Informacion y entropia

Informacion: $I(e) = -\log_2 P(e)\ \text{bits}$

Entropia de una fuente: $H(S) = -\sum_{s \in S} P(s) \cdot \log_2 P_S(s)$

Entropia bajo equiprobabilidad: $H(S) = \log_2 |S|$

Bit: Cantidad de información obtenida al especificar una de dos posibles alternativas igualmente probables

Fuente de memoria nula: Fuente en la que cada emisión es estadísticamente independiente

Codigos

Que es un codigo?

- Un alfabeto es un conjunto de símbolos.
- Dado un alfabeto fuente Σ, un código es una correspondencia entre todas las secuencias posibles de símbolos de Σ a secuencias de símbolos de otro alfabeto X (alfabeto código).
- Muchas veces son utilizados a los efectos de lograr una representación más eficiente de la información (i.e., para eliminar redundancia).

Código bloque y código no singular

Un **código bloque** es aquél que asigna cada símbolo de Σ a una secuencia fija de símbolos de X : $C: \Sigma \rightarrow X^*$

Un código C se dice **no singular** si todas sus palabras son distintas (i.e., si C es una función inyectiva).

Código instantáneo y código unívocamente decodificable

Un código es **unívocamente decodificable** si ninguna tira de símbolos del alfabeto código admite más de una única decodificación.

- Definición más formal: si su extensión de orden n es no singular ∀n ∈ N.

Un código es **instantáneo** cuando es posible decodificar las palabras sin necesidad de conocer los símbolos que la suceden.

- Condición necesaria y suficiente: ser **libre de prefijos**, no codificar ningún símbolo como prefijo de otro.

Teorema: Codigo instantaneo => codigo univocamente decodificable

Longitud de codigo

Dado un código C sobre una fuente S, la **longitud media** de C, L(C), se define como $L(C) = \sum_{s \in S} |C(s)| \cdot P_S(s)$

Un código se dice **óptimo** si no existe un código para la misma fuente con menor longitud media. En otras palabras, utiliza en promedio el menor número posible de bits para codificar un mensaje.

Teorema: Codificacion sin perdida de informacion $H(S) \leq L(C)$

Todo código que satisface esto se dice que codifica **sin pérdida de información.**

Capacidad y teorema de Shannon

La **capacidad** C de un canal es la velocidad teórica máxima de transmisión, y viene dada por el...

Teorema de Shannon: $C = B \cdot \log_2(1 + \text{SNR})$

- *B:* Ancho de banda (Hz)
- *SNR:* Relacion señal-ruido (veces)

Delay, propagación y transmisión

El **delay** representa el tiempo total que tardamos en enviar información de un punto a otro: $\text{Delay} = T_{tx} + T_{prop}$

- T_{tx}: tiempo de transmision = | datos | / V_{tx}
- T_{prop}: tiempo de propagacion = D / V_{prop}

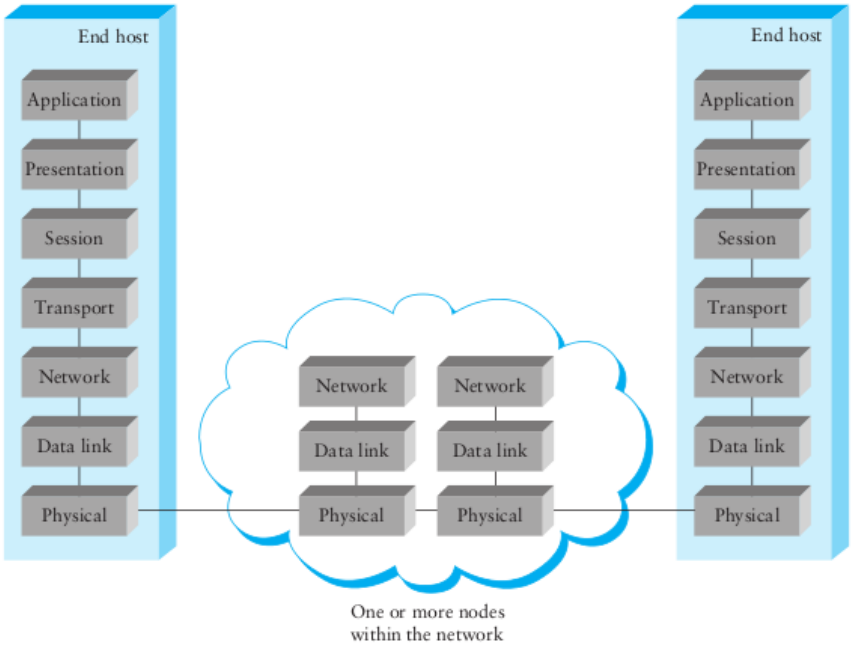
Capacidad de volumen

La **Capacidad de volumen** la cantidad de bits que entran en el medio desde que se envía el primer bit hasta que éste llega al receptor: $C_{vol} = \text{Delay} * V_{tx}$

Protocolos punto a punto

Arquitectura en capas

Las comunicaciones se dan en capas que se brindan servicios entre sí



Protocolos punto a punto

Conceptos

- **Caño serial** (no hay desorden)
- **Sujeto a rudio impulsivo** Lo que se recibe puede no ser lo que se envió (error de transmisión)

Objetivos

- **Framing -** Encapsular los bits en frames agregando información de control - Cómo los codifico/decodifico?
- **Proveer servicio a la capa superior** - ¿Confiable o no confiable?
- **Control de Errores** - ¿Se produjo algún error? ¿Que hacemos con los errores?
- **Control de Flujo** - (Más adelante: en nivel de transporte)

Framing

Cómo se separan los frames en un tren de bits consecutivos?

- *Largo fijo*
- *Largo en el encabezado*
- *Delimitadores con bit-stuffing*
- *(Violación de código)*

Eficiencia de frame: $\eta_{frame} = \frac{largo\ de\ los\ datos}{largo\ total\ del\ frame}$

<u>Frames de largo fijo:</u> Probabilidad de error: La probabilidad de que el frame llegue bien depende del largo del frame.

