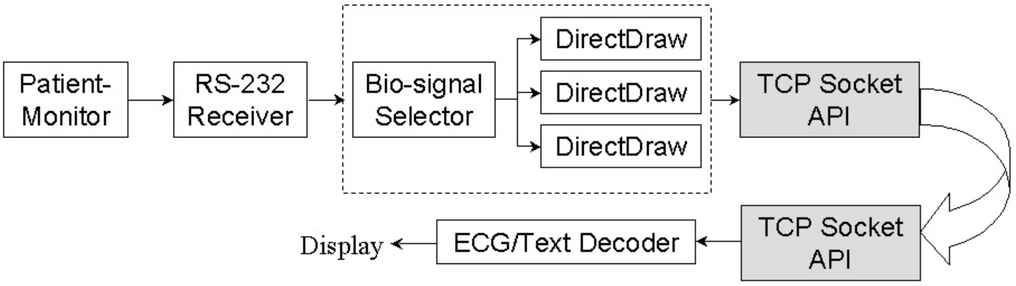


Figure 4.5 Data Flow of Biosignal Transmission Module

4.2.2 Socket Communication Module

The software configuration for the socket communication focuses on the exchange of patient data using IPv6 multicast. Windows Socket 2.2 is used to construct the API function of the socket communication for IPv6. The end systems for each the patient and the specialist terminal are equipped with the IPv6 Technology Preview for Windows 2000. The IPv6 Technology Preview for Windows 2000 is to help the Winsock developers who are developing network applications using Winsock programming interfaces [16]. Computers loaded with this IPv6 Technology Preview are able to use both of the Internet protocols: IPv4 and IPv6.

For the IPv6 PC routers, we used computers loaded with FreeBSD 4.3. It is dual-stacked so that it can encapsulate the arrived IPv6 packets using IPv4 carriers. It also can support the multicast routing protocol when used as a PC router of IPv6 multicast. PIM-DM was used as the routing protocol in the experiment. Winsock API functions for the UDP socket communication are implemented under Platform SDK using Visual C++ 6.0 and Microsoft

IPv6 Technology Preview. We used both TCP and UDP sockets in the system trying not to change any part that is regardless of multicast. Therefore when the center program is activated, the patient's terminal make a connection with the center program and sends the data using TCP connection. Other two monitoring terminals join the multicast group using the multicast address of the PC router, and when the data come from the patient's terminal to the center, it delivers data to the PC router. Finally, the PC router duplicates the data received and transmits them to the multicast group members (Figure 4.5).

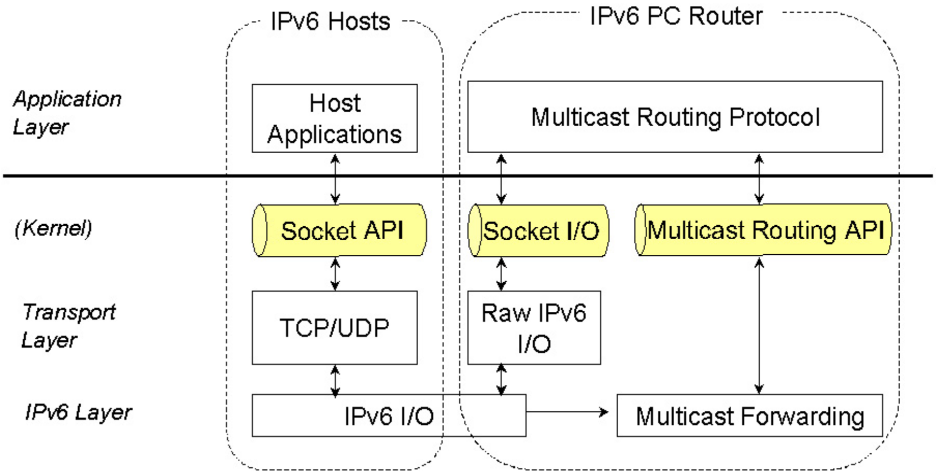


Figure 4.6 Socket Communication Mechanisms

4.3 Test-bed Configuration

Since UDP is a connectionless data delivery method, there is no guarantee that all the multicast packets that the patient sent to the group of monitors successfully delivered to the group members. In order to show the possibility of IPv6 multicast I tested the designed system in local network as shown in Figure 4.4. As you can see in the figure, three IPv6-enabled end systems were connected to each other using a FreeBSD PC router.

