韓國컴퓨터情報學會 論文誌
第14卷 第4號, 2009. 4.

2009-14-4-4-4

휴대용 단말기를 위한 실시간 무선 영상 음성 전송 기술

윤 경 섭*

Real-time wireless Audio/Video Transmission Technique for Handheld Devices

Kyung Seob Yoon*

요 약

무선 인터넷의 속도 향상과 휴대용 단말기의 컴퓨팅 파워 향상으로 이동 중에도 다양한 멀티미디어 서비스를 이용할 수 있게 되었다. 그러나, 휴대용 단말기를 통한 인터넷 사용을 위해서는 액세스 포인트(Access Point) 장비가 요구되고, 해당 장비에서는 동적으로 가상 네트워크 주소를 할당하는 방식을 사용하기 때문에, 휴대용 단말기 간의 직접 통신이 요구되는 서비스 즉, 1:1 음성 또는 화상 통화 및 메신저 서비스 등은 거의 제공되지 않고 있는 실정이다. 또한 이러한 서비스를 제공하는 측에서도, 중앙에 중계 서버를 두어 단말에서 전송되는 패킷을 수집, 반대편 단말로 중계하도록 구현함으로써, 실시간 멀티미디어 데이터 등 대량의 데이터 전송이 요구되는 서비스에서는 중계 서버의 전송량과 이를 감당하기 위한 비용이 증가하게 되어, 실시간 대용량 데이터 전송 서비스는 제한적으로 제공되고 있는 실정이다. 이 연구에서는 TCP/UDP Hole Punching이라는 기술을 이용하여 휴대용 단말기나 가상 개인 네트워크 주소를 사용하는 장비들 간의 실시간 멀티미디어 데이터 직접 전송이 가능한 P2P 서비스를 구현하였다.

Abstract

Improvement of Wireless internet and handheld devices makes it possible that users can use various multimedia services. But, access point devices are needed while using handheld devices, and those devices use virtual network address for networking. For that reason, end-users hardly use the 1:1 voice or video chat, and messenger service that require direct communications between devices. Also, service providers need central server for relaying packets from terminals to others, the traffic and costs of relaying go high, so real-time massive data transmission services are restrictively provided. In this study, we apply TCP/UDP hole punching technique to those applications. And we implement service that supports real-time multimedia direct transmission between equipments that use virtual network addresses.

▸ Keyword : 홀펀칭(hole punching), 무선 P2P(wireless P2P), 실시간 전송(real-time transmission)

• 제1저자 : 윤경섭
• 투고일 : 2009. 03. 16, 심사일 : 2009. 03. 18, 게재확정일 : 2009. 04. 02.
* 인하공업전문대학 컴퓨터정보과 교수

Ⅰ. 서 론

많은 P2P 응용 프로그램들은 TCP 프로토콜을 통해 서비스를 제공하고 있지만, 최근 들어 보안이나 네트워크 회선 공유를 위한 인터넷 공유기, 무선 액세스 포인트 등의 NAT (Network Address Translator) 장비들과 파이어 월 (Firewall) 등의 사용이 증가함에 따라 피어 간의 직접 접속이 불가능하여 서비스 제공이 어려운 실정이다[1, 2].

휴대용 무선 단말기들의 경우, 공인된 고정 네트워크 주소를 갖는 것이 불가능하기 때문에, 네트워크에 물리적으로 연결되어 고정 네트워크 주소를 갖는 서버가 제공하는 서비스를 이용하거나, 서비스를 제공하고자 하는 경우에는 고정 네트워크 주소를 갖는 중계 서버(Relay Server)들을 두어 데이터를 중계하도록 하여야만 한다.

현재 대부분의 P2P 서비스들이 NAT 장비의 사용을 고려하고 있지 않거나, 지원이 부족한 상태이며, 최근에 개발되고 있는 서비스들도 중계 서버를 사용하거나, 고정 공인 네트워크 주소를 통한 방식으로 서비스되고 있으며, 일부는 사용자들에게 NAT 장비의 설정을 재구성하도록 권고하고 있다.

