

Proyecto 4 Capa de Red (parte 2)

Elementos a implementar

1. **Dirección IP**. (parte 1)
2. **Subred**. (parte 1)
3. **Máscara** de **subred**. (parte 1)
4. **Protocolo ARP**. (parte 1)
5. **IP Packet**. (parte 1)
6. **Router** (**enrutador**): Es un dispositivo de capa de red que viene varios puertos (similar al **hub** y al **switch**). El **router** tiene internamente una **tabla de rutas** que le permite enrutar los **paquetes** entre diferentes **subredes**.
 - **NOTA**: todos los **broadcasts** que lleguen a un **router** por alguno de sus puertos, “mueren” ahí. O sea, los **routers** nunca propagan un **broadcast** de ningún tipo.
7. **Tabla de rutas**. Cada **router** (o **host**) tiene un tabla (que se le configura a través de comandos que veremos en la próxima sección) que permite decidir la ruta correcta para cada **paquete**, basado en el **IP de destino**.
 - Cada **ruta** contiene los siguientes campos:
 - **Destination**: un número **IP** que puede ser de un **host** o de una **subred**
 - **Mask**: la **máscara** que se aplicará para comparar con el **destination**
 - **Gateway**: un número **IP** de otro **router** al cual se enviará el **paquete**. Este valor será cero (0.0.0.0), en el caso en el que la comunicación entre el **host** y el destino sea “directa”. (**Comunicación directa** significa que no pasa por ningún otro **router**, que están en la misma **subred**)
 - **Interface**: un número entre 1 y N (cantidad de puertos del **router** o **host**).
 - Las **rutas** se ejecutan en orden de **prioridad**. La prioridad la da la cantidad de unos de la **máscara**.
 - La **máscara** 255.255.255.255 (todo en 1) es la más prioritaria.
 - La **máscara** 0.0.0.0 (todo en 0) es la última en ejecutarse.
 - Cuando llega un paquete a un **router**, se analiza el **IP** de destino de ese **paquete** y se analizan cada una de las **rutas** (en orden de prioridad) hasta que una de ellas sea capaz de **enrutar** ese **paquete** hacia su destino.
 - Para determinar cual **ruta** es la que puede **enrutar** ese **paquete** hacia su **IP** de destino:
 1. Se coge una **ruta** R
 2. Se le hace un AND al **IP** de destino con el campo **máscara** de la **ruta** R

3. Se compara el resultado del AND con el campo destination de la **ruta** R
4. Si el resultado es igual, el **paquete** se envía por la interfaz correspondiente (especificada en el campo *inteface* de la **ruta**) hacia el **IP** especificado en el campo *Gateway*.
5. Si el resultado del AND no coincide con el *destination* se pasa a evaluar la próxima **ruta**.
6. Si ninguna **ruta** puede **enrutar** dicho **paquete** se le envía el **host** de origen un **paquete ICMP** con “*destination host unreachable*”.

8. **Tabla de rutas** de un **host**:

- Además de los **routers**, los **hosts** también van a tener una **tabla de rutas**, aunque tengan una sola interfaz de red.
- La **tabla de rutas** más simple que puede tener un **host** contiene solamente 2 **rutas**:
 - una **ruta** para especificar la **subred** en la que se encuentra el **host**
 - y la **ruta por defecto**
- **Ejemplo:**
 - un **host** con **IP** 10.6.122.40 y **máscara** 255.255.255.0 es un **host** que se encuentra en la **subred** 10.6.122.0 y que en esa **subred** debe haber un **gateway** que por lo general es 10.6.122.1.
 - 10.6.122.0 255.255.255.0 0.0.0.0 1
 - 0.0.0.0 0.0.0.0 10.6.122.1 1
 - esto se puede leer de la siguiente forma:
 - todos los **paquetes** que vayan a un **IP** de la **subred** 10.6.122.0 se envían directo
 - todos los **paquetes** que no sean para la **subred** 10.6.122.0 se envían a 10.6.122.1 que él debe saber cómo llegar a ese otro **IP**.

9. **Protocolo ICMP**: es un protocolo de capa de red, que usa la misma estructura de los **paquetes IP** pero con un **payload** de 1 byte. (**ICMP** viene de las siglas en inglés *Internet Control Message Protocol*)

- En el **paquete IP** hay un campo Protocolo que habíamos dejado en cero. A partir de ahora si el campo protocolo vale 1 (00000001) significa que nos encontramos con un **paquete IP** que contiene un **mensaje ICMP**.
- El campo *payload size* en el caso de los mensajes **ICMP** siempre va a ser 1.
- El *payload* va a ser un byte, y va a ser una de las siguientes opciones:
 - 0 → **echo reply** (la respuesta a un *Ping*, también conocido como *Pong*)
 - 3 → **destination host unreachable**
 - 8 → **echo request** (lo que conocemos como *Ping*)

- 11 → **time exceeded** (tiene que ver con el campo *TTL* que todavía no estamos usando)

Instrucciones

A continuación de muestran las posibles instrucciones adicionales que se incluyen en este 4to proyecto (parte 2) que puede tener el **script.txt**

