2021 Computer Network Lab

acpang@csie.ntu.edu.tw
http://www.csie.ntu.edu.tw/~acpang
Prof. Ai-Chun Pang
Graduate Institute of Networking & Multimedia,
Dept. of Computer Science and Engineering
National Taiwan University

LAB1
Trace route concept and implementation

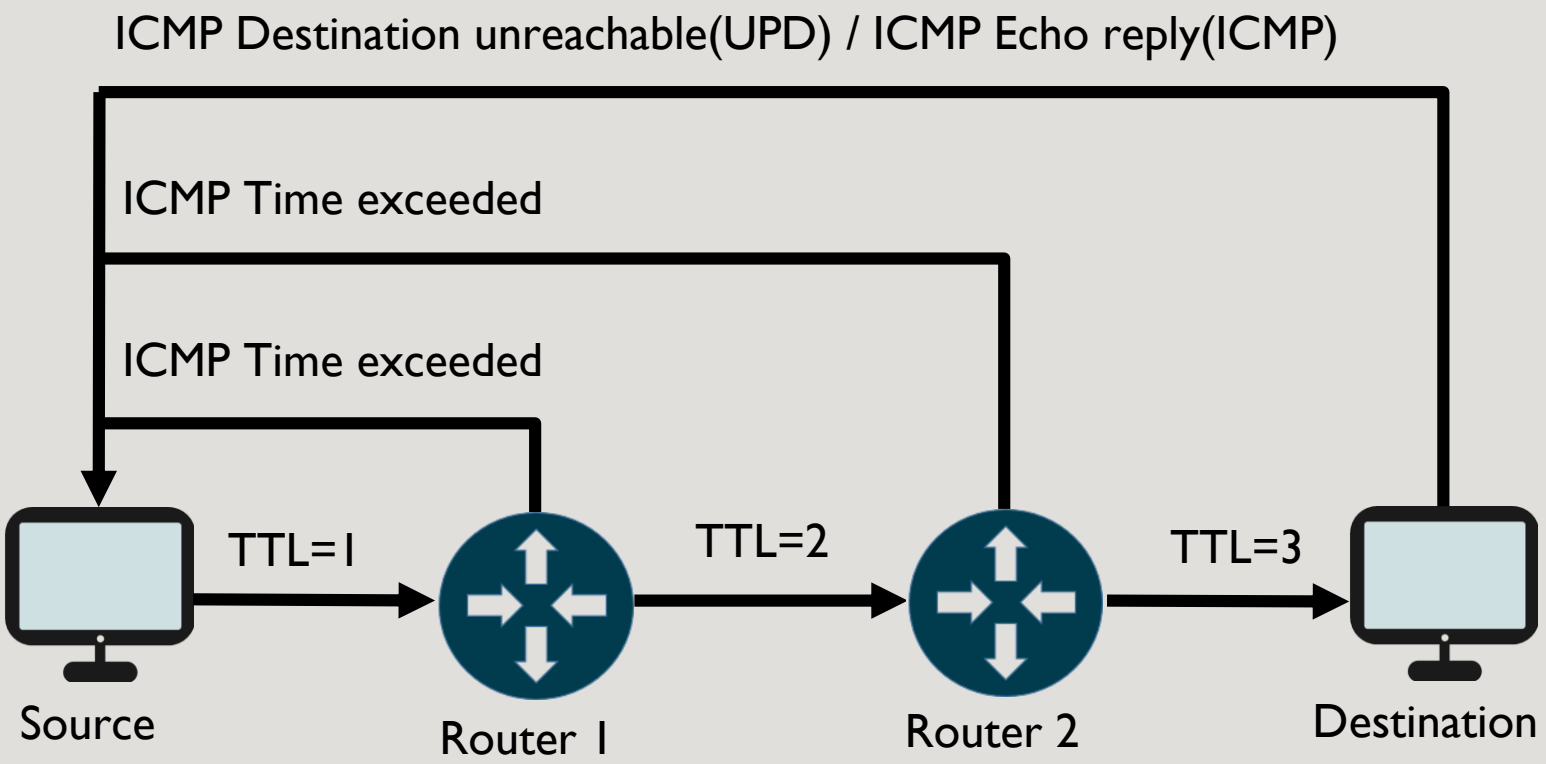
Outline

- Introduction
- Environment
- Procedure
- Grading Policy
- Regulation
- Deadline