그러나, 실시간 멀티미디어 서비스의 경우에는 중계 서버를 통한 서비스는 중앙 서버에 전송량이 집중하게 되며, 휴대용 단말기 장비를 사용하는 경우 NAT 장비를 다른 장비들과 공유해서 사용하는 경우가 대부분이므로 NAT 장비의 설정을 변경하는 방법은 적합하지 못하다[3, 4].

이러한 문제들을 해결하기 위해 최근의 솔루션들은 UDP 상의 터널링(tunneling)을 설정하기 위해 SIP(Session Initiation Protocol)을 사용하거나, UPnP(Univeral PnP) 기법을 사용한다거나, IPv6를 기반으로 동작하기도 한다. 이러한 방식들은 확장성이 부족하고 휴대형 단말기를 고려하고 있지 않으며, 새로운 네트워크 인프라를 요구하거나 비표준 방식의 네트워크 인터페이스와 프로토콜을 요구한다.

이 연구에서는 NAT 장비의 사용과 무관하게 피어 간의 직접 TCP 접속을 구현하여 멀티미디어 서비스를 제공할 수 있는 방식을 제안한다.

Ⅱ. NAT를 통한 데이터 전송의 문제점

2.1 기존 P2P 서비스에서의 NAT에 대한 고려

많은 P2P 응용들이 NAT를 통한 데이터 전송에서 발생하는 문제점들을 언급하고, 여러 가지 방식으로 해결책을 제시하고 있다. 그러한 응용들에서 NAT를 통한 데이터 전송을 해결하는 방안은 다음과 같다.

1) 한쪽의 피어만 NAT 장비를 사용하는 경우, NAT 장비를 사용하는 쪽에서 네트워크 연결을 시도한다[5, 6]. 이는 불완전한 해결책이 될 수 밖에 없는데, 만일 양쪽의 피어 모두 NAT 장비를 사용하는 경우 통신 자체가 불가능하다.
2) 사용자가 NAT 장비의 설정을 변경하여 특정 포트로 전달되는 모든 요구들을 NAT 장비에 물려 있는 특정 컴퓨터에 전달하도록 한다[7]. 이 방식 또한 하나의 NAT 장비에 물려 있는 여러 대의 장비들이 동일한 서비스를 사용하거나, 사용자가 NAT 장비의 설정을 변경할 수 있을만한 능력과 권한이 부족하다면 사용할 수 없다.
3) 중앙에 중계 서버를 두어 P2P 통신에서 발생되는 데이터를 중계하도록 한다[8]. 이 방식은 서비스 초기에는 적용하기 쉽고 안정된 방식이기는 하지만, 서비스의 확장을 고려하는 경우 중앙 서버의 컴퓨팅 파워와 네트워크 전송량의 증가를 위해 막대한 비용을 소모하게 된다.
4) NAT 장비를 사용하지 않으면서 고정 네트워크 주소를 갖는 장비에 해당 응용의 인스턴스를 실행시켜 TCP 접속을 중계하도록 한다[9, 10]. 이 경우 발생되는 모든 데이터를 중계하는 것이 아니라, 최초의 접속 시도만 중계하도록 함으로써 네트워크 전송량의 문제나 컴퓨팅 파워는 해결할 수 있지만, 역시 P2P 서비스를 제공하는 것과는 별도로 접속을 위한 특정 장비가 존재해야 한다.
5) 피어간 UDP 포트를 교환할 수 있는 특정한 SIP(session initiation protocol)을 사용함으로써 UDP 상에서 TCP 터널을 구성한다[11]. 이 방식은 표준 TCP 스택과 인터페이스를 사용하지 않으며, 많은 네트워크에서 UDP를 허용하지 않기 때문에 쉽게 적용할 수 없다.

이 연구에서는 4)번 방식을 확장, P2P 사용자 인증과 과금을 위한 멀티미디어 실시간 전송을 위한 서비스를 구현한다.

2.2 구현 시 고려사항

1절에서 언급한 내용들과 실시간 멀티미디어 P2P 전송 서비스를 구현하기 위해 고려할 사항은 다음과 같은 것들이 있다.

