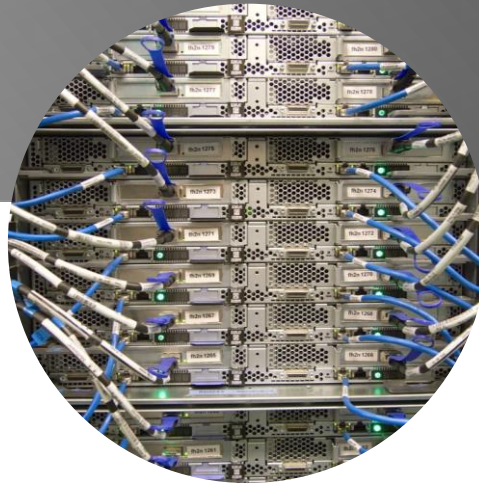


Redes de computadores 2022 -1 (11310052)

David Felipe Celeita Rodriguez



Universidad del
Rosario

Escuela de Ingeniería,
Ciencia y Tecnología

Probability is expectation founded upon partial knowledge. A perfect acquaintance with all the circumstances affecting the occurrence of an event would change expectation into certainty, and leave nether room nor demand for a theory of probabilities.

George Boole



Programa

Fecha (Sesión)	Tema
Sesión 1-2 24 Ene – 28 Ene	Introducción a redes de computadores Parte 1
Sesión 3-4 31 Ene – 4 Feb	Introducción a redes de computadores Parte 2
Sesión 5-6 7 Feb – 11 Feb	Capa de aplicación Parte 1
Sesión 7-8 14 Feb – 18 Feb	Capa de aplicación Parte 2
Sesión 9-10 21 Feb – 25 Feb	Capa de transporte Parte 1
Sesión 10 21 Feb – 25 Feb	PARCIAL 1



Programa

Fecha (Sesión)	Tema
Sesión 11-12 28 Feb – 4 Mar	Capa de transporte Parte 2
Sesión 13-14 7 Mar – 11 Mar	Capa de red Parte 1 (Plano de datos)
Sesión 15-16 14 Mar – 18 Mar	Capa de red Parte 2 (Plano de datos)
Sesión 17-20 21 Mar – 25 Mar	Capa de red Parte 3 (Plano de control)
Sesión 17-20 28 Mar – 1 Abr	Capa de red Parte 4 (Plano de control)
Sesión 20 28 Mar – 1 Abr	PARCIAL 2



Capítulo 4: Capa de red

Capa de red: descripción general

- plano de datos

- plano de control

¿Qué hay dentro de un enrutador?

- puertos de entrada, conmutación, puertos de salida

- gestión de búfer, scheduling

IP: el Protocolo de Internet

- formato de datagrama

- Direccionamiento

- Traducción de Direcciones de RedIPv6

Reenvío generalizado, SDN

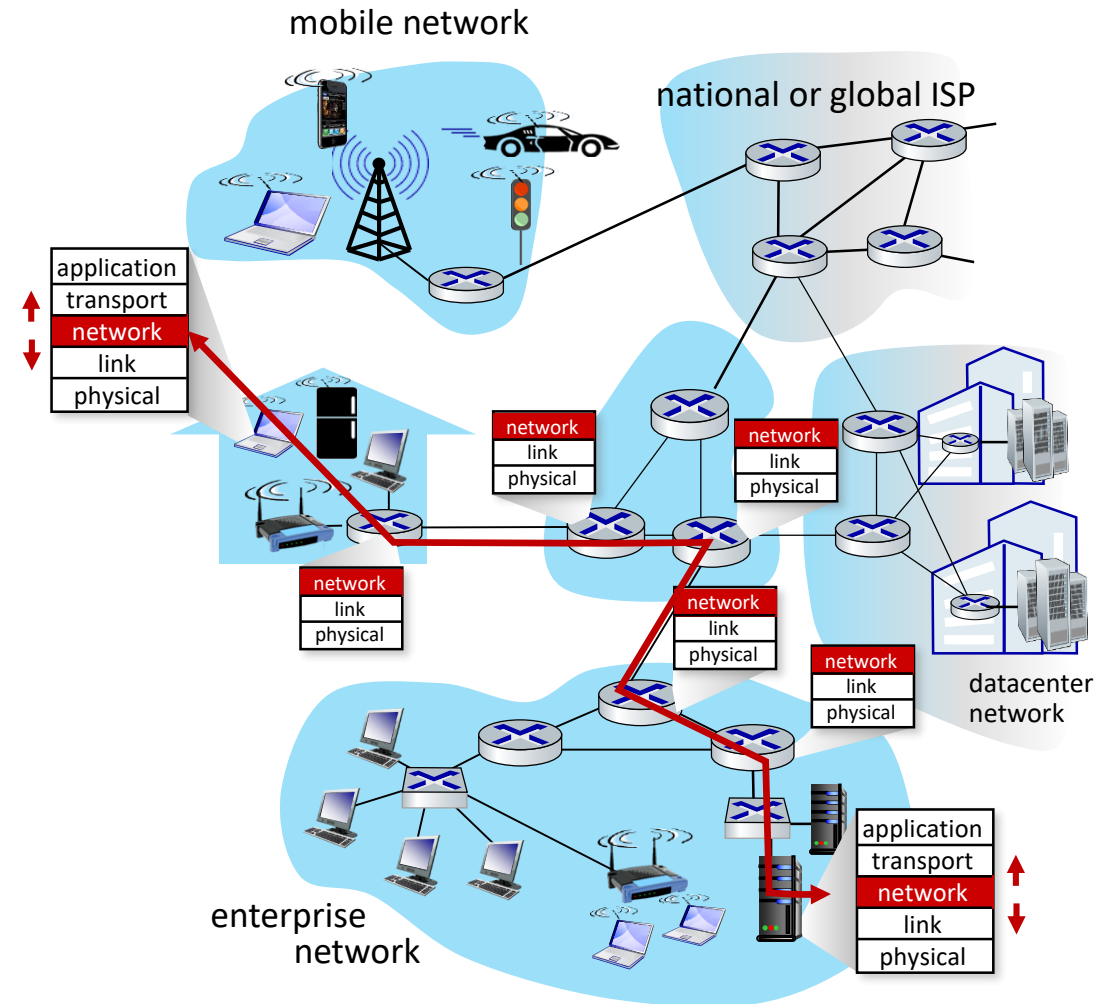
- Match+action

- OpenFlow: match+action en operación

Middleboxes

Servicios y protocolos de la capa de red

- Transportar el segmento entre hosts desde remitente a receptor
 - **sender:** encapsula segmentos en datagramas, pasa a la capa de enlace
 - **receiver:** entrega segmentos al protocolo de capa de transporte
- protocolos de capa de red en *todos los dispositivos de Internet*: hosts, routers
- **Routers(enrutadores):**
 - examina los campos de encabezado en todos los datagramas IP que lo atraviesan
 - mueve datagramas de los puertos de entrada a los puertos de salida para transferir datagramas (cifrados) a lo largo de la ruta de extremo a extremo



Capa de red

Funciones principales:

- *Forwarding (re-enviar)*: mover paquetes desde el enlace de entrada de un enrutador al enlace de salida del enrutador apropiado
- *Routing (enrutamiento)*: determinar la ruta tomada por los paquetes desde el origen al destino
 - algoritmos de enrutamiento

analogía: roadtrip

- *Forwarding (re-enviar)*: proceso de pasar por un solo intercambio de autopista
- *Routing (enrutamiento)*: proceso de planificación del viaje desde el origen hasta el destino



forwarding



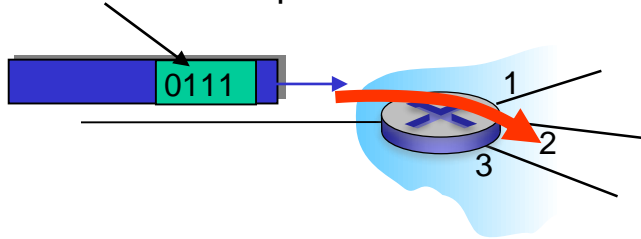
routing

Planos de la capa de red

Plano de datos:

- *local*, función por cada router
- Determina ¿cómo el datagrama que llega al puerto de entrada del enrutador se reenvía al puerto de salida del enrutador?

