

# Computer Network

## Term Project 2

- Building a Small Network -

학	과	컴퓨터공학과
학	번	201211704
이	름	김기홍
제	출 일	2016.06.06



## <목 차>

1장 프로젝트 개요 .....	5
2장 실습 환경 구축 .....	5
2.1 구축 환경 .....	5
2.2 라우터 .....	5
2.3 가상 머신 .....	5
2.4 각종 서버 .....	6
3장 네트워크 구성 .....	7
3.1 LAN 토폴로지 .....	7
3.2 구성도 .....	9
4장 서버 구축 과정 .....	10
4.1 가위바위보 Server .....	10
4.2 Mail Server .....	10
4.3 Web Server .....	10
4.4 DNS Server .....	10
4.5 FTP Server .....	11
4.6 DHCP Server .....	11
5장 동작 예시 .....	12
5.1 라우터 간 통신 확인 .....	12
5.2 종단 시스템 간 통신 확인 .....	16
5.3 각 서버의 동작 확인 .....	18
6장 WireShark를 통한 프로토콜 분석 .....	30
7장 프로젝트 평가 및 소감 .....	38
별첨 .....	39

## <그림 목차>

그림 3.1.1 스타 토폴로지 .....	7
그림 3.1.2 물리적인 토폴로지 .....	7
그림 3.1.3 액세스 포트와 트렁크 포트 .....	8
그림 3.2.1 전체적인 구성도 .....	9
그림 5.1.1 OSPF 라우팅 테이블 .....	12
그림 5.1.2 RIP 라우팅 테이블 .....	13
그림 5.1.3 BGP 라우팅 테이블 .....	14
그림 5.1.4 Static 라우팅 테이블 .....	15
그림 5.2.1 패킷 경로 추적(같은 네트워크) .....	16
그림 5.2.2 패킷 경로 추적(다른 네트워크) .....	17
그림 5.3.1 가위바위보 Server .....	18
그림 5.3.2 가위바위보 Server 동작 화면 .....	19
그림 5.3.3 Player1 접속 화면 .....	20
그림 5.3.4 Player2 접속 화면 .....	20
그림 5.3.5 Player1 결과 화면 .....	21
그림 5.3.6 Player2 결과 화면 .....	21
그림 5.3.7 가위바위보 Server 결과 화면 .....	22
그림 5.3.8 메일 서버 접속 화면 .....	23
그림 5.3.9 메일 전송 화면 .....	23
그림 5.3.10 메일 확인 화면 .....	24
그림 5.3.11 Web Server에 있는 HTML 문서 .....	24
그림 5.3.12 Web Browser에서 HTML 문서 출력 .....	25
그림 5.3.13 FTP 서버에 있는 Test 파일 .....	25
그림 5.3.14 FTP 서버에 접속하여 Test 파일을 읽는 화면 .....	26
그림 5.3.15 nslookup으로 살펴본 도메인 .....	26

그림 5.3.16 도메인으로 접속한 Web서버 .....	27
그림 5.3.17 도메인으로 접속한 FTP서버 .....	27
그림 5.3.18 dhcp 설정 값 (range 10.0.0.10~ 10.0.0.20) .....	28
그림 5.3.19 dhcp 사용 설정 .....	28
그림 5.3.20 IP 주소를 자동으로 할당받는 화면 .....	29
그림 6.1.1 Web Server 접속 시 교환되는 패킷 .....	30
그림 6.1.2 HTTP 요청 메시지 .....	30
그림 6.1.3 HTTP 응답 메시지 .....	30
그림 6.2.1 FTP Server 접속 시 교환되는 패킷 .....	31
그림 6.2.2 FTP CWD 명령 및 응답 메시지 .....	31
그림 6.2.3 Mail Server 접속 시 교환되는 패킷 .....	32
그림 6.4.1 DNS Server 접속 시 교환되는 패킷 .....	32
그림 6.4.2 DNS query 패킷 정보 .....	32
그림 6.4.3 DNS resoponse 패킷 정보 .....	33
그림 6.5.1 동적 IP를 할당받기 위해 교환되는 패킷 .....	33
그림 6.5.2 DHCP Discover 패킷 정보 .....	34
그림 6.5.3 DHCP Offer 패킷 정보 .....	35
그림 6.5.4 DHCP Request 패킷 정보 .....	36
그림 6.5.5 DHCP ACK 패킷 정보 .....	37

## 별첨

그림-mesh 방식의 LAN 구성도 .....	39
그림-연결에 이상이 없는 경우 OSPF 라우팅 테이블 .....	40
그림-연결에 이상이 없는 경우 OSPF neighbor .....	40
그림-R4로의 경로 추적 .....	40
그림-하나의 인터페이스가 다운되었을 경우 OSPF 라우팅 테이블 .....	41
그림-하나의 인터페이스가 다운되었을 경우 OSPF neighbor .....	41
그림-R4로의 경로 추적 .....	41

그림-R1에서 R4로의 경로 추적 .....	42
그림-두 개의 인터페이스가 다운되었을 경우 OSPF 라우팅 테이블 ...	43
그림-두 개의 인터페이스가 다운되었을 경우 OSPF neighbor .....	43
그림-R3로의 경로 추적 .....	43
그림-R4로의 경로 추적 .....	43
그림-R1에서 R4, R1에서 R3로의 경로 추적 .....	44

## 1장. 프로젝트 개요

이번 Term Project2의 주제는 라우터로 연결된 소규모 네트워크를 구성하고 인터넷의 각종 서버를 구축하여 동작을 확인하는 것이다. 본 프로젝트는 소규모 네트워크를 구성하기 위해 네트워크의 전반적인 이해와 인터넷의 여러 서버들의 동작 이해, 그리고 가상머신의 이해 등을 돕고자 진행되었다. 이러한 학습의 장점은 구축을 통한 실질적인 결과가 눈에 보임으로써 프로젝트 진행에 지루함이 없다는 점이다.

해당 보고서는 전체적인 소규모 네트워크의 구성과 동작 예시에 중점을 두었으며 세부적인 서버 구축 방법, 패킷 분석 결과 등을 간단하게 기술한다. 또한 마지막에 프로젝트 진행에 대해 자가 평가한 도표도 제시되어있다.

## 2장. 실습 환경 구축

### 2.1 구축 환경

본 프로젝트의 주제인 소규모 네트워크 구축은 2.7GHz 쿼드코어, 4GB RAM, Windows 7 Professional-64bit 환경에서 이루어졌다.

### 2.2 라우터

PC에서 가상적으로 시스코 라우터를 동작시키는 프로그램인 다이나믹스를 사용하였고 다이나믹스에서 특정한 라우터와 인터페이스를 구성하고 동작시켜주는 프로그램으로 다이나젠을 사용하였다. 시스코 라우터 중에서도 Cisco 3600 Series의 Cisco 3660을 사용하였으며, 사용한 모듈로는 1)NM-16ESW와 2)NM-4T이다. 그리고 각 라우터에 접속하기 위해 SecureCRT에서 Telnet을 사용함으로써 보다 간편하게 여러 터미널을 관리했다.

### 2.3 가상 머신

소규모 네트워크의 종단 시스템을 구축하기 위해 VMware Workstation 7.1을 사용하였으며 게스트 OS로는 Windows XP Professional-32bit, Linux Mint mate-32bit, ubuntu 12.0.4 server-32bit, CentOS 6.7 minimal-32bit를

---

1) 이더넷 인터페이스 16개를 추가로 장착 가능한 모듈

2) 시리얼 인터페이스 4개를 추가로 장착 가능한 모듈

사용하였다. 게스트 OS는 주로 과거 버전을 사용하였는데, 이는 본 프로젝트를 진행한 컴퓨터의 자원 부족 현상을 완화하기 위함이다.

## 2.4 각종 서버

구축한 서버의 종류에는 TermProject1의 서버 프로그램을 제외하고 Mail Server, Web Server, DNS Server, FTP Server, DHCP Server가 있다.

Web Server와 FTP Server는 Windows XP의 IIS(Internet Information Service)를 사용하여 구축하였고, DNS Server는 Ubuntu 환경에서 bind9를 통해, Mail Server는 Ubuntu(Linux Mint) 환경에서 Postfix를 통해, DHCP Server는 CentOS 환경에서 각각 구축하였다.

서버 종류	구축 환경 및 방법
Mail Server	Ubuntu(Linux Mint Mate) 32bit - Postfix
Web Server	Windows XP Professional 32bit - IIS(Internet Information Service)
DNS Server	Ubuntu 12.0.4 server 32bit - bind9
FTP Server	Windows XP Professional 32bit - IIS(Internet Information Service)
DHCP Server	CentOS 6.7 minimal 32bit - dhcp

### 3장. 네트워크 구성

#### 3.1 LAN 토폴로지

본 프로젝트에서는 다음 그림 3.1.1과 같이 4대의 라우터를 이더넷을 이용하여 스타(star) 형태의 토폴로지로 구성하였다.

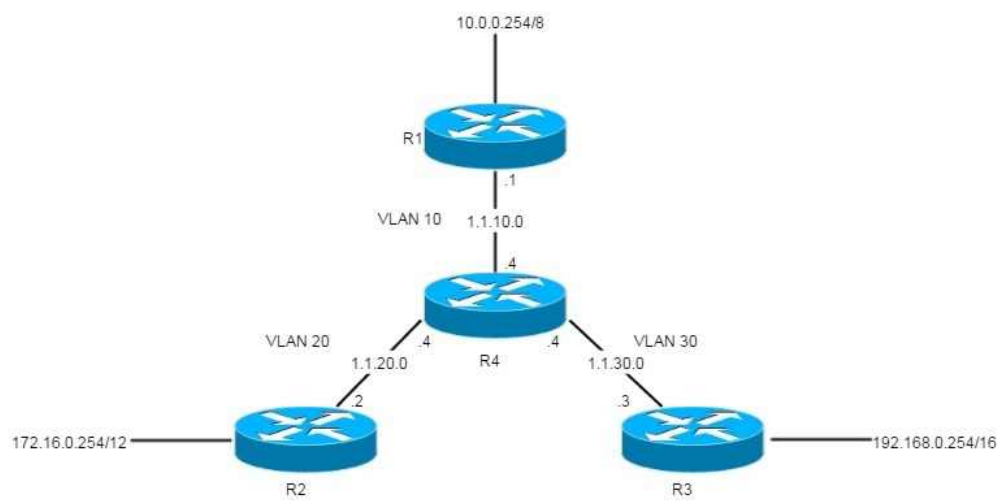


그림 3.1.1 스타 토폴로지

그림 3.1.1을 보면 R4에서 다른 라우터로의 연결이 VLAN으로 연결되어 있음을 알 수 있다. 즉, 그림 3.1.1의 물리적인 토폴로지는 다음 그림 3.1.2와 같다.

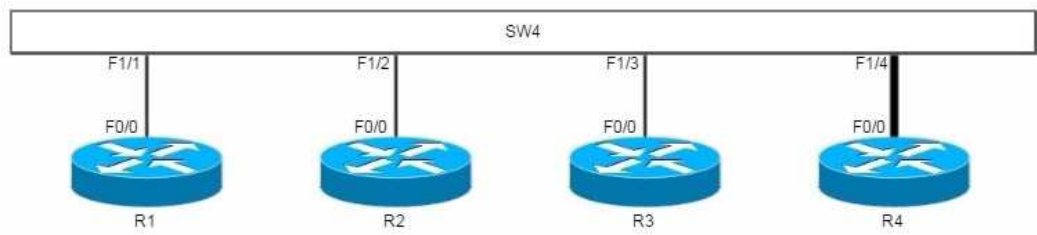


그림 3.1.2 물리적인 토폴로지



각 라우터의 연결은 SW4을 거치며 R1, R2, R3의 F0/0 인터페이스와 연결되는 스위치 측의 포트들은 각각 하나의 VLAN에만 소속되는 액세스 포트<sup>3)</sup>이고, R4의 F0/0 인터페이스와 연결되는 스위치 측의 F1/4 포트는 동시에 복수개의 VLAN 연결을 지원하는 트렁크 포트<sup>4)</sup>로 설정해야한다.