Detección y Corrección de errores

- *Bit de paridad*
- *CRC*
- *Checksum*
- *Hamming*
- *Reed-Solomon*
- *MD5*

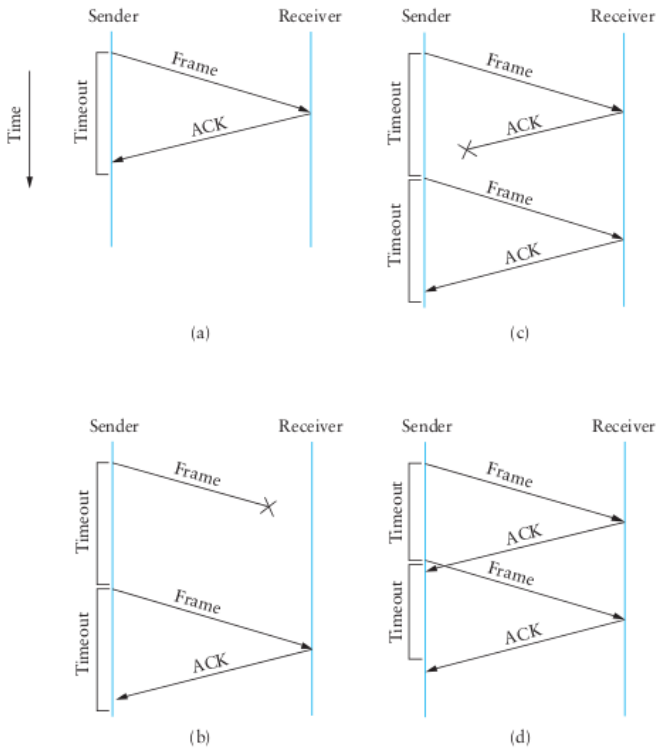
Retransmisiones

- *Explícitas* (mensajes de control específicos para pedir un datos nuevamente)
- *Implícitas* (cuando ocurre un time-out se asume que el dato se perdió)

Tipos de servicio

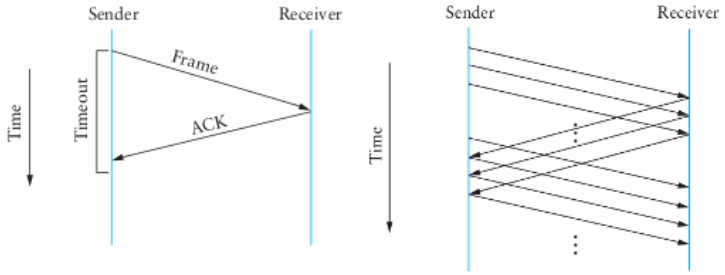
- <u>*Sin conexión y sin reconocimiento:*</u> Los datos se envían sin necesidad de saber si llegan bien.
- <u>*Sin conexión y con reconocimiento:*</u> Los datos se envían y se asegura la correcta recepción mediante el aviso explícito (ACKs)
- <u>*Orientado a conexión:*</u> Además de asegurar la correcta recepción de los datos. Se mantiene un estado de conexión (una sesión)

Transmision confiable: Stop and wait



- Cada Frame debe ser reconocido por el receptor
- Surge la necesidad de secuenciar los frames para evitar el problema de las reencarnaciones
- Para Stop & Wait se necesita poder secuenciar 2 frames.

Transmisión confiable: Sliding Window



- *Ventana de emision:* $SWS = V_{tx} * RTT/|Frame|$
- *Enviar segun:*

$UltimoFrameEnviado \leq UltimoFrameReconocido + SWS$

Debemos aumentar la ventana de emisión para aprovechar mejor el canal:

$$SWS = V_{tx} * RTT/|Frame|$$

Si el receptor dispone de buffers, se puede usar SACK. Sino, no aporta a la eficiencia del protocolo.

$$RWS = \begin{cases} SWS & \text{Si hay SACK} \\ 1 & \text{Si no} \end{cases}$$

Y para distinguir reencarnaciones

$$\#frames \geq SWS + RWS$$

Eficiencia

¿Cuánto tiempo se está transmitiendo con respecto al tiempo bloqueado esperando?

$$Eficiencia = \frac{T_{tx}(V)}{RTT(F)}$$

Con T_{tx} *(V)* el tiempo de transmisión de una ventana y *RTT (F)* el tiempo de ida y vuelta de un frame.

Aumentar la eficiencia es estar bloqueado lo menos posible.

Medios Compartidos

Acceso Compartido: Un medio físico para varios hosts

Ethernet

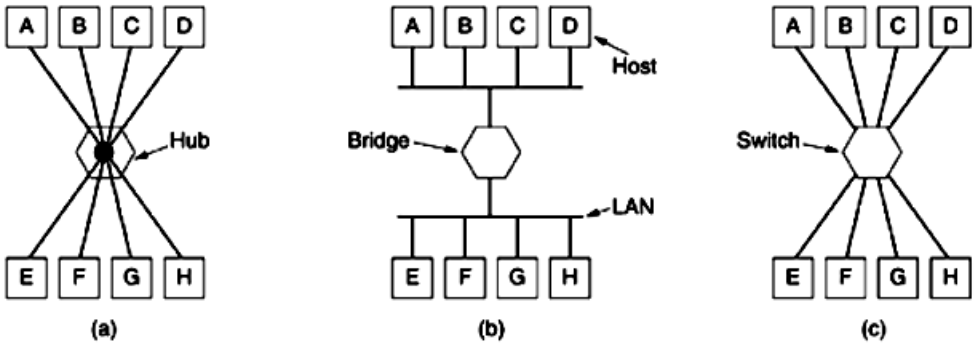
- IEEE 802.3
- Formato de frame
- CSMA/CD
- Exponential backoff

Learning Bridge y Spanning Tree Protocol (STP)

- Formato BPDU
- Algoritmos

LAN

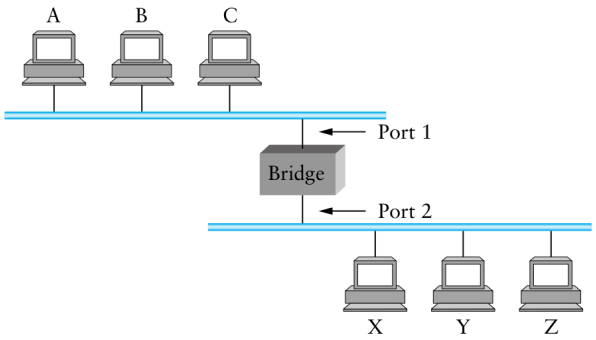
- Conectar enlaces por razones de: heterogeneidad, distancia, aislamiento, redundancia, seguridad, eficiencia, escalabilidad.
- Distintos tipos de multiplexores. Se pueden caracterizar por la capa o nivel en que trabajan.
  - *Físico:* Repetidores y hubs.
  - *Enlace:* Bridges y switches.
  - *Red:* Routers. Gateways?



