

## T1

Hay 3 redes físicas de tecnología Ethernet

Al estar conectadas entre sí, constituyen una interred

En una interred, cada red física se asocia a una red IP

En este caso, las redes IP se identifican por los 24 primeros bits (representados como tres dígitos decimales); el último decimal es siempre cero.

Las redes son 10.0.0.0, 10.0.1.0 y 10.0.2.0

Dentro de cada red, le asignamos una dirección IP (de la red correspondiente) a cada placa ethernet conectada a la red (al switch). Para eso variamos el último decimal de la IP

Notar que un equipo tiene tantas IP como puntos de conexión tenga (en este caso, placas ethernet)

Cada placa ethernet, tiene una dirección (MAC) única, de 48 bits

Tener en cuenta que la MAC es una dirección del link layer (ethernet) y la IP es una dirección de interred

## T2

### NIVEL APLICACION

En HOST 3 se está ejecutando una aplicación que recibe caracteres de la red (de cualquier equipo en la intranet) y los muestra en pantalla.

Esta aplicación, ni bien se inicia, solicita al SO usar protocolo de transporte UDP, y escuchar lo que le llega en el puerto 700.

Todos los demás equipos conocen que este programa usa UDP, puerto 700, y saben además que está corriendo en HOST 3, cuya IP es 10.0.2.20

En HOST2 se ejecuta un programa que quiere enviar los caracteres "AB" para que en HOST 3 sean mostrados en pantalla.

Este programa pide al SO poder enviar caracteres usando protocolo UDP. No pide un puerto específico, el SO le asigna el primero disponible, 11111.

El usuario en HOST2 tipea "AB", el programa los lee de teclado y los envía al destino. Para esto usa una primitiva de nivel aplicación a nivel transporte.

SIMPLIFICADO: esta primitiva sería un `data.request`, y como parámetros debería pasar al nivel UDP: el texto a enviar ("AB"), el puerto de destino (700) y la IP del host de destino (10.0.2.20).

Por ejemplo: `send(700, 10.0.2.20, AB)`

El nivel UDP recibe la solicitud de enviar datos, los datos a enviar, la IP de destino y el puerto de destino.

## T3

### NIVEL UDP

El nivel UDP recibe la solicitud de enviar datos, los datos a enviar, la IP de destino y el port de destino.

UDP construye los datos a enviar al UDP en el host de destino (frame de nivel UDP). Conoce el port de origen (11111); el port de destino y los datos se los paso la aplicación. La direccion IP no esta incluida en el frame UDP, se pasa al nivel IP.

UDP coloca la longitud de su bloque (10) y calcula el checksum para que el otro UDP verifique si los datos fueron bien recibidos (sin errores)

Finalmente, UDP solicita al nivel IP, enviar el bloque construido al destino 10.0.2.20 (direccion pasada por la aplicación).

Esta solicitud corresponde a la interfaz entre el nivel IP y el nivel transporte (UDP). En las implementaciones, todo esto esta dentro del SO, y por lo tanto no es evidente (se implementa mediante buffers compartidos por los niveles).

Por simplicidad, suponemos que se hace a traves de un send (seria tambien un data.request)  
Send(10.0.2.20, bloque\_construido)

## NIVEL IP

Ip ya puede armar su bloque (datagram).

Pone como dato el bloque UDP, la IP de origen es la de este host, la de destino la que le paso UDP. La longitud la calcula sumando el header y los datos a enviar.

Pone el TTL adecuado (suponemos 64) y calcula el checksum para el header

T4

## NIVEL IP

El nivel IP es el responsable del ruteo: en funcion de la direccion IP de destino, debe decidir a que router enviar el datagram para que en algun momento llegue a destino

Para determinar esto, consulta la tabla de reenvio (o ruteo): selecciona una fila de acuerdo a la red a la que este destinado el datagram a enviar.

La tabla indica:

si hay algo que sea para la red 10.0.0.0, hay que enviarlo directamente al destino, porque esta en en una red local a este host. Por ejemplo, si hay que enviar a 10.0.0.20, no necesito pasarle el datagram a ningun router, se lo envio al host directamente.

Si hay algo destinado a cualquier otra red (se conoce como ruta por defecto o default route), la tabla indica que hay que mandarlo al router 10.0.0.1, y este router se arregla para enviarlo al siguiente o (si es adyacente a el) al destino

En todos los casos, en la tabla encontramos una indicacion de la interfaz (placa ethernet) por la que hay que mandar el datagram.

En resumen, para saber por que interfaz y a que equipo hay que mandar el datagram, tomamos su direccion de destino y vemos de que red se trata (en este caso los 3 primeros bytes) y en base a esto, tratamos de encontrar una entrada en la tabla; si no la encontramos, usamos la ruta por defecto.

En este punto sabemos que hay que enviar el datagram al router 10.0.0.1, y que ese envio es a traves de la interfaz eth0 (que es la unica que tenemos)

T5

### **NIVEL IP**

En este punto sabemos que hay que enviar el datagram al router 10.0.0.1, y que ese envio es a traves de la interfaz eth0 (que es la unica que tenemos)

IP solicita al nivel link layer (ethernet) que envíe el datagram.

Que debe agregar ethernet?

1-los datos a enviar los tiene, es el datagram IP

2-el checksum lo puede calcular una vez que tenga todo el resto

3-el tipo de protocolo lo sabe, porque el que le hace el requerimiento es IP

4-la direccion ethernet de origen la conoce, porque es la de esta pala ethernet.

5-la direccion ethernet de destino NO la conoce, y debe ser suministrada por IP.

Procedimiento para que IP pueda conocer la direccion ethernet de destino(ARP)

IP en el HOST1 desea conocer la direccion ethernet del ROUTER1, cuya IP conoce y es 10.0.0.1

Envia un frame ARP (ARP request), con destino a todos los equipos en la red ethernet. Este frame solicita que aquel nodo cuya IP es 10.0.0.1 le responda indicandole su direccion ethernet (ARP reply)

T6

### **NIVEL ETHERNET (LINK LAYER)**

En Host 2:

- Arma el frame ethernet con los datos pasados por IP
- Envia el frame, que es entregado por el switch al router 1

En Router 1

- La placa ethernet del router 1 reconoce que la direccion de destino es la de ella:
  - desencapsula el datagram IP
  - lo entrega al nivel IP (sabe que es IP por el tipo de protocolo)

### **NIVEL IP (en Router 1)**

- Extrae la direccion de destino del datagram que le llego (10.0.2.20)
- Como esa direccion IP NO es de este equipo, debe reenviarlo para que llegue a destino
- Consulta su tabla de ruteo, la red de destino (10.0.2.0 ) no coincide con ninguna entrada en la tabla, por lo tanto usa la entrada por defecto. Debe enviar el datagram al proximo router: 10.0.1.2 a traves de eth1

Pide al nivel ethernet de eth1, que envíe el datagram (debe tambien usarse ARP para descubrir cual es la ethernet asociada a 10.0.1.2)

Tener en cuenta que el datagram IP NO se modifica

T7

**En los pasos del 6 al 10 se muestra la operación del router 2 y del host 5.**

**El el caso de host 3, el nivel IP determina que la direccion IP de destino del datagram es para el equipo, por lo tanto, mira el campo tipo de protocolo en IP (que indica que el protocolo de nivel transporte es UDP), y lo pasa a UDP.**

**UDP, a su vez, chequea por errores (opcional) y termina pasando los datos al nivel de aplicación. Podria haber varias aplicaciones usando UDP, pero UDP la pasa a la correcta mirando el port de destino**