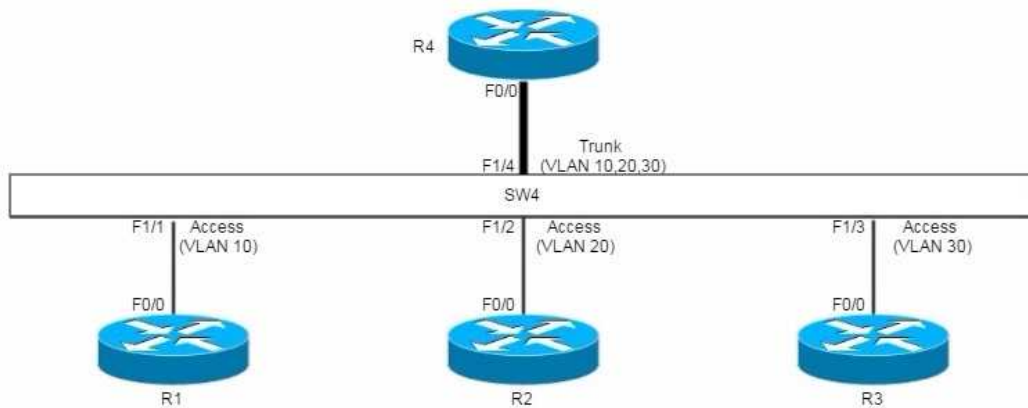


그림 3.1.3 액세스 포트와 트렁크 포트

이와 같은 구성을 한 이유는 R4에 NM-4T 모듈을 장착하여 각 라우터를 시리얼 포트로 연결하게 되면 복수 개의 인터페이스가 필요하게 되는데, 이를 방지하기 위함이 크다. 그리고 VLAN의 장점이 많기 때문에 VLAN 및 트렁킹 설정의 이해를 높이기 위함도 있다.

3) 하나의 VLAN에만 소속되는 포트

4) 복수개의 VLAN에 소속되는 포트

## 3.2 구성도

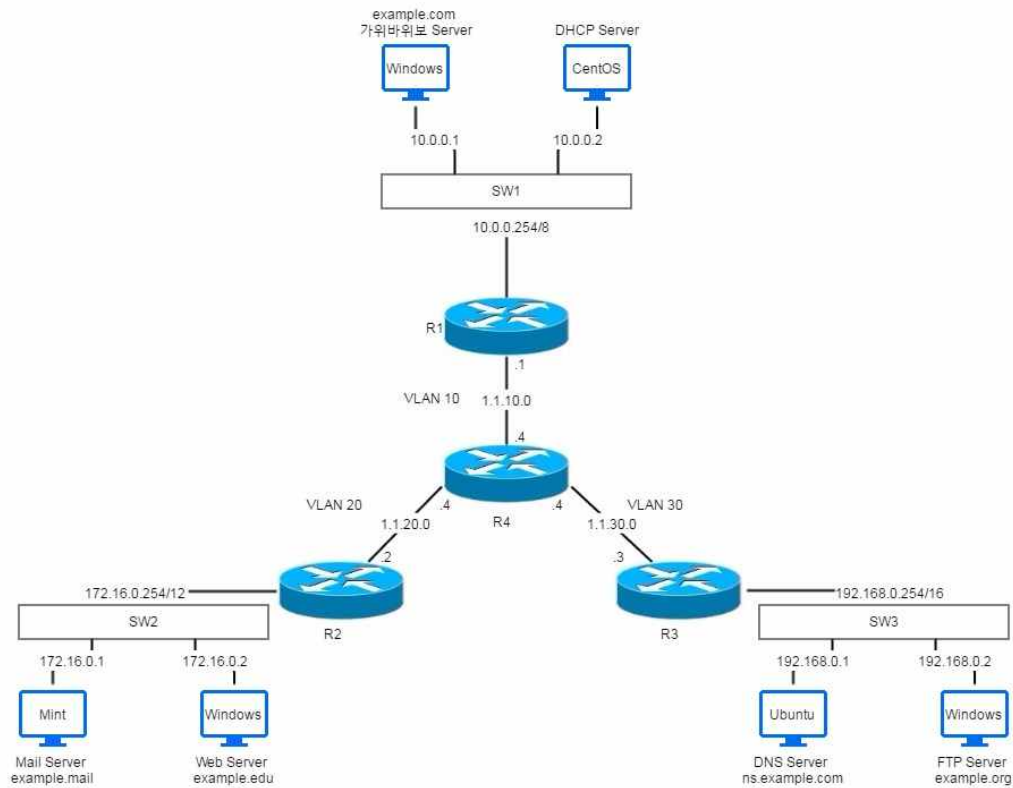


그림 3.2.1 전체적인 구성도

본 프로젝트에서 구축한 소규모 네트워크의 전체적인 구성도는 그림 3.2.1과 같다. 각 종단 시스템은 서로 통신이 가능하며 DHCP Server가 구축된 호스트를 제외하고는 각자 도메인을 가지고 있다. 그리고 DHCP Server가 있는 10.0.0.0 네트워크의 IP는 동적으로 할당이 가능하다.

## 4장. 서버 구축 과정

### 4.1 가위바위보 Server

가위바위보 Server는 지난 TermProject1의 산출 프로그램이다. 가위바위보 Server의 IP를 구축된 소규모 네트워크의 적당한 IP로 설정하고 Server 및 Client 소스 파일을 Release 모드로 컴파일하여 배포하였다. 이미 로컬에서 동작이 검증된 프로그램이기에 구축에 큰 어려운 점은 없었다.

### 4.2 Mail Server

Mail Server는 우분투 기반의 Linux Mint 환경에서 Postfix를 사용해 구축하였다. 그리고 메일 서버에 있는 메일을 가져오기 위한 IMAP, POP3 서버로 dovecot를 사용하였다. 그 다음 SASL을 이용한 SMTP 인증을 위한 설정을 했는데 굉장히 복잡했다. 설치하는 apt-get을 통해 쉽게 하였지만, 설정에 있어서 구축한 서버 중 가장 어려웠던 서버이다.

### 4.3 Web Server

Web Server는 과거에 아파치를 통해 구축해본 경험이 있으므로 이번에는 IIS(Internet Information Service)를 사용하여 구축해보기로 하였다. Windows XP Professional 이상에서 기본적으로 IIS를 제공하는데 클릭 몇 번으로 여러 가지 서버를 한 번에 설치할 수 있고, 또한 GUI 제공으로 인해 쉽게 설정 할 수 있었다.

### 4.4 DNS Server

구축한 서버 중에서 구축하는 데 있어서 가장 재미있는 서버가 아니었나 생각해본다. 구축 환경은 Ubuntu이고, bind9를 사용하였다. DNS 서버를 직접 구축해봄으로써 forward name resolution<sup>5)</sup>과 reverse name resolution<sup>6)</sup>의 개념을 배웠고 더 나아가 각 방법의 zone을 만드는 방법 또한 학습하였다.

---

5) 정방향 이름 풀이, 도메인을 ip주소로 변환하는 과정

6) 역방향 이름 풀이, ip주소를 도메인으로 변환하는 과정

## 4.5 FTP Server

FTP Server 또한 Windows에서 제공하는 IIS를 통해 구축하였으며, 처음 구축해봄에도 불구하고 어려운 점이 없었다.

## 4.6 DHCP Server

DHCP Server는 CentOS 환경에서 구축하였다. 설정은 비교적 쉬운 편이었고 동작도 잘 확인하였다. 하지만 문제점은 IP가 동적으로 할당됨에 따라 DNS 서버의 내용이 무효화 된다는 점이다. 이 문제점을 해결하기 위해서는 Dynamic DNS(DDNS)를 설정해야한다.

## 5장. 동작 예시

### 5.1 라우터 간 통신 확인

라우터 간 통신은 ping을 통해 확인할 수 있다. ping을 통한 통신 확인은 각 라우터에서 다른 모든 라우터로 해보아야 확실하겠지만, 여기서는 여러 라우팅 프로토콜 간 라우팅 테이블에 중점을 두기로 하고 ping 테스트는 간단하게 선보인다.

#### 5.1.1 OSPF

라우터	OSPF 라우팅 테이블
R1	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0 O    1.1.20.0 [110/2] via 1.1.10.4, 04:13:56, FastEthernet0/0 O    1.1.30.0 [110/2] via 1.1.10.4, 04:13:56, FastEthernet0/0 C    10.0.0.0/8 is directly connected, FastEthernet0/1 O    172.16.0.0/12 [110/3] via 1.1.10.4, 04:13:56, FastEthernet0/0 O    192.168.0.0/16 [110/3] via 1.1.10.4, 04:13:56, FastEthernet0/0 </pre>
R2	<pre> 1.0.0.0/24 is subnetted, 3 subnets O    1.1.10.0 [110/2] via 1.1.20.4, 04:03:02, FastEthernet0/0 C    1.1.20.0 is directly connected, FastEthernet0/0 O    1.1.30.0 [110/2] via 1.1.20.4, 04:03:02, FastEthernet0/0 O    10.0.0.0/8 [110/3] via 1.1.20.4, 04:03:02, FastEthernet0/0 C    172.16.0.0/12 is directly connected, FastEthernet0/1 O    192.168.0.0/16 [110/3] via 1.1.20.4, 04:03:02, FastEthernet0/0 </pre>
R3	<pre> 1.0.0.0/24 is subnetted, 3 subnets O    1.1.10.0 [110/2] via 1.1.30.4, 04:23:42, FastEthernet0/0 O    1.1.20.0 [110/2] via 1.1.30.4, 04:23:42, FastEthernet0/0 C    1.1.30.0 is directly connected, FastEthernet0/0 O    10.0.0.0/8 [110/3] via 1.1.30.4, 04:23:42, FastEthernet0/0 O    172.16.0.0/12 [110/3] via 1.1.30.4, 04:23:43, FastEthernet0/0 C    192.168.0.0/16 is directly connected, FastEthernet0/1 </pre>
R4	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0.1 C    1.1.20.0 is directly connected, FastEthernet0/0.2 C    1.1.30.0 is directly connected, FastEthernet0/0.3 O    10.0.0.0/8 [110/2] via 1.1.10.1, 04:03:05, FastEthernet0/0.1 O    172.16.0.0/12 [110/2] via 1.1.20.2, 04:03:05, FastEthernet0/0.2 O    192.168.0.0/16 [110/2] via 1.1.30.3, 04:03:05, FastEthernet0/0.3 </pre>
<pre> R1#ping 172.16.0.254 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 172.16.0.254, timeout is 2 seconds: !!!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 12/19/40 ms </pre>	

그림 5.1.1 OSPF 라우팅 테이블

## 5.1.2 RIP

라우터	RIP 라우팅 테이블
R1	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0 R    1.1.20.0 [120/1] via 1.1.10.4, 00:00:19, FastEthernet0/0 R    1.1.30.0 [120/1] via 1.1.10.4, 00:00:19, FastEthernet0/0 C    10.0.0.0/8 is directly connected, FastEthernet0/1 R    172.16.0.0/12 [120/2] via 1.1.10.4, 00:00:19, FastEthernet0/0 R    192.168.0.0/16 [120/2] via 1.1.10.4, 00:00:19, FastEthernet0/0 </pre>
R2	<pre> 1.0.0.0/24 is subnetted, 3 subnets R    1.1.10.0 [120/1] via 1.1.20.4, 00:00:24, FastEthernet0/0 C    1.1.20.0 is directly connected, FastEthernet0/0 R    1.1.30.0 [120/1] via 1.1.20.4, 00:00:24, FastEthernet0/0 R    10.0.0.0/8 [120/2] via 1.1.20.4, 00:00:24, FastEthernet0/0 C    172.16.0.0/12 is directly connected, FastEthernet0/1 R    192.168.0.0/16 [120/2] via 1.1.20.4, 00:00:24, FastEthernet0/0 </pre>
R3	<pre> 1.0.0.0/24 is subnetted, 3 subnets R    1.1.10.0 [120/1] via 1.1.30.4, 00:00:03, FastEthernet0/0 R    1.1.20.0 [120/1] via 1.1.30.4, 00:00:03, FastEthernet0/0 C    1.1.30.0 is directly connected, FastEthernet0/0 R    10.0.0.0/8 [120/2] via 1.1.30.4, 00:00:03, FastEthernet0/0 R    172.16.0.0/12 [120/2] via 1.1.30.4, 00:00:03, FastEthernet0/0 C    192.168.0.0/16 is directly connected, FastEthernet0/1 </pre>
R4	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0.1 C    1.1.20.0 is directly connected, FastEthernet0/0.2 C    1.1.30.0 is directly connected, FastEthernet0/0.3 R    10.0.0.0/8 [120/1] via 1.1.10.1, 00:00:11, FastEthernet0/0.1 R    172.16.0.0/12 [120/1] via 1.1.20.2, 00:00:21, FastEthernet0/0.2 R    192.168.0.0/16 [120/1] via 1.1.30.3, 00:00:14, FastEthernet0/0.3 </pre>
<pre> R2#ping 192.168.0.254  Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 192.168.0.254, timeout is 2 seconds: !!!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 8/29/100 ms </pre>	