- **<time> create router <name> <number-of-ports>**
 - Crea un **router** con esa cantidad de puertos
 - **Nota:** El **IP** de cada interfaz y las **MAC** se asignan con los comandos **ip** y **mac** respectivamente.
 - **ejemplo:** `10 create router rt 4`
- **<time> ping <host> <ip-address>**
 - El **host** correspondiente envia **4 mensajes ICMP** (*ping*) al **IP** especificado.
 - Cada *ping* se envia **100 ms** después del anterior.
 - **ejemplo:** `30 ping pc1 10.10.10.10`
- **<time> route reset <name>**
 - borra la **tabla de rutas** completa de un **router**.
- **<time> route add <name> <destination> <mask> <gateway> <interface>**
 - adiciona una nueva **ruta** a la **tabla** del **router** especificado
 - **ejemplo:** `10 route add rt 10.6.122.0 255.255.255.0 10.6.122.1 1`
- **<time> route delete <name> <destination> <mask> <gateway> <interface>**
 - elimina una **ruta** a la **tabla** del **router** especificado.
 - **nota:** el comando **route delete** tiene que tener exactamente los mismos campos especificados en el comando **route add** de lo contrario el comando no hace nada.
 - **ejemplos:**
 - `10 route add rt 10.6.1.0 255.255.255.0 10.6.1.1 1 # se crea la ruta`
 - `10 route delete rt 10.6.1.0 255.255.255.0 # este no hace nada`
 - `10 route add rt 10.6.1.0 255.255.255.0 10.6.1.1 3 # no hace nada`
 - `10 route add rt 10.6.1.0 255.255.255.0 10.6.1.1 1 # este funciona`

Transmisión de información.

- Cuando un **host** reciba un mensaje **ping** (**paquete IP** con protocolo **ICMP** y *payload* = 8) debe responder un **pong** (**paquete IP** con protocolo **ICMP** y *payload* 0)
- Cuando un **host** recibe un **mensaje ICMP** con *payload* = 3 (*destination host unreachable*) debe detener el envío de paquetes hacia ese **IP** debido a que ese **IP** no se encuentra en la red y/o no es accesible por las rutas configuradas.
- A partir de ahora, cuando se haga un **send packet** ese **paquete** puede pasar por varios **routers**, en cada uno de ellos se evaluará la **tabla de rutas** correspondiente para determinar el próximo paso del **paquete**.

Salida

- En el fichero **_payload.txt** se escribirán, en caso de que un **host** reciba un **mensaje ICMP**, escribirá como hasta ahora el *payload* (que es 1 *byte*) y a continuación el significado del *payload*.
 - Si recibe un 0 escribe todo lo de siempre y ademas un “*echo reply*”

- Si recibe un 8 escribe “*echo request*”
- ... y así con todos los demás **mensajes ICMP**

Entregables

Archivo **steve_rodgers_311_natasha_romanov_312.zip** subido al EVEA que contenga:

- **README.txt** con información para compilar y ejecutar el programa
- [1] archivo **script.txt**
- [2] archivo **config.txt**
- [1] directorio **docs/**
 - [1] decisiones tomadas en la capa física
 - [2] decisiones tomadas en la capa de enlace
 - [3] algoritmos de detección y corrección de errores
 - [4] decisiones tomadas en la capa de red
 - [1] cualquier otra cosa que nos ayude a entender mejor el proyecto
- directorio **output/**
 - [1] un archivo **.txt** por cada **dispositivo**
 - [2] un archivo **_data.txt** por cada **host**
 - [4] un archivo **_payload.txt** por cada **host**

Elementos Configurables

- [1] **signal_time** = <time in ms>
- [3] **error_detection** = <algorithm name>

Elementos Físicos

- [1] **Cable**
- [1] **Computadora / Host**
- [1] **Concentrador / Hub**
- [2] **Cable Duplex**
- [2] **Conmutador / Switch**
- [4] **Enrutador / Router**

Conceptos

- [2] Dirección **MAC** (Dirección **MAC de broadcast**)
- [2] **Trama / Frame**
- [3] Detección y Corrección de **Errores**
- [4] Dirección **IP** y **Máscara** (Dirección de **Subred**, Dirección de **broadcast**)
- [4] Protocolo **ARP**
- [4] **Paquete IP / IP Packet**
- [4] **Tabla de rutas**
- [4] **Ruta**
- [4] Protocolo **ICMP**

Instrucciones

- [1] <time> **create host** <name>
- [1] <time> **create hub** <name> <number-of-ports>
- [1] <time> **connect** <port1> <port2>
- [1] <time> **disconnect** <port>
- [1] <time> **send** <host-name> <data>
- [2] <time> **create switch** <name> <number-of-ports>
- [2] ~~<time> mac <host-name> <mac-address>~~
- [2] <time> **send_frame** <host-name> <mac-address> <data>
- [4] ~~<time> mac <host-name>[:<interface>] <mac-address>~~
- [4] <time> **ip** <host-name>[:<interface>] <ip-address> <mask>

- [4] <time> **send_packet** <host-name> <ip-address> <data>
- [4] <time> **create router** <name> <number-of-ports>
- [4] <time> **ping** <host-name> <ip-address>
- [4] <time> **route reset** <name>
- [4] <time> **route add** <name> <destination> <mask> <gateway> <interface>
- [4] <time> **route delete** <name> <destination> <mask> <gateway> <interface>