Valores llegando
Encabezado de pkt

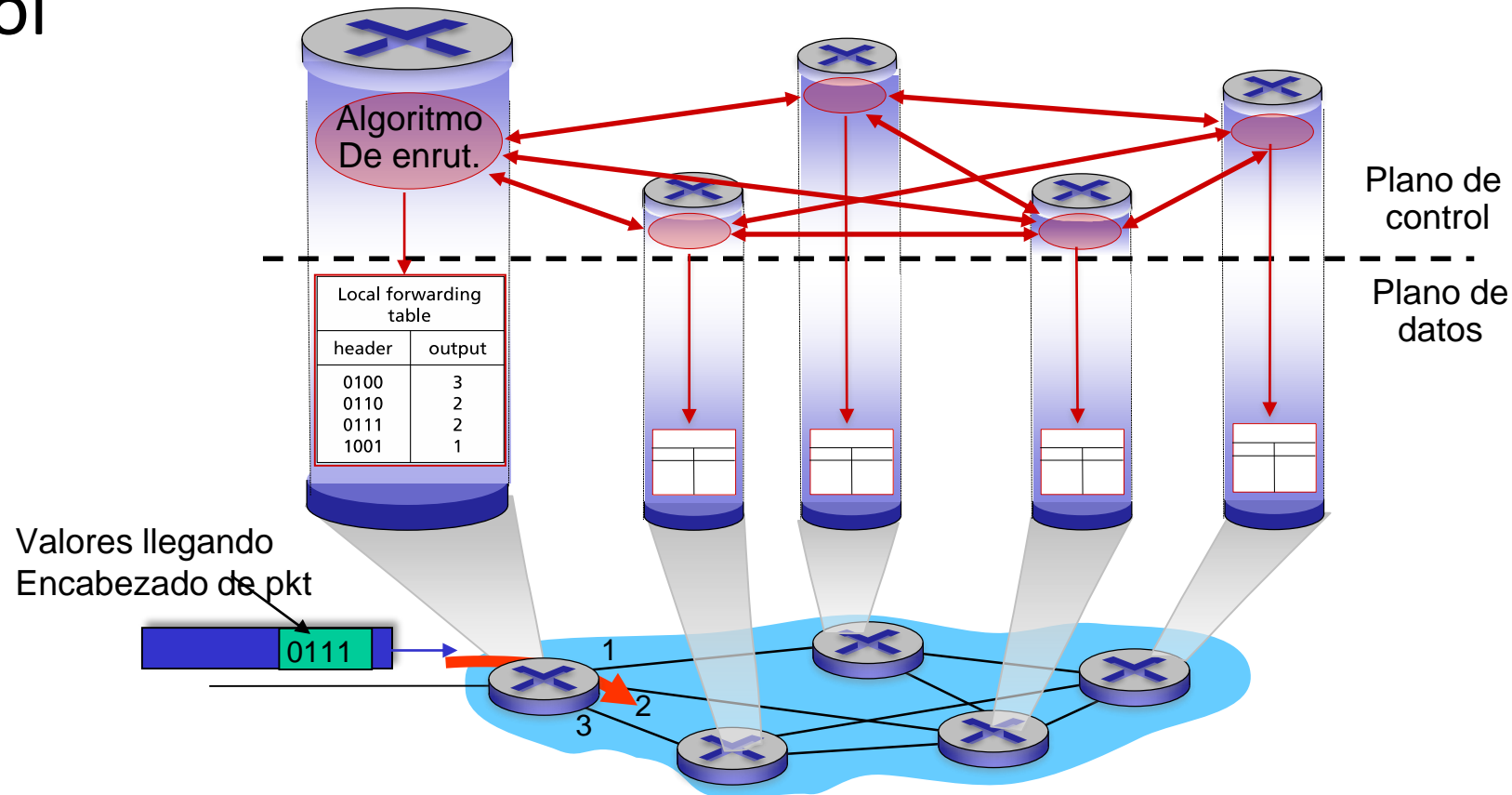


Plano de control

- Lógica de toda la *red amplia*
- Determina ¿cómo se enruta el datagrama entre los enrutadores a lo largo de la ruta de extremo a extremo desde el host de origen al host de destino?
- Dos metodologías en el plano de control:
 - *Algoritmos de enrutamiento tradicional*: implementada en los routers
 - *software-defined networking (SDN)*: implementada (remotamente) en servidores