Definicion: Conjunto de estaciones que comparten dominio de broadcast.

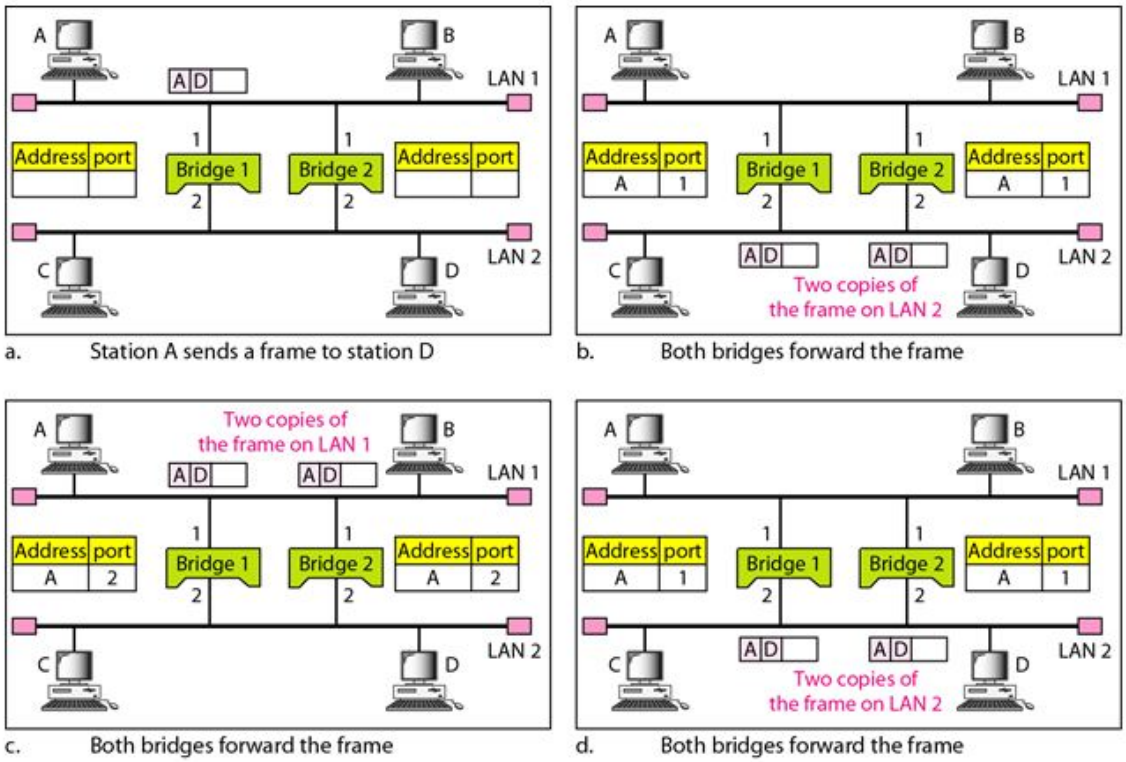
Learning bridge

Los switchs aprenden =>
Relacionan direcciones (i.e.: MAC) con interfaz en función del tráfico en la LAN.



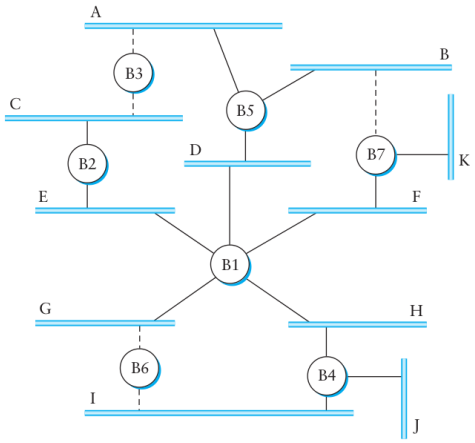
Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

Topologias con ciclos

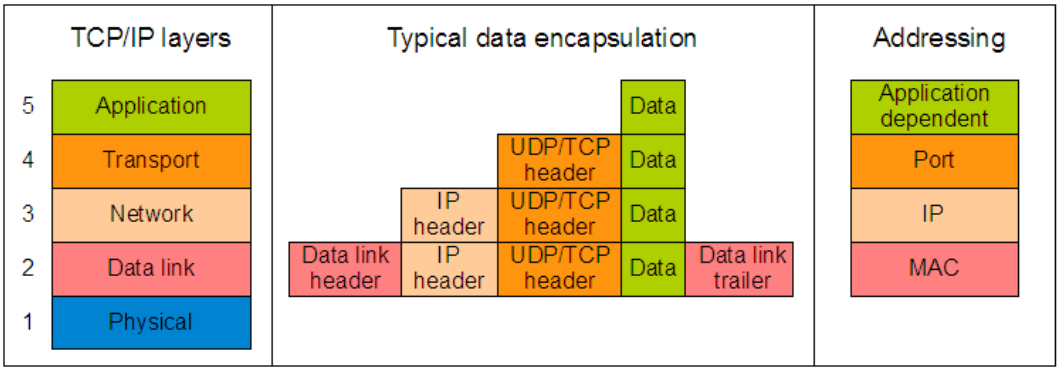


Spanning tree protocol

Sacar ciclos

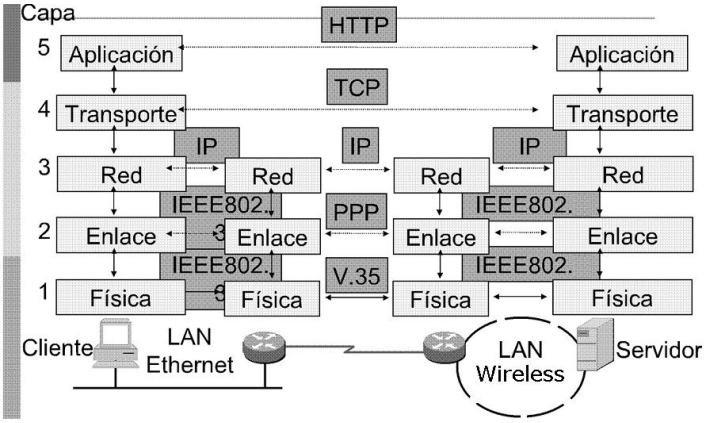


Internetworking IP (pt. 1)

