5. 전송계층, UDP프로토콜과 공격

41526: 네트워크보안

강문수
(mskang@chosun.ac.kr)

컴퓨터네트워크 연구실
(Computer Networks Lab.)

5장 목차

5.1 개요

- Transport layer protocol
- Port number
- UDP protocol
- Attacks using UDP

전송계층 프로토콜

Properties	TCP	UDP
Connections	√	
Packet boundary		√
Reliability	√	
Ordering	√	
Speed		Faster
Broadcast		√

포트 번호: 왜 필요한가?

- Analogy

	Mailing Address
IP address	Apartment building's street address
Port number	Apartment number

- IP Address: address of machines
- Port number: address of applications (within a machine

포트번호(Port Number)

- Well-known ports: **0 – 1023**
  - ftp (20, 21), ssh (22), telnet (23), smtp (25), DNS (53), http (80), https (443)
  - Super-user privilege needed, why?
- Less well-known ports: **1024 – 49151**
  - OpenVPN (1194), Microsoft SQL server (1433), Docker (2375-2377)
- Private ports: **49152 – 65535**
  - Source port number

5.2 UDP 헤더와 프로토콜

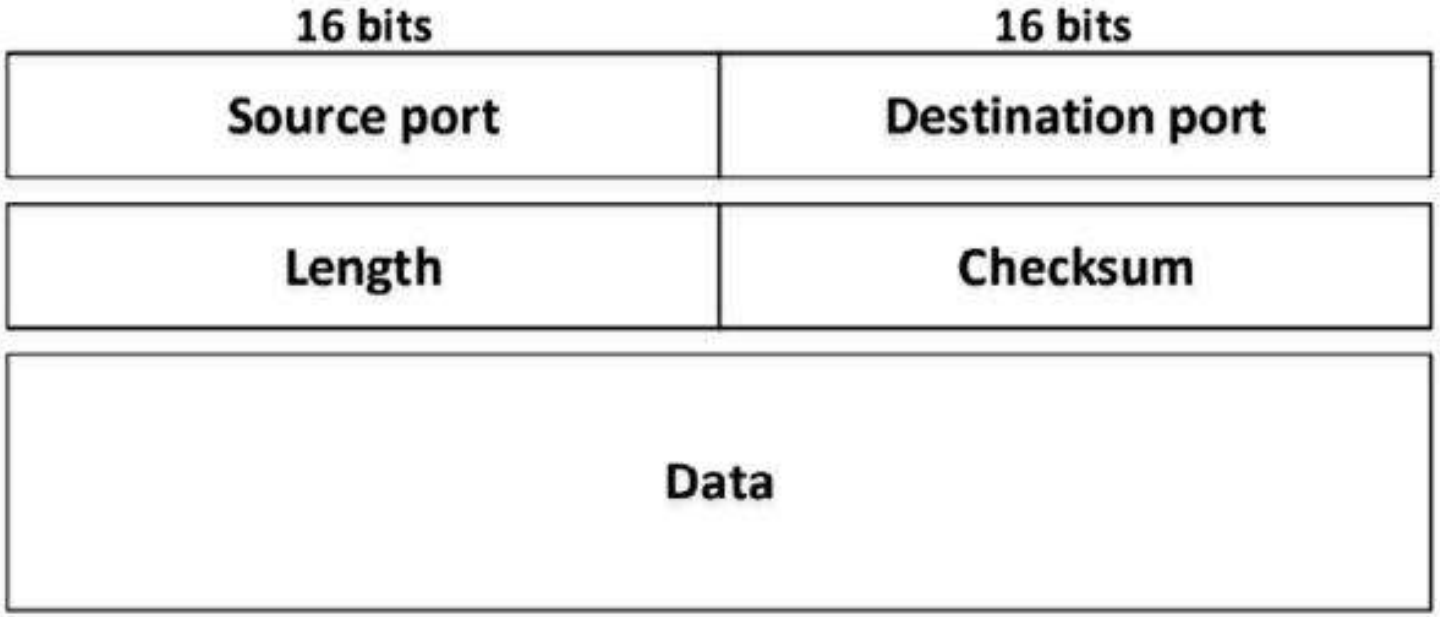


그림 5.1: UDP 헤더 형식

5.2.2 UDP 클라이언트 포로그램

```
#!/usr/bin/python3

import socket

IP   = "10.0.2.7"
PORT = 9090
data = b'Hello, World!'

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(data, (IP, PORT))
```