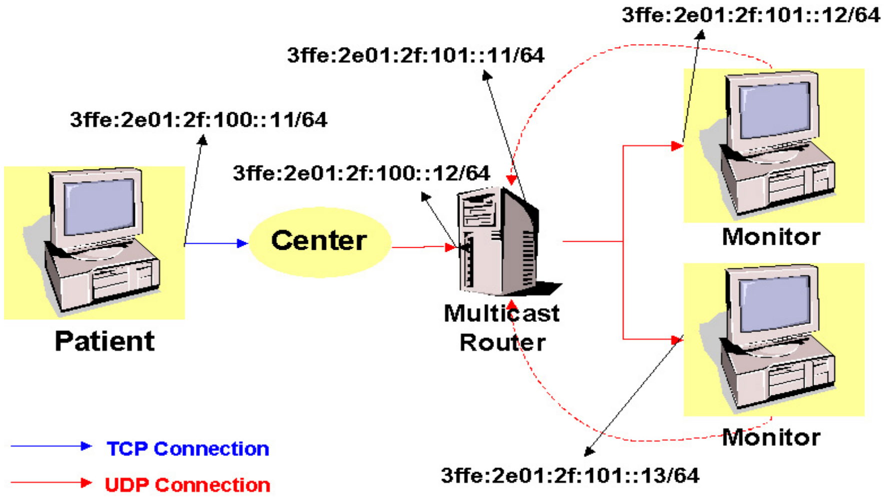


Figure 4.7 The Local Test-bed Configuration

UDP transmission is used only where it has something to do with the multicast. Other than that, it used the existing TCP connection. UDP can reduce the transmission and processing time, but since it is connectionless, unreliable transmission protocol it is better not to use UDP socket in delivering medical data when not related to the multicast.

Chapter 5

Experimentation & Result

5.1 Clinical Experimentation

As mentioned earlier in the paper, an expert consultation from more than one medical doctor is required for a proper treatment of a patient due to the subdivision of medical specialties. Under a serious emergency situation, the emergency medical technician has difficulty in taking care of the patient alone. An advice from a corresponding subspecialist is needed for a pertinent treatment. To examine the system operability and efficacy in such clinical cases, we have done experiments in two patient treatments: transcranial foreign body patient, mental change patient. The IPv6 multicast telemedicine system was set up in an emergency room of Yonsei Medical Center, in a office of a radiologist, and in a office of a neurosurgeon. Three subspecialists conducted a simultaneous multi-consultation, giving comments to the emergency physician on the proper treatment of the patient.

5.1.1 Patient with a Penetrating Injury in Cranium

A patient who had fatal penetrating injury in cranium was brought to the emergency room of Yonsei Medical Center with a serious damage in his head (Figure 5.1(a)). Emergency physicians examined patient's mental and physical state and contacted a neurosurgeon who stays in his office (Figure 5.1(b)) with the designed telemedicine system. The neurosurgeon was able to examine the patient's physical state remotely with the high-quality video and the biosignal. The radiologist can examine the exact anatomical location of the damage based on the patient's x-ray image obtainable through the PACS system (Figure 5.1(c)). The radiologist gave his opinion about the patient's status with the radiological image data. Therefore with the designed multicast telemedicine system, three specialists (emergency physician, neurosurgeon, and radiologist) were able to have a multi-consultation regarding an urgent patient and give more prompt and exact diagnosis.