그림 5.1.2 RIP 라우팅 테이블



### 5.1.3 BGP

라우터	BGP 라우팅 테이블
R1	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0 B    1.1.20.0 [20/0] via 1.1.10.4, 00:05:57 B    1.1.30.0 [20/0] via 1.1.10.4, 00:05:57 C    10.0.0.0/8 is directly connected, FastEthernet0/1 B    172.16.0.0/12 [20/0] via 1.1.10.4, 00:05:56 B    192.168.0.0/16 [20/0] via 1.1.10.4, 00:05:57 </pre>
R2	<pre> 1.0.0.0/24 is subnetted, 3 subnets B    1.1.10.0 [20/0] via 1.1.20.4, 00:06:02 C    1.1.20.0 is directly connected, FastEthernet0/0 B    1.1.30.0 [20/0] via 1.1.20.4, 00:06:02 B    10.0.0.0/8 [20/0] via 1.1.20.4, 00:06:02 C    172.16.0.0/12 is directly connected, FastEthernet0/1 B    192.168.0.0/16 [20/0] via 1.1.20.4, 00:06:02 </pre>
R3	<pre> 1.0.0.0/24 is subnetted, 3 subnets B    1.1.10.0 [20/0] via 1.1.30.4, 00:06:04 B    1.1.20.0 [20/0] via 1.1.30.4, 00:06:04 C    1.1.30.0 is directly connected, FastEthernet0/0 B    10.0.0.0/8 [20/0] via 1.1.30.4, 00:06:04 B    172.16.0.0/12 [20/0] via 1.1.30.4, 00:06:04 C    192.168.0.0/16 is directly connected, FastEthernet0/1 </pre>
R4	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0.1 C    1.1.20.0 is directly connected, FastEthernet0/0.2 C    1.1.30.0 is directly connected, FastEthernet0/0.3 B    10.0.0.0/8 [20/0] via 1.1.10.1, 00:06:06 B    172.16.0.0/12 [20/0] via 1.1.20.2, 00:06:06 B    192.168.0.0/16 [20/0] via 1.1.30.3, 00:06:06 </pre>
<pre> R3#ping 10.0.0.254 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 10.0.0.254, timeout is 2 seconds: !!!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 8/50/80 ms </pre>	

그림 5.1.3 BGP 라우팅 테이블

## 5.1.4 Static

라우터	Static 라우팅 테이블
R1	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0 S    1.1.20.0 is directly connected, FastEthernet0/0 S    1.1.30.0 is directly connected, FastEthernet0/0 C    10.0.0.0/8 is directly connected, FastEthernet0/1 S    172.16.0.0/12 is directly connected, FastEthernet0/0 S    192.168.0.0/16 is directly connected, FastEthernet0/0 </pre>
R2	<pre> 1.0.0.0/24 is subnetted, 3 subnets S    1.1.10.0 is directly connected, FastEthernet0/0 C    1.1.20.0 is directly connected, FastEthernet0/0 S    1.1.30.0 is directly connected, FastEthernet0/0 S    10.0.0.0/8 is directly connected, FastEthernet0/0 C    172.16.0.0/12 is directly connected, FastEthernet0/1 S    192.168.0.0/16 is directly connected, FastEthernet0/0 </pre>
R3	<pre> 1.0.0.0/24 is subnetted, 3 subnets S    1.1.10.0 is directly connected, FastEthernet0/0 S    1.1.20.0 is directly connected, FastEthernet0/0 C    1.1.30.0 is directly connected, FastEthernet0/0 S    10.0.0.0/8 is directly connected, FastEthernet0/0 S    172.16.0.0/12 is directly connected, FastEthernet0/0 C    192.168.0.0/16 is directly connected, FastEthernet0/1 </pre>
R4	<pre> 1.0.0.0/24 is subnetted, 3 subnets C    1.1.10.0 is directly connected, FastEthernet0/0.1 C    1.1.20.0 is directly connected, FastEthernet0/0.2 C    1.1.30.0 is directly connected, FastEthernet0/0.3 S    10.0.0.0/8 is directly connected, FastEthernet0/0.1 S    172.16.0.0/12 is directly connected, FastEthernet0/0.2 S    192.168.0.0/16 is directly connected, FastEthernet0/0.3 </pre>
<pre> R4#ping 192.168.0.254 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 192.168.0.254, timeout is 2 seconds: !!!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 8/20/60 ms </pre>	

그림 5.1.4 Static 라우팅 테이블



## 5.2 종단 시스템 간 통신 확인

종단 시스템 간 통신 확인은 `tracert/traceroute`(Windows/Linux)<sup>7)</sup>를 통해 패킷이 전달되는 경로를 추적하고, 그 결과를 간단하게 살펴보도록 한다.

### 5.2.1 같은 네트워크에 속하는 종단 시스템

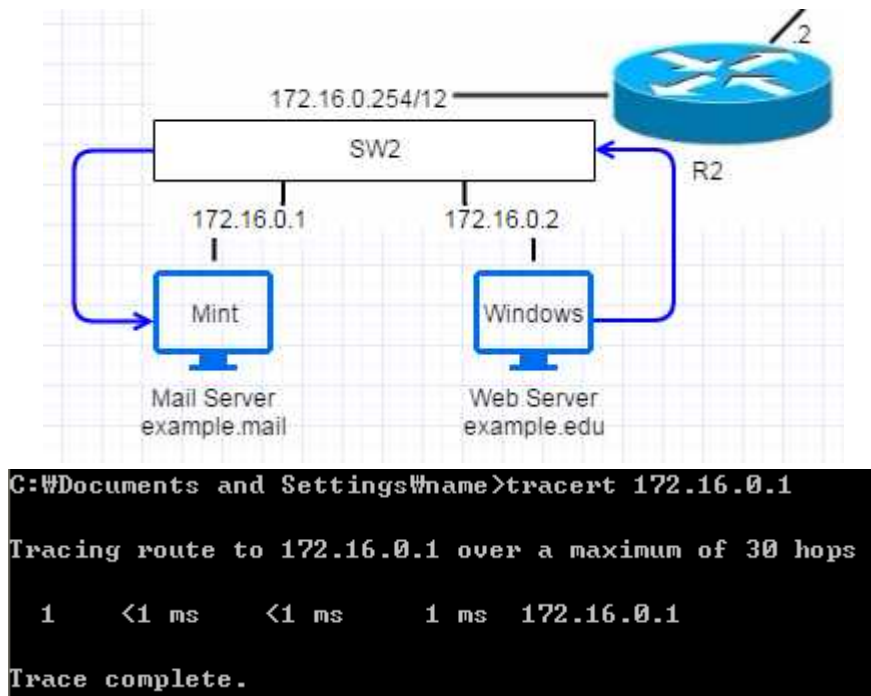


그림 5.2.1 패킷 경로 추적(같은 네트워크)

7) 지정된 호스트에 도달할 때까지 통과하는 경로의 정보와 각 경로에서의 지연 시간을 추적하는 명령.

## 5.2.2 다른 네트워크에 속하는 종단 시스템

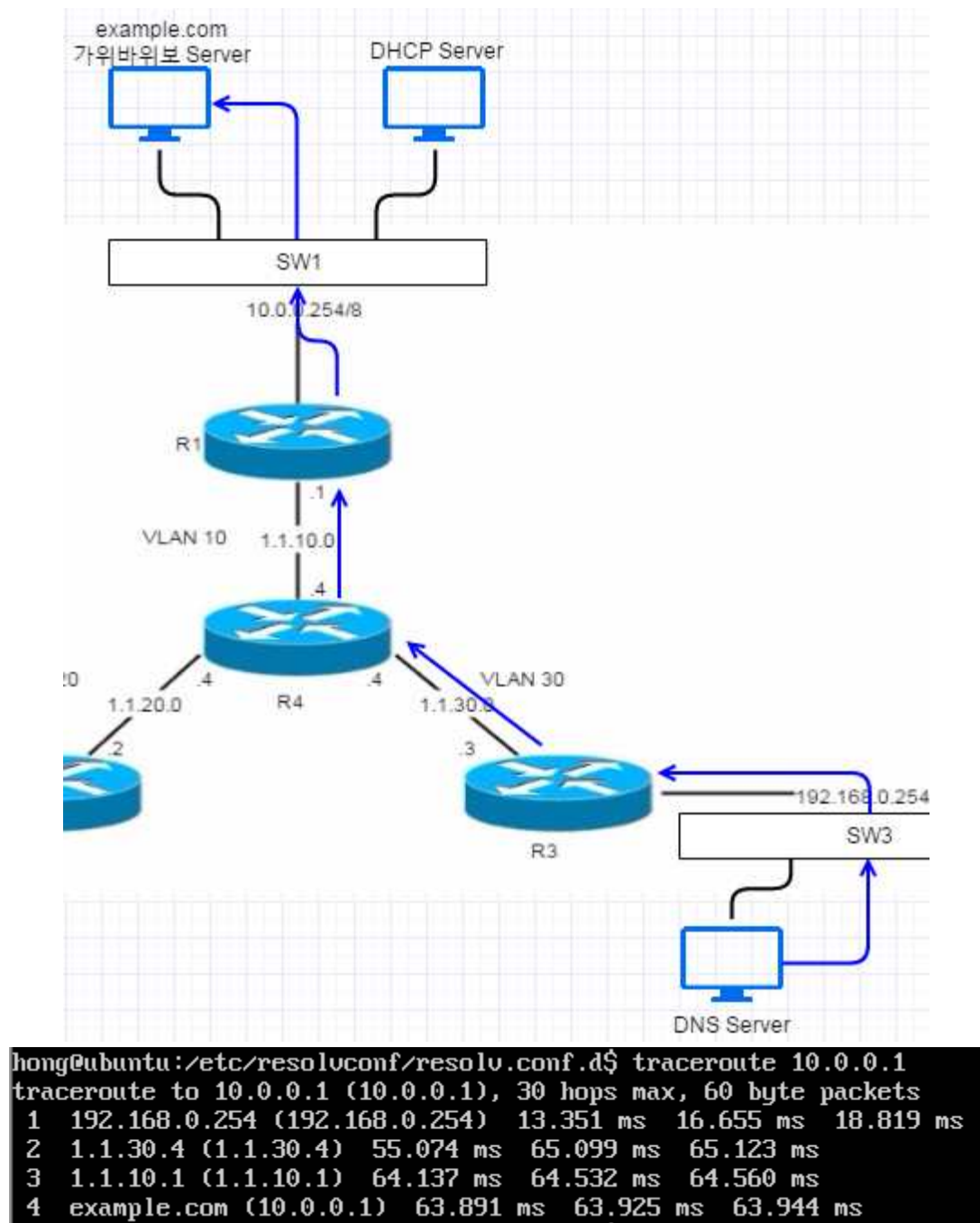


그림 5.2.2 패킷 경로 추적(다른 네트워크)

## 5.3 각 서버의 동작 확인

### 5.3.1 가위바위보 Server

가위바위보 Server의 동작 확인은 그림 5.3.1과 같이 두 명의 Player가 하나의 서버에 접속하여 가위바위보 게임을 하는 시나리오로 보인다.

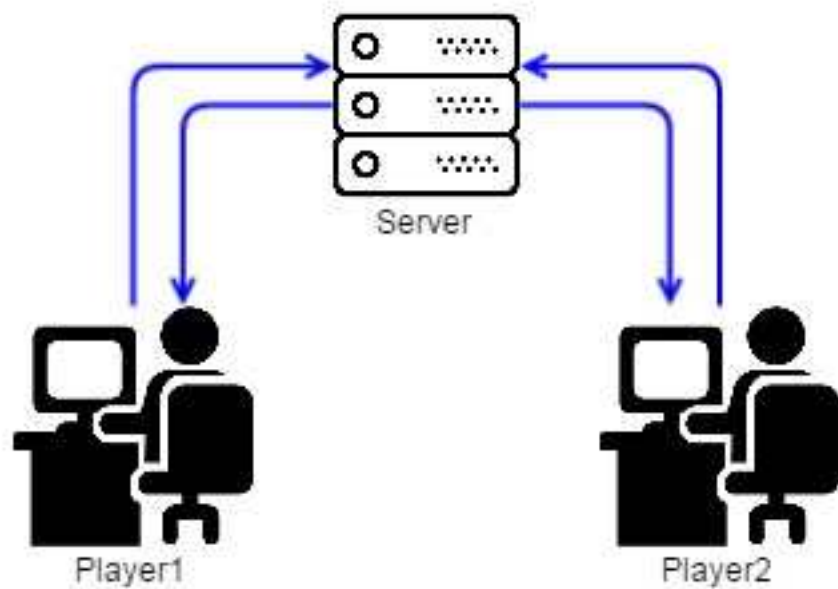


그림 5.3.1 가위바위보 Server

먼저 서버를 동작 시킨다. 그림 5.3.2는 가위바위보 Server를 동작시키고 Player들의 연결을 기다리고 있는 화면이다.