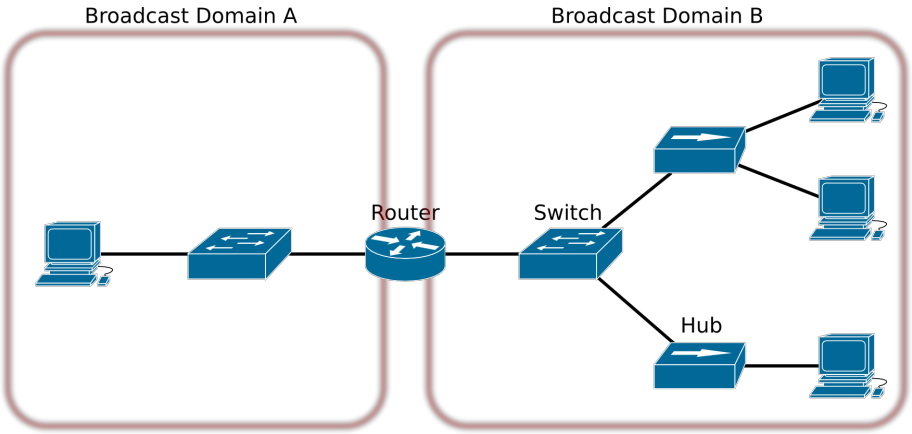


Capas, encapsulamiento y direccionamiento

Ejemplo: acceso a servidor web



Internetworking



Problemas cuando se conectan redes

- *Heterogeneidad*
  - Los usuarios de un tipo de red quieren ser capaces de comunicarse con usuarios de otro tipo de redes.
  - Establecer conectividad entre hosts de dos redes diferentes puede requerir atravesar varias otras redes intermedias, cada una de las cuales puede ser a su vez de otro tipo.
- *Escalabilidad*
  - Ruteo: ¿Cómo podemos encontrar un camino eficiente a través de una red con millones o quizás billones de nodos?
  - Direccionamiento: Es la tarea de proveer indentificadores adecuados para todos esos nodos.

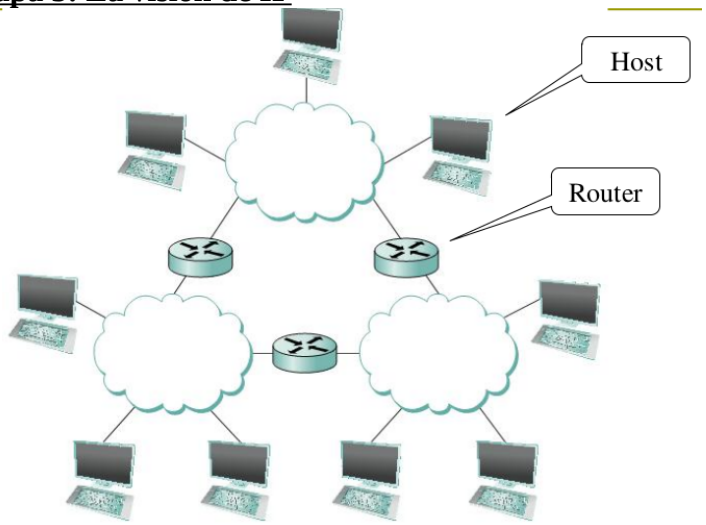
Modelo de servicio de IP

- Sin conexión (basado en datagramas).
- Best-effort delivery “mejor esfuerzo” (servicio no confiable):
  - Los paquetes se pierden.
  - Los paquetes pueden ser entregados fuera de orden al destino.
  - Los paquetes se pueden retrasar por un tiempo largo en la red.
- Similar al envío de mensajes de texto o SMS.

- Define un esquema de direccionamiento global (las direcciones IP son globalmente únicas en la red).

Modelo de capas para IP

Capa 3: La vision de IP



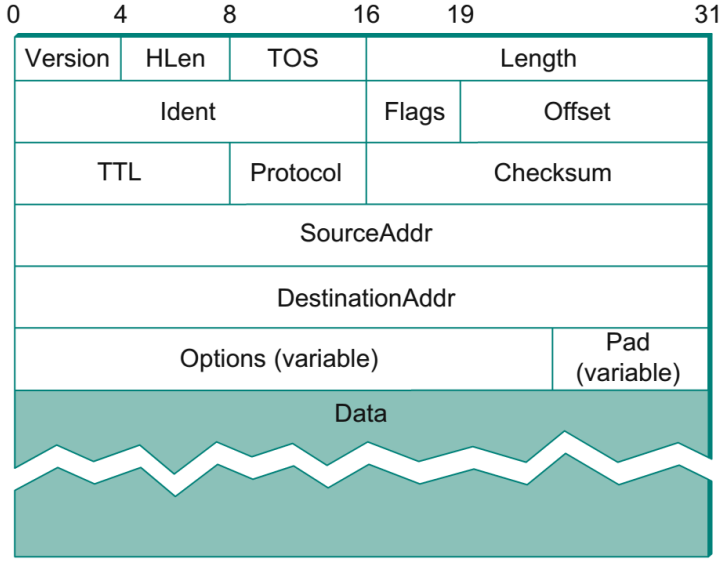
Situación de los protocolos de Internet en el modelo de capas

- El protocolo IP (a nivel de red) es el ‘pegamento’ que mantiene unida la Internet.
- Es capaz de funcionar sobre una gran diversidad de protocolos a nivel de enlace y de medios físicos.
- Un slogan popular en las reuniones de Internet es ‘IP over everything’ indicando la flexibilidad de IP que se adapta a cualquier medio físico y protocolo del nivel de enlace.