Source Port Number

- Application does not specify one
  - ➲ OS will assign a random source IP
  - ➲ Common for most client programs
- Application specifies one
  - ➲ not common for client
  - ➲ needed for server

```
udp = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
udp.bind(("0.0.0.0", 9999))
```

5.2.3 UDP Server Program

```
#!/usr/bin/python3

import socket

IP   = "0.0.0.0"
PORT = 9090

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP, PORT))

while True:
  data, (ip, port) = sock.recvfrom(1024)
  print("Sender: {} and Port: {}".format(ip, port))
  print("Received message: {}".format(data))
```

UDP Applications

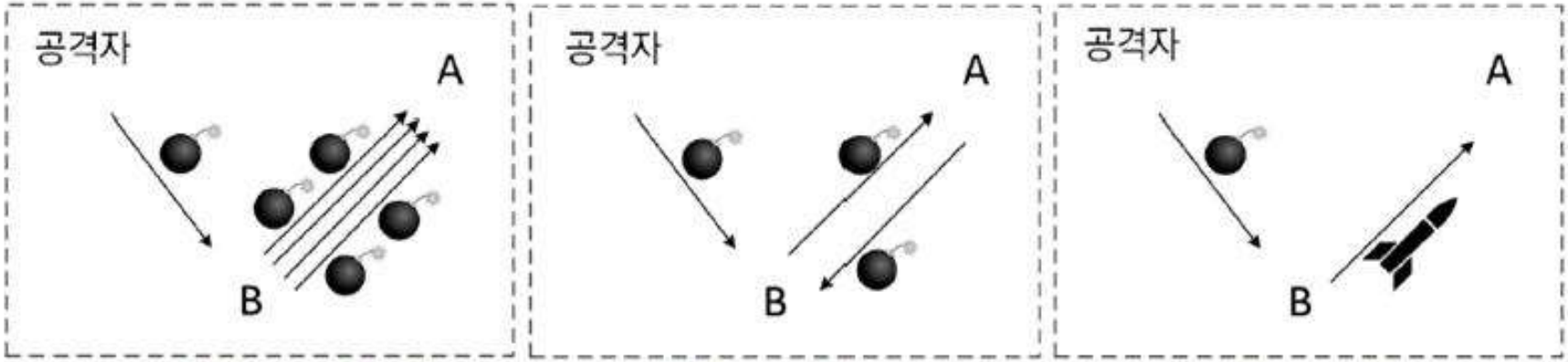
- DNS Protocol
  - Port number: 53
- Video/Audio Streaming, Skype, Zoom
  - Netflix and YouTube use TCP (no need for real time)
- Real-Time Applications

Question

UDP는 순서를 관리하지 않으며 패킷 손실을 처리하지 않는다. 응용 프로그램이 패킷 손실과 순서에 관심이 있다면 여전히 UDP를 사용할 수 있는가?

5.3 UDP를 이용한 공격

- Mostly used for Denial-Of-Service(DOS) Attacks
- Strategies: magnify attacking power