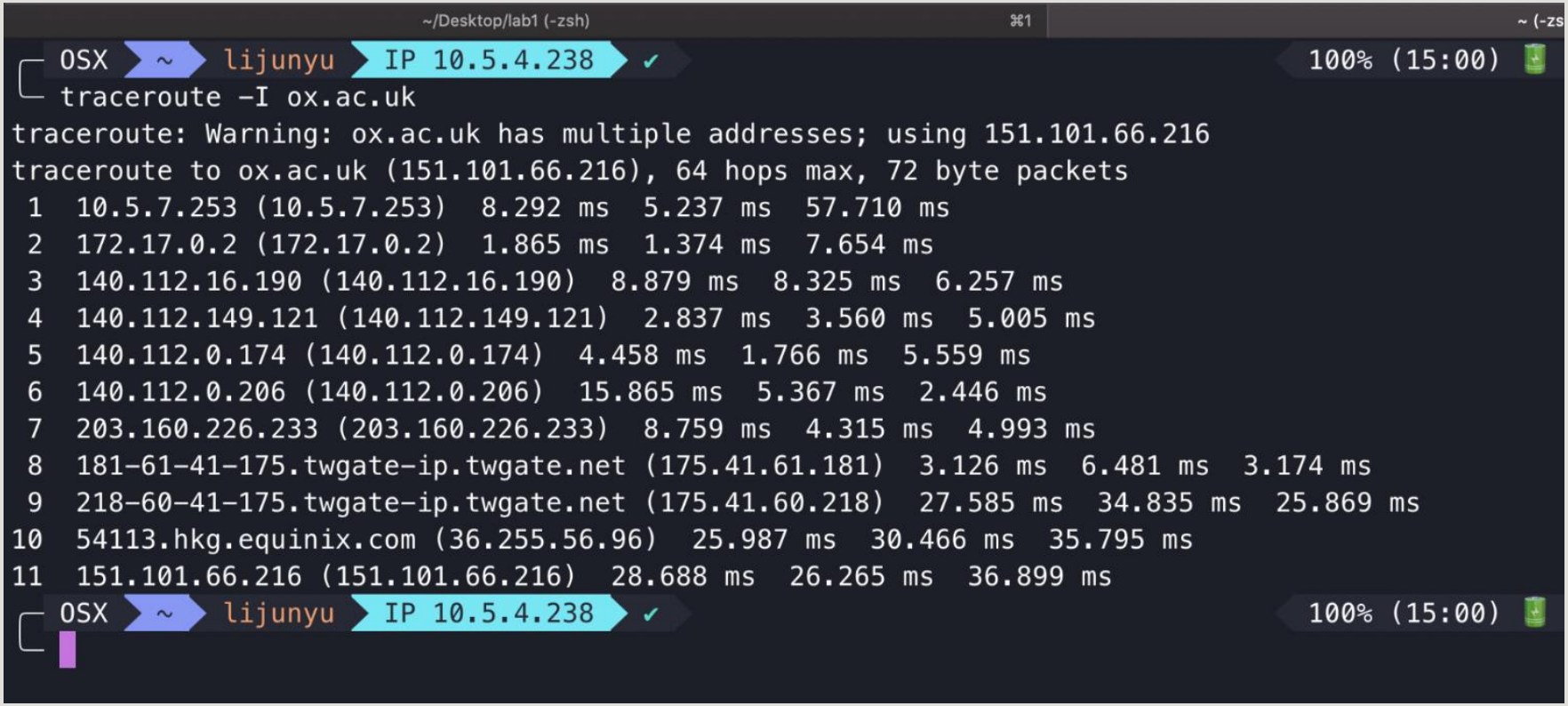
Introduction

- By setting the TTL number in the IP header, we can "trace" the routes from source to destination with RTT.
- Try to send and receive ICMP packets.
- Besides, you should analyze why traceroute cannot show the path of the full route and explain how to detect and defend traceroute.