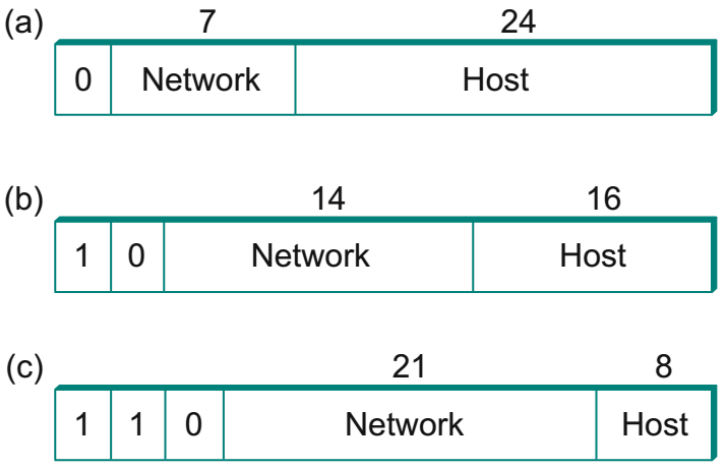
Direcciones IP

- Cada host y router en Internet tiene al menos una dirección IP.
- En realidad las direcciones se asignan a las interfaces. Por ejemplo, si un host tiene varias interfaces (host „multihomed“) cada una tendrá una dirección IP.
- Las direcciones IP tienen una longitud de 4 bytes (32 bits) y se suelen representar como cuatro números decimales separados por puntos (notación dot), ej.: 147.156.135.22
- En principio cada uno de los cuatro bytes puede tener cualquier número entre 0 y 255, aunque algunas direcciones están reservadas.
- Son globalmente unicas y jerarquicas

Ubicacion de direcciones en el datagrama IP



Clases de direcciones

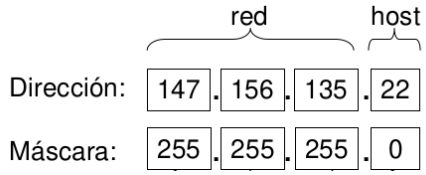


Direcciones y mascaras

- Los hosts y routers interpretan las direcciones IP separándolas en dos partes, la de red y la de host:

Red (n bits)	**Host** (32-n bits)

- La longitud de cada parte se indica mediante un parámetro denominado máscara de red.
- La máscara tiene también una longitud de 32 bits y está formada por un conjunto de unos seguido de ceros. Los unos indican la parte red.
- Como la dirección IP, la máscara se expresa mediante cuatro números decimales separados por puntos, ej.: 255.255.255.0
- Al configurar la dirección IP de una interfaz hay que especificar la máscara utilizada. Por ejemplo:



  Esta interfaz está en una red con 256 direcciones, desde la 147.156.135.0 hasta la 147.156.135.255

Uso de la primera y ultima direcciones de red

Cuando tenemos una red, por ejemplo la 40.40.25.0 con máscara 255.255.255.0:

- La primera dirección posible (40.40.25.0) identifica la red.
- La última dirección posible (40.40.25.255) es la de broadcast en esa red.
- El rango asignable en este caso sería desde 40.40.25.1 hasta 40.40.25.254.
- No se puede asignar a una interfaz ni la primera ni la última direcciones de cada red.

Asignación de dirección IP a un host

La asignación de direcciones y máscaras puede hacerse:

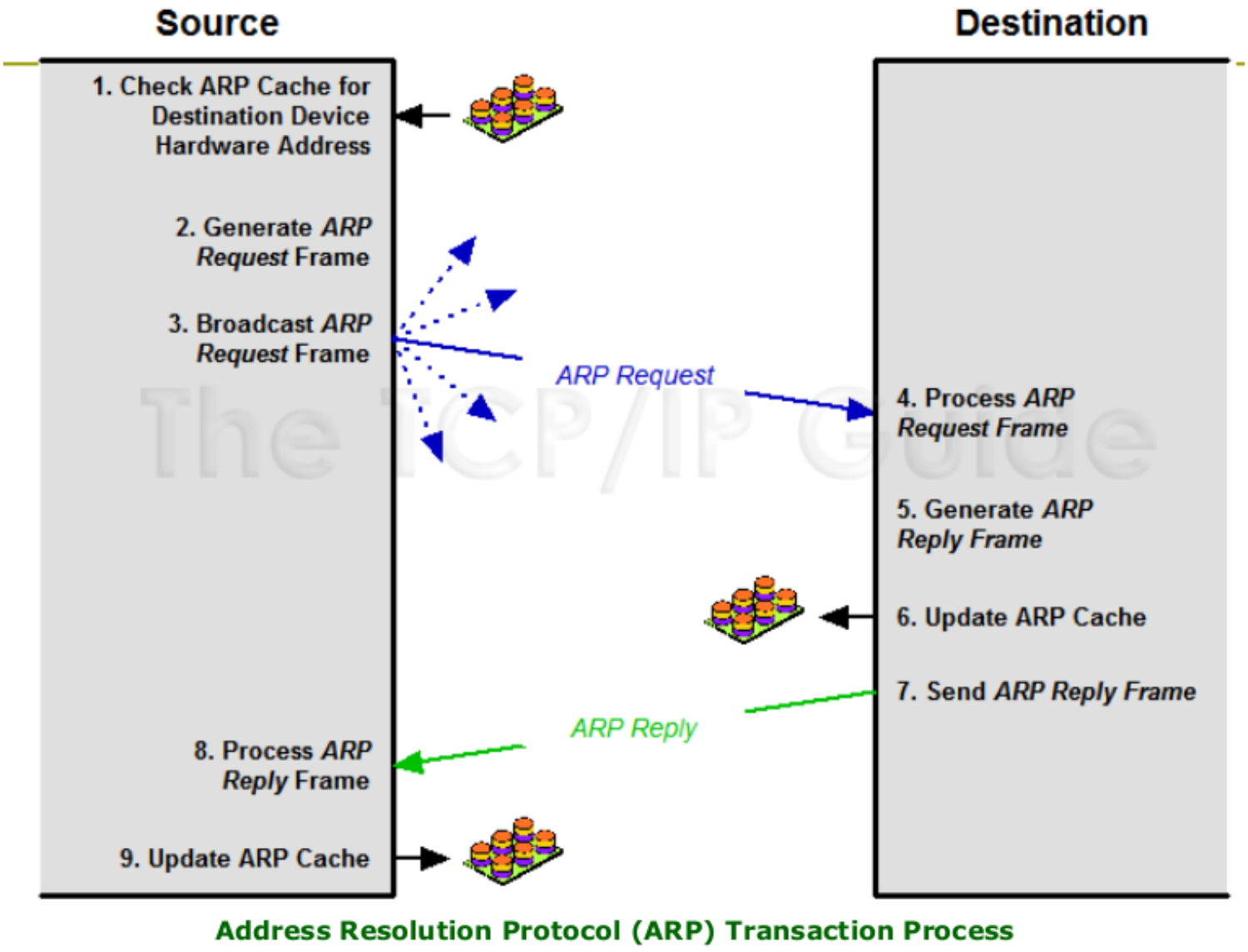
- Por configuración manual en el propio equipo.
- Automáticamente, mediante un protocolo de asignación de direcciones desde un servidor: típicamente DHCP.

