

PROGRAMMING ASSIGNMENT III

Te-Chuan Chiu

PING FUNCTION

PING OVERVIEW

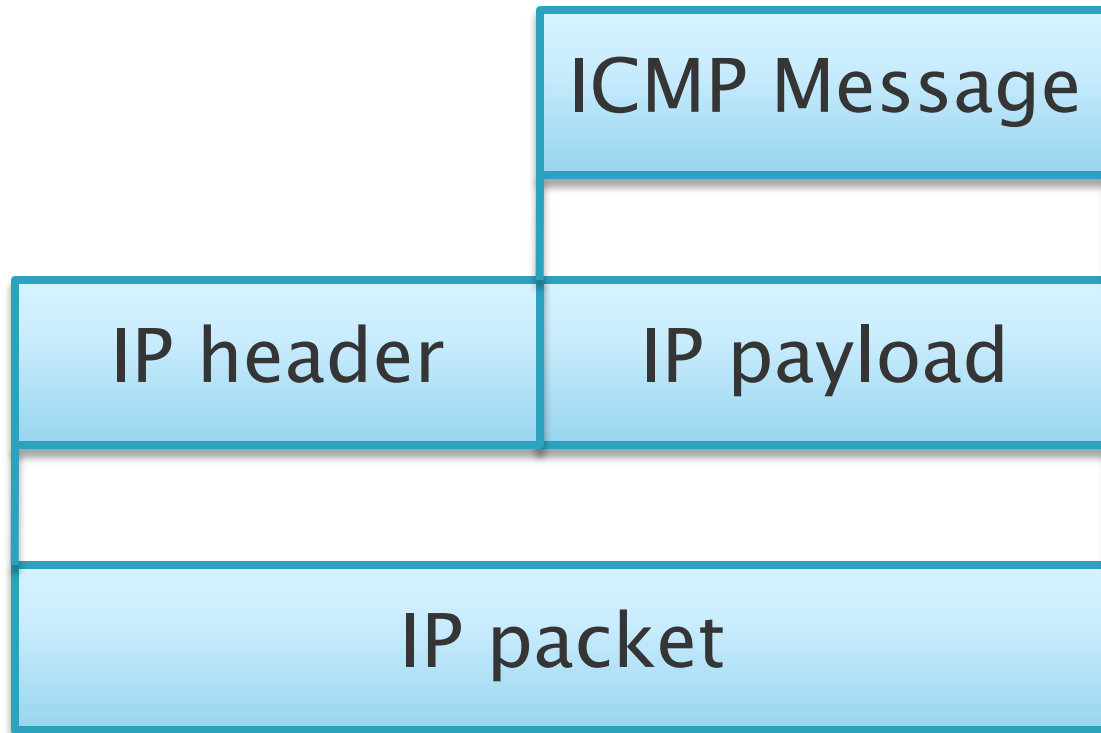
- ▶ Defined by RFC 792
- ▶ Whether the destination exists or not
- ▶ The information is retrieved from **ICMP protocol**



INTRODUCTION TO ICMP

- ▶ Internet control message protocol
- ▶ Hosts and routers exchange information with each other
- ▶ Most use is for **error reporting**
 - Get an error message such as “Destination network unreachable” when running on a telnet session
- ▶ **Network layer** – included in IP payload