1) 확장성 - 서비스의 추가나 변경, 규모 확장 시의 비용을 최소화 할 수 있는 방안을 제공해야 한다.
2) 표준 인터페이스 - 특정 하드웨어나 프로토콜 스택의 추가 없이 기본 TCP 인터페이스만으로 서비스 제공이 가능해야 한다.

3) 이동성 - 휴대용 단말기의 사용을 고려할 때, 각 피어는 한 네트워크에서 다른 네트워크로 이동하며 통신하게 되므로 그러한 경우 NAT 장비의 재설정이나 포트의 매핑, UDP 또는 UPnP 서비스를 사용하지 않고도 서비스 제공이 가능해야 한다.

III. 서비스 설계

3.1 서비스 아키텍쳐

제안하는 아키텍쳐에서 P2P 서비스는 다음과 같은 순서로 제공된다.

1) 서비스를 시작하는 피어에서 인증 서버로 로그인하며 네트워크 주소를 등록한다.
2) 서비스를 받을 피어에서도 인증 서버로 로그인하며 네트워크 주소를 등록한다.
3) 서비스를 시작하는 피어에서 서비스를 받을 피어 쪽으로 접속을 시도한다.
   A. 서비스를 받을 피어에서 NAT 장비를 사용하는 경우 고정 네트워크 주소를 갖지 않기 때문에 직접 접속 시도는 실패하게 된다.
   B. 이를 방지하기 위해 TCP Hole Punching을 사용한다.
4) 직접 접속이 성공하게 되면 이후부터는 인증 서버와 무관하게 피어 간의 데이터 전송이 발생하게 된다.

다음 그림 1은 제안하는 P2P 서비스 아키텍쳐를 나타낸다.

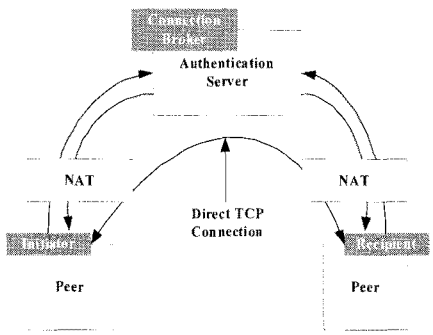

그림 1. 서비스 아키텍쳐
Fig. 1. Service Architecture

3.2 TCP Hole Punching

다음 그림 2는 Hole Punching 메커니즘을 나타내고 있다.

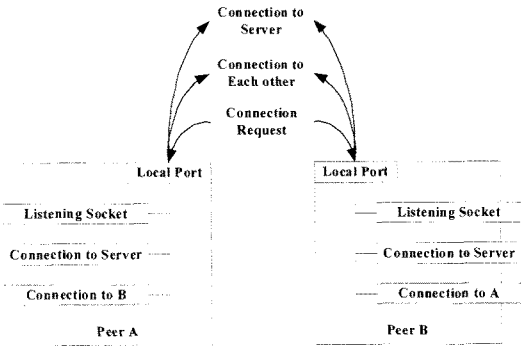

그림 2. TCP Hole Punching 메커니즘
Fig. 2. TCP Hole Punching Mechanism

피어 B로 TCP 접속을 시도하는 경우, A와 B는 이미 고정 네트워크 주소를 갖는 서버에 TCP 접속이 이루어져 있어야 한다. 서버는 각 피어의 고정 네트워크 주소와 포트 정보, 그리고 가상 네트워크 주소와 포트 정보를 기록한다. 이는 UDP 서비스에서와 유사한데, 프로토콜 수준에서의 TCP Hole Punching은 UDP와 거의 동일하게 동작한다.

1) 피어 A는 서버와 TCP 세션을 활성화하고, 서버 측에 피어 B와 접속을 시도한다는 사실을 알린다.
2) 서버는 A에게 B의 공인 네트워크 주소와 가상 네트워크 주소를 알려주고, 동시에 B에게도 A의 공인 및 가상 네트워크 주소를 알려 준다.
3) A와 B가 서버에 등록한 것과 동일한 TCP 포트로 상대방의 공인 네트워크 주소로의 접속을 시도한다. 이는 비 동기적으로 발생하게 되며, 각 피어에서는 동일한 포트로 외부에서 들어오는 접속 요구를 대기하게 된다.
4) A와 B는 접속 시도가 성공할 때까지 대기하거나, 접속 요청이 들어올 때까지 대기한다. 만일 접속 시도가 실패하게 되면, 피어에서는 약간의 대기 시간을 두고 난 후 다시 접속을 시도한다.
5) TCP 접속이 이루어지면 각 피어는 서로를 인증하게 되며, 인증 작업이 실패하게 되면 해당 접속을 제거하고, 다른 시도가 성공할 때까지 대기한다.