Description



Description

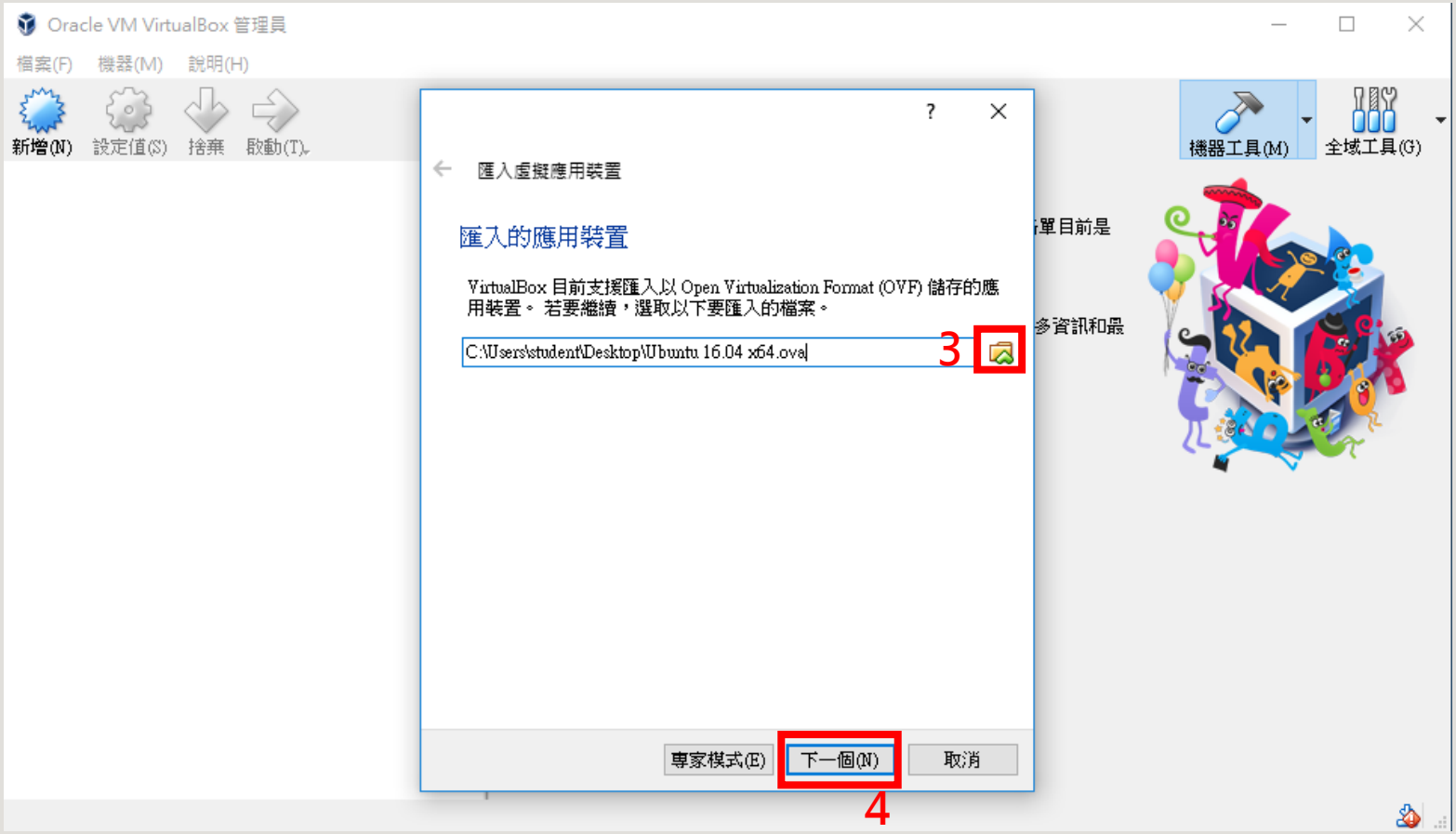


Environment

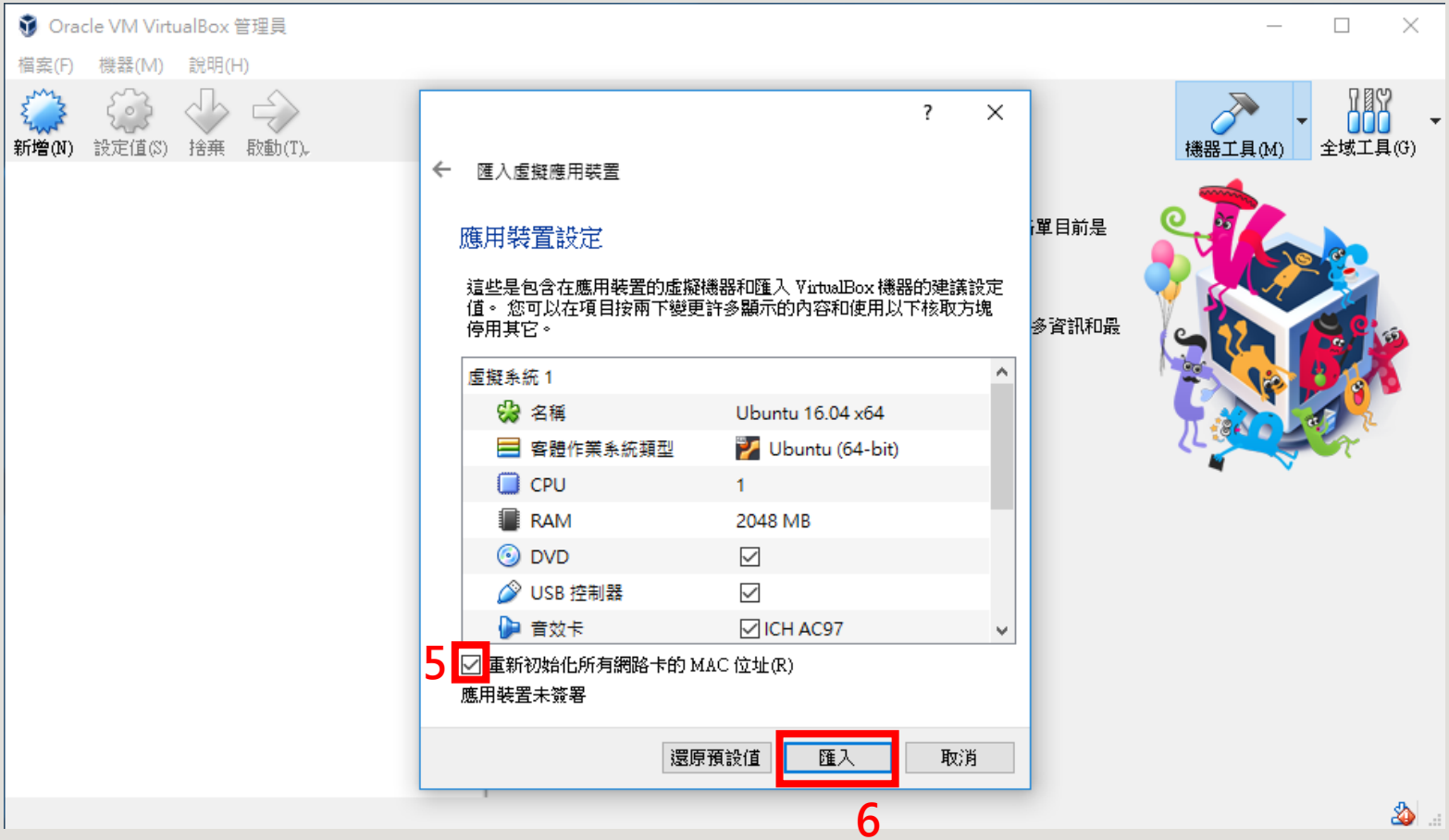
- Language: C / C++
- OS: ubuntu(recommended)
- Virtual box ova file, password: cnlab2021
  - Dropbox: https://reurl.cc/Kx011q
  - Google drive: https://reurl.cc/Q7XqA9
  - USB
- You can start with our sample code in NTU COOL.