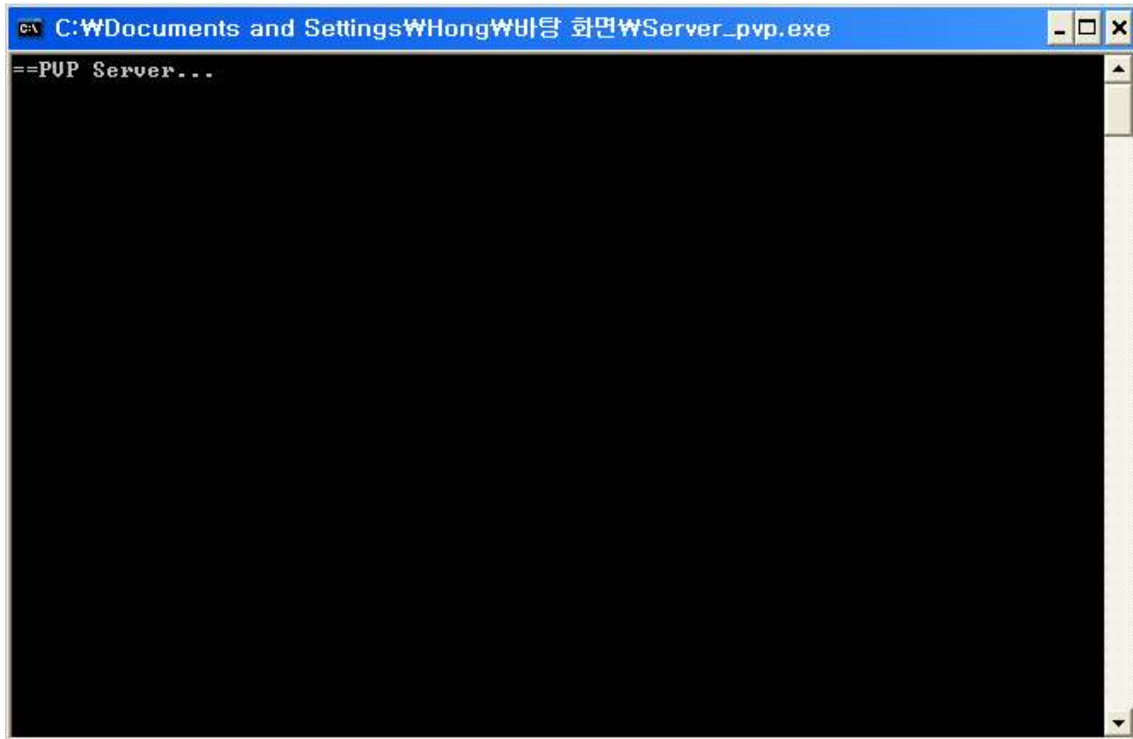


그림 5.3.2 가위바위보 Server 동작 화면

그림 5.3.3은 Player1이 먼저 접속하여 Player2를 대기하는 화면이고, 그림 5.3.4는 Player2도 접속하여 연결이 완료된 화면이다. 가위바위보 선택을 기다리고 있다.

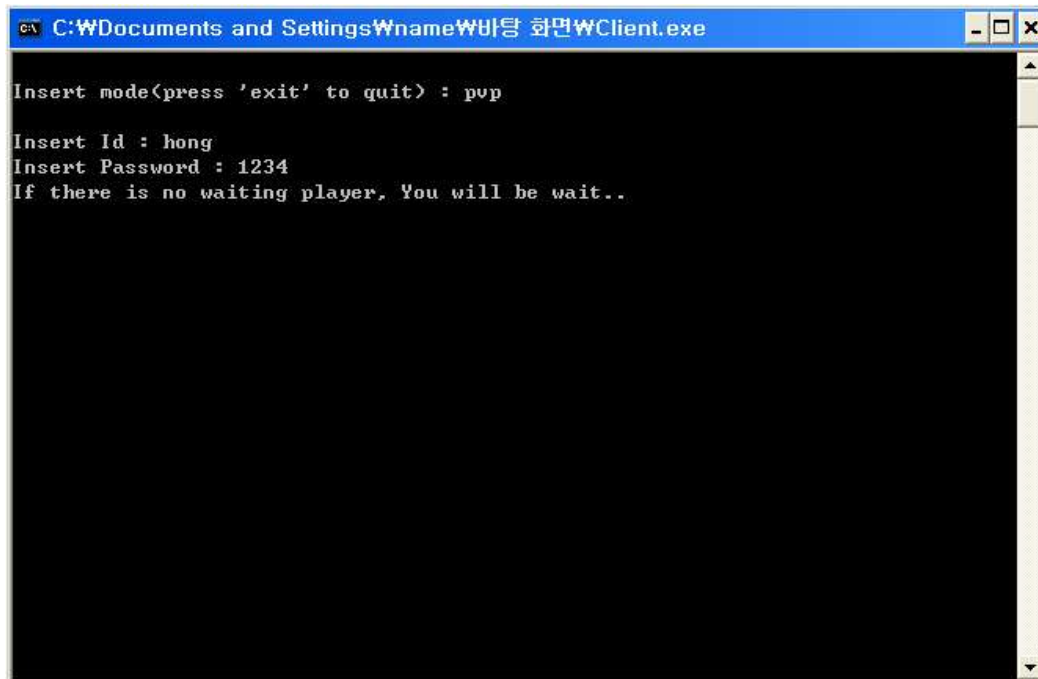


그림 5.3.3 Player1 접속 화면

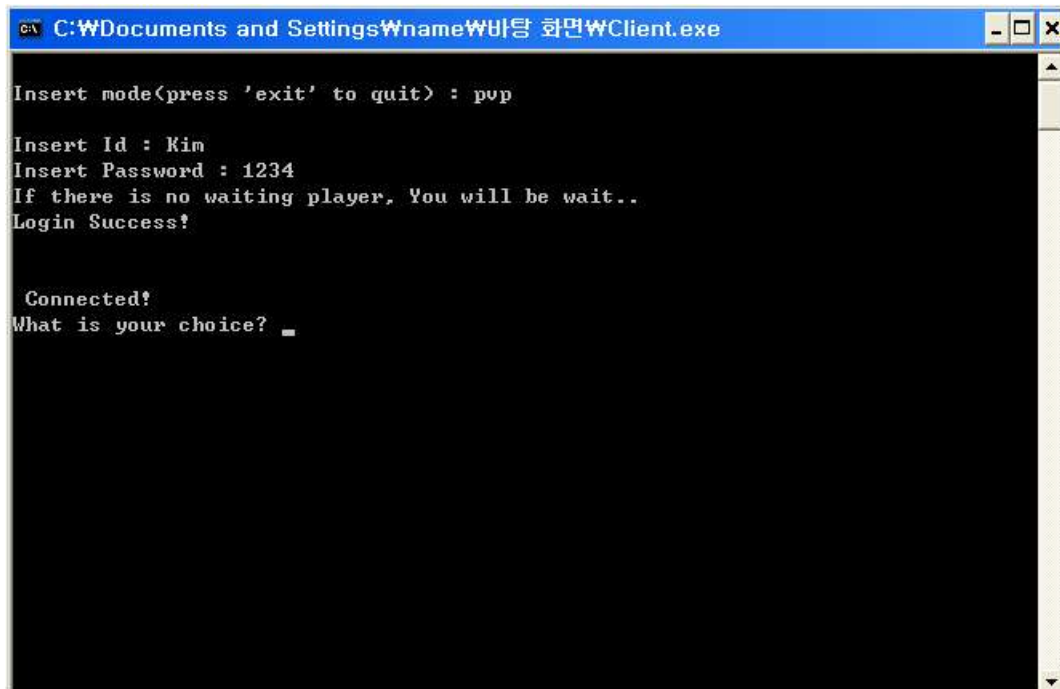
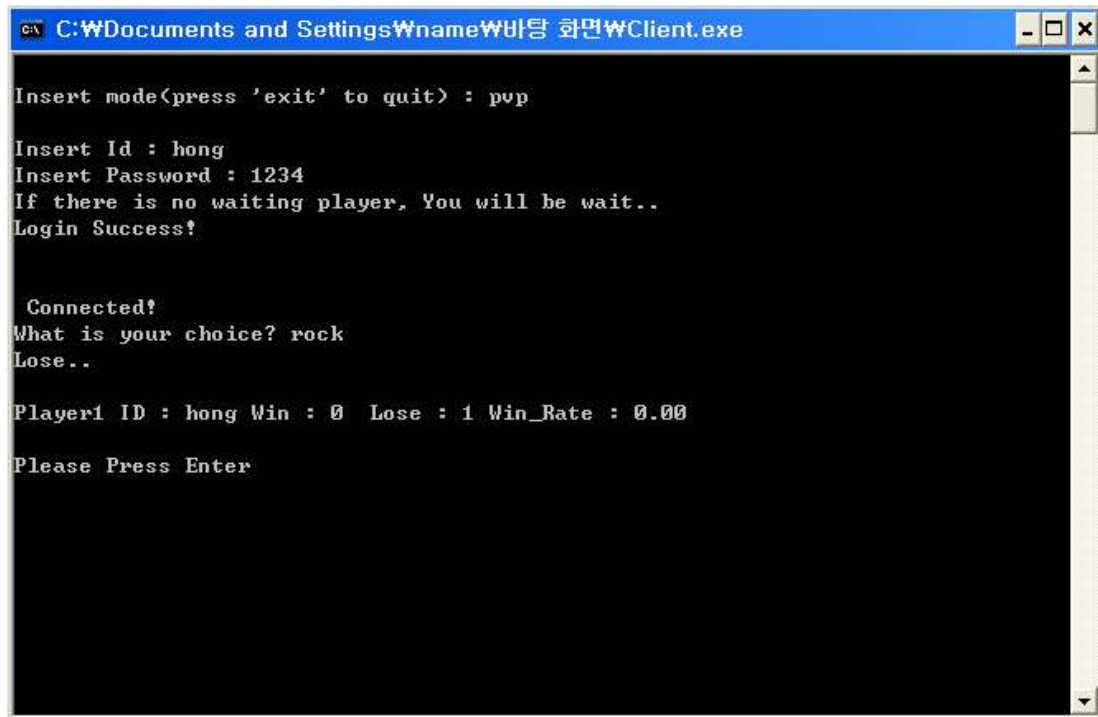


그림 5.3.4 Player2 접속 화면

그림 5.3.5와 그림 5.3.6는 두 Player가 가위바위보 선택을 마친 후 서버로부터 결과를 받아 출력하고 있는 화면이다.



```
C:\WDocuments and Settings\Wname\바탕 화면\WClient.exe

Insert mode<press 'exit' to quit> : pvp

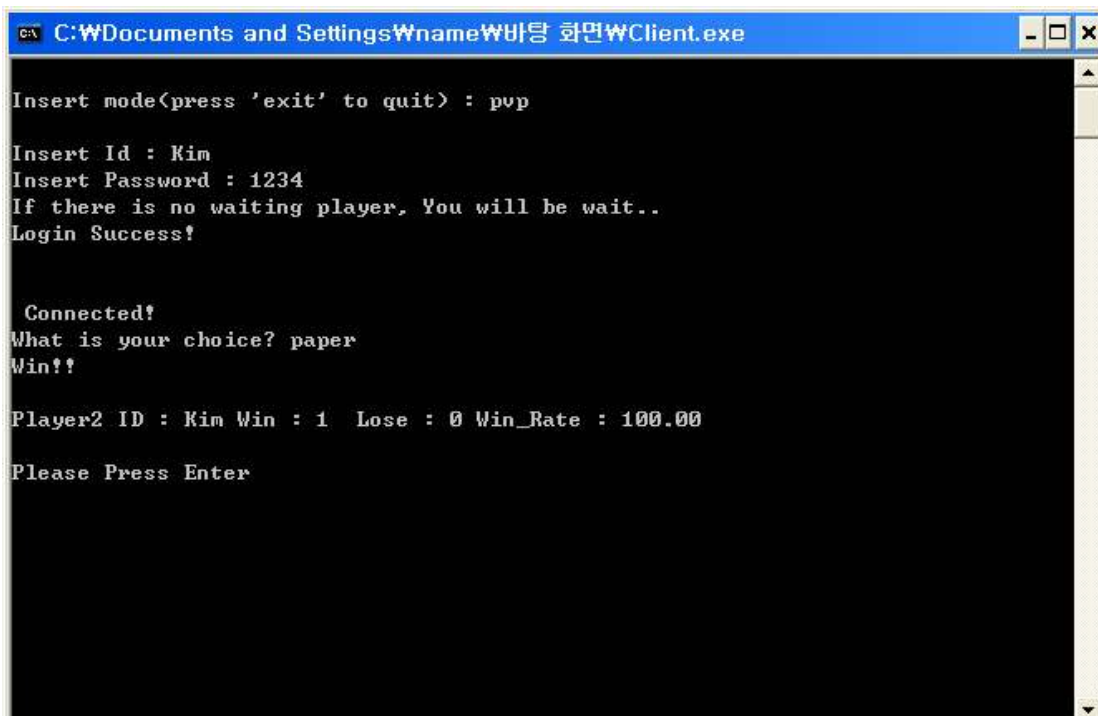
Insert Id : hong
Insert Password : 1234
If there is no waiting player, You will be wait..
Login Success!

Connected!
What is your choice? rock
Lose..

Player1 ID : hong Win : 0 Lose : 1 Win_Rate : 0.00

Please Press Enter
```