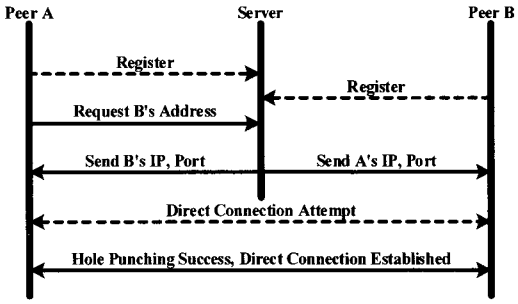

그림 3. Hole Punching 타이밍 다이어그램
Fig. 3. Hole Punching Timing Diagram

3.3 Hole Punching 구현

Hole Punching의 구현은 BSD 소켓의 SO_REUSEADDR 옵션을 통해 이루어진다. 본 구현에서는 WinCE 운영체제가 동작하는 단말기를 대상으로 하였으므로 Winsock32 2.0 기준의 라이브러리 함수와 MFC 클래스 라이브러리를 사용하였다.

외부의 접속 요청을 받아들이기 위한 listenSock과 해당 소켓의 참조를 저장하기 위한 accSock, 그리고 데이터 전송을 위한 chatSock을 생성한 후, 소켓에 핸들을 부여하지 않은 상태에서 소켓 핸들을 정해진 포트로 생성한 후, 필요에 따라 각 소켓에 해당 핸들을 부여하여 사용한다. 다음 알고리즘 1은 하나의 포트에 여러 개의 소켓을 생성하기 위한 알고리즘이다.

알고리즘 1. 중첩된 소켓의 생성
Algorithm 1. Overlapped Sockets Creation

```
STEP 1: Create socket handle;
STEP 2: Attach that handle to Socket ;
STEP 3: Set Socket's option with
        SO_REUSEADDR;
STEP 4: Bind Socket to predefined port;
```

중첩된 소켓을 사용하여 hole punching을 시도하려면 양쪽의 피어에서 동시에 접속을 시도해야 한다. 다음 알고리즘 2는 클라이언트 측에서 연결이 이루어질 때까지 hole punching을 시도하는 알고리즘이다.

알고리즘 2. Hole Punching 시도
Algorithm 2. Trying Hole Punching

```
STEP 1: Until connection established;
STEP 2: Overlap chatSock to pre-created socket
        using algorithm 1;
STEP 3: Try to connect to peer;
    STEP 3.1: If success,
        break out this loop;
    STEP 3.2: Else if unsuccessful,
        increase counter;
STEP 4: Check counter;
    STEP 4.1: If counter too big,
      connection failed and break out;
    STEP 4.2: else
      wait some second and GOTO STEP1;
```

접속이 성공적으로 완료되면 서버와의 접속은 accSock을 이용하여 유지한 상태에서 chatSock을 통해 데이터 전송과 수신을 시작한다.

IV. 프로토콜 정의

피어간의 접속을 위하여 시스템은 2 단계로 구현된다.

4.1 서버로 로긴 후 피어들의 접속, 관리를 위한 제어 채널을 생성하는 단계

Step 1. 피어 A는 피어 B를 초청하기 위한 커맨드를 서버로 전송한다(command INVITE).

Step 2. 서버는 해당 커맨드를 피어 B로 전송한다(command INVITE).

Step 3. 피어 B는 피어 A의 초청 요구를 수락하는 응답을 서버로 전송한다(command INVITE-RESULT).

Step 4. 서버는 피어 A에게 B에서의 초청 수락 응답을 전송한다(command INVITE-RESULT).

Step 5. 이 단계는 중앙 중계 서버를 이용한 시스템에서도 구현되어 있는 부분이며, 이 이후부터 Hole Punching 처리가 시작된다.