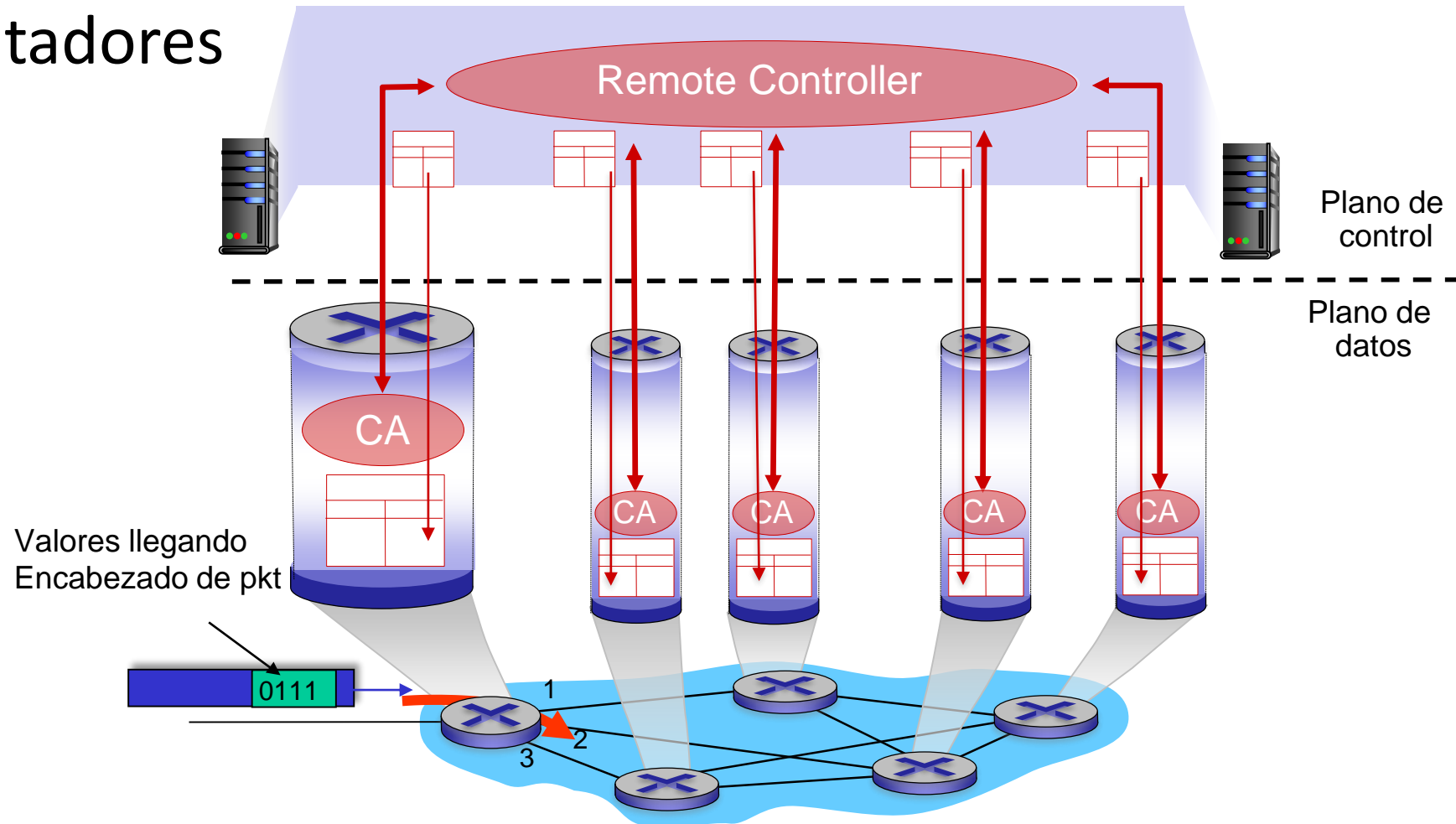
Plano de control por cada router

Los componentes individuales del algoritmo de enrutamiento *en todos y cada uno de los enrutadores* interactúan en el plano de control



Software-Defined Networking (SDN) control plane

El controlador remoto calcula e instala tablas de reenvío en enrutadores



Modelo de servicios – Capa de red

¿Qué *modelo de servicio* se usa para el "canal" que transporta datagramas del remitente al receptor?

Ejemplo de servicio para datagramas individuales:

- Entrega garantizada
- Tiempo de entrega garantizado con menos de 40 mseg de retraso

Ejemplo de Servicio para un "flujo" de datagramas

- Entrega de datagramas en orden
- Ancho de banda mínimo garantizado para fluir
- Restricciones sobre cambios en el espaciado entre paquetes

Modelo de servicios – Capa de red

Arquitectura De red	Modelo de Servicio	¿Quality of Service (QoS) garantizado?			
		Bandwidth	Loss	Order	Timing
Internet	“mejor esfuerzo”	ninguna	no	no	no

Modelo de Servicio “best-effort” en Internet

No garantiza:

- i. entrega exitosa de datagramas a destino
- ii. tiempo o orden de entrega
- iii. ancho de banda disponible para el flujo de extremo a extremo



Modelo de servicios – Capa de red

Arquitectura De red	Modelo de Servicio	¿Quality of Service (QoS) garantizado?				
		Bandwidth	Loss	Order	Timing	
Internet	best-effort	ninguna	no	no	no	Sin indicador de congestión
ATM	Constant Bit Rate	Tasa constante	Si	Si	Si	Sin congestión
Asynchronous Transfer Mode						
ATM	Available Bit Rate	Min. garantizado	no	Si	no	Con indicador de congestión
Internet	Intserv Guaranteed (RFC 1633)	Si	Si	Si	Si	
Internet	Diffserv (RFC 2475)	Posible	Posible	Posible	no	

Sin embargo el modelo de servicio “best-effort” se caracteriza por:

- la **simplicidad del mecanismo** ha permitido que Internet sea implementado y adoptado ampliamente
- el **aprovisionamiento suficiente de ancho de banda** permite que el rendimiento de las aplicaciones en tiempo real (p. ej., voz interactiva, video) sea "suficientemente bueno" para "la mayor parte del tiempo"
- Los **servicios distribuidos de capa de aplicación replicados** (centros de datos, redes de distribución de contenido) que se conectan cerca de las redes de los clientes permiten que los servicios se brinden desde múltiples ubicaciones.
- El control de la congestión de los servicios "elásticos" ayuda
Puede surgir un buen debate acerca de la efectividad de este modelo de servicio



Capítulo 4: Capa de red

Capa de red: descripción general

plano de datos

plano de control

¿Qué hay dentro de un enrutador?