Normalmente le asignamos además al host un router por defecto („puerta de enlace predeterminada“ o „default gateway“). No es obligatorio.

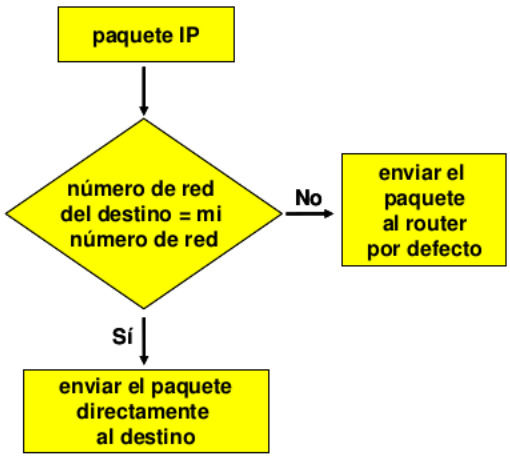
Enrutamiento en un host

Desde el punto de vista de un host el mundo se divide en dos partes: sus vecinos (los que tienen la misma dirección de red) y el resto del mundo. Con sus vecinos habla directamente, con los demás lo hace a través del router.

ARP



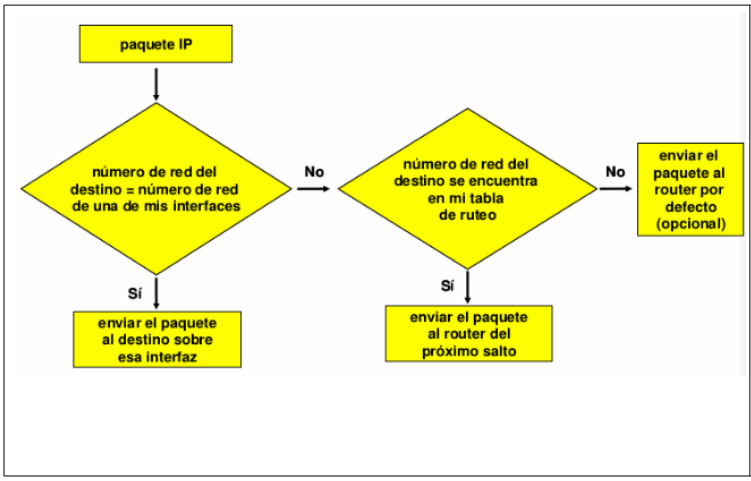
Enrutamiento en un host

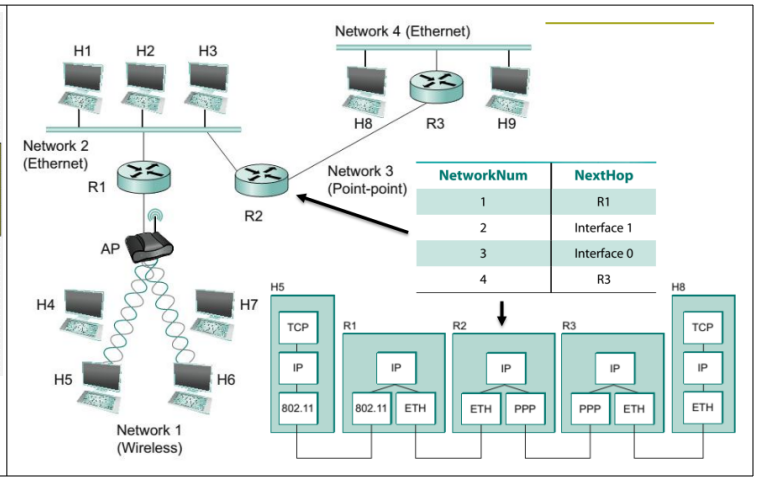


Objetivo: Transportar los paquetes IP desde el origen al destino.

- Cada paquete "viaja" de router en router.
- El paquete se enruta de acuerdo con su dirección de destino.
- Las direcciones origen y destino no se modifican en el proceso.
- Para transportar el paquete al destino, cada router mantiene una tabla de la forma general: **red destino | proximo salto**
- Las tablas pueden ser cargadas de manera manual por el administrador de la red (enrutamiento estático) o de manera automática mediante algoritmos de ruteo (enrutamiento dinámico).

Enrutamiento en la red





Notacion

Network	Next hop
172.16.5.0/24	IF 0/1
10.4.2.0/27	IF 0/0
192.168.2.0/26	10.4.2.25
Default	10.4.2.25

Network (Red): Red destino

Next hop (Proximo salto):

- Interface de salida, si la red destino se encuentra directamente conectada a esta interface, o
- Direccion IP del proximo salto, si la red destino es una red remota

Ruteo

Forwarding:	**Routing:**
Consiste en seleccionar un puerto de salida basándose en la dirección destino y en la tabla de ruteo.	Proceso por el cual se construye la tabla de ruteo.

Formas de enrutamiento

Estatico:	**Dinamico:**
Genera carga y tiempo de administración de red en redes grandes, debe configurarse **manualmente** el enrutamiento en cada router de la red.	No genera mucha carga administrativa porque los routers **aprenden** a enrutarse de los demás routers de la red.
Los routers: • No comparten su tabla de enrutamiento con los routers vecinos. • No tienen capacidad de reacción ante un **fallo/cambio** en la red.	Los routers: • Comparten su tabla de enrutamiento con los routers vecinos. • Tienen capacidad de reacción ante un **fallo/cambio** en la red.

Algoritmos (distribuidos) de enrutamiento

Distance Vector – RIP:

1. Cada nodo construye un arreglo unidimensional conteniendo la *distancia* a todos los demás nodos.
2. Al iniciar se asume que cada nodo conoce a sus vecinos, cuya distancia es 1. Al resto se asigna ∞.
3. Distribuye el arreglo a todos sus vecinos inmediatos.
4. Por cada mensaje que se recibe, se actualiza la tabla de distancia sumando 1 a los nodos alcanzados por el vecino. Si esa distancia es menor que la conocida, se aprende, si no, se descarta.