4.2 Hole Punching을 시도하는 단계

1) 클라이언트 측

Step 1.

A. 피어 A는 PeerChannel이라는 이름의 소켓을 생성하고, 지역 포트로 바인드한다.

B. 피어 A는 PeerChannel을 이용하여 서버의 Hole Punching 포트로 접속한다.

C. 피어 A ListenChannel이라는 이름의 소켓을 생성하고, 위에서 생성한 지역 포트를 통해 요구를 받아들이도록 설정한다.

Step 2.

A. PeerChannel이 서버와 접속되면, 피어 A는 PeerChannel을 통해 서버로 피어 B의 공인 네트워크 주소와 포트를 요청한다.

Step 3.

A. 피어 A는 서버로부터 피어 B의 공인 네트워크 주소와 포트 정보를 수신한다.

B. 피어 A는 서버와의 접속을 종료한다.

C. 피어 A는 PeerChannel을 사용하여 피어 B의 공인 네트워크 주소와 포트를 사용하여 피어 B로 접속을 시도한다(이 과정에서 피어 B도 피어 A의 공인 네트워크 주소와 포트를 사용하여 피어 A로의 접속을 시도한다).

Step 4.

A. 이 단계에서는 다음과 같은 두 가지 경우가 발생할 수 있다.

1. 피어 A의 PeerChannel이 피어 B와의 접속에 성공한다.
2. 피어 A의 ListenChannel이 피어 B로부터의 접속 요청을 수락한다.

B. 피어 A는 두 가지 경우 중 먼저 성공한 접속을 유지한다.

2) 서버 측

서버는 Hole Punching 요구를 피어 A에게서 받게 되면 동시에 피어 B로부터 동일한 요청이 들어올 때까지 대기한다. 요청이 들어오면 공인 네트워크 주소와 포트 정보를 추출하여 피어 A와 피어 B에 동시에 서로의 정보를 전송한다.

표 1. 피어와 인증 서버간 프로토콜 정의
Table 1. Protocol definition between Peer to Server

Peer	Server
Command: REGISTER	Command: REGISTER RESULT
Params: StationType, User, Password, Group	Params:User, Result : {Failed, Existed, Pwd required, Success},
Command: LOGIN	Command: LOGIN RESULT
Params: Status, StationType, User, Password	Params: Result : {not found, Wrong password, Success}
Command: LOGOUT	
Params: Status, StationType, User, Password	
Command: INVITE	Command: INVITE RESULT
Params: Invitation, Buddy	Params: Invitation, Buddy
Command: HOLEPUNCHING	Command: HOLEPUNCHING
Params:HolePunchParam (request), Buddy, UserIP, Port	Params: HolePunchParam (reply),Buddy, UserIP, Port
	Command: UPDATE FRIENDLIST
	Params: UserCount
Command: ENABLE AUDIO	Command: CREATEMEDIACHANNEL
Params: Enable	Params: ServerMediaSenderPort ServerMediaReceiverPort
Command: ENABLE VIDEO	
Params: Enable	
Command: NOTIFY	Command: NOTIFY
Params: ElapseTime, Notify: {ping, timeout,server died}	Params:ElapseTime, Notify: {ping, timeout,server died}

4.3 서비스를 위한 프로토콜 정의

표 1은 피어와 인증 서버 간에 정의되는 프로토콜을 도식화한 것이다. 서비스 제공을 위한 프로토콜은 피어간 통신을 위한 프로토콜의 경우 서비스에 종속적인 프로토콜로 정의하여 사용하게 되며, 실시간 멀티미디어 서비스의 경우 RTP 등의 프로토콜을 사용할 수 있다. 이 장에서는 제안하는 서비스를 위한 피어와 인증 서버간의 프로토콜을 정의한다.

피어와 인증 서버간 프로토콜로 정의되는 제어 패킷 구조는 그림 4와 같다.