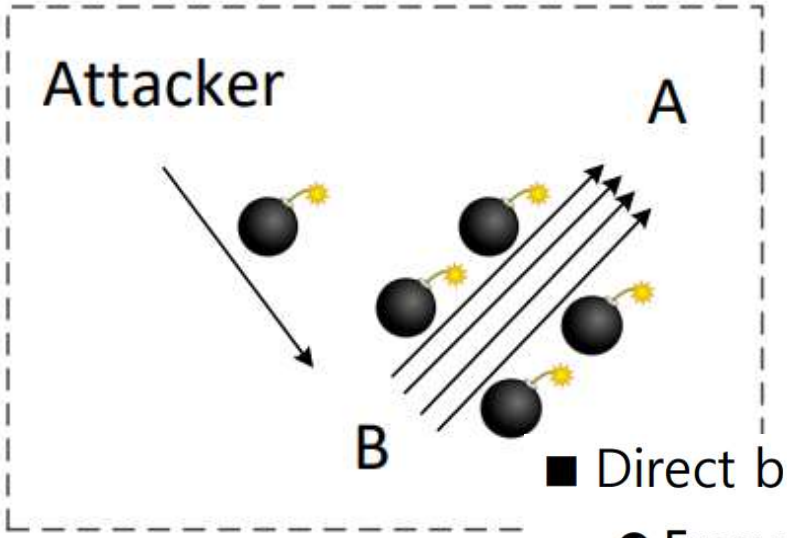
(a) 수류탄 하나를 여러 수류탄으로 전환

(b) 재생 가능한 수류탄 만들기

(c) 수류탄을 미사일로 전환

그림 5.2: 공격자가 자신의 힘을 확대하는 방법을 보여주는 비유

5.3.1 Fraggle 공격: 하나의 수류탄을 여러 수류탄으로 만들기



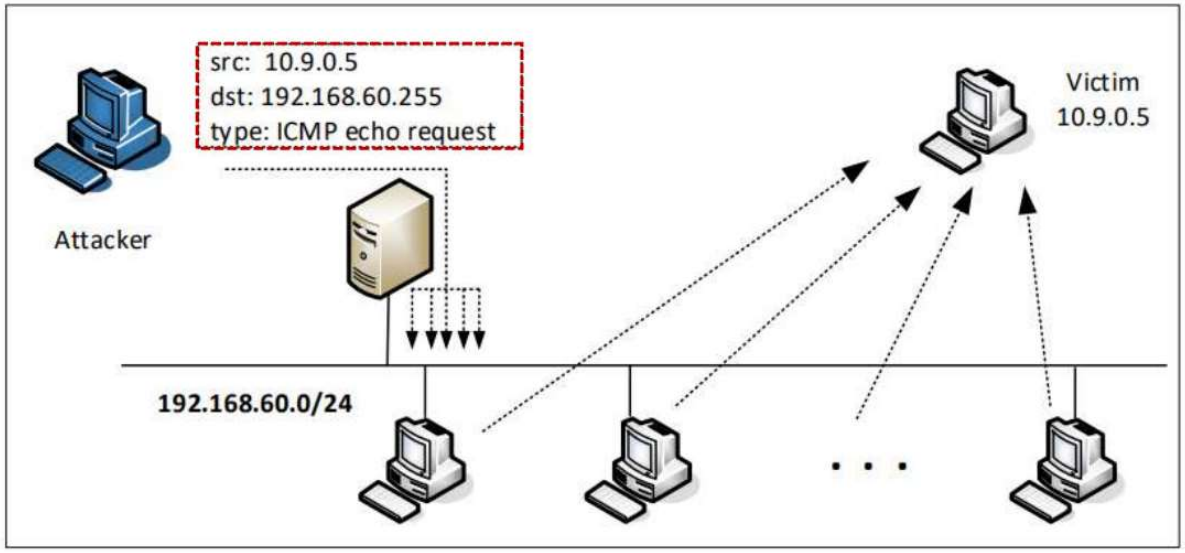
■ Example: Smurf Attack (ICMP),

■ Fraggle Attack (UDP)

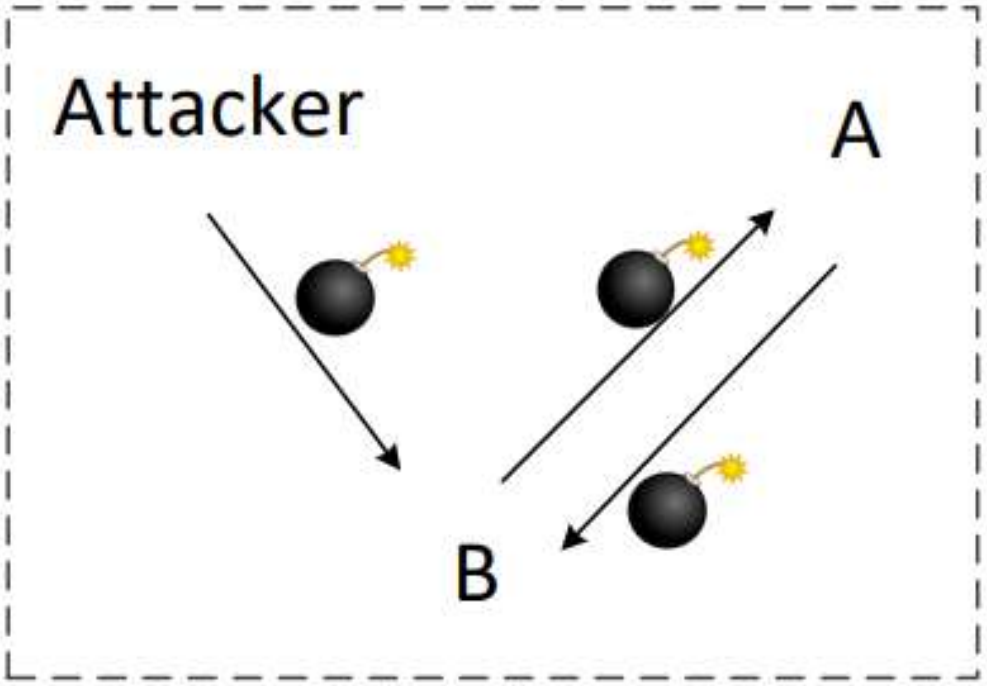
➲ UDP 에코 서비스(포트 7)