Environment

Environment

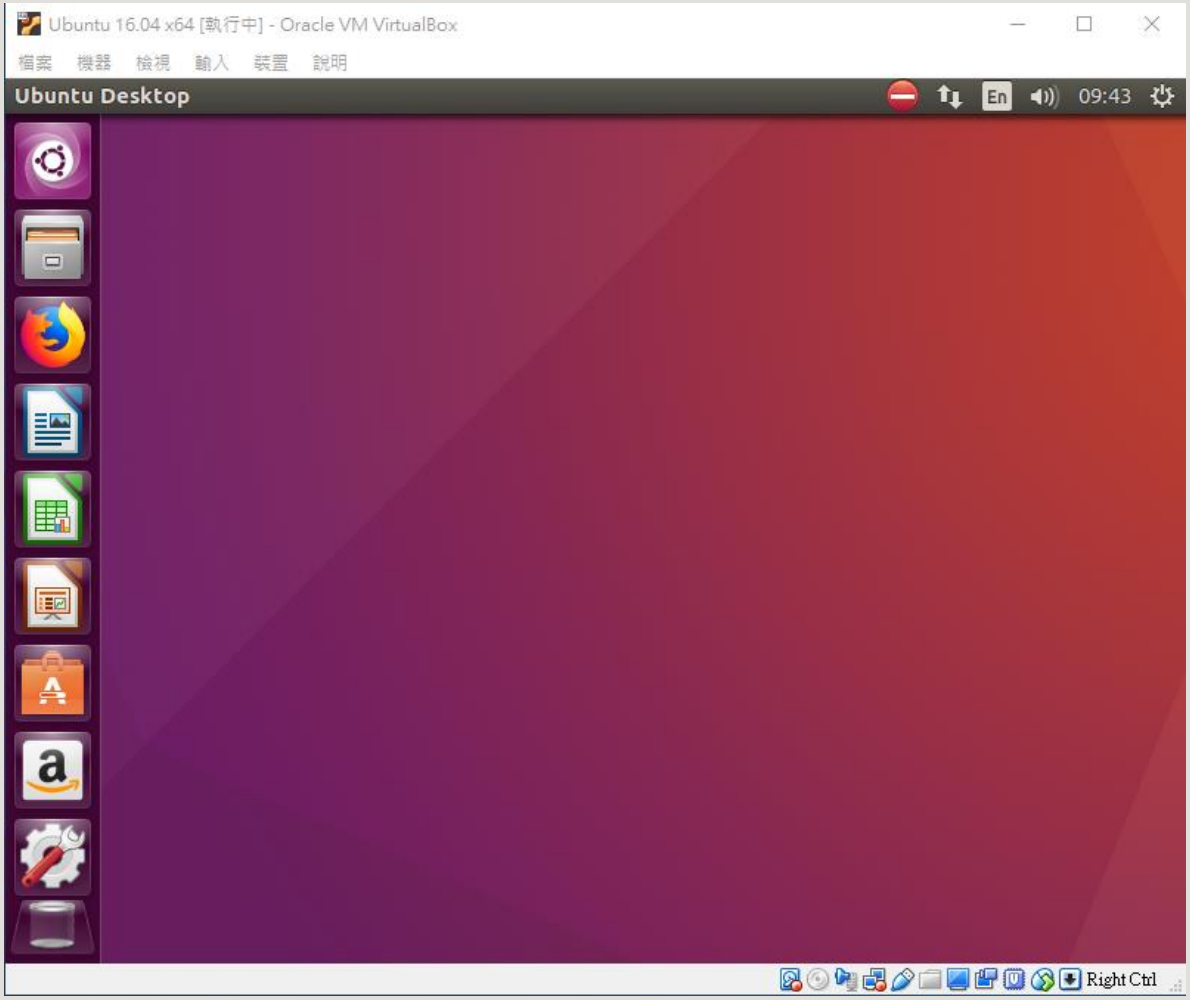


Environment



Environment

Environment



Step

- Open a socket
- Create an ICMP packet
- Send and receive
- Extract the packet
- Print the result

Step

- Open a socket
- Create an ICMP packet
- Send and receive
- Extract the packet
- Print the result

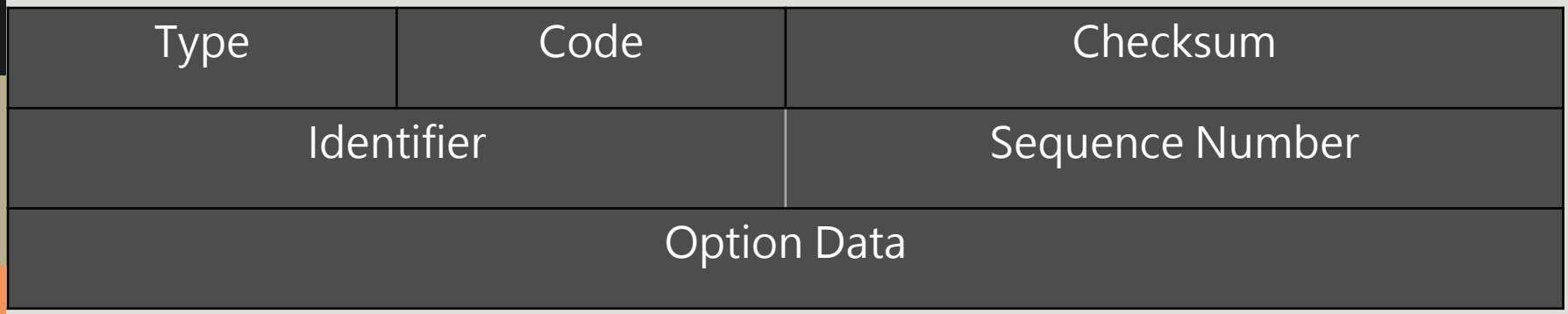
Step - Open a socket

- int socket(int domain, int type, int protocol)
  - domain：IPv4 (Inter Protocol Version 4)
  - type：The communication semantics, we used raw type here
  - protocol：ICMP
- int setsockopt(int socket, int level, int option_name, const void *option_value, socklen_t option_len);
  - Setting Time To Live value

Step

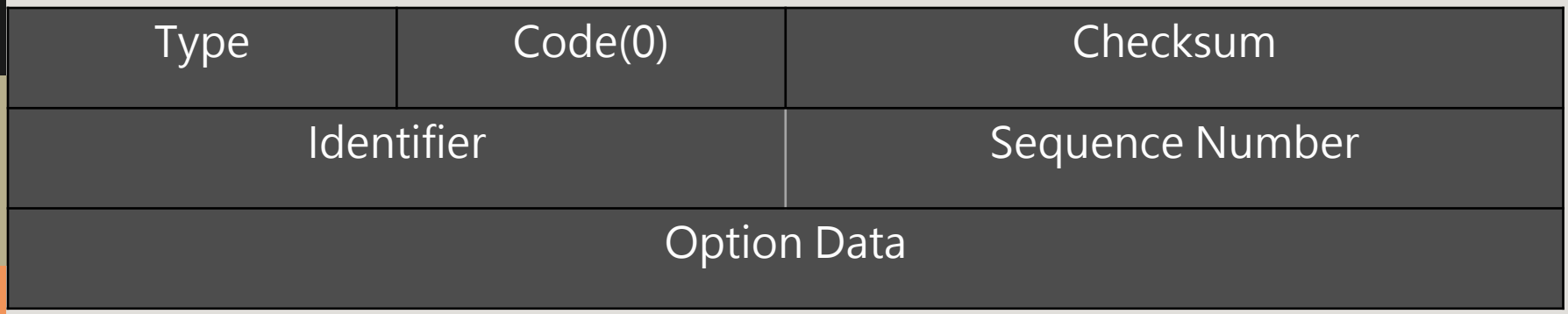
- Open a socket
- Create an ICMP packet
- Send and receive
- Extract the packet
- Print the result