그림 5.3.5 Player1 결과 화면



```
C:\WDocuments and Settings\Wname\바탕 화면\WClient.exe

Insert mode<press 'exit' to quit> : pvp

Insert Id : Kim
Insert Password : 1234
If there is no waiting player, You will be wait..
Login Success!

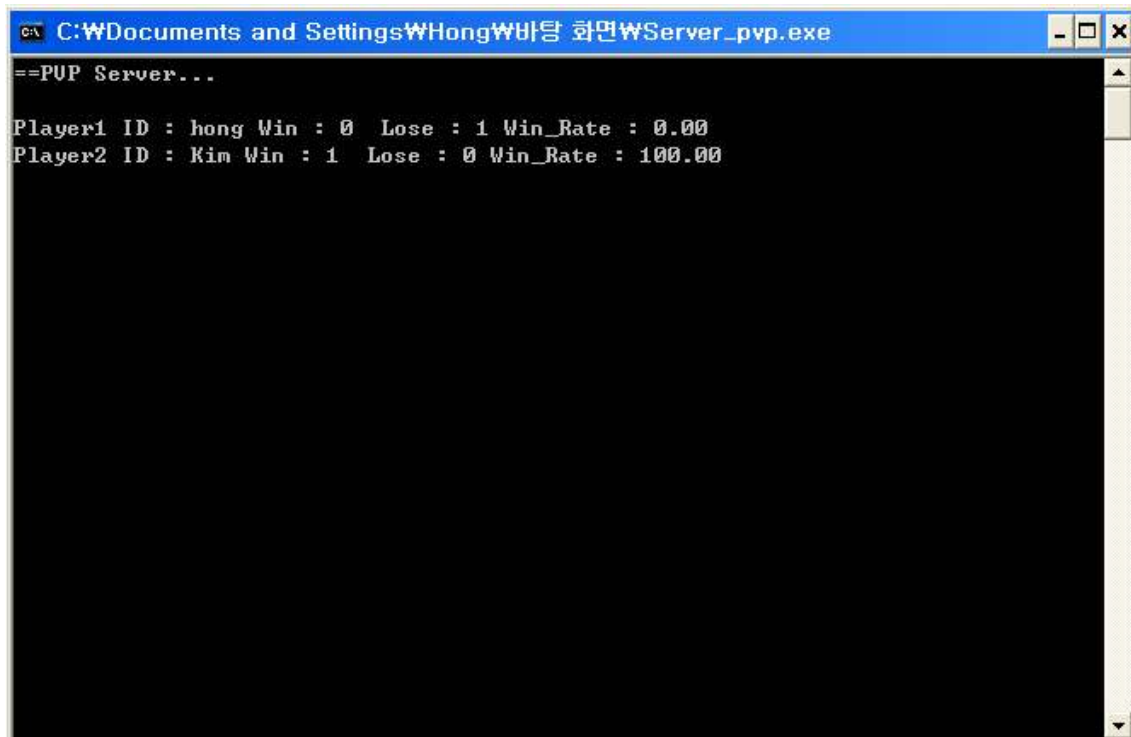
Connected!
What is your choice? paper
Win!!

Player2 ID : Kim Win : 1 Lose : 0 Win_Rate : 100.00

Please Press Enter
```

그림 5.3.6 Player2 결과 화면

이후 그림 5.3.7과 같이 서버는 Player의 로그인 정보와 승패 결과를 저장하고 출력한다.



```
C:\Documents and Settings\WHong\바탕 화면\WServer_pvp.exe
==PVP Server...
Player1 ID : hong Win : 0 Lose : 1 Win_Rate : 0.00
Player2 ID : Kim Win : 1 Lose : 0 Win_Rate : 100.00
```

그림 5.3.7 가위바위보 Server 결과 화면

### 5.3.2 Mail Server

Mail Server의 동작은 telnet을 이용하여 다른 호스트에서 메일 서버가 구축된 호스트로 메일을 보냄으로써 확인한다.

```
hong@ubuntu:~$ telnet 172.16.0.1 25
Trying 172.16.0.1...
Connected to 172.16.0.1.
Escape character is '^I'.
220 example.mail ESMTP Postfix (Ubuntu)
-
```

그림 5.3.8 메일 서버 접속 화면

그림 5.3.8은 우분투 환경에서 telnet 25번 포트를 사용하여 원격으로 접속하는 것을 볼 수 있다.

```
ehlo hi
250-example.mail
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: <rkfaorl000@naver.com>
250 2.1.0 Ok
rcpt to: <hong@example.mail>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: This is...

This is a Test Mail.
.
250 2.0.0 Ok: queued as A1E86101846
quit
221 2.0.0 Bye
Connection closed by foreign host.
hong@ubuntu:~$ _
```

그림 5.3.9 메일 전송 화면

이후 그림 5.3.9에서처럼 정해진 프로토콜에 따라 메일을 전송한다.



```

hong@hong-virtual-machine ~/Maildir/new $ ls
1463926997.V801I23881M248110.hong-virtual-machine
hong@hong-virtual-machine ~/Maildir/new $ cat 1463926997.V801I23881M248110.hong-virtual-machine
Return-Path: <rkfaor1000@naver.com>
X-Original-To: hong@example.mail
Delivered-To: hong@example.mail
Received: from hi (ns.example.com [192.168.0.1])
        by example.mail (Postfix) with ESMTP id A1E86101846
        for <hong@example.mail>; Sun, 22 May 2016 23:22:25 +0900 (KST)
Subject: This is...

This is a Test Mail.
hong@hong-virtual-machine ~/Maildir/new $

```

그림 5.3.10 메일 확인 화면

그림 5.3.10에서는 수신된 메일을 확인하는 화면을 보여준다.

### 5.3.3 Web Server

다른 호스트의 Web browser를 이용해 Web Server에 있는 HTML 문서를 화면에 출력함으로써 동작을 확인한다.

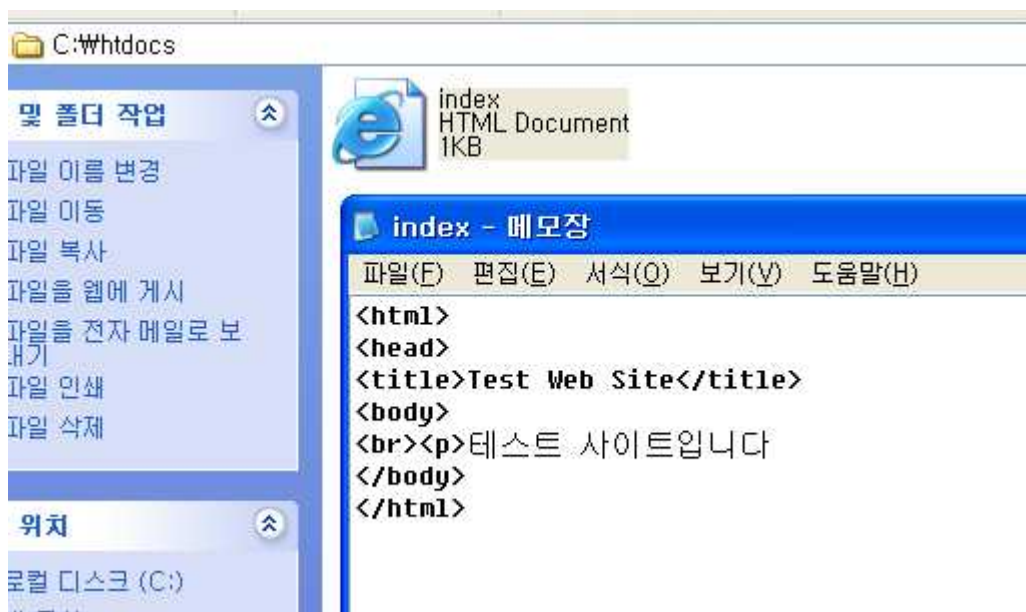


그림 5.3.11 Web Server에 있는 HTML 문서



그림 5.3.12 Web Browser에서 HTML 문서 출력

### 5.3.4 FTP Server

다른 호스트를 이용하여 FTP Server로 접속하여 파일 내용을 확인함으로써 서버 동작을 확인한다.

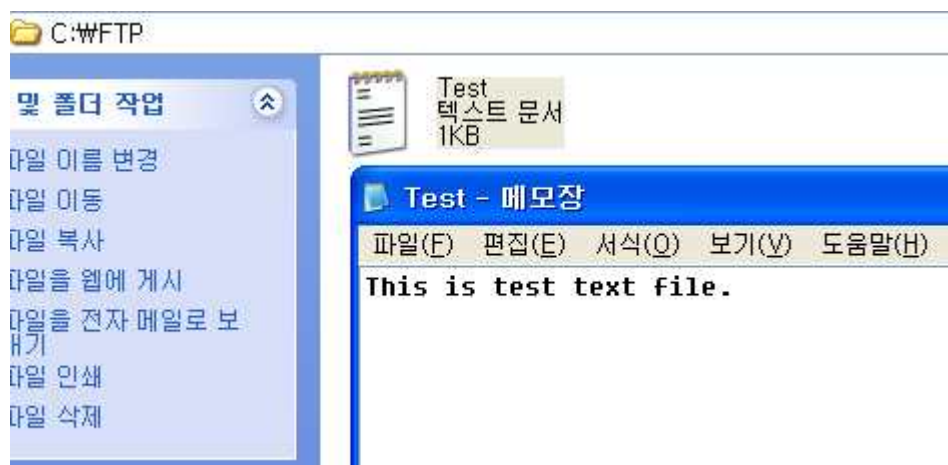


그림 5.3.13 FTP 서버에 있는 Test 파일

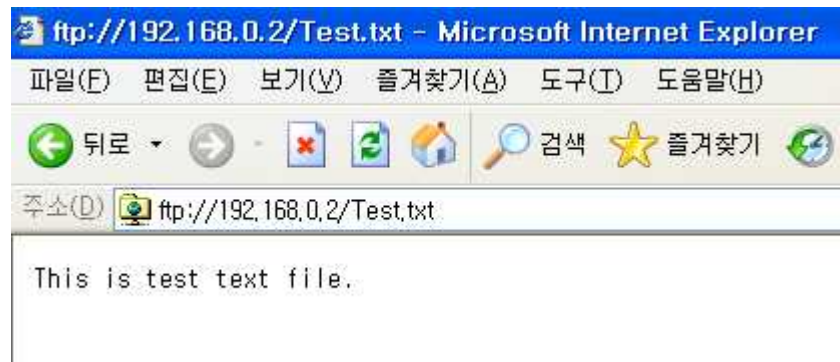


그림 5.3.14 FTP 서버에 접속하여 Test 파일을 읽는 화면

### 5.3.5 DNS Server

nslookup 명령어를 통해 각 도메인을 살펴보고, 다른 서버들에 적용함으로써 동작을 확인한다.

ns.example.com	<pre>&gt; ns.example.com Server: ns.example.com Address: 192.168.0.1  Name: ns.example.com Address: 192.168.0.1</pre>
example.org	<pre>&gt; example.org Server: ns.example.com Address: 192.168.0.1  Name: example.org Address: 192.168.0.2</pre>
example.com	<pre>&gt; example.com Server: ns.example.com Address: 192.168.0.1  Name: example.com Address: 10.0.0.1</pre>
example.mail	<pre>&gt; example.mail Server: ns.example.com Address: 192.168.0.1  Name: example.mail Address: 172.16.0.1</pre>
example.edu	<pre>&gt; example.edu Server: ns.example.com Address: 192.168.0.1  Name: example.edu Address: 172.16.0.2</pre>

