



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. en computación	2003-1	5070	Redes de área local

PRÁCTICA No.	LABORATORIO DE		DURACIÓN (HORA)
7	NOMBRE DE LA PRÁCTICA	Adaptar códigos libres de ICMP para conocer el desempeño de una red	2 horas

1. INTRODUCCIÓN

ICMP es un protocolo de capa de red que auxilia las funciones de enrutamiento, permitiendo transmitir mensajes de control entre usuarios (host) y enrutadores.

ICMP está definido por la IETF (Internet Engineering Task Force). Los protocolos IP e ICMP pertenecen a la capa 3 del modelo OSI. El protocolo IP es un protocolo con servicio orientado a no conexión y tiene a su cargo un gran número de funciones como lo son: Reenvío de paquetes, Fragmentación, Direccionamiento.

El protocolo ICMP puede ser visto como accesorio el cual se encarga de los mensajes de error y control del protocolo IP. Mping es un programa de uso y distribución libre que generaliza la función del ping. En el código fuente del mping: mping.c se puede modificar directamente el campo "type" del protocolo ICMP, en la rutina "PINGER". Por default esta "echo request" (el valor 8). Este valor provoca que el host destino a que se envía el paquete IP devuelva al host fuente el encabezado del paquete.

2. OBJETIVO (COMPETENCIA)

Que el alumno manipule los sockets de ICMP de manera que pueda acceder a cualquiera de sus funciones como usuario final.

Formuló M.C. Marlene Angulo M.C. Marco Turrubiarres	Revisó M.C. Gloria E. Chávez	Aprobó	Autorizó M.C. Miguel ángel Martínez
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad

3. FUNDAMENTO

4. PROCEDIMIENTO (DESCRIPCIÓN)

A)	EQUIPO NECESARIO	MATERIAL DE APOYO
	Un nodo de interconexión (Hub ó switch) 2 computadoras Cable de red	Lenguaje de programación estructurada: ansi c. Ethereal en ambas computadoras Sistema operativo linux Código libre mping.c

B) DESARROLLO DE LA PRÁCTICA

1.- Compilar mping.c

Para compilar y crear el ejecutable:

```
gcc mping.c -o pingv2
```

Para ejecutar el programa:

```
./pingv2
```

si no se da ninguna instrucción por default despliega un help que muestra las opciones.

Este programa mide los retardos RTT de un host fuente a un destino, p.ej.

```
./pingv2 -c 5 yahoo.com
```

mide los retardos RTT desde el host donde se ejecuta el comando al destino solicitado, en este caso yahoo.com. La opción `-c 5` es para solicitar solamente 5 mediciones.

2.- Modificar mping.c de manera guiada

En el código fuente del mping: mping.c se puede modificar directamente el campo "type" del protocolo ICMP, en la rutina "PINGER". Por default esta "echo request" (el valor 8). Este valor provoca que el host destino a que se envía el paquete IP devuelva al host fuente el encabezado del paquete.

```
./ping2 -c 5 yahoo.com
```

```
PING yahoo.com : 56 data bytes
```

```
64 bytes from yahoo.com : icmp_seq=1 time=101.725 msec
```

```
64 bytes from yahoo.com : icmp_seq=2 time=96.789 msec
```

```
64 bytes from yahoo.com : icmp_seq=3 time=96.270 msec
```

```
64 bytes from yahoo.com : icmp_seq=4 time=93.769 msec
```

```
64 bytes from yahoo.com : icmp_seq=5 time=96.078 msec
```

```
---MPING Statistics5---
packets transmitted to each host
yahoo.com: 5 packets received, 0% packet loss
roundtrip (msec) min/avg/max/cubesum
= 93.769/96.926/101.725/1751109706.1088877555
```

Si cambiamos este valor al 13 (time stamp request) se le pide al host destino que además de devolver de regreso el encabezado del paquete, le grabe el tiempo maquina en que recibio el paquete y el tiempo en que lo esta devolviendo: "rtime" y "ttime".

```
icp>icmp_type = 13;
```

Para leer los datos de tiempo que envia el host destino hay que convertir de hexadecimal a decimal y enviar imprimir a pantalla estos valores. Esta modificación se encuentra en la rutina: PR_PACK.

```
if (icp>icmp_type ==14)
{
gettimeofday(&atv, NULL);
printf("\n src_time: %d",icp>icmp_otime);
num = icp>icmp_rtime;
V=(num & 0xFF000000) >> 24;
V=V+((num & 0x00FF0000) >> 8);
V=V+((num & 0x0000FF00) << 8);
V=V+((num & 0x000000FF) << 24);
printf(" \n rtime: %ld,\n",V);
num=0; V=0;
num = icp >icmp_ttime;
V=(num & 0xFF000000) >> 24;
V=V+((num & 0x00FF0000) >> 8);
V=V+((num & 0x0000FF00) << 8);
V=V+((num & 0x000000FF) << 24);
}
```

Al compilar y crear el ejecutable del programa modificado:

```
gcc mpingv2.c dis
```

y al correr al destino: 148.231.1.3

```
./dis -c 3 148.231.1.3
```

Este sera el resultado en pantalla:

```
src_time: 20935225
```

rttime: 6028577,
ttime: 6028577,
src2_time: 20935263.000000

src_time: 20935228
rttime: 6028586,
ttime: 6028586,
src2_time: 20935280.000000

src_time: 20935350
rttime: 6028700,
ttime: 6028700,
src2_time: 20935387.000000

son 3 mediciones de los diferentes tiempos por los que paso el paquete IP.

C) CÁLCULOS Y REPORTE

El reporte deberá contener el listado del programa mping y comentarios respecto a lo que realiza cada función (pinger y pinger6, PR_PACK y PR_PACK 6, TVSUB, FINIS y CHECKSUM), así como comentarios de la modificación (documentación), para esta práctica se evaluará la habilidad del alumno para establecer comunicación entre dos computadoras.

5. RESULTADOS Y CONCLUSIONES

Serán evaluados por el maestro el programa con preguntas individuales a cada miembro del equipo los resultados deberán ser comprobados mediante Ethereal considerando el código del paquete ICMP que se generó.

6. ANEXOS

- 1.- Estandar RFC 792 <http://www.ietf.org/rfc/rfc792.txt>
- 2.- Código libre mping.c <http://yaqui.mx.l.uabc.mx/~mangulo/mping.c>

7. REFERENCIAS

Behrouz A. Forouzan, "Data Communications and Networking", DeAnza College, Cuarta Edición, ISBN: 0072967757, 2007.
Douglas Comer, "TCP/IP" Ed. Prentice Hall.