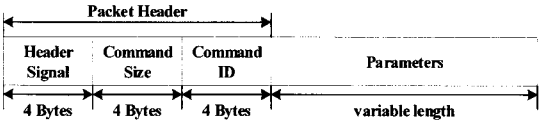

그림 4. 제어 패킷 구조
Fig. 4. Control Packet Structure

V. 실험

정의한 프로토콜과 서비스의 실험을 위해 무선 단말간의 접속 및 유,무선 단말간 접속 환경을 구성하여 실험을 진행하였으며, 국내 시판중인 NAT 장비들이 정의한 프로토콜을 제공할 수 있는가를 테스트하기 위하여 BUFFALO 사의 유무선 공유기 5종, CISCO-linksys의 유무선 공유기 4종, 넷기어와 다산-SMC의 제품 각 6종씩을 준비하여 각 장비 별로 환경을 구성하여 실험을 진행하였다.

무선 인증 서버로는 Intel Xeon 1.6 GHz 듀얼 PC 서버를 사용하였으며, 휴대용 단말로는 HP iPaq hx2790과 HP iPaq rw6100 모델을 사용하였고, 2.6 GHz 노트북 및 공인 네트워크 주소를 갖는 3GHz PC와 가상 네트워크 주소를 갖는 3GHz PC를 사용하였다.

시뮬레이션을 통한 스트레스 테스트에서는 인증 서버 한 대에 평균 5000여 개 정도의 피어가 접속이 가능하였으며, 5000개의 피어가 접속한 상황에서도 인증 서버는 정상적으로 동작하는 것을 확인할 수 있었다.

이것은 피어와 서버 간의 접속을 영구적으로 유지하는 프로토콜이 아닌, 상대방 피어와의 접속을 위한 서버 접속만 허용하는 프로토콜을 사용하므로, 피어와 서버와의 트래픽의 문제나 병목 현상 등이 발생할 이유가 없기 때문이며, 이론상으로는 동일 시간대에 서버 당 약 3만개의 피어가 접속이 가능할 것으로 예측되었다.

피어간의 데이터 통신에 있어서는 텍스트 데이터의 경우 전혀 문제가 없었으나, 실시간 화상 채팅의 경우 동영상과 음성을 동시에 인코딩하여 전송하는 실험에서 휴대용 단말기의 경우 컴퓨팅 파워 부족으로 가끔 과열되는 현상이 발생했다. 노트북이나 PC의 경우는 NAT를 거치는 경우나, 거치지 않는 경우나 서비스 이용에 전혀 차이가 없었다.

NAT 장비의 서비스 제공 가능성을 알아보기 위한 호환성 테스트에서는 일반 사용자가 사용한다는 것을 전제로 기본 설정을 변경하지 않은 상태에서 실험을 진행하였다. 기본 설정 상태에서도 대부분의 NAT 장비가 hole punching 기법을 허용하며, 극히 일부 장비의 경우에도 포트 포워딩(port forwarding) 설정 이후에는 실험에 사용된 전체 NAT 장비가 이 연구에서 정의한 프로토콜과 호환성을 제공함을 알 수 있었다.

표 2. 장비별 서비스 호환성
Table 2. Service Compatibility among Equipments

NAT	hole punching	port forwarding 후
BUFFALO	4/5	5/5
CISCO-linksys	4/4	4/4
Netgear	5/6	6/6
다산-SMC	6/6	6/6
전체	19/21	21/21

실험 결과, 일반적으로 사용되는 Windows OS 기반의 단말기와 Linux OS 기반의 NAT 장비간의 서비스 호환성은 충분한 것으로 보여지며, 일반 사용자들이 구매하여 사용하게 되는 단말기와 유무선 공유기 장비들이 추가적인 설정 변경 없이 실시간 무선 멀티미디어 전송이 가능하므로 주위에 무선 접속 지점(access point)이 설치된 지역에서는 일반 화상 전송 휴대폰에 비해 저렴한 비용으로 멀티미디어 전송이 가능하다.

또한 행사장이나 보안이 필요한 창고 등의 건물, 전파 방해만 없다면 병원 등에서도 휴대성이 용이하고, 다기능을 제공하는 무선 단말기를 사용하여 항시적으로 접속을 유지하며 통신할 수 있으므로 활용될 여지가 많음을 알 수 있다.