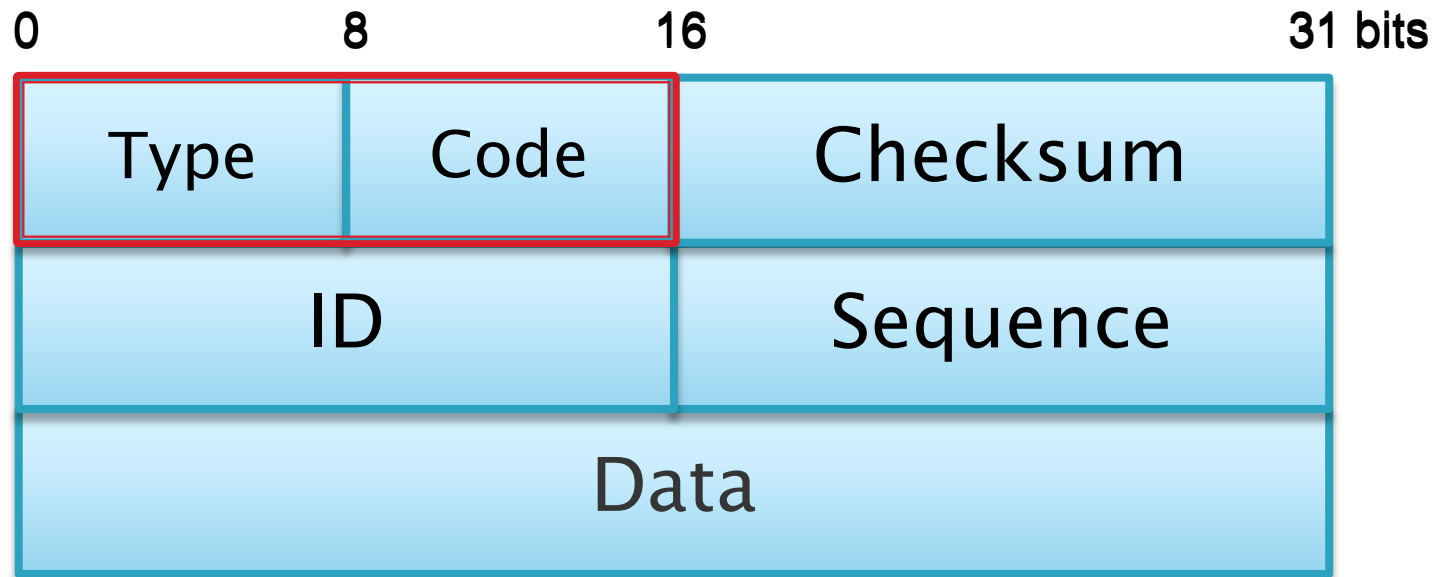
IP v.s. ICMP



IP PACKET

	0-3	4-7	8-15	16-18	19-31 bits
IP HEADER	Version	Header Length	Differentiated Services	Total Length	
	Identification			Flags	Fragment Offset
	Time to Live		Protocol	Header Checksum	
	Source Address				
	Destination Address				
IP PAYLOAD	(Up to 1024 bytes)				

ICMP MESSAGE



- ▶ ID – This field contains an ID value Ex. PID
- ▶ Sequence – Sequence number is set 0
Sequence number plus one for next packet

ICMP TYPE & CODE

Type	Code	Description
0	0	Echo reply (ping)
3	1	Destination network unreachable
3	2	Destination host unreachable
8	0	Echo request (ping)
11	0	TTL expired

CHECKSUM(1 / 2)

- ▶ Defined by RFC 1071
 - Divide ICMP message into 16-bit integer fragments and sum up
 - If the summation may produce carry, add the carry bit to the summation result
 - Do 1's complement to obtain the checksum

CHECKSUM(2 / 2)

0 8 16 31 bits

Type(8)	Code(0)	Checksum(0)
ID(65520)		Sequence(0)
TEST		

8 & 0	00001000	00000000
0	00000000	00000000
65520	11111111	11110000
0	00000000	00000000
T & E	01010100	01000101
S & T	01010011	01010100