puertos de entrada, conmutación, puertos de salida

gestión de búfer, scheduling

IP: el Protocolo de Internet

formato de datagrama

Direccionamiento

Traducción de Direcciones de RedIPv6

Reenvío generalizado, SDN

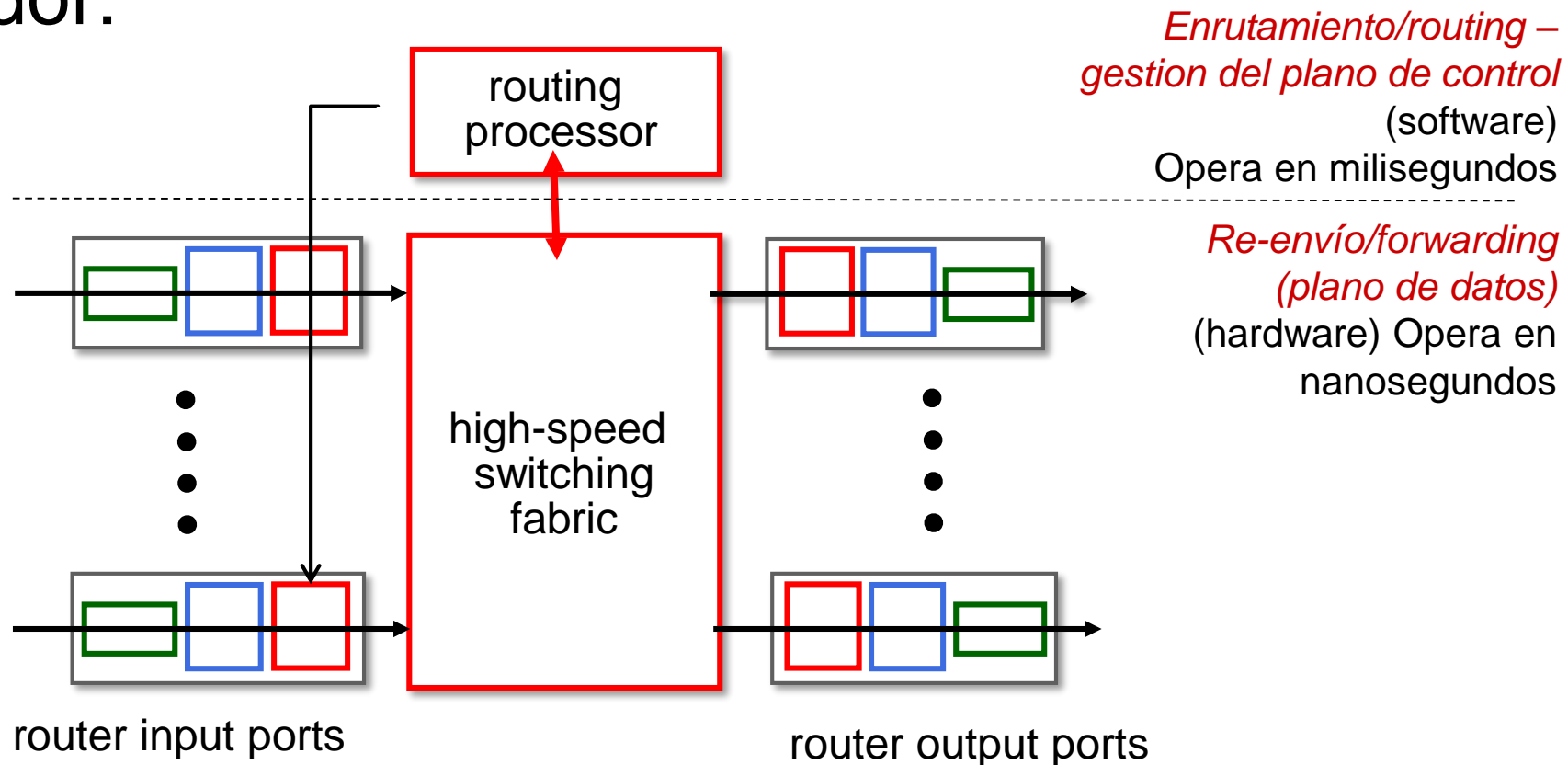
Match+action

OpenFlow: match+action en operación

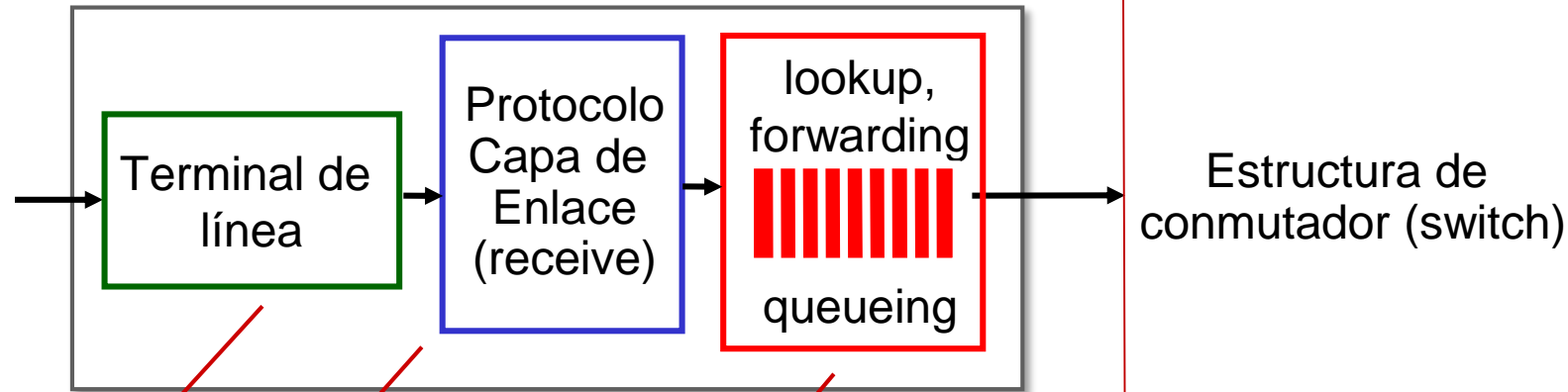
Middleboxes

Arquitectura de un enrutador

vista de alto nivel de la arquitectura genérica del enrutador:



Funciones de puertos de entrada



Capa física:

Recepción de bits

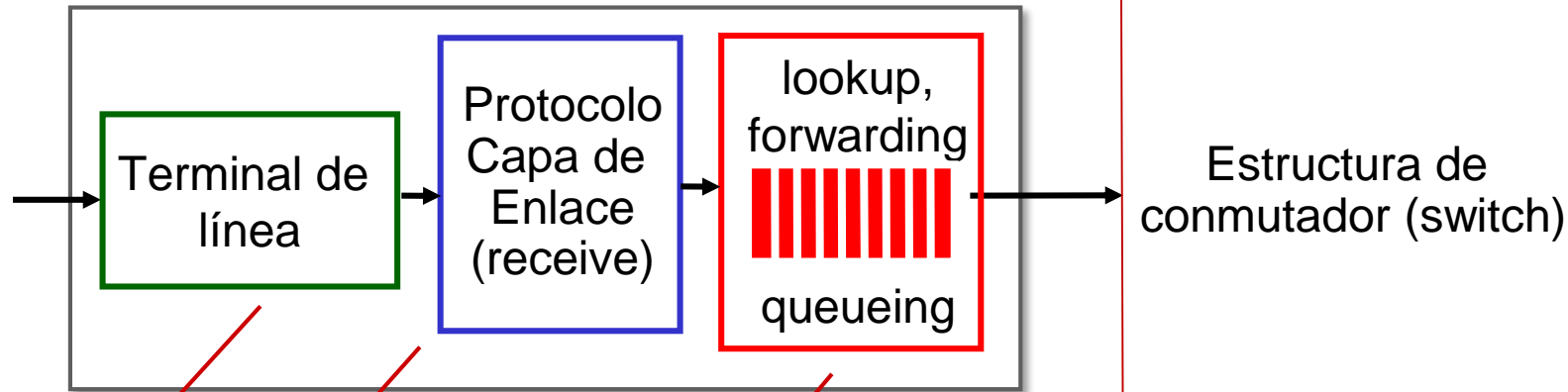
Capa de enlace:

e.g., Ethernet
(Siguiendo módulo)

switching descentralizado :

- usando valores de campo de encabezado, puerto de salida de búsqueda (lookup) usando la tabla de reenvío en la memoria del puerto de entrada ("*match plus action*")
- **Objetivo:** procesamiento completo del puerto de entrada a "velocidad de línea"
- **Input port queueing** (cola del puerto de entrada): si los datagramas llegan más rápido que la velocidad de reenvío a la estructura del conmutador

Funciones de puertos de entrada



Capa física:

Recepción de bits

Capa de enlace:

e.g., Ethernet
(Siguiendo módulo)

switching descentralizado :

- usando valores de campo de encabezado, puerto de salida de búsqueda (lookup) usando la tabla de reenvío en la memoria del puerto de entrada ("*match plus action*")
- **Reenvío basado en destino:** reenvío basado solo en la dirección IP de destino (tradicional)
- **Reenvío generalizado:** reenvío basado en cualquier conjunto de valores de campo de encabezado

Destination-based forwarding

<i>forwarding table</i>	
Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010000 00000100	n 3
11001000 00010111 00010000 00000111	
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

¿Qué pasa cuando no se pueden dividir las direcciones muy bien?

Longest prefix matching

longest prefix match

cuando busque la entrada de la tabla de reenvío para la dirección de destino dada, use el prefijo de dirección **más largo** que coincida con la dirección de destino.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

Ejemplos:

11001000 00010111 00010110 10100001 Cuál interfaz?

11001000 00010111 00011000 10101010 Cuál interfaz?

Longest prefix matching

longest prefix match

cuando busque la entrada de la tabla de reenvío para la dirección de destino dada, use el prefijo de dirección **más largo** que coincida con la dirección de destino.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 match! 1 00011*** *****	2
otherwise	3

Ejemplos:

11001000 00010111 00010110	10100001	Cuál interfaz?
11001000 00010111 00011000	10101010	Cuál interfaz?