Step - Create an ICMP packet



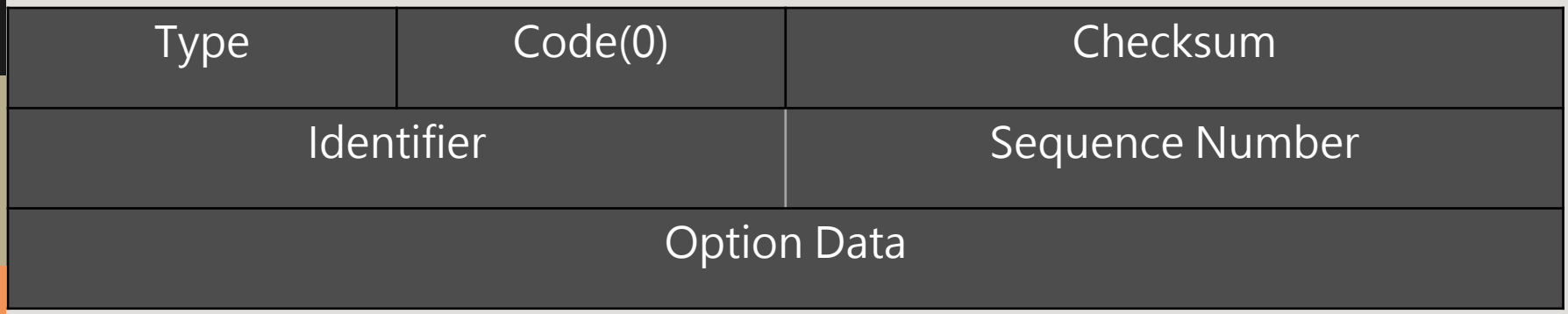
- Type :
  - 0 : Echo reply
  - 8 : Echo request
- Code :
  - ICMP subtype (e.g. Error code)
  - Filled with 0 in this case

Step - Create an ICMP packet



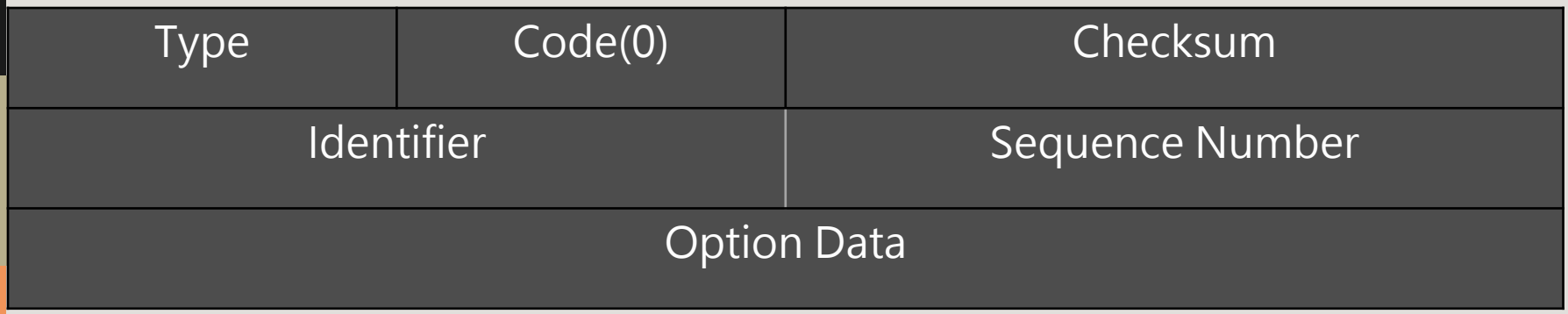
- Identifier：2 Bytes
- Sequence Number：2 Bytes
- Identifier and Sequence Number can be used by the client to match the timestamp reply with the timestamp request

Step - Create an ICMP packet



- Optional Data：can be used to identify whether Echo-Reply and Echo-Request is a couple
- In this experience, the Optional Data field can be empty.