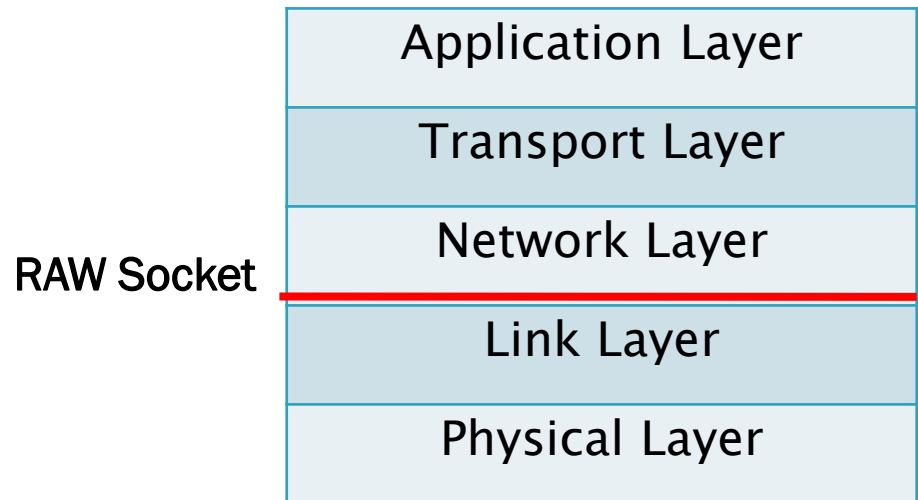
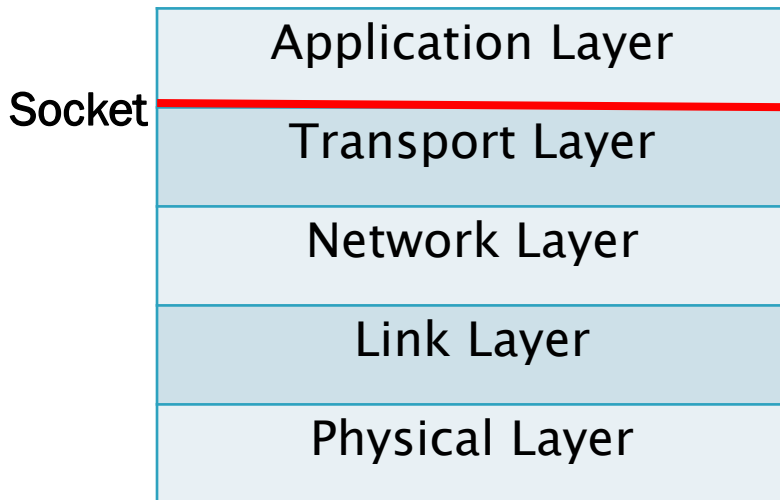
Sum	¹ 10101111	10001001
	10101111	10001010

(A red arrow points from the carry bit '1' of the first sum row to the second sum row.)

Checksum	01010000	01110101
----------	----------	----------

RAW SOCKET

- ▶ Super user privilege is required for raw socket
- ▶ Read and write ICMPv4, IGMPv4, and ICMPv6 packets



RAW SOCKET CREATION

```
int sockfd;  
sockfd = socket(AF_INET, SOCK_RAW, protocol);  
where protocol is one of the constants, IPPROTO_XXX,  
defined by including the <netinet/in.h> header, such as IPPROTO_ICMP
```

```
#include <sys/socket.h>
```

```
int setsockopt(int socket, int level, int option_name,  
               const void *option_value, socklen_t option_len);
```

Ex:

```
const int ttl = 20;  
if (setsockopt(sockfd, IPPROTO_IP, IP_TTL, &ttl, sizeof(ttl)) < 0)    error
```

RAW SOCKET READ OR WRITE

```
#include <sys/socket.h>
```

```
ssize_t recvfrom(int sockfd, void* buf, size_t nbytes, int flags,  
struct sockaddr* addr, socklen_t addrlen);
```