Longest prefix matching

longest prefix match

cuando busque la entrada de la tabla de reenvío para la dirección de destino dada, use el prefijo de dirección **más largo** que coincida con la dirección de destino.

Destination Address Range				Link interface
11001000	00010111	00010***	*****	0
11001000	00010111	00011000	*****	1
11001000	00010111	00011***	*****	2
otherwise				3

match!

Ejemplos:

11001000	00010111	00010110	10100001	Cuál interfaz?
11001000	00010111	00011000	10101010	Cuál interfaz?

Longest prefix matching

longest prefix match

cuando busque la entrada de la tabla de reenvío para la dirección de destino dada, use el prefijo de dirección **más largo** que coincida con la dirección de destino.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

match!

ejemplos:

11001000 00010111 00010110 10100001

Cuál interfaz?

11001000 00010111 00011000 10101010

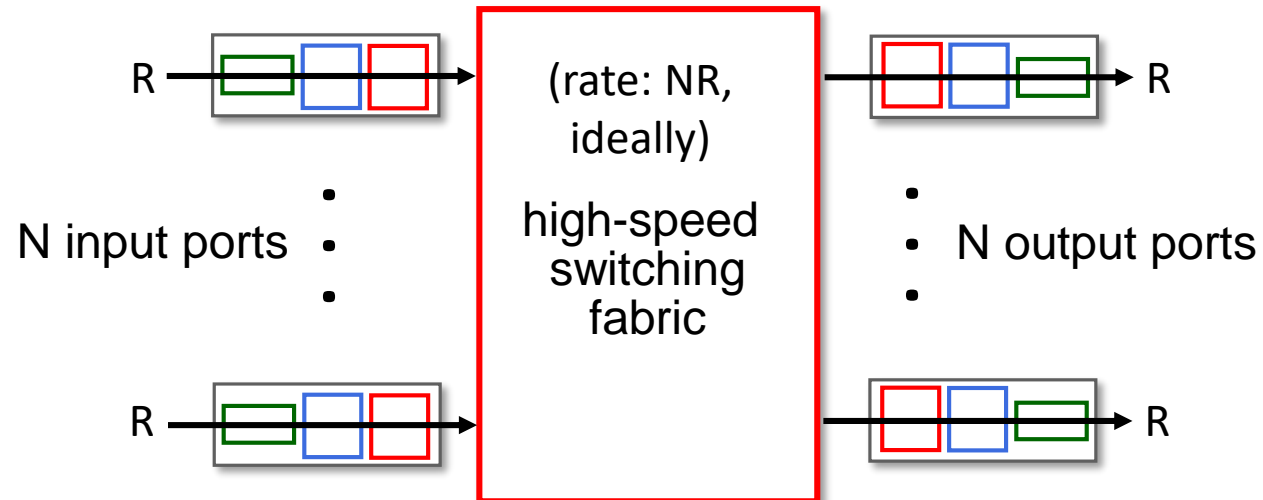
Cuál interfaz?

Longest prefix matching

- veremos por qué Longest prefix matching se usa comunmente, cuando estudiemos el direccionamiento
- Longest prefix matching: a menudo se realiza utilizando memorias direccionables de contenido ternario (TCAM - Ternary Content Addressable Memories)
 - **contenido direccionable:** presenta la dirección a TCAM: recupera la dirección en un ciclo de reloj, independientemente del tamaño de la tabla
 - Cisco Catalyst (fabricante): ~ 1 millón de entradas de la tabla de enrutamiento en TCAM

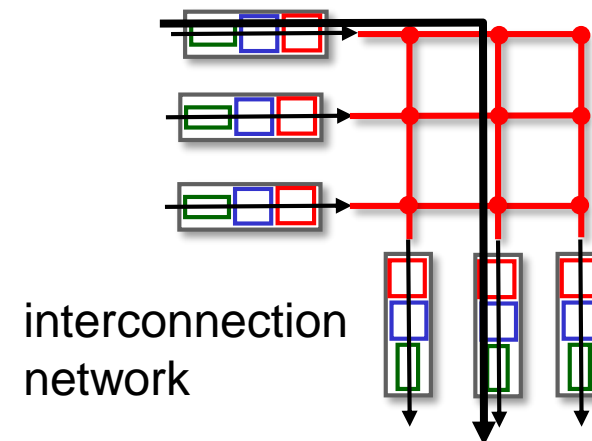
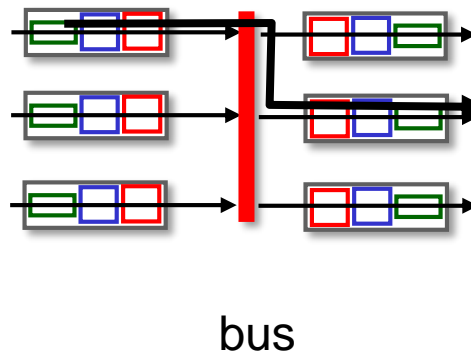
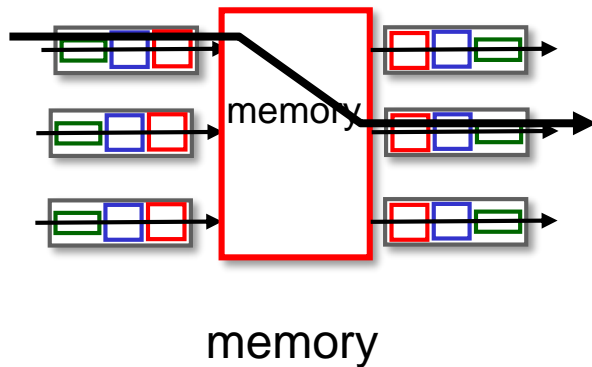
Estructura de conmutador

- transferir el paquete desde el enlace de entrada al enlace de salida apropiado
- **switching rate:** velocidad a la que los paquetes se pueden transferir de las entradas a las salidas
 - a menudo medido como múltiplo de la tasa de línea de entrada / salida
 - N entradas: tasa de conmutación N veces la tasa de línea deseable



Estructura de conmutador

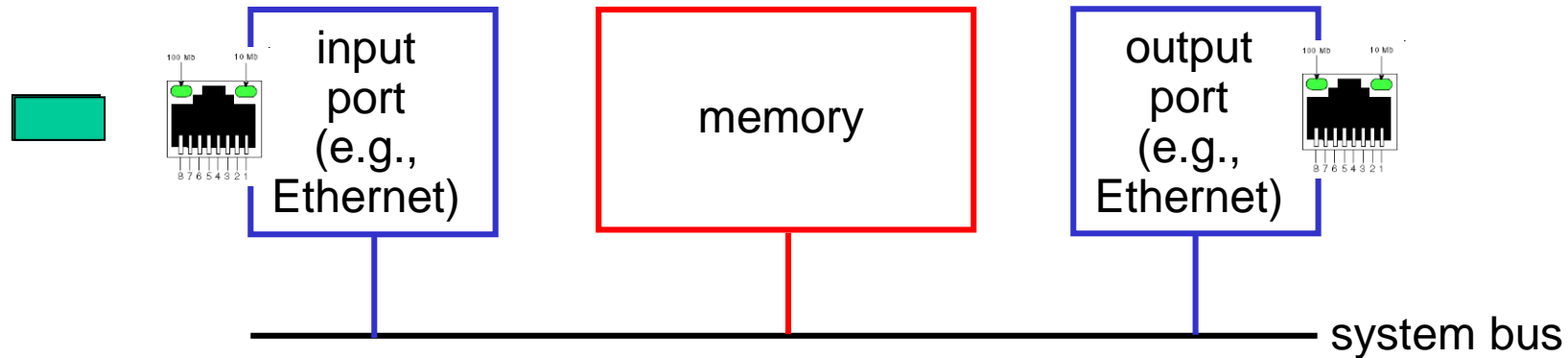
- transferir el paquete desde el enlace de entrada al enlace de salida apropiado
- **switching rate:** velocidad a la que los paquetes se pueden transferir de las entradas a las salidas
 - a menudo medido como múltiplo de la tasa de línea de entrada / salida
 - N entradas: tasa de conmutación N veces la tasa de línea deseable
- Tres categorías de fabricación:



Tipo 1: Switching via memory

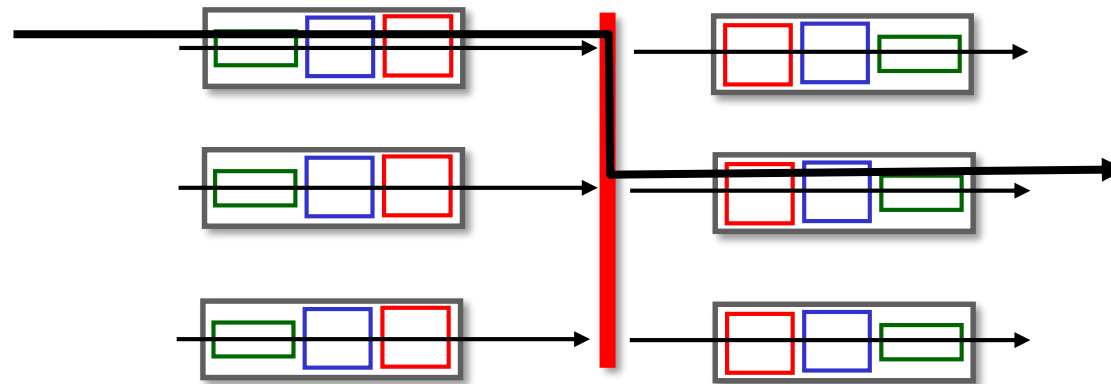
Primera generación de routers:

- computadoras tradicionales con conmutación bajo control directo de la CPU
- paquete copiado a la memoria del sistema
- velocidad limitada por el ancho de banda de la memoria (2 cruces de bus por datagrama)



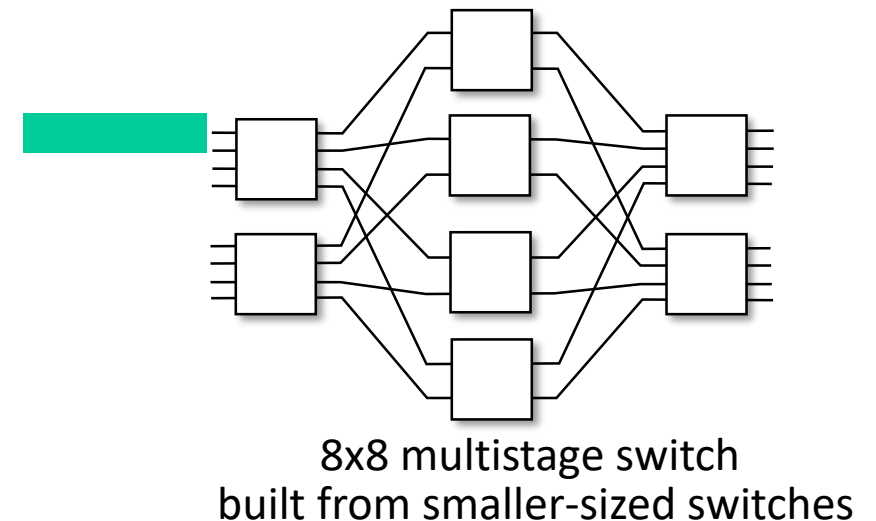
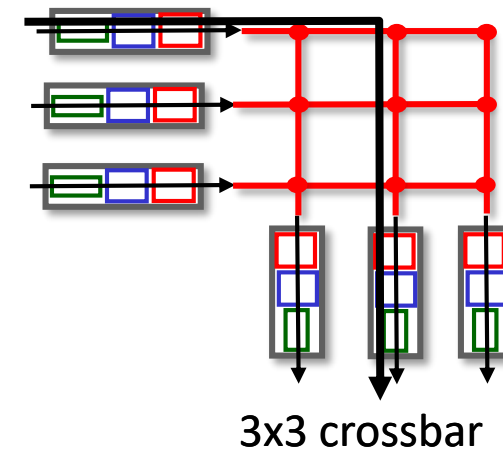
Tipo 2: Switching via a bus

- datagrama de la memoria del puerto de entrada a la memoria del puerto de salida a través de un bus compartido
- *contención del bus*: velocidad de conmutación limitada por el ancho de banda del bus
- Bus de 32 Gbps, Cisco 5600: velocidad suficiente para enrutadores de acceso



Tipo 3: Switching via interconnection network

- Crossbar, Clos Networks, otras redes de interconexión desarrolladas inicialmente para conectar procesadores en multiprocesador
- **multistage switch:** *conmutación $n \times n$* de múltiples etapas de interruptores más pequeños
- **Aprovecha el paralelismo:**
 - fragmentar el datagrama en celdas de longitud fija al ingresar
 - cambiar celdas a través del conmutador, reensamblar el datagrama en la salida