Propiedades:

- En ausencia de cambios en la topología, solo toma unos pocos intercambios de mensajes entre los vecinos antes de que cada nodo logre completar su tabla.
- Se dice que el proceso converge cuando todos los nodos obtienen una tabla de forwarding consistente.
- El algoritmo es distribuido, entonces ningún nodo tiene toda la información de la tabla, pero tiene una vision consistente de la red.
- Actualizaciones:
  - **Periódicas,** envían un update automáticamente en un intervalo de tiempo, aún si nada ha cambiado.
  - Disparados por cualquier mensaje que llegue que genere un **cambio** en la tabla, entonces la nueva tabla es transmitida.
  - Ante la **caída** de un nodo

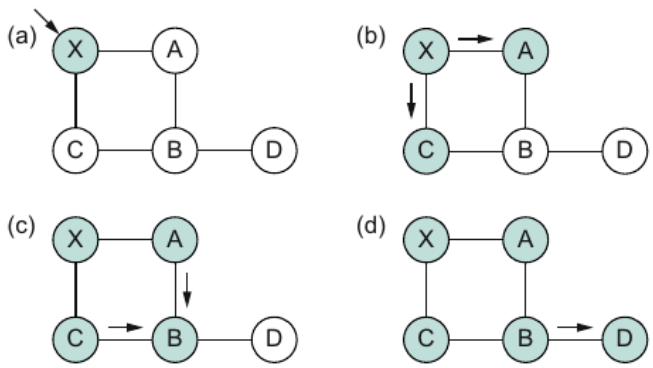
Link State - OSPF:

Se asume, igual que en distance-vector, que cada nodo conoce el estado y costo del enlace con todos sus vecinos.

La idea consiste en **diseminar** el conocimiento que tiene cada nodo sobre sus vecinos, a todos los nodos de la red.

Cada nodo tendrá entonces en su tabla, información suficiente para construir un mapa completo de la red, pudiendo encontrar el **camino mínimo** a cualquier nodo.

Reliable flooding:

Es el proceso por el cual nos aseguramos que todos los nodos participantes del protocolo de ruteo consigan una copia del link state de todos los otros nodos. La idea es enviar a todos mis vecinos mi información de link state y toda la información que reciba de mis vecinos también enviarla a todos los vecinos. Este proceso continua hasta que la información haya llegado a todos los nodos. ¿Cómo nos aseguramos que la inundación sea confiable (que todos reciban la copia mas reciente)?	

LSP:

Cada nodo crea un paquete LSP (link state package) que contiene los siquientes campos:

- ID del nodo que crea el paquete
- lista de todos los vecinos conectados a ese nodo y sus respectivos costos.
- número de secuencia
- tiempo de vida del paquete.

Los dos primeros campos son justamente la información necesaria para poder armar el grafo de la red.

Los últimos 2 son para poder realizar la inundación confiable.

Como funciona reliable flooding con LSP:

- Los nodos almacenan los LSP que reciben en sus tablas

- El pasaje de LSP entre vecinos se asegura mediante mecanismos de ACK y retransmisión.
- Si recibo un LSP de X y **NO** lo tenia almacenado, lo almaceno y lo propago.
- Si recibo un LSP de X y **SI** lo tenía verifico el número de secuencia
  - si el recién llegado es más nuevo (mayor número de secuencia) que el almacenado, me lo quedo y lo retransmito a todos menos al que me lo envió.
  - si el recién llegado es más viejo o igual que el almacenado, lo descarto.
- El hecho de no volver a enviar el paquete de vuelta al que me lo envió, ayuda a que la inundación termine.
- Como los nodos envían la información a todos sus vecinos conectados, entonces la información más reciente eventualmente alcanzará a todos los nodos.

LSP (cont):

- Al igual que RIP, los LSP se generan o bien periódicamente (en el orden de horas) o ante un cambio en la topología.
- El número de secuencia en los LSP no se supone que se agoten (64 bits). En caso de reboot vuelven a 0, pero si encuentra un LSP suyo, actualiza su número de secuencia.
- El campo TTL de los LSP se decrementa cada vez que se recibe y se inunda al resto de los vecinos.
- EL campo TTL también sirve para añejar LPS almacenados en las tablas de los nodos (cada un determinado tiempo se decrementa)
- Los LSP que agotan un TTL se inundan para avisar que esa información es vieja y hay que descartarla.

Tabla de forwarding:

Terminada la inundación, sé que todos los nodos de la red recibieron al menos 1 LSP de cada uno del resto de los nodos de la red. Esto significa que disponemos de la información suficiente para armar un grafo de la red completa.

Para armar la tabla de forwarding OSPF utiliza una variante del algoritmo de caminos mínimos de Djikstra llamada forward search.

Highlights de RIP y OSPF

Distance-Vector (RIP)	**Link-State (OSPF)**
Envía Informacion de TODA la red SOLO a sus vecinos directos	Envía Informacion SOLO de sus vecinos directos a TODA la red
< *Destino, distancia* >	< ID nodo, vecinos, SeqNum, TTL >

Internetworking IP (pt.2)

ICMP (Internet Control Message Protocol)

- ICMP permite **reportar** diversas incidencias o situaciones excepcionales que pueden producirse en el envío de un datagrama.
- Todos los mensajes ICMP se envían en datagramas IP (valor 1 en el campo protocolo).
- ICMP permite implementar dos **herramientas** fundamentales para **diagnosticar problemas** en la red:
-

Comando	Mensaje	Significado	Que nos dice?
Ping	Echo request / Echo reply	Sirve para comprobar la accesibilidad de la IP remota. Por cada paquete enviado se recibe una respuesta con el tiempo de ida y vuelta	permite verificar la conectividad IP entre dos puntos de la red
Traceroute	Time exceeded	Datagrama descartado por agotamiento del TTL	permite determinar la ruta efectuada por un paquete hacia un destino determinado en la red.

Direcciones IP especiales

Dirección	Significado	Ejemplo
255.255.255.255	Broadcast en la LAN propia	255.255.255.255
Parte Host a ceros	Identifica una red	147.156.0.0 255.255.0.0
Parte Host a unos	Broadcast en una red remota	147.156.255.255 255.255.0.0
127.0.0.1	Dirección Loopback (para pruebas)	127.0.0.1

Direcciones IP privadas

Existen tres rangos de direcciones IP que han sido declarados como **privados**. Las organizaciones pueden utilizarlos internamente como deseen. La única regla es que los paquetes que contienen estas direcciones **no pueden aparecer en Internet**. Los tres rangos reservados son:

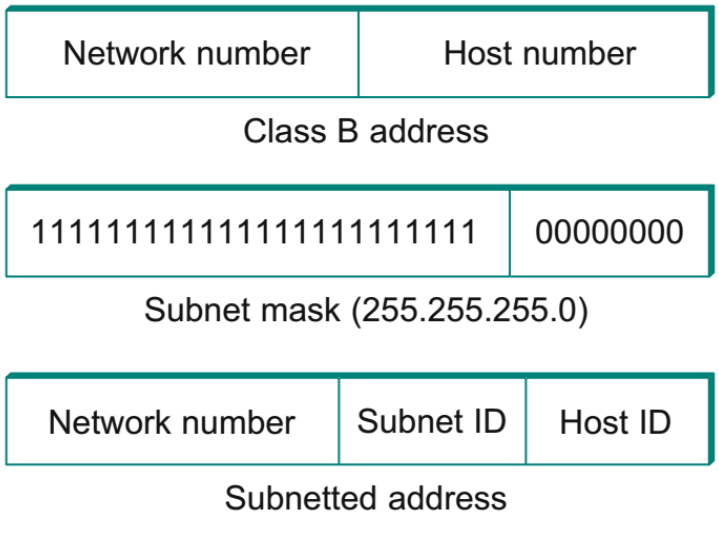
10.0.0.0 – 10.255.255.255/8 (16.777.216)

172.16.0.0 – 172.31.255.255/12 (1.048.576)

192.168.0.0 – 192.168.255.255/16 (65.536)

Subredes IP

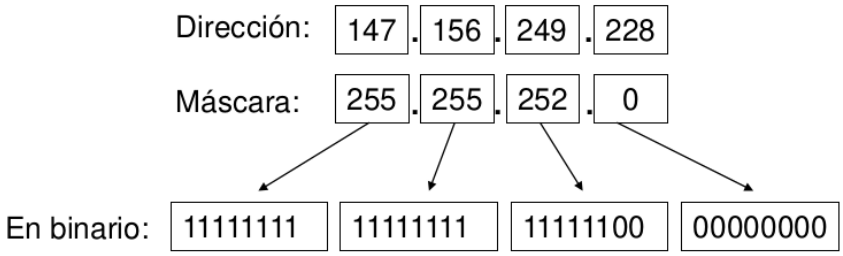
A menudo la red de una organización está a su vez formada por varias redes. En estos casos suele ser conveniente partir de una red grande que dividimos en trozos más pequeños llamados subredes.



Mascaras

Mascaras que no son multiplo de 8

Las máscaras no siempre son de 8, 16 o 24 bits. En estos casos la separación de la parte red y la parte host no es tan evidente, aunque el mecanismo es el mismo:



Posibles valores de mascara

En las máscaras los bits a 1 siempre han de estar contiguos empezando por la izquierda. No se utiliza por ejemplo la máscara 255.255.0.255

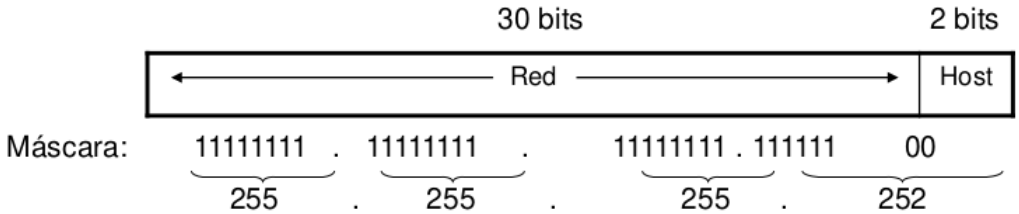
Bits de máscara (n)	Binario	Decimal
0	00000000	**0**
1	10000000	0 + 128 = **128**
2	11000000	128 + 64 = **192**
3	11100000	192 + 32 = **224**
4	11110000	224 + 16 = **240**
5	11111000	240 + 8 = **248**
6	11111100	248 + 4 = **252**
7	11111110	252 + 2 = **254**
8	11111111	254 + 1 = **255**

Notacion concisa de mascaras

- en lugar de expresarla con números decimales se puede indicar su longitud en bits (entre 0 y 32).
- La interfaz “40.40.0.1 255.255.255.0” se convierte en “40.40.0.1/24”
- La ruta “A 20.0.0.0 255.0.0.0 por 90.0.0.2” se convierte en “A 20.0.0.0/8 por 90.0.0.2”

Mini redes

La red más pequeña que podemos hacer es la de máscara de 30 bits:



En este caso obtenemos cuatro direcciones, de las cuales solo podemos usar dos. Estas redes se suelen utilizar en enlaces punto a punto ya que en este caso solo se necesitan dos direcciones.

Ruta por defecto

- En muchos casos al indicar las rutas en un router hay muchas que son accesibles por la misma dirección, y no es cómodo especificarlas una a una.
- El uso de ruta por defecto **no es obligatorio.**

- Para esto se puede utilizar la llamada “ruta por defecto” que se le aplica al paquete cuando no se le aplica ninguna de las otras rutas definidas.
- Un caso típico es cuando un router conecta una o varias redes entre sí y hay una única salida a Internet.
- La ruta por defecto tiene la sintaxis:

  A 0.0.0.0 0.0.0.0 por <dirección del router por defecto>

  Por ejemplo si el router por defecto es 20.0.0.1: A 0.0.0.0 0.0.0.0 por 20.0.0.1

  O en notación concisa: A 0.0.0.0/0 por 20.0.0.1

Mascaras de tamaño variable

- A menudo interesa dividir una red en **subredes de diferentes tamaños.**
- Para esto se utilizan **máscaras de tamaño variable**, es decir la división red/host no es igual en todas las subredes.
- Aunque las subredes pueden tener diferente tamaño **no pueden solaparse** (existirían direcciones duplicadas).
- La visión que tenemos de las subredes puede variar. Por ejemplo lo que en un sitio de la red se ve como una subred /22 (1024 direcciones) puede dividirse en varias /24 (256 direcciones) cuando nos acercamos.

Orden de forwarding o enrutamiento

- Es posible que **haya varias rutas válidas** para un **mismo paquete**. Por ejemplo la ruta por defecto es aplicable en principio a cualquier paquete.
- Se revisan primero las rutas de máscara más larga. Este criterio garantiza que se aplicarán **primero** las rutas **más específicas** y **luego** las **más generales**. Así por ejemplo las rutas host (/32) se aplican en primer lugar y la ruta por defecto (/0) se aplican en último lugar.
- Ejemplo:
  - Un router recibe un datagrama con destino 200.40.1.1
  - La búsqueda en la tabla encuentra dos entradas:
    - 200.40.1.0/24
    - 200.40.0.0/16
  - La ruta que se usará es la 200.40.1.0/24