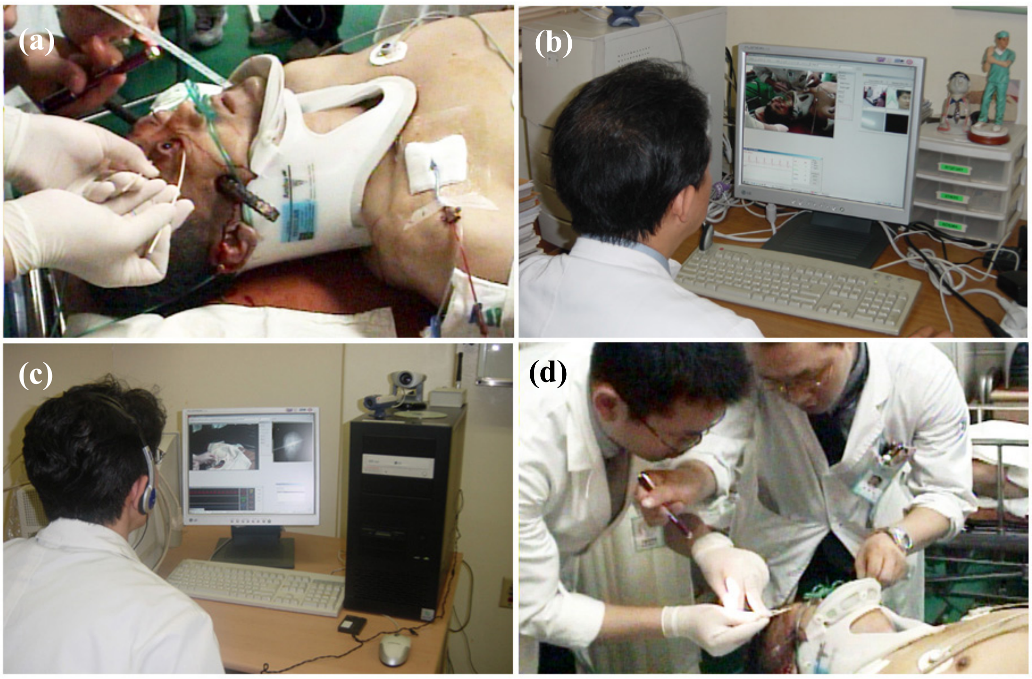


Figure 5.1 (a) Patient with a Penetrating Injury in Cranium, (b) Neurosurgeon, (c) Radiologist, (d) Emergency Physicians

5.1.2 Patient with Deteriorated Mentation

An unconscious patient arrived at the emergency room. The emergency doctor examined patient's condition and reported a drowsy mental state to the radiologist. The radiologist advised CT scan and diagnosed a small amount of subdural hemorrhage of the patient. He was able to give a direct comment to a neurosurgeon. The neurosurgeon and the emergency doctor could successfully make a management plan about the patient's treatment.

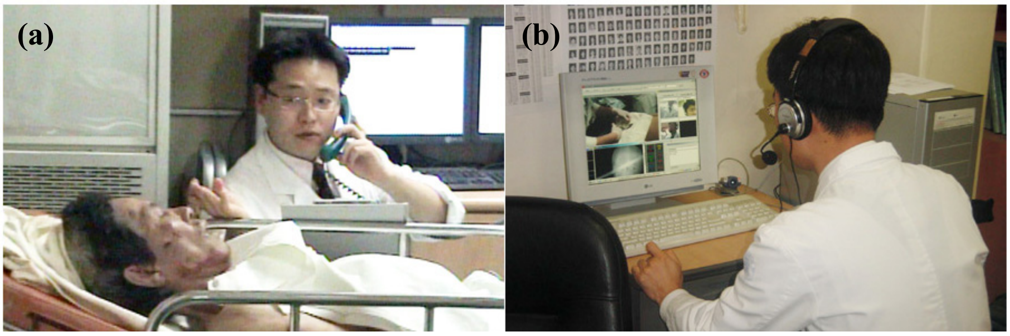


Figure 5.2 (a) Patient with Deteriorated Mentation, (b) Radiologist

As demonstrated in the previous experiments, multimedia telemedicine system based on IPv6 multicast enables a simultaneous multi-consultation between more than two specialists improving the decision-making process of patient care. It helps decreasing the possibility of misdiagnosis and makes it less troublesome to find out patient's disposition. Provided with the patient's multimedia data, such as the high-quality video and the biosignal, specialists can interpret patient's condition more exactly and promptly compared with when receiving a call about the patient.

5.2 Technical Experiment

To verify the stable operation of the system, we measured the data rate of high-quality video. The video was encoded in MPEG-2 and MPEG-4 with temporal resolution of 30frames/sec, spatial resolution of 720×480. The result is as shown in Table 5.1.