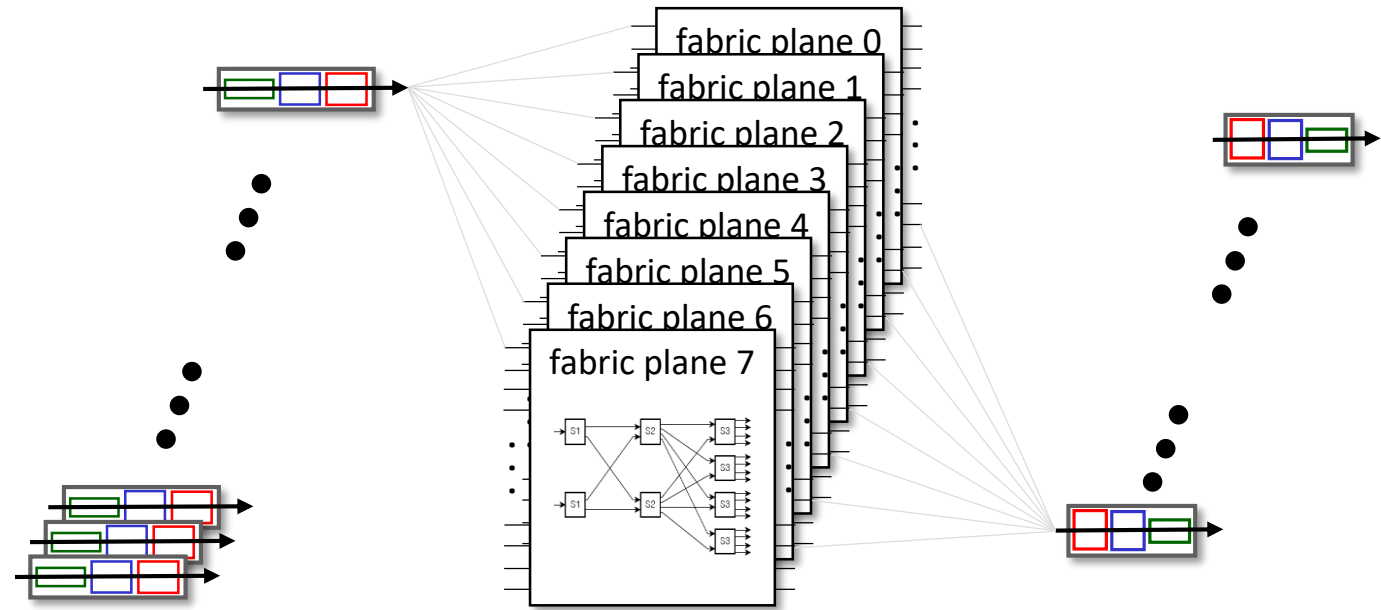


Tipo 3: Switching via interconnection network

- escalamiento, utilizando múltiples "planos" de conmutación en paralelo (aceleración, escalamiento mediante paralelismo)

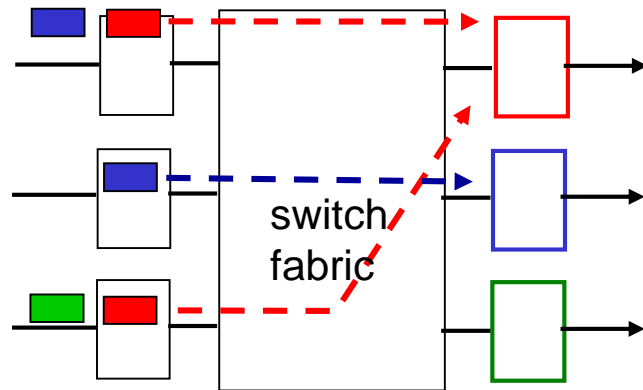
- **Ejemplo:** Router Cisco CRS

- Unidad básica: 8 planos de conmutación
- Cada plano: 3-stage interconnection network
- Capacidad de conmutación hasta 100's Tbps

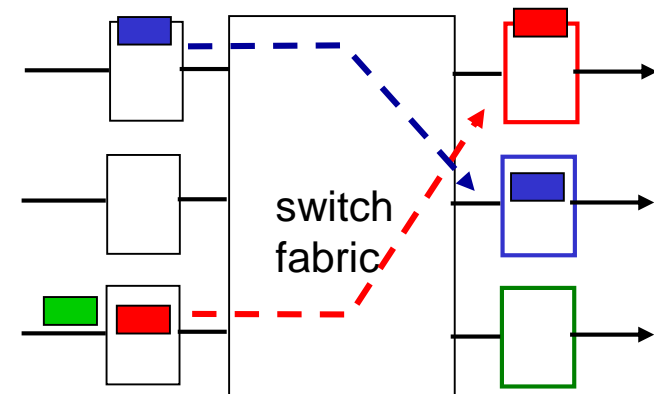


Colas en el puerto de entrada (queueing)

- Si la estructura de conmutación es más lenta que los puertos de entrada combinados -> pueden producirse colas en las colas de entrada
 - queueing delay + loss: retrasos en la cola y pérdidas debido al desbordamiento del búfer de entrada (**input buffer overflow**)
- **Head-of-the-Line (HOL) blocking:** El datagrama en cola al frente de la cola evita que otros en la cola avancen

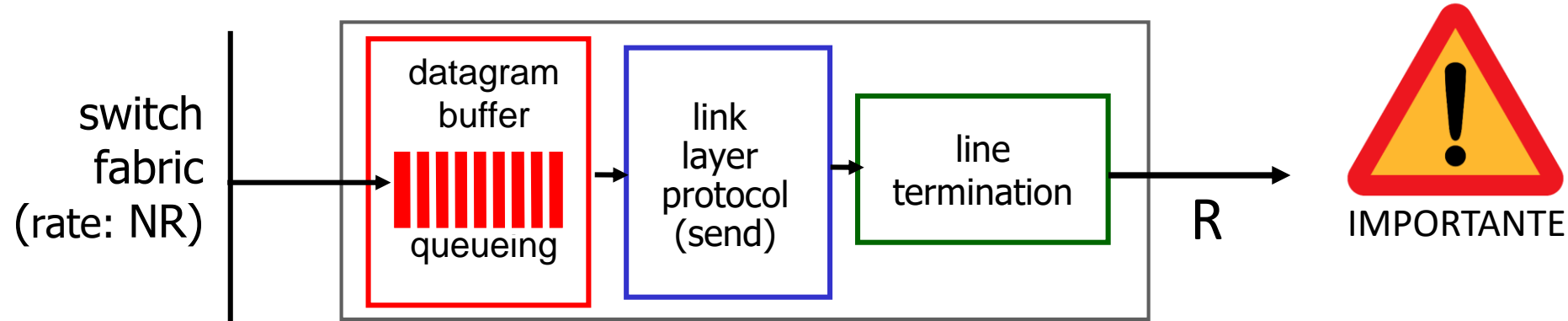


Contención del puerto de salida: solo se puede transferir un datagrama rojo. El paquete rojo inferior está *bloqueado*



un paquete más tarde: el paquete verde experimenta bloqueo HOL

Colas en el puerto de salida (queueing)