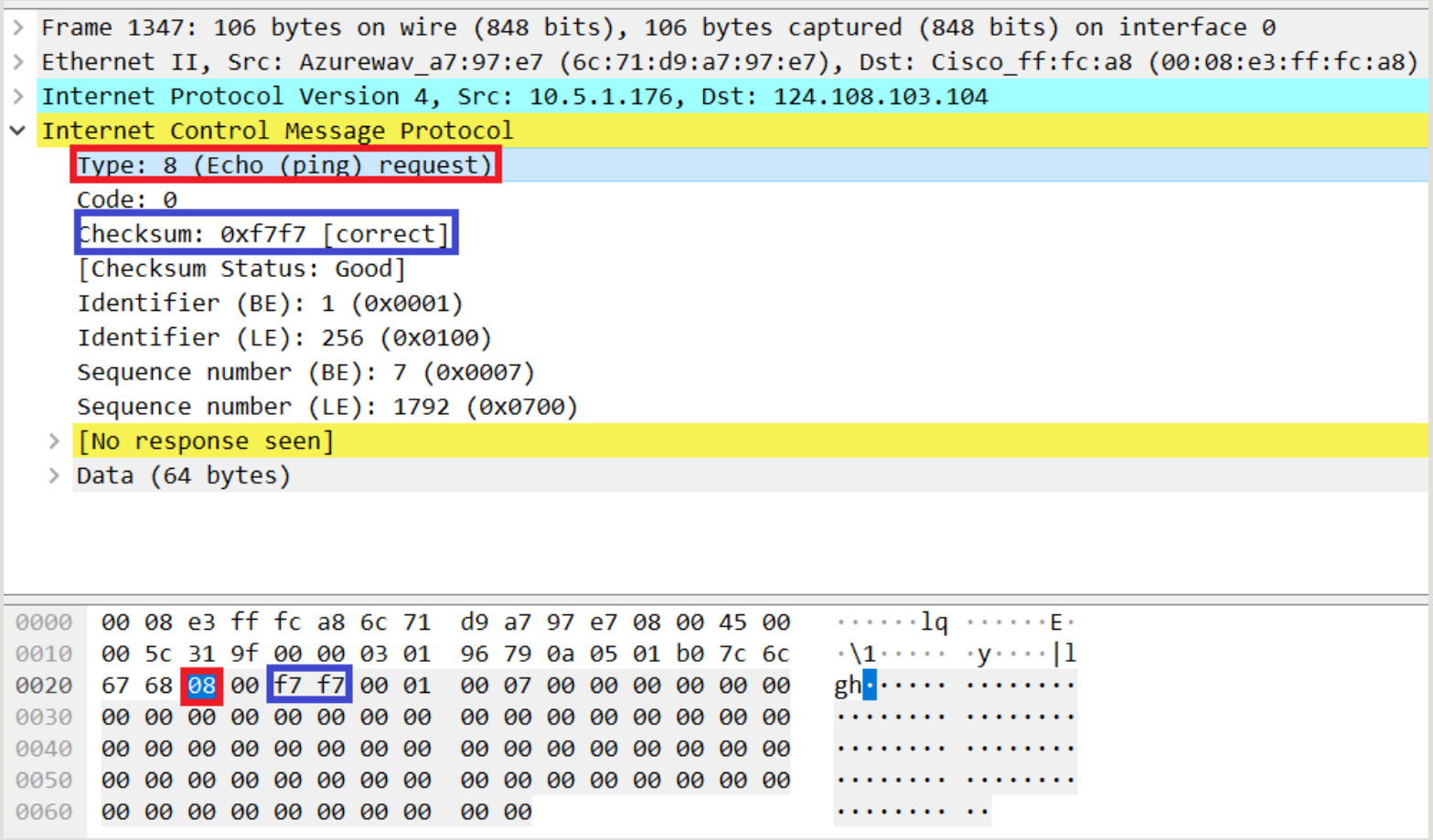
Step - Create an ICMP packet



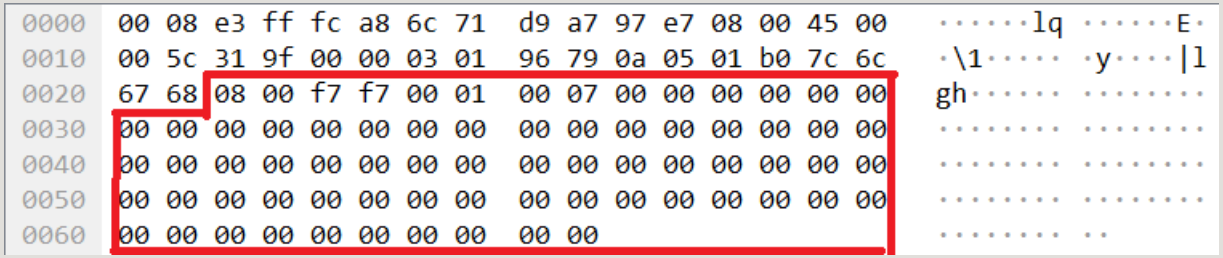
- Type： unsigned 8 bits
- Code ： unsigned 8 bits
- Checksum ：unsigned 16 bits
- Identifier：unsigned 16 bits
- Sequence Number：signed 16 bits

Step - Create an ICMP packet

- Wireshark



Step - Checksum



- Sum up each half-word (excluding checksum)
  - 0800 + 0001 + 0007 + 0000 + … + 0000 = 0000 0808
- Sum of the higher bit and lower bit
  - 0000 0808 => 0000 + 0808 = 0808
- Ones complement of the sum (1 to 0，0 to 1)
  - 0808 => F7F7

Step

- Open a socket
- Create an ICMP packet
- Send and receive
- Extract the packet
- Print the result