	One-to-one Communication	One-to-two Communication	One-to-three Communication
MPEG-2 30frames/sec (720×480)	4.3 Mbps	8.1 Mbps	13.0 Mbps
MPEG-4 30frames/sec (720×480)	0.5 Mbps	1.1 Mbps	1.6 Mbps

Table 5.1 Data Rate of High-Quality Video as the Number of Users Increase (Without Multicast)

As shown in the table, the data transmission rate of high-quality video linearly increases as the number of users increase. From these data we can infer that as the number of users increase the bandwidth consumption of the system will increase geometrically, thus requiring large network bandwidth. Not many kinds of commercial network support such large bandwidth, and this will cause a serious degradation of the designed system.

Since multicasting is for transmitting same kind of data to a large number of users, there has to be a bandwidth efficiency when communicating. To prove this theory we measured the data rate of high-quality video when multicasting (Table 5.2). As expected the bandwidth consumption did not increase as the number of multicast group members increase. From these result we can conclude that multicasting is very efficient when there are limited bandwidth and large number of simultaneous users. With multicast we can transmit same kind of data to many users without any bandwidth squandering.

	One-to-one Communication	One-to-two Communication	One-to-three Communication
MPEG-2 30frames/sec (720×480)	4.3 Mbps	4.5 Mbps	4.3 Mbps
MPEG-4 30frames/sec (720×480)	0.5 Mbps	0.5 Mbps	0.5 Mbps

Table 5.2 Data Rate of High-Quality Video as the Number of Receivers Increase when Multicasting

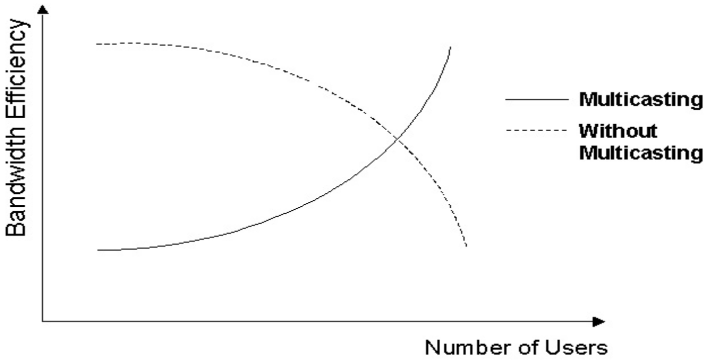


Figure 5.3 Comparison of Bandwidth Efficiency

Chapter 6

Conclusion and Discussion

In this paper we introduce a multimedia telemedicine system based on IPv6 multicast. This system is implemented on account of two facts. First, IPv6 is a newly developed Internet protocol that will solve the address shortage problem of existing IPv4 thus expected to become a dominant network protocol in near future. It has many advanced features that IPv4 does not have such as QoS (Quality of Service) control, built-in security, auto-configuration, multicast, and etc. Currently various kinds of IPv6 applications are being developed in many fields, and the medical application, too, should be prepared for the new network environment yet to come. Secondly, multicasting can be very bandwidth efficient when there is large number of users in a simultaneous communication. Instead of duplicating the same data according to the number of receivers (Figure 6.1), the multicast router receives one copy of required data and transmits them to the subscribed users (Figure 6.2).