■ Direct broadcast address 이용

➲ Example: 192.168.60.255 for network 192.168.60.0/24



5.3.2 UDP Ping Pong: 재생 가능한 수류탄 만들기



■ Example: UDP Ping Pong Attack

UDP Ping Pong 공격: 취약한 서버

```
#!/usr/bin/python3

import socket

IP   = "0.0.0.0"
PORT = 9090

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP, PORT))

while True:
  data, (ip, port) = sock.recvfrom(1024)
  print("Sender: {} and Port: {}".format(ip, port))
  print("Received message: {}".format(data))

  # Send back a "thank you" note
  sock.sendto(b'Thank you!', (ip, port))
```

UDP Ping Pong 공격: 공격

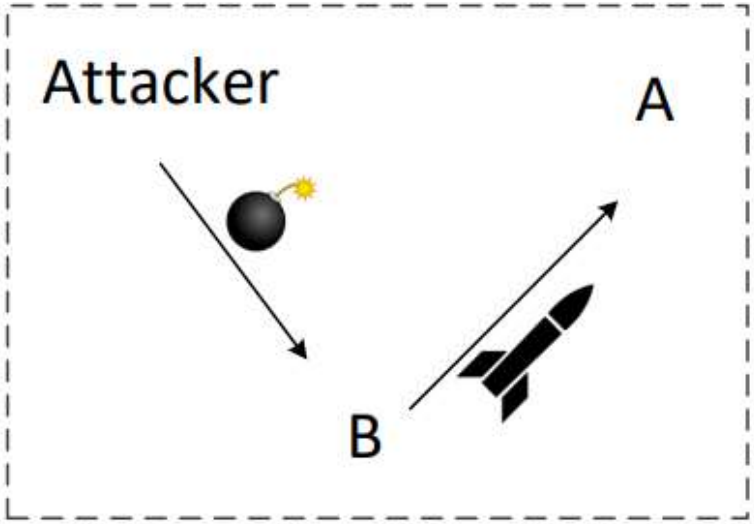
```
from scapy.all import *

ip   = IP(src="10.9.0.5", dst="10.9.0.6")
udp  = UDP(sport=9090, dport=9090)
data = "Let the Ping Pong game start!\n"
pkt  = ip/udp/data
send(pkt, verbose=0)
```

공격 결과 (타임스탬프를 보고 탁구공이 얼마나 빠른지 확인하시오.):

```
02:44:58.837942 IP 10.9.0.5.9090 > 10.9.0.6.9090: UDP, ...
02:44:58.837994 IP 10.9.0.6.9090 > 10.9.0.5.9090: UDP, ...
02:44:58.838218 IP 10.9.0.5.9090 > 10.9.0.6.9090: UDP, ...
02:44:58.838298 IP 10.9.0.6.9090 > 10.9.0.5.9090: UDP, ...
02:44:58.840450 IP 10.9.0.5.9090 > 10.9.0.6.9090: UDP, ...
02:44:58.840560 IP 10.9.0.6.9090 > 10.9.0.5.9090: UDP, ...
```

5.3.3 UDP 증폭 공격: 수류탄을 미사일로 전환



■ UDP 증폭(Amplification) 공격

- ➲ 공격자가 A인 것처럼 하여, 특정 크기의 UDP패킷을 B에게 전송
- ➲ B는 자신의 수신한 UDP패킷보다 훨씬 더 큰 UDP패킷을 생성하여 A에게 전송 → UDP 증폭
- ➲ 많은 종류의 UDP서비스들이 확대효과를 가지고 있음.

대역폭 증폭 인수

Protocol	Bandwidth Amplification Factor	Vulnerable Command
DNS	28 to 54	see: TA13-088A [4]
NTP	556.9	see: TA14-013A [5]
SNMPv2	6.3	GetBulk request
NetBIOS	3.8	Name resolution
SSDP	30.8	SEARCH request
CharGEN	358.8	Character generation request
QOTD	140.3	Quote request
BitTorrent	3.8	File search
Kad	16.3	Peer list exchange
Quake Network Protocol	63.9	Server info exchange
Steam Protocol	5.5	Server info exchange
Multicast DNS (mDNS)	2 to 10	Unicast query
RIPv1	131.24	Malformed request
Portmap (RPCbind)	7 to 28	Malformed request
LDAP	46 to 55	Malformed request [6]

Source: Christian Rossow

Q & A