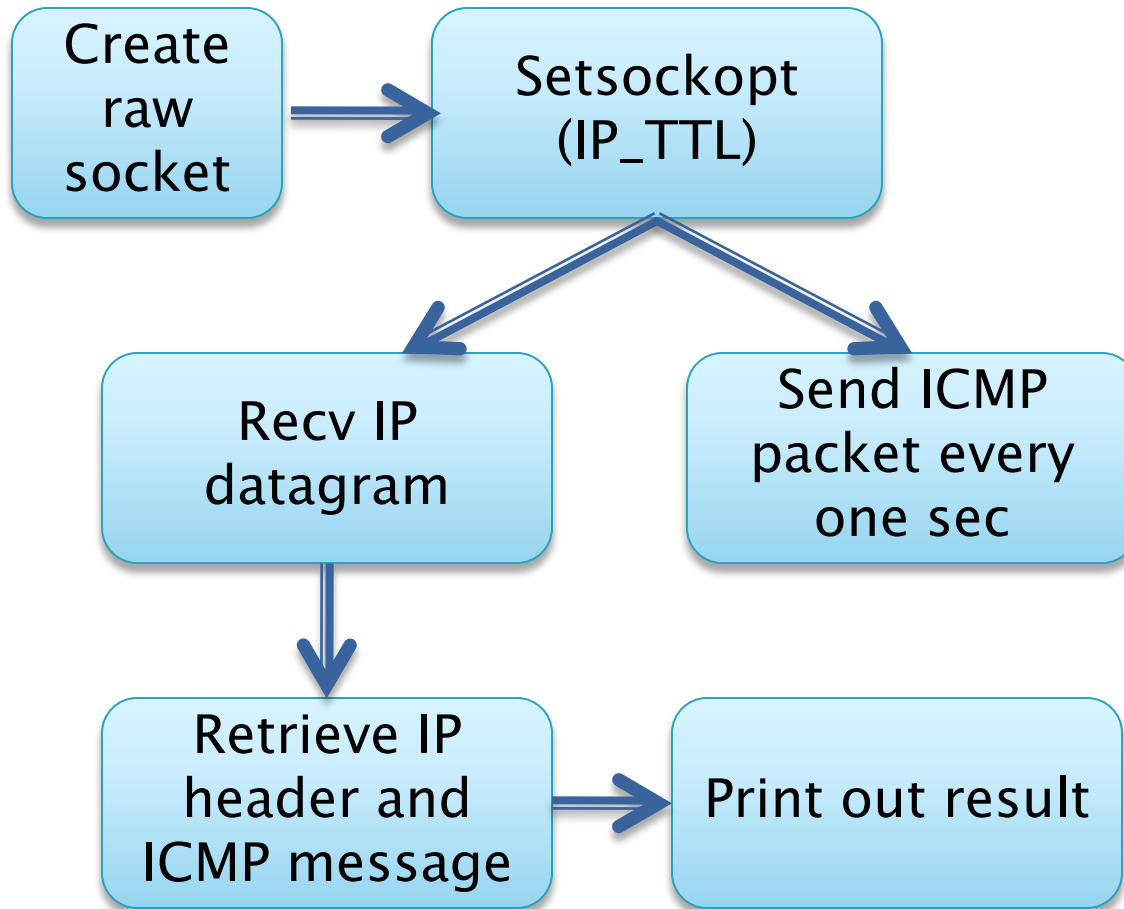
Return :length of message in bytes, -1 on error

```
#include <sys/socket.h>
```

```
ssize_t sendto(int sockfd, const void* buf, size_t nbytes, int flags,  
const struct sockaddr* destaddr, socklen_t destlen);
```