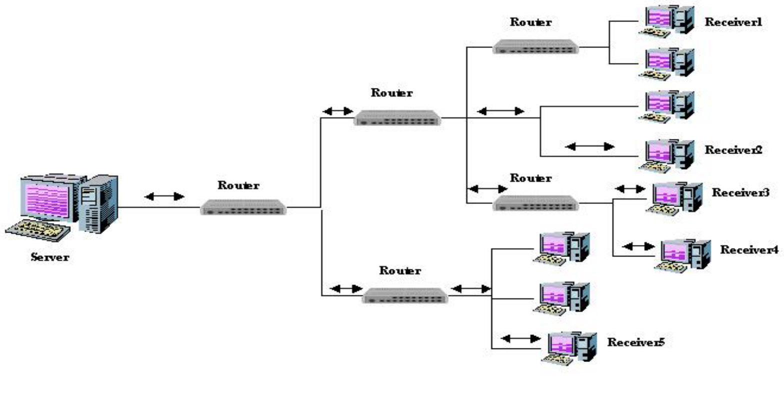


Figure 6.1 Data Communication without Multicasting

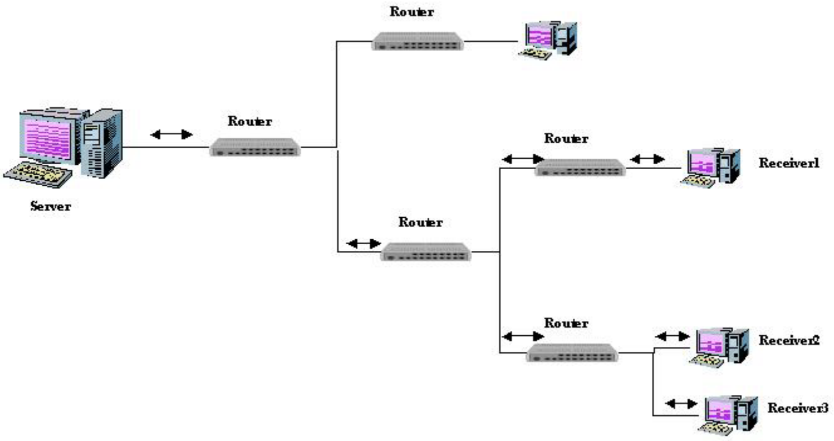


Figure 6.2 Data Communication when Multicasting

In order to describe patient's state, we used a high-quality video and the biosignal transmission. Since multicast must use UDP as the transport layer protocol, it does not guarantee a reliable data transmission. Packets can be lost or have errors while communication. If the patient's biosignal data is lost or has errors, it can affect the treatment of a patient in a fatal way. However, the high-quality video data of a patient is not affected by the packet loss or errors as long as it is distinguishable and running in real-time. Therefore we transmitted the video data in UDP and the signal data in TCP respectively. Rigidly only the high-quality video data is in multicast, and the biosignal data is being transmitted using a multi-connection method. Since most of the network bandwidth is consumed by the video data, this topology does not make large difference in system performance.

To show the system operability we conducted clinical and technical experiments in Yonsei Medical Center network. Being able to give patients a multi-consultation of required subspecialists in a limited time, emergency doctors could give the patient more exact and prompt treatment. With multicast it was possible to transmit high-bandwidth medical multimedia data in a limited network bandwidth without any performance degradation. This bandwidth-efficient data communication scheme, multicast, is expected to be useful in a larger multimedia data communication

References

[1] Smith, P., Ipsky: IPv6 for the aeronautical telecommunications network, *Digital Avionics Systems,*

The 20th Conference, Oct 14-18, 2001.

[2] Lee Ling Chuan, Jumari, K., Ismail, M. Implementation of video conferencing over IPv6 on the Linux platform, *SCOReD Student Conference*, July 16-17, 2002.

[3] Han-Chieh Chao, Yen-Ming Chu, Ma-Tai Lin. The cellular mobile IPv6 using low latency handoff algorithm for the packet-based cellular network. *ICCE 2000 Digest of Technical Papers*, 2000.

[4] Janos Mohacsi, Szabolcs Szigeti, Tamas Maray. Testing IPv6 implementations. *Computer Networks and ISDN Systems*, 1998.

[5] S. Ju, T. Cai, J. Yong, X. Zhang, Telemedicine System Integrated with Multimedia.

[6] E. Kyriacou, S. Pavlopoulos, D. Koutsouris, A.S. Andreou, C. Pattichis, C. Schizas, Multipurpose Health Care Telemedicine System, IEEE 23rd EMBS International Conference, Oct (2001).

[7] Hinden, R., and S. Deering. “IP Version 6 Addressing Architecture.” RFC 2373, July 1998.

[8] David Makofske, Kevin Almeroth, “Multicast Sockets: Practical Guide for Programmers,” Morgan Kaufmann Publishers.

[9] S. Deering, C. Partridge, D. Waitzman, “Distance Vector Multicast Routing Protocol,” RFC 1075, Nov.1998

[10] J. Moy, “Multicast Extensions to OSPF,” RFC 1584, Mar. 1994

[11] A. Adams, J. Nicholas, W. Siadak, “Protocol Independent Multicast-Dense Mode (PIM-DM): Protocol Specification (Revised),” draft-ietf-pim-dm-new-v2-01.txt, Feb.