Step - Send and receive

- ssize_t sendto(int sockfd, const void *buf, size_t len, int flags,const struct sockaddr *dest_addr, socklen_t addrlen);
- ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen);
- Calculate the response time
  - int gettimeofday(struct timeval *tv, struct timezone *tz);
  - Timeout handling

Step

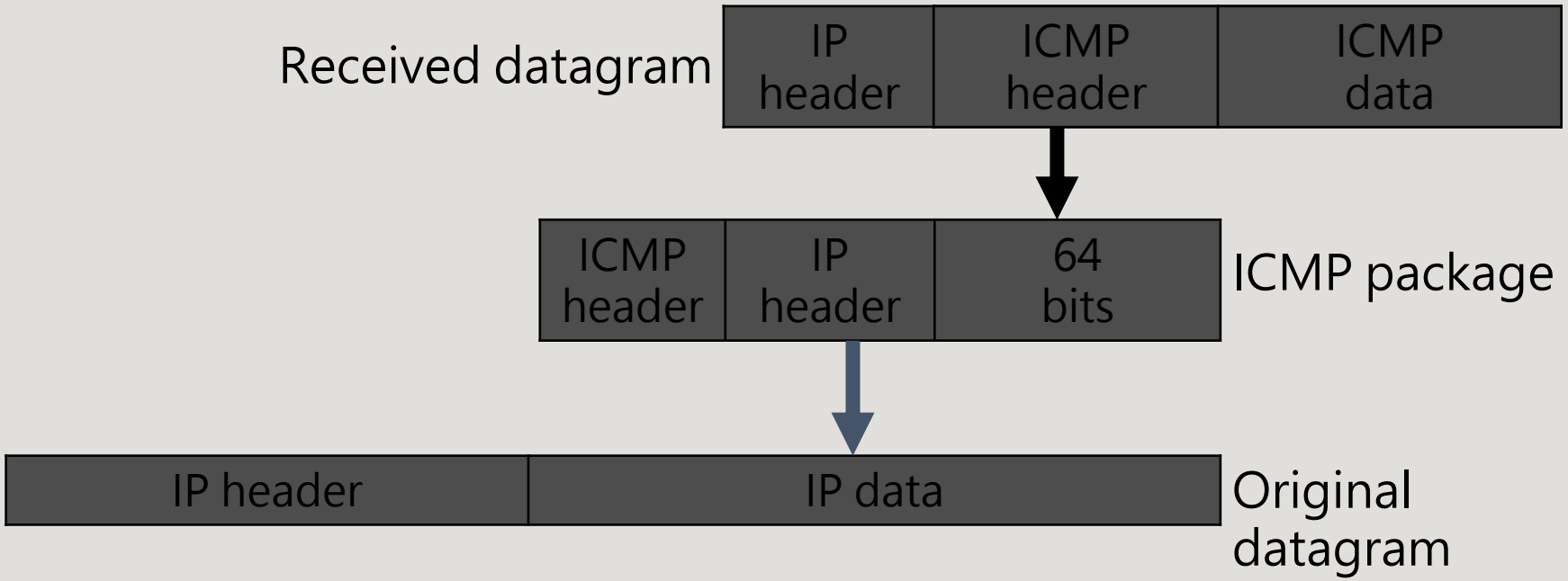
- Open a socket
- Create an ICMP packet
- Send and receive
- Extract the packet
- Print the result

Step - Extract the packet

Type	Code	Checksum
Undefined or other (32 bits)		
Data Section (IP header + 64 bits of original data)		

- Check ICMP type is either time exceeded or ICMP_ECHOREPLY
- Extract the original ICMP packet in the received IP packet
- Check identifier and sequence number in the original

Step - Extract the packet



Step

- Open a socket
- Create an ICMP packet
- Send and receive
- Extract the packet
- Print the result

Step - Print the result

- Linux traceroute format
  - <ttl, IP, RTT, RTT, RTT>
  - Timeout
  - No response / not reachable

Grading Policy

- Experiment：(50%)
  - Localhost (10%)
  - Nearest router (10%)
  - No response (handling time out) (10%)
  - Handle DNS lookup (10%)
  - Using TCP and UDP to implement traceroute (by options) (10%)

Grading Policy

- Report：(50%)
  - Environment (5%)
  - How to detect and defend traceroute (5%)
  - Why traceroute cannot show the full route (10%)
  - Why the result may not always be the same (10%)
  - Compare the results between local and foreign, and explain what causes the difference. (10%)
  - Explain the difference by using TCP, UDP, and ICMP. (10%)

Demo

- Traceroute localhost(127.0.0.1)
- Traceroute wireless AP
- Traceroute by hostname
- Traceroute by TCP or UDP

Regulation

- No cheating.
- You can only include libraries that appear in our sample code.
- List reference in your report.

Deadline

- 3 / 18 Demo
- 3 / 25 23:59:00
  - Submit report and source code to NTU cool
  - Report has to be in pdf format