Return :number of bytes sent if OK, -1 on error

PING FLOWCHART

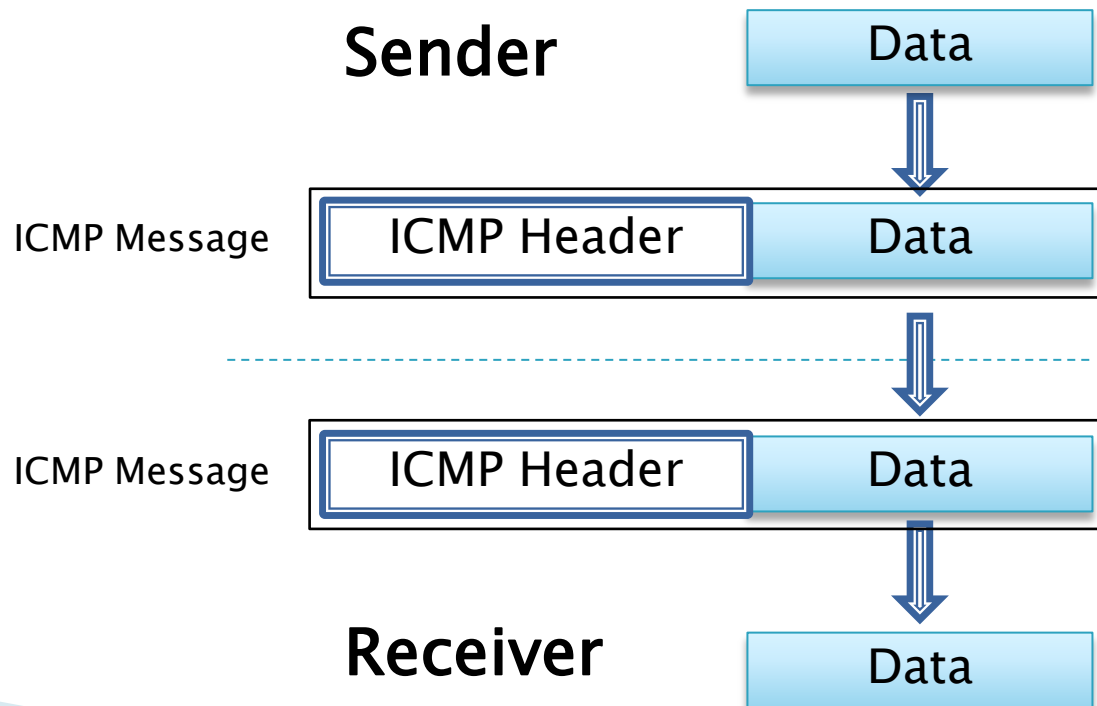


PING TALK

–by Ping tunnel method

PING TUNNEL

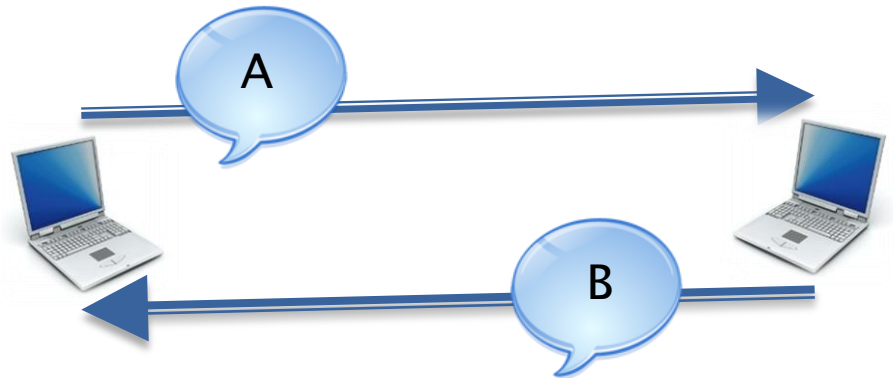
- ▶ Encapsulate data in ping packet's data section
 - Can pass through firewalls without being detected