그림 5.3.15 nslookup으로 살펴본 도메인



그림 5.3.16 도메인으로 접속한 Web서버

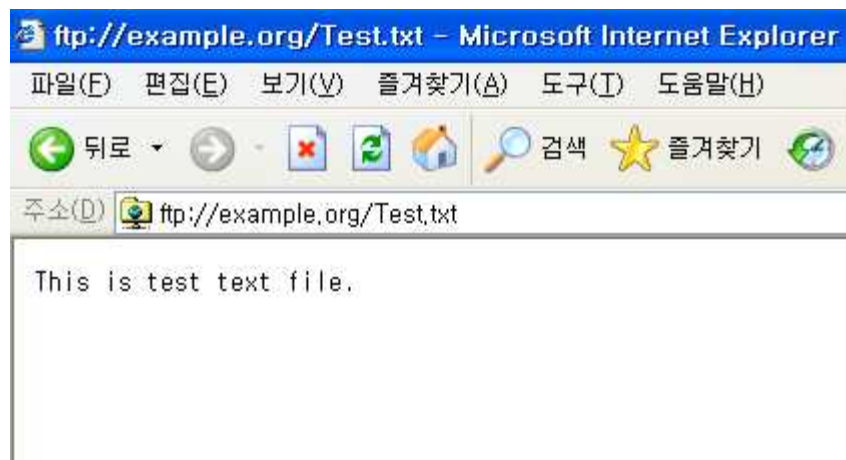


그림 5.3.17 도메인으로 접속한 FTP서버

### 5.3.6 DHCP Server

DHCP Server의 동작은 먼저 서버의 dhcp 설정 값을 살펴본 후에 클라이언트에서 아이피 주소 할당이 잘 되었는지를 확인한다.

```
# A slightly different configuration for an internal subnet.
subnet 10.0.0.0 netmask 255.0.0.0 {
  _ range 10.0.0.10 10.0.0.20;
  option domain-name-servers 192.168.0.1;
  option domain-name "ns.example.com";
  option routers 10.0.0.254;
  option broadcast-address 10.255.255.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

그림 5.3.18 dhcp 설정 값 (range 10.0.0.10~ 10.0.0.20)

The image shows a DHCP client configuration window. It has two main sections. The first section is for IP address settings, with a radio button selected for '자동으로 IP 주소 받기(O)' (Obtain IP address automatically). Below it, there are three input fields for '다음 IP 주소 사용(S):' (Use the following IP address): 'IP 주소(I):', '서브넷 마스크(U):', and '기본 게이트웨이(D):'. The second section is for DNS settings, with a radio button selected for '자동으로 DNS 서버 주소 받기(B)' (Obtain DNS server address automatically). Below it, there are two input fields for '다음 DNS 서버 주소 사용(E):' (Use the following DNS server address): '기본 설정 DNS 서버(P):' and '보조 DNS 서버(A):'. All input fields are currently empty.

그림 5.3.19 dhcp 사용 설정

```
C:\Documents and Settings\HONG>ipconfig /release

Windows IP Configuration

Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 0.0.0.0
    Subnet Mask . . . . .             : 0.0.0.0
    Default Gateway . . . . .         : 

C:\Documents and Settings\HONG>ipconfig /renew

Windows IP Configuration

Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix  . : ns.example.com
    IP Address. . . . .             : 10.0.0.10
    Subnet Mask . . . . .           : 255.0.0.0
    Default Gateway . . . . .       : 10.0.0.254
```

그림 5.3.20 IP 주소를 자동으로 할당받는 화면

## 6장 Wireshark를 통한 프로토콜 분석

### 6.1 HTTP 분석

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.1	172.16.0.2	TCP	62	1379 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	0.052254	172.16.0.2	10.0.0.1	TCP	62	80 → 1379 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1
3	0.053272	10.0.0.1	172.16.0.2	TCP	54	1379 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.053273	10.0.0.1	172.16.0.2	HTTP	421	GET / HTTP/1.1
5	0.100619	172.16.0.2	10.0.0.1	HTTP	243	HTTP/1.1 304 Not Modified
6	0.319419	10.0.0.1	172.16.0.2	TCP	54	1379 → 80 [ACK] Seq=368 Ack=190 Win=65346 Len=0
7	65.100258	10.0.0.1	172.16.0.2	TCP	54	1379 → 80 [RST, ACK] Seq=368 Ack=190 Win=0 Len=0

그림 6.1.1 Web Server 접속 시 교환되는 패킷

그림 6.1.1에서 보이는 것처럼 최초 Web Server 접속 시 먼저 TCP 연결을 위한 핸드셰이킹 단계를 거친다. 그 후 HTTP 프로토콜을 사용하여 HTML 문서를 요청하고 그에 응답하는 모습을 보인다. HTTP 프로토콜을 그림 6.1.2와 그림 6.1.3에서 조금 더 자세하게 살펴보자.

```
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, */*\r\n
  Accept-Language: ko\r\n
  Accept-Encoding: gzip, deflate\r\n
  If-Modified-Since: Wed, 18 May 2016 08:09:38 GMT\r\n
  If-None-Match: "fa54519fdbcb0d11:a22"\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)\r\n
  Host: example.edu\r\n
  Connection: Keep-Alive\r\n
\r\n
[Full request URI: http://example.edu/]
[HTTP request 1/1]
[Response in frame: 11]
```

그림 6.1.2 HTTP 요청 메시지

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
  Server: Microsoft-IIS/5.1\r\n
  Date: Sun, 22 May 2016 09:03:12 GMT\r\n
  Content-Location: http://example.edu/index.html\r\n
  ETag: "fa54519fdbcb0d11:a22"\r\n
  Content-Length: 0\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.251610000 seconds]
[Request in frame: 9]
```

그림 6.1.3 HTTP 응답 메시지



## 6.2 FTP 분석

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.1	192.168.0.2	TCP	62	1459 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
2	0.000014	192.168.0.2	10.0.0.1	TCP	62	21 → 1459 [SYN, ACK] Seq=0 Ack=1 Min=65535 Len=0 MSS=1460 SACK_PERM=1
3	0.090125	10.0.0.1	192.168.0.2	TCP	54	1459 → 21 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.164071	192.168.0.2	10.0.0.1	FTP	81	Response: 200-Microsoft FTP Service
5	0.332914	10.0.0.1	192.168.0.2	TCP	54	1459 → 21 [ACK] Seq=1 Ack=28 Win=65508 Len=0
6	0.383092	192.168.0.2	10.0.0.1	FTP	159	Response: -----
7	0.383278	10.0.0.1	192.168.0.2	FTP	70	Request: USER anonymous
8	0.397031	192.168.0.2	10.0.0.1	FTP	126	Response: 331 Anonymous access allowed, send identity (e-mail name) as password.
9	0.397213	10.0.0.1	192.168.0.2	FTP	68	Request: PASS IEUser@
10	0.454100	192.168.0.2	10.0.0.1	FTP	75	Response: 230-Login Successful
11	0.600928	10.0.0.1	192.168.0.2	TCP	54	1459 → 21 [ACK] Seq=31 Ack=226 Win=65310 Len=0
12	0.707657	192.168.0.2	10.0.0.1	FTP	99	Response: Welcome!
13	0.707971	10.0.0.1	192.168.0.2	FTP	68	Request: opts utf8 on
14	0.723138	192.168.0.2	10.0.0.1	FTP	98	Response: 500 'OPTS utf8 on': command not understood
15	0.723384	10.0.0.1	192.168.0.2	FTP	60	Request: syst
16	0.737066	192.168.0.2	10.0.0.1	FTP	70	Response: 215 Windows_NT
17	0.737223	10.0.0.1	192.168.0.2	FTP	65	Request: site help
18	0.781367	192.168.0.2	10.0.0.1	FTP	214	Response: 214-The following SITE commands are recognized(* ==>'s unimplemented).
19	0.781578	10.0.0.1	192.168.0.2	FTP	59	Request: PAD
20	0.842052	192.168.0.2	10.0.0.1	FTP	85	Response: 257 "/" is current directory.
21	0.938139	10.0.0.1	192.168.0.2	FTP	60	Request: noop
22	0.976125	192.168.0.2	10.0.0.1	FTP	84	Response: 200 NOOP command successful.
23	0.997580	10.0.0.1	192.168.0.2	FTP	61	Request: CWD /
24	1.027233	192.168.0.2	10.0.0.1	FTP	83	Response: 250 CWD command successful.
25	1.207966	10.0.0.1	192.168.0.2	TCP	54	1459 → 21 [ACK] Seq=80 Ack=581 Win=64955 Len=0
26	1.232729	10.0.0.1	192.168.0.2	FTP	62	Request: TYPE A
27	1.246054	192.168.0.2	10.0.0.1	FTP	74	Response: 200 Type set to A.
28	1.246517	10.0.0.1	192.168.0.2	FTP	60	Request: PASV
29	1.347178	192.168.0.2	10.0.0.1	FTP	102	Response: 227 Entering Passive Mode (192,168,0,2,4,100).
30	1.347471	10.0.0.1	192.168.0.2	TCP	62	1460 → 1132 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
31	1.361118	192.168.0.2	10.0.0.1	TCP	62	1132 → 1460 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1
32	1.361312	10.0.0.1	192.168.0.2	TCP	54	1460 → 1132 [ACK] Seq=1 Ack=1 Win=65535 Len=0
33	1.361490	10.0.0.1	192.168.0.2	FTP	60	Request: LIST
34	1.400759	192.168.0.2	10.0.0.1	FTP	100	Response: 125 Data connection already open; Transfer starting.
35	1.578865	192.168.0.2	10.0.0.1	FTP-DATA	103	FTP Data: 49 bytes
36	1.579141	192.168.0.2	10.0.0.1	TCP	54	1132 → 1460 [FIN, ACK] Seq=50 Ack=1 Win=65535 Len=0
37	1.579238	10.0.0.1	192.168.0.2	TCP	54	1460 → 1132 [ACK] Seq=1 Ack=51 Win=65486 Len=0
38	1.579325	10.0.0.1	192.168.0.2	TCP	54	1460 → 1132 [FIN, ACK] Seq=1 Ack=51 Win=65486 Len=0
39	1.630603	192.168.0.2	10.0.0.1	TCP	54	1132 → 1460 [ACK] Seq=51 Ack=2 Win=65535 Len=0
40	1.646007	10.0.0.1	192.168.0.2	TCP	54	1459 → 21 [ACK] Seq=100 Ack=703 Win=64833 Len=0
41	1.662105	192.168.0.2	10.0.0.1	FTP	78	Response: 226 Transfer complete.
42	1.864009	10.0.0.1	192.168.0.2	TCP	54	1459 → 21 [ACK] Seq=100 Ack=727 Win=64809 Len=0

그림 6.2.1 FTP Server 접속 시 교환되는 패킷

그림 6.2.1을 살펴보면 클라이언트 측에서 8)FTP 명령어를 전송하면 서버 측에서 응답하는 방식을 취하고 있다.

아래 그림 6.2.2와 그림 6.2.3에서 CWD 명령과 그에 해당하는 응답에 대한 패킷을 한 번 자세하게 살펴보자.

File Transfer Protocol (FTP)	File Transfer Protocol (FTP)
▲ CWD /\r\n Request command: CWD Request arg: /	▲ 250 CWD command successful.\r\n Response code: Requested file action okay, completed (250) Response arg: CWD command successful.

그림 6.2.2 FTP CWD 명령 및 응답 메시지

8) [https://en.wikipedia.org/wiki/List\\_of\\_FTP\\_commands](https://en.wikipedia.org/wiki/List_of_FTP_commands) 참조