VI. 결론

이 연구에서는 휴대용 단말기에서 직접 TCP 접속을 통한 P2P 서비스를 위한 프로토콜을 정의하고, 정의한 프로토콜을 기반으로 한 실시간 멀티미디어 데이터 전송 응용으로 화상 채팅 응용을 구현하여 해당 프로토콜을 통해 P2P 접속이 가능한가를 실험하였다.

제안한 프로토콜은 하드웨어 장비의 추가나 설정 변경이 전혀 요구되지 않으며, 표준 프로토콜 스택을 사용하여 이후의 서비스 변경이나 추가 등을 지원할 수 있도록 확장성을 보장한다. 또한 중계 서버를 두는 방식에 비해 피어간의 접속 및 데이터 전송 비용을 획기적으로 절감하였다.

그러나, 휴대용 단말기에서의 원활한 실시간 멀티미디어 전송 서비스를 위해서는 프로토콜 등 기반 환경의 구축뿐만 아니라, 해당 서비스를 원활하게 사용할 수 있는 휴대용 단말기의 컴퓨팅 파워 향상이 더 시급한 문제로 여겨진다.

제안한 프로토콜은 이후 휴대용 단말기 사용의 활성화가 이루어진 경우 기존 네트워크 망 구성이나 하드웨어 설정의 변경 없이 쉽게 사용할 수 있으며, NAT 장비를 사용하는 환경에서 실시간 멀티미디어 전송 등 다양한 서비스의 기본 프로토콜로 사용될 수 있다.

참고문헌

[1] K. Egevang & P. Francis, *"The IP Network Address Translator (NAT),"* IETF RFC 1631.

[2] P. Srisuresh & K. Egevang, *"Traditional IP Network Address Translator,"* IETF RFC 3022.

[3] J. Rosenberg, et al., *"STUN - Simple Traversal of User Datagram Protocol Through Network Address Translators,"* IETF RFC 3489.

[4] M. Deshpande, et al., "Flashback: A Peer-to-Peer Web Server for Flash Crowds," 27th International Conference on Distributed Computing Systems, pp. 15, 2007.

[5] K. Kimotsuki, et al., "Construction of P2P-VPN Utilizing UDP Hole Punching Method," IEICE Tech. Rep., Vol. 106, No. 577, NS2006-196, pp. 197-200, March, 2007.

[6] K. Suh, et al., "Characterizing and Detecting Skype-Relayed Traffic," Proceedings of Infocom, 2006.

[7] M. Wu, et al., "A Scalable Port Forwarding for P2P-Based Wi-Fi Applications," Lecture Notes in Computer Science, Vol 4138, pp. 26-37, Springer Berlin/Heidelberg, 2006.

[8] G. Schiele, et al., "Requirements of Peer-to-Peer Based Massively Multiplayer Online Gaming," Proceedings of the 7th International Workshop on Global and Peer-to-Peer Computing, 2007.

[9] A. Wacker, et al., "Towards an Authentication Service for Peer-to-Peer Based Massively Multiuser Virtual Environments," International Journal of Advanced Media and Communication, Vol. 2, No. 4, pp. 364-379, 2008.

[10] Taein Hwang, Hojin Park, and Jinwook Chung, "Personal Mobile A/V Control Point for Home-to-Home Media Streaming," IEEE Transactions on Consumer Electronics, Vol. 54, Issue 1, pp. 87-92, 2008.

[11] N. Steinleitner, et al., "Implementation and Performance Study of a New NAT/Firewall Signaling Protocol," Proceedings of the International Workshop on Assurance in Distributed Systems and Networks, 2006.

저 자 소 개

윤 경 섭

1982 인하대학교 전자계산학과 학사
1984 인하대학교 전자계산학과 석사
1995 인하대학교 전자계산학과 박사
2000 ~ 2001 University of Cincinnati in USA, 방문교수
2004 ~ 현재 인천부천김포이업종교류연합회 IT교류회, 자문교수
2005 ~ 2006 인천교원단체총연합회, 부회장
2009 ~ 현재 한국교원단체총연합회, 이사
1987 ~ 현재 인하공업전문대학 컴퓨터정보과, 교수
관심분야 : e-Learning, e-Business, IT Services