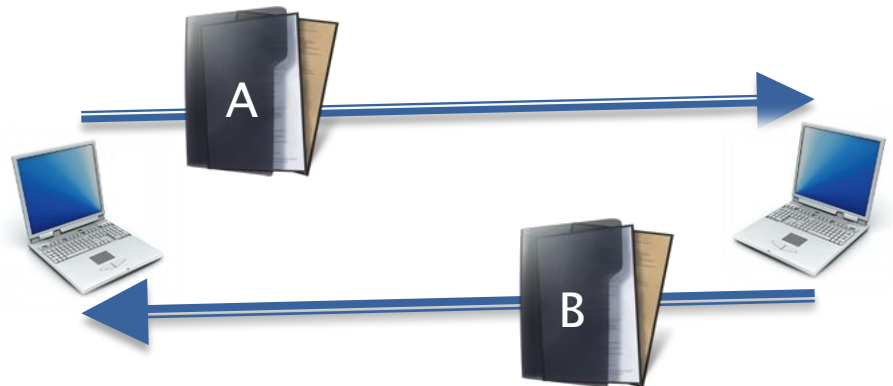
SCHEME (1 / 2)

- ▶ Use Ping tunnel to :

Send text message



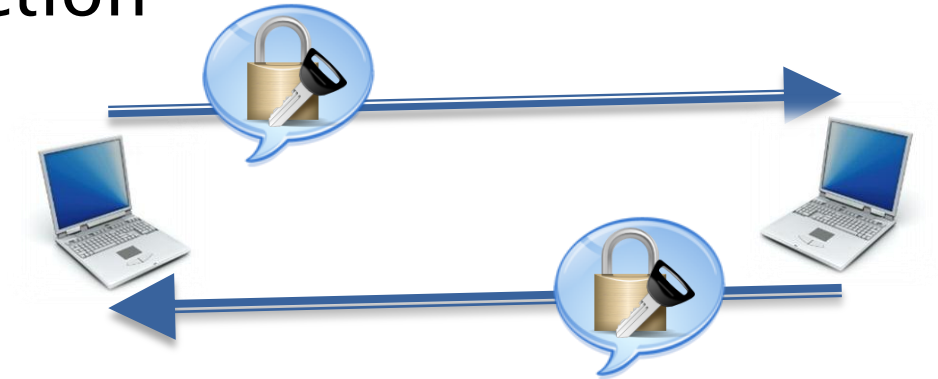
Transmit file



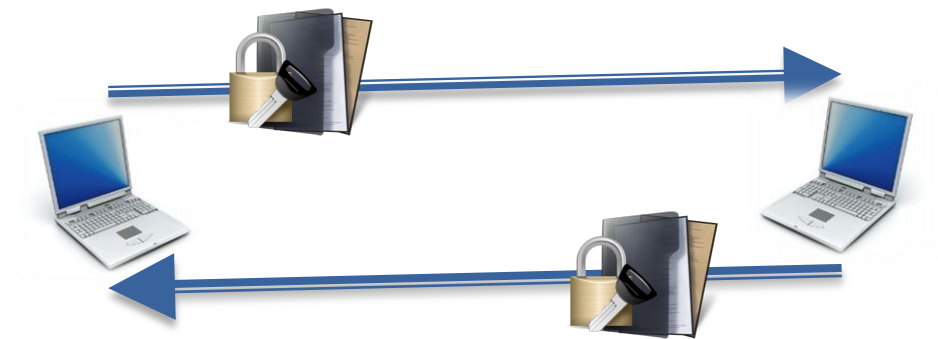
SCHEME (2/2)

- ▶ Add Security protection

- ▶ Sender:
Use base-64
encoding




- ▶ Receiver:
Use base-64
decoding



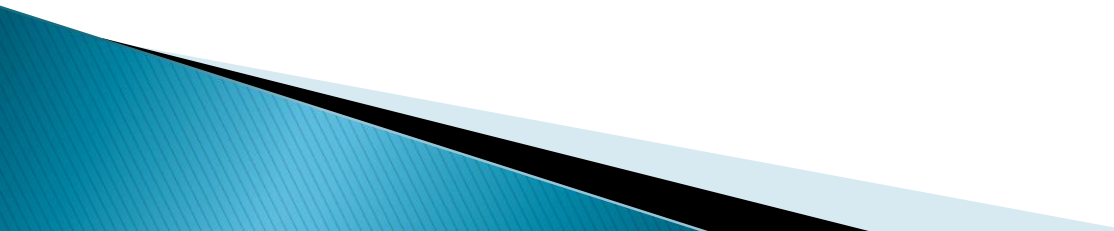
PING REQUIREMENTS(1 / 2)