2002

[12] A. Ballardie, “Core Based Trees (CBT) Multicast Routing Architecture,” RFC 2201, Sep.1997

[13] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M.Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, “Protocol Independent Multicast-Sparse Mode (PIM-SM) Protocol Specification/” RFC 2362, June 1998

[14] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, “Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised),” draft-ietf-pim-sm-v2-new-05.txt, Mar. 2002

[15] W. Fenner, “Internet Group Management Protocol, Version 2,” RFC 2236, Nov. 1997

[16] B. Cain, S. Deering, B. Fenner, I. Kouvelas, Ajiy Thyagarajan, “Internet Group Management Protocol, Version 3,” RFC 3376, Oct. 2002

[17] J. H. Jeong, S. Y. Lee, Y. J. Kim, “Construction of IPv6 Multicast Networks and Installation of IPv6 Multicast Applications,” Technical Document, IPv6 Forum Korea 2002-001.

[18] Mark A. Miller, P.E., “Implementing IPv6, Second Edition: Supporting the Next Generation Protocols,” M&T Books, 1999.

[19] MSDN, URL: http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/ReadMe.asp

국문요약

IPv6 멀티캐스트 기반 멀티미디어 원격진료 시스템의 개발

지금까지 시간과 공간의 제약을 극복하고 원거리의 환자를 진단, 처치할 수 있는 다양한 종류의 원격진료 시스템이 개발되었다. 환자의 상태를 나타내기 위하여 고화질 비디오, 생체신호, 방사선 영상, 의무기록 등의 멀티미디어 데이터를 송수신하는 시스템이 여러 종류의 네트워크에서 테스트 되었다. 그러나 지금까지의 원격진료 시스템은 모두 현재의 인터넷 프로토콜인 IPv4 기반의 일대일 통신 프로그램이었다. 네트워크 기술이 지속적으로 발전함에 따라 원격진료 시스템도 이제는 단순히 환자와 전문의 양자간의 데이터 교환을 넘어서 다자간의 데이터 공유가 가능해졌다. 멀티캐스트라고 하는 이 다자간 통신은 여러 명의 동시다발적인 통신이 가능하기 때문에 여럿이 모여 의견을 교환하고 의사결정을 하는 데에 효과적이다. 본 연구에서는 차세대 인터넷 프로토콜 IPv6 기반의 다자간 통신을 이용한 멀티미디어 원격진료 시스템을 구현하고, 실제 데이터의 공유가 정상적으로 이루어지는지 환경을 꾸며 실험하였다.

IPv6 멀티캐스트를 이용한 멀티미디어 원격진료 시스템은 환자의 상태를 나타내기 위하여 고화질비디오, 생체신호의 멀티미디어 데이터를 공유하며 공유자들간의 효과적인 커뮤니케이션이 이루어지도록 하였다. 고화질 비디오 신호는 사용하는 네트워크 대역폭에 따라 MPEG-2와 MPEG-4로 압축하여 전송할 수 있도록 하였으며, UDP 소켓을 기반으로 다중 수신자에게 동시에 멀티캐스팅 되도록 설계하였다. 생체신호는 신뢰성 확보를 위해 TCP를 기반으로 데이터를 송수신하도록 했다.

시스템의 활용성을 테스트 하기 위해서 각각 임상 실험과 기술 실험을 수행하였으며 임상에서는 동시에 여러 명의 전문의의 진료를 필요로 하는 외상 환자와 뇌혈관성 질환 환자의 진료에 효과적인 것을 확인할 수 있었다. 또 시스템이 동작할 때 소비되는 네트워크 대역폭을 측정하였는데, 대부분의 시스템 대역폭을 차지하는 고화질 비디오의 압축방법과 frame rate를 달리하면서 측정한 결

과 다중 수신자의 수가 증가하여도 일정한 대역폭으로 통신이 이루어지는 것을 확인하였다. 따라서 IPv6 멀티캐스트를 이용하면 한정된 네트워크 bandwidth 내에서도 여러 사람에게 동시에 data 전송을 할 수 있음을 확인하였다.

IPv6 멀티캐스트 기반의 멀티미디어 원격진료 시스템을 개발함으로써 여러 전문의의 multi-consultation이 가능해짐에 따라 환자의 보다 정확하고 빠른 진료가 가능해졌다. 또 높은 대역폭을 요구하는 의료 데이터를 동시에 여러 수신자에게 효과적으로 전송할 수 있는 방법을 제시하였다.

핵심이 되는 말: IPv6, 멀티캐스트, 원격진료 시스템, 고화질 비디오, 생체신호, 실시간 전송