## 6.3 SMTP 분석

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.1	172.16.0.1	TCP	74	41401 → 25 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=5658614 TSecr=0 WS=32
2	0.008002	172.16.0.1	192.168.0.1	TCP	74	25 → 41401 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=5658614 TSecr=5658614 WS=8
3	0.023549	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=1 Ack=1 Win=29216 Len=0 TSval=5658623 TSecr=3175809
4	0.305544	172.16.0.1	192.168.0.1	SMTP	107	S: 250 example.mail [SMTP Postfix (Ubuntu)]
5	0.381653	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=1 Ack=42 Win=29216 Len=0 TSval=5658713 TSecr=3175899
6	3.949727	192.168.0.1	172.16.0.1	SMTP	75	C: ehlo hi
7	3.949985	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [ACK] Seq=42 Ack=10 Win=14480 Len=0 TSval=3176797 TSecr=5659605
8	3.950144	172.16.0.1	192.168.0.1	SMTP	245	S: 250 example.mail   250 PIPELINING   250 SIZE 10240000   250 VRFY   250 ETRN   250 STARTTLS   250 AUTH PLAIN LOGIN   250 AUTH=PLAIN LOGIN   250 ENHANCEDSTATUS=
9	4.837852	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=10 Ack=221 Win=30272 Len=0 TSval=5659617 TSecr=3176797
10	13.331296	192.168.0.1	172.16.0.1	SMTP	101	C: mail from: <rkfaor1000@naver.com>
11	13.345006	172.16.0.1	192.168.0.1	SMTP	80	S: 250 2.1.0 Ok
12	13.400222	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=45 Ack=235 Win=30272 Len=0 TSval=5661959 TSecr=3179146
13	21.172446	192.168.0.1	172.16.0.1	SMTP	96	C: rcpt to: <hong@example.mail>
14	21.222677	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [ACK] Seq=235 Ack=75 Win=14480 Len=0 TSval=3181115 TSecr=5663910
15	21.226965	172.16.0.1	192.168.0.1	SMTP	80	S: 250 2.1.5 Ok
16	21.270292	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=75 Ack=240 Win=30272 Len=0 TSval=5663928 TSecr=3181116
17	22.476540	192.168.0.1	172.16.0.1	SMTP	72	C: data
18	22.476702	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [ACK] Seq=240 Ack=81 Win=14480 Len=0 TSval=3181428 TSecr=5664237
19	22.477903	172.16.0.1	192.168.0.1	SMTP	103	S: 354 End data with <R>[LF],<R>[LF]
20	22.534026	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=81 Ack=286 Win=30272 Len=0 TSval=5664250 TSecr=3181428
21	31.576788	192.168.0.1	172.16.0.1	SMTP	88	C: DATA fragment, 22 bytes
22	31.617655	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [ACK] Seq=286 Ack=103 Win=14480 Len=0 TSval=3183714 TSecr=5666511
23	33.292078	192.168.0.1	172.16.0.1	DNS	69	This is a text Mail.
24	33.293841	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [ACK] Seq=286 Ack=106 Win=14480 Len=0 TSval=3184133 TSecr=5666931
25	33.328538	172.16.0.1	192.168.0.1	SMTP	103	S: 250 2.0.0 Ok: queued as C1B0D181846
26	33.376772	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=106 Ack=323 Win=30272 Len=0 TSval=5666961 TSecr=3184141
27	35.686038	192.168.0.1	172.16.0.1	SMTP	72	C: quit
28	35.686430	172.16.0.1	192.168.0.1	SMTP	81	S: 221 2.0.0 Bye
29	35.686529	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [FIN, ACK] Seq=338 Ack=112 Win=14480 Len=0 TSval=3184711 TSecr=5667518
30	35.654096	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [ACK] Seq=112 Ack=338 Win=30272 Len=0 TSval=5667531 TSecr=3184711
31	35.696043	192.168.0.1	172.16.0.1	TCP	66	41401 → 25 [FIN, ACK] Seq=112 Ack=339 Win=30272 Len=0 TSval=5667531 TSecr=3184711
32	35.696195	172.16.0.1	192.168.0.1	TCP	66	25 → 41401 [ACK] Seq=339 Ack=113 Win=14480 Len=0 TSval=3184733 TSecr=5667531

그림 6.2.3 Mail Server 접속 시 교환되는 패킷

그림 6.3.1에서 SMTP 패킷을 보면 메일 클라이언트와 서버 간에 교환되는 메시지가 telnet으로 접속한 터미널에서의 내용과 같은 것을 볼 수 있다.

## 6.4 DNS 분석

7	20.234781	10.0.0.1	192.168.0.1	DNS	71	Standard query 0x853f A example.edu
8	20.250534	192.168.0.1	10.0.0.1	DNS	129	Standard query response 0x853f A example.edu A 172.16.0.2 NS example.edu AAAA ::1

그림 6.4.1 DNS Server 접속 시 교환되는 패킷

그림 6.4.1에서 보이는 바와 같이 DNS 프로토콜은 아주 간단하다. 질의와 응답으로 이루어진다. 다음 그림 6.4.2와 그림 6.4.3에서 조금 더 자세하게 살펴보자.

▶ Frame 4: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0 ▶ Ethernet II, Src: Vmware_c0:8c:d8 (00:0c:29:c0:8c:d8), Dst: cc:04:12:6c:00:01 (cc:04:12:6c:00:01) ▶ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.0.1 ▶ User Datagram Protocol, Src Port: 1025 (1025), Dst Port: 53 (53) ▶ Domain Name System (query) [Response In: 5] Transaction ID: 0xd134 ▶ Flags: 0x0100 Standard query Questions: 1 Answer RRs: 0 Authority RRs: 0 Additional RRs: 0 ▶ Queries ▶ example.edu: type A, class IN
--

그림 6.4.2 DNS query 패킷 정보

```

> Frame 5: 129 bytes on wire (1032 bits), 129 bytes captured (1032 bits) on interface 0
> Ethernet II, Src: cc:04:12:6c:00:01 (cc:04:12:6c:00:01), Dst: Vmware_c0:8c:d8 (00:0c:29:c0:8c:d8)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 10.0.0.1
> User Datagram Protocol, Src Port: 53 (53), Dst Port: 1025 (1025)
  Domain Name System (response)
    [Request In: 4]
    [Time: 0.020212000 seconds]
    Transaction ID: 0xd134
    > Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 1
    > Queries
    < Answers
      > example.edu: type A, class IN, addr 172.16.0.2
    > Authoritative nameservers
    > Additional records

```

그림 6.4.3 DNS response 패킷 정보

## 6.5 DHCP 분석

1 0.000000	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x976b48b7
2 0.032131	10.0.0.2	10.0.0.1	DHCP	342 DHCP Offer - Transaction ID 0x976b48b7
3 0.037237	0.0.0.0	255.255.255.255	DHCP	342 DHCP Request - Transaction ID 0x976b48b7
4 0.062533	10.0.0.2	10.0.0.1	DHCP	342 DHCP ACK - Transaction ID 0x976b48b7

그림 6.5.1 동적 IP를 할당받기 위해 교환되는 패킷

그림 6.5.1을 보면 IP주소가 없는 DHCP 클라이언트(0.0.0.0)가 브로드캐스트를 통하여 DHCP 서버를 'Discover'한다. 이를 감지한 DHCP 서버는 10.0.0.1이라는 IP주소를 'Offer' 하고 다시 클라이언트는 'Request'한 뒤 서버는 이에 'ACK' 응답을 보낸다.

아래 그림에서 하나하나 자세히 살펴보자.

## -Discover

Bootstrap Protocol (Discover)
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0x6fc3463d
Seconds elapsed: 0
▶ Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: Vmware_c0:8c:d8 (00:0c:29:c0:8c:d8)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
▶ Option: (53) DHCP Message Type (Discover)
▶ Option: (116) DHCP Auto-Configuration
▶ Option: (61) Client identifier
▶ Option: (50) Requested IP Address
▶ Option: (12) Host Name
▶ Option: (60) Vendor class identifier
▶ Option: (55) Parameter Request List
▶ Option: (43) Vendor-Specific Information
▶ Option: (255) End
Padding: 000000000000

그림 6.5.2 DHCP Discover 패킷 정보

## -Offer

```
└─ Bootstrap Protocol (Offer)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x6fc3463d
  Seconds elapsed: 0
  ▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 10.0.0.1
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: Vmware_c0:8c:d8 (00:0c:29:c0:8c:d8)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  ▶ Option: (53) DHCP Message Type (Offer)
  ▶ Option: (54) DHCP Server Identifier
  ▶ Option: (51) IP Address Lease Time
  ▶ Option: (1) Subnet Mask
  ▶ Option: (15) Domain Name
  ▶ Option: (3) Router
  ▶ Option: (6) Domain Name Server
  ▶ Option: (255) End
  Padding: 00000000000000000000
```

그림 6.5.3 DHCP Offer 패킷 정보

## -Request

- ▣ Bootstrap Protocol (Request)
  - Message type: Boot Request (1)
  - Hardware type: Ethernet (0x01)
  - Hardware address length: 6
  - Hops: 0
  - Transaction ID: 0x6fc3463d
  - Seconds elapsed: 0
  - ▷ Bootp flags: 0x0000 (Unicast)
  - Client IP address: 0.0.0.0
  - Your (client) IP address: 0.0.0.0
  - Next server IP address: 0.0.0.0
  - Relay agent IP address: 0.0.0.0
  - Client MAC address: Vmware\_c0:8c:d8 (00:0c:29:c0:8c:d8)
  - Client hardware address padding: 00000000000000000000
  - Server host name not given
  - Boot file name not given
  - Magic cookie: DHCP
  - ▷ Option: (53) DHCP Message Type (Request)
  - ▷ Option: (61) Client identifier
  - ▷ Option: (50) Requested IP Address
  - ▷ Option: (54) DHCP Server Identifier
  - ▷ Option: (12) Host Name
  - ▷ Option: (60) Vendor class identifier
  - ▷ Option: (55) Parameter Request List
  - ▷ Option: (43) Vendor-Specific Information
  - ▷ Option: (255) End
  - Padding: 0000

그림 6.5.4 DHCP Request 패킷 정보



## -ACK

```
└─ Bootstrap Protocol (ACK)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x6fc3463d
  Seconds elapsed: 0
  ▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 10.0.0.1
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: Vmware_c0:8c:d8 (00:0c:29:c0:8c:d8)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  ▶ Option: (53) DHCP Message Type (ACK)
  ▶ Option: (54) DHCP Server Identifier
  ▶ Option: (51) IP Address Lease Time
  ▶ Option: (1) Subnet Mask
  ▶ Option: (15) Domain Name
  ▶ Option: (3) Router
  ▶ Option: (6) Domain Name Server
  ▶ Option: (255) End
  Padding: 00000000000000000000
```

그림 6.5.5 DHCP ACK 패킷 정보

## 7장. 프로젝트 평가 및 소감

해당 프로젝트를 진행하면서 네트워크의 전반적인 이해부터 라우터 설정, 라우팅 알고리즘, 여러 가지 서버 구축 방법, 프로토콜의 조금 더 현실적인 이해 등 정말 다양한 것들을 배울 수 있었다.

하지만 배운 점이 많은 만큼 역시나 이번에도 아쉬운 점 또한 많다. 이번 장에서는 본 프로젝트를 진행하면서 스스로 아쉬웠던 점을 간단하게 기술하고, 제시된 평가표를 작성하여 첨부하도록 한다. (별첨 뒷부분)

### -DDNS 설정

DHCP를 사용하는 호스트의 도메인은 IP가 바뀌면 DNS서버의 내용도 갱신되어야 한다. 이를 자동으로 해주는 서비스가 DDNS인데, 관련 설정을 하다 설정들이 꼬여서 재설치를 해야 하는 바람에 계속 진행하지 못했다. 없어서는 안 되는 설정이기 때문에 더욱 아쉽다.

### -Mail 서버

Mail 서버는 사실 인터넷의 설정 방법을 그대로 따라했을 뿐 제대로 이해하지 못해서 아직까지 여운이 남는다. 호스트 간 메일을 주고받는 동작 예를 시도해보고 싶었지만, 설정 과정 중 인증 절차 등의 이해 부족으로 조금 더 학습할 시간이 필요할 것 같다.

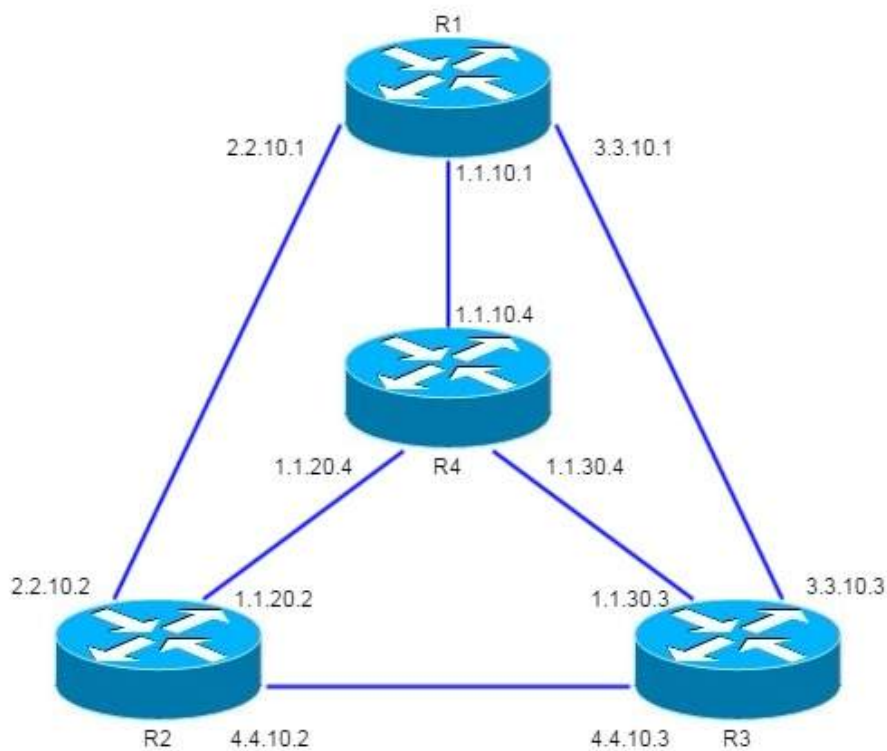
### -라우터 고장 Test

최초에 네트워크 구성을 단일 경로로 만들어서 고장 Test를 실험해 볼 수 없었던 점이 너무 아쉽다. 고장 Test는 보고서 작성을 마친 후 간단한 네트워크를 구축하여 뒷부분에 별첨하도록 할 예정이다.