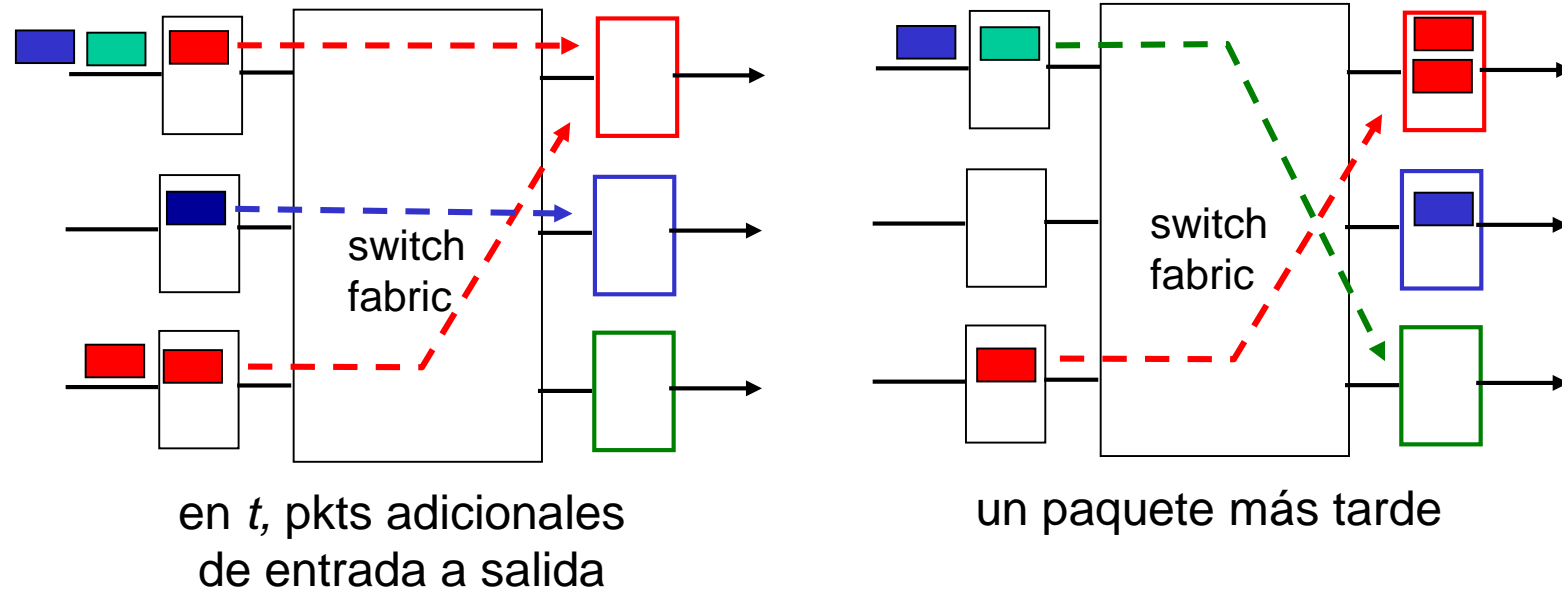


- **Buffering** necesario cuando los datagramas llegan de la estructura más rápido que la velocidad de transmisión del enlace.
- **Drop policy:** ¿Qué datagramas eliminar si no hay búfer disponible?
- **Scheduling discipline** elige entre los datagramas en cola para la transmisión

Los datagramas se pueden perder debido a la congestión, la falta de búferes

Programación prioritaria: obtiene el mejor rendimiento-> neutralidad de la red

Colas en el puerto de salida (queueing)



- almacenamiento en búfer cuando la tasa de llegada a través del interruptor excede la velocidad de la línea de salida
- *queueing delay + loss: retrasos en la cola y pérdidas debido al desbordamiento del búfer de salida (output buffer overflow)*

¿Cuánto se debe almacenar en el buffer?

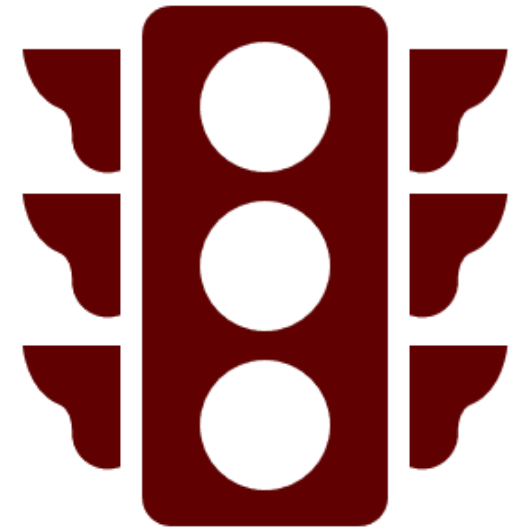
- RFC 3439 regla general: almacenamiento en búfer promedio igual al RTT "típico" (digamos 250 mseg) multiplicado por la capacidad del enlace C
- p. ej., C = enlace de 10 Gbps: búfer de 2,5 Gbit
 - recomendación más reciente: con N flujos, buffering igual a

$$Buffering = \frac{RTT * C}{\sqrt{N}}$$

- pero demasiado almacenamiento en búfer puede aumentar los retrasos (especialmente en los enrutadores domésticos)
- RTTs excesivos: bajo rendimiento para aplicaciones en tiempo real, respuesta TCP lenta
- Tener en cuenta el **control de congestión basado en demoras (delay-based congestion control)**: "mantener el enlace de cuello de botella lo suficientemente lleno (ocupado) pero no muy completo"

Queuing (Generalidades)

- First-in-first-out (FIFO)
- Last-in-first-out (LIFO)
- Service in random order (SIRO)
- Shortest processing time first (SPT)
- Service according to priority (PR)



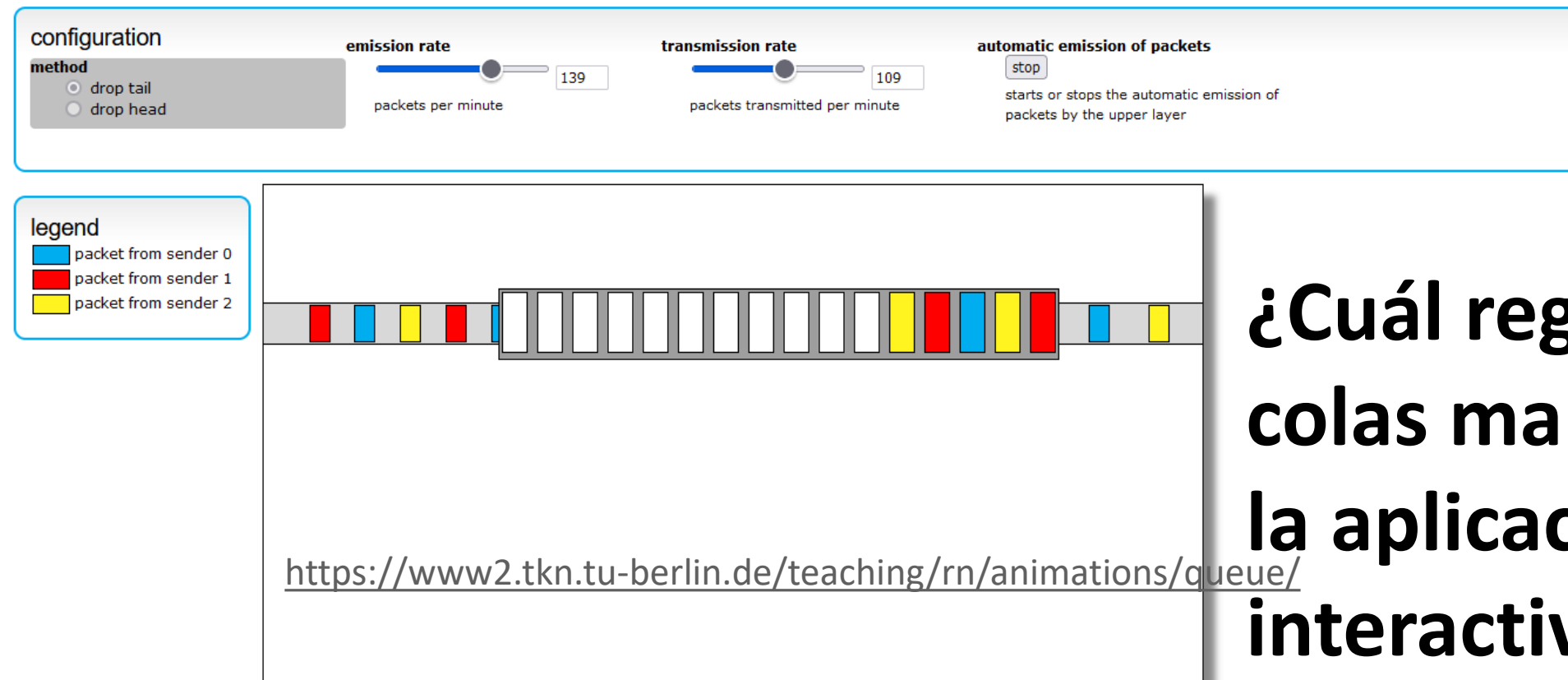
HANDS-ON (recordando)

Queuing and Loss

This animation illustrates queuing delay and packet loss.