- ▶ ping <host or IP> -c <Number of request>
-s <Send buffer size>
-t <TTL>
- ▶ Output example:
 - Root~> ping www.google.com.tw -c 2 -s 200
 - Pinging www.google.com.tw(72.14.203.147) with 200 bytes of data:
 - Reply from 72.14.203.147 : seq = 0 byte = 200 RTT = 32 msec
 - Reply from 72.14.203.147 : seq = 1 byte = 200 RTT = 32 msec
 - Root~> ping www.google.com.tw -t 5
 - Pinging www.google.com.tw(72.14.203.14) with 200 bytes of data:
 - Reply from 72.14.203.147 : TTL expired in transit.

PING REQUIREMENTS(2 / 2)

- ▶ User can decide number of request, send buffer size and TTL
 - ▶ Be able to ping domain name and IP address
 - ▶ If host can't reach, then print error message
 - ▶ Deal with TTL expire
 - ▶ RTT – Round trip time
- 

PING TALK REQUIREMENTS

- ▶ Be able to chat with each other(text message) by sending ping packet
 - ▶ Available of transmitting picture with each other by sending ping packet
 - ▶ Implement base-64 encoding/decoding on data packet between sender and receiver
 - ▶ TA will use **Wireshark** to check your encoding method (You can try to test by yourself!)
- 

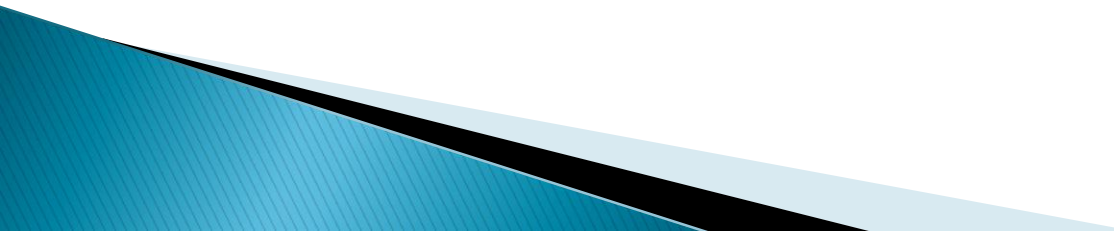
FORMAT REQUIREMENTS

- ▶ Your program should be ...
 - Must be implemented with **C / C++ language**
 - Read data from file (i.e. file I/O)
- ▶ Naming
 - b97902xxx_hw3_pingtalk.**c/cpp**
- ▶ Compression
 - `tar -zcvf b97902xxx_hw3.tar.gz b97902xxx_hw3/`

GRADING POLICIES

- ▶ Ping function – (30%)
- ▶ Ping talk – (40%)
 - Text – (10%) Picture – (20%)
 - Base64 Encoding/Decoding – (10%)
- ▶ Clarity of your code (comments!) – (5%)
- ▶ Demo – (15%)
- ▶ Report – (10%)
 - Execution instruction
 - Ping Talk Flowchart
 - What you do, and how you do it
 - Challenging issues and solutions

REMINDERS

- ▶ Do not cheat! You cheat, you fail!
 - ▶ Do not copy source codes from the Internet
 - ▶ Ask TA if you have any question, except for debugging
 - ▶ For language other than C / C++ : 30% off
- 

DEADLINE

- ▶ Homework due
 - **2011/06/29 23:59:59 +0800**
 - Start your work as early as possible
- ▶ Demo
 - **2011/06/30 & 2011/07/01**
 -
- ▶ For late submission (before demo) – **30% off**
- ▶ For late submission (after demo) – **0**

HAPPY CODING FINALLY ^o^