## 별첨

### -라우터 고장 Test

라우터 고장 Test를 실험하기 위해 기존 네트워크의 구성을 변경하여 라우터 4개가 mesh 형태로 구성하였다. 각각의 인터페이스에 할당한 IP 주소는 임의적으로 설정하였다.



mesh 방식의 LAN 구성도

이렇게 구성하면 임의의 라우터에서 다른 링크가 다 고장 나더라도 링크가 하나만 살아있다면 다른 모든 라우터와 연결이 가능하다.

(다른 라우터에 연결된 링크는 고장 나지 않았다는 일반적인 상황에서 적용된다.)

고장 Test는 R1에서 인터페이스를 다운 시켰을 때 라우팅 테이블의 변화를 살펴보고, 각각의 상황에서 패킷이 다른 라우터로 전달되는 경로를 추적한다. 라우팅 프로토콜은 OSPF를 사용한다.



### (1) 연결에 이상이 없는 경우

```
1.0.0.0/24 is subnetted, 3 subnets
C    1.1.10.0 is directly connected, FastEthernet0/0
O    1.1.20.0 [110/2] via 2.2.10.2, 00:35:52, FastEthernet1/2
      [110/2] via 1.1.10.4, 00:35:52, FastEthernet0/0
O    1.1.30.0 [110/2] via 3.3.10.3, 00:35:52, FastEthernet1/3
      [110/2] via 1.1.10.4, 00:35:52, FastEthernet0/0
2.0.0.0/24 is subnetted, 1 subnets
C    2.2.10.0 is directly connected, FastEthernet1/2
C    3.0.0.0/24 is subnetted, 1 subnets
C    3.3.10.0 is directly connected, FastEthernet1/3
O    4.0.0.0/24 is subnetted, 1 subnets
      4.4.10.0 [110/2] via 3.3.10.3, 00:35:52, FastEthernet1/3
              [110/2] via 2.2.10.2, 00:35:52, FastEthernet1/2
```

연결에 이상이 없는 경우 OSPF 라우팅 테이블

각각의 네트워크에 접근할 수 있는 여러 경로가 라우팅 테이블에 기록되는 그림이다.

Neighbor	ID	Pri	State	Dead Time	Address	Interface
3.3.3.3		0	FULL/ -	00:01:48	3.3.10.3	FastEthernet1/3
4.4.4.4		0	FULL/ -	00:00:35	1.1.10.4	FastEthernet0/0
2.2.2.2		0	FULL/ -	00:00:37	2.2.10.2	FastEthernet1/2

연결에 이상이 없는 경우 OSPF neighbor

다른 세 라우터와의 상태가 FULL이므로 인접 OSPF 라우터와 통신이 잘 되고 있음을 확인할 수 있다.

```
R1#traceroute 1.1.10.4
Type escape sequence to abort.
Tracing the route to 1.1.10.4
 0 1.1.10.4 40 msec * 20 msec
```

R4로의 경로 추적

직접 연결되어 있기 때문에 패킷이 곧바로 전달되는 것을 볼 수 있다.

(2) 하나의 인터페이스가 다운되었을 경우 (R1과 R4를 잇는 인터페이스 다운)

```

1.0.0.0/24 is subnetted, 3 subnets
O   1.1.10.0 [110/3] via 3.3.10.3, 00:01:22, FastEthernet1/3
O   1.1.20.0 [110/3] via 2.2.10.2, 00:01:22, FastEthernet1/2
O   1.1.30.0 [110/2] via 3.3.10.3, 00:01:22, FastEthernet1/3
2.0.0.0/24 is subnetted, 1 subnets
C   2.2.10.0 is directly connected, FastEthernet1/2
3.0.0.0/24 is subnetted, 1 subnets
C   3.3.10.0 is directly connected, FastEthernet1/3
4.0.0.0/24 is subnetted, 1 subnets
O   4.4.10.0 [110/2] via 3.3.10.3, 00:01:22, FastEthernet1/3
O   4.4.10.0 [110/2] via 2.2.10.2, 00:01:22, FastEthernet1/2

```

하나의 인터페이스가 다운되었을 경우 OSPF 라우팅 테이블

위 그림에서 빨간 네모로 표시된 곳을 보면 C에서 O로 바뀐 것을 볼 수 있고, 해당 네트워크로 가는 2가지의 경로가 표시됨을 알 수 있다.

Neighbor ID	Pri	State	Dead Time	Address	Interface
3.3.3.3	0	FULL/ -	00:01:41	3.3.10.3	FastEthernet1/3
2.2.2.2	0	FULL/ -	00:00:30	2.2.10.2	FastEthernet1/2

하나의 인터페이스가 다운되었을 경우 OSPF neighbor

다른 세 라우터 중 R4가 없어진 것을 볼 수 있는데, 이는 인터페이스가 다운됨으로써 R4의 Hello 패킷을 받을 수 없기 때문이다.

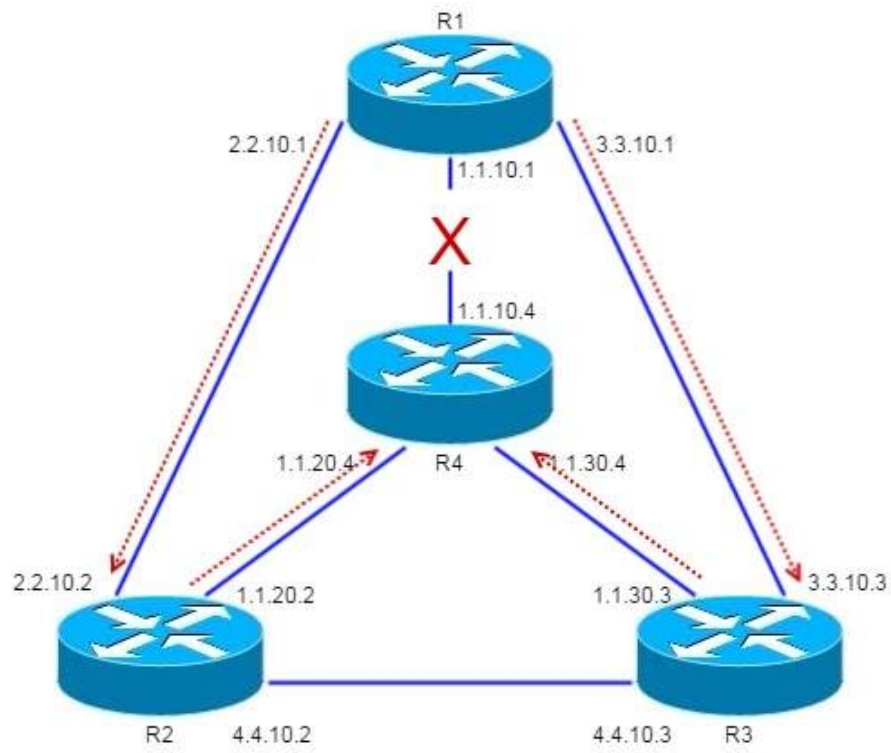
```

R1#traceroute 1.1.10.4
Type escape sequence to abort.
Tracing the route to 1.1.10.4
 0 3.3.10.3 32 msec
 1 2.2.10.2 16 msec
 2 3.3.10.3 12 msec
 3 1.1.20.4 32 msec
 4 1.1.30.4 24 msec *

```

R4로의 경로 추적

연결이 끊어진 R4로 경로를 추적해 본 결과를 나타내는 그림이다. 조금 더 보기 쉬운 그림으로 살펴보자.



R1에서 R4로의 경로 추적

### (3) 두 개의 인터페이스가 다운되었을 경우 (R1과 R4, R1과 R3를 잇는 인터페이스 다운)

```

1.0.0.0/24 is subnetted, 3 subnets
O   1.1.10.0 [110/3] via 2.2.10.2, 00:00:01, FastEthernet1/2
O   1.1.20.0 [110/2] via 2.2.10.2, 00:00:01, FastEthernet1/2
O   1.1.30.0 [110/3] via 2.2.10.2, 00:00:01, FastEthernet1/2
2.0.0.0/24 is subnetted, 1 subnets
C   2.2.10.0 is directly connected, FastEthernet1/2
3.0.0.0/24 is subnetted, 1 subnets
O   3.3.10.0 [110/3] via 2.2.10.2, 00:00:01, FastEthernet1/2
4.0.0.0/24 is subnetted, 1 subnets
O   4.4.10.0 [110/2] via 2.2.10.2, 00:00:01, FastEthernet1/2

```

두 개의 인터페이스가 다운되었을 경우 OSPF 라우팅 테이블

R1에서 다른 라우터로의 모든 경로가 R2를 거쳐서 나가는 것을 확인할 수 있다.

Neighbor ID	Pri	State	Dead Time	Address	Interface	
2.2.2.2	0	FULL/	-	00:00:31	2.2.10.2	FastEthernet1/2

두 개의 인터페이스가 다운되었을 경우 OSPF neighbor

인터페이스가 다운되지 않은 R2에게서만 Hello 패킷을 받을 수 있기 때문에 R2의 정보만 neighbor로 출력된다.

R1#traceroute 1.1.30.3

Type escape sequence to abort.  
Tracing the route to 1.1.30.3

```

1 2.2.10.2 60 msec 36 msec 8 msec
2 4.4.10.3 160 msec * 28 msec

```

R3로의 경로 추적

R1#traceroute 1.1.10.4

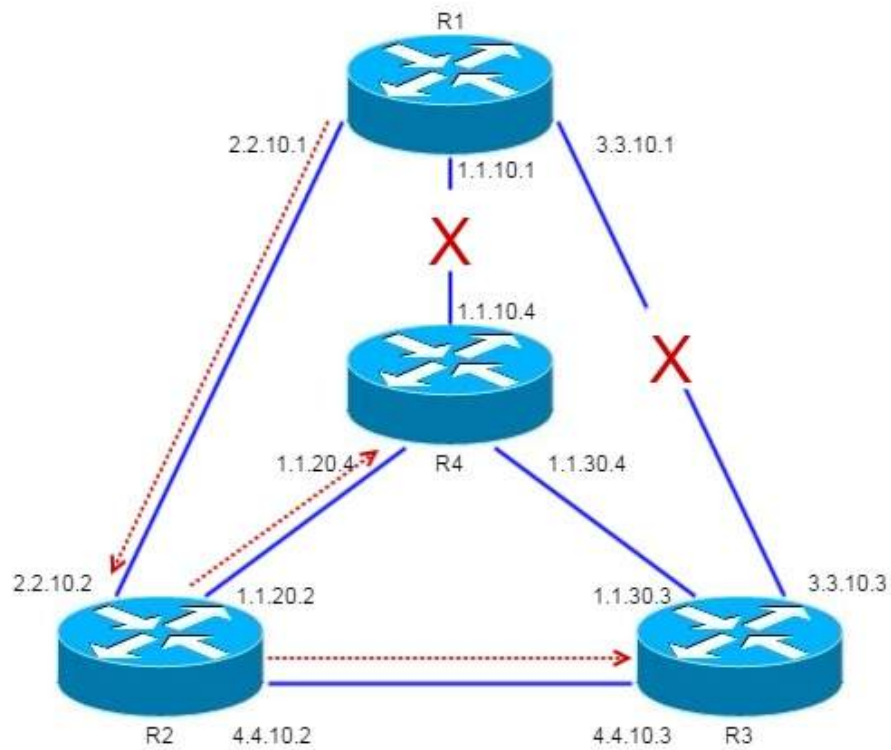
Type escape sequence to abort.  
Tracing the route to 1.1.10.4

```

1 2.2.10.2 44 msec 16 msec 8 msec
2 1.1.20.4 36 msec * 24 msec

```

R4로의 경로 추적



R1에서 R4, R1에서 R3로의 경로 추적

라우팅 프로토콜의 종류 중 하나인 OSPF를 통해 라우팅 테이블이 자동으로 갱신되며, 패킷이 전송되는 경로가 동적으로 변한다는 것을 확인할 수 있었다.

컴퓨터 네트워크 term project 평가표

항목	세부내용	평가(1-5점)	비고
1. 개발환경	Dynamips	5	구축하는 데 있어서 아주 편리한 기능 및 쉬운 수단 제공
	Dynagen	5	
	VMware	5	
	SecureCRT	5	
2. 네트워크 구성	4개의 Router	5	VLAN Test O
	Static Routing	5	
	RIP	5	
	OSPF	5	
	BGP	5	
	VLAN 및 TRUNK 설정	5	
	고장 test	5	
3. 서버 동작	DNS Server	4	DDNS 설정 X
	DHCP Server	5	
	Web Server	5	
	Mail Server	4	완벽한 구축 X
	FTP Server	5	
	가위바위보 Server	5	
4. 프로토콜 이해	Wireshark test	5	
5. 종합평가	보고서 작성	4	그림이 많아 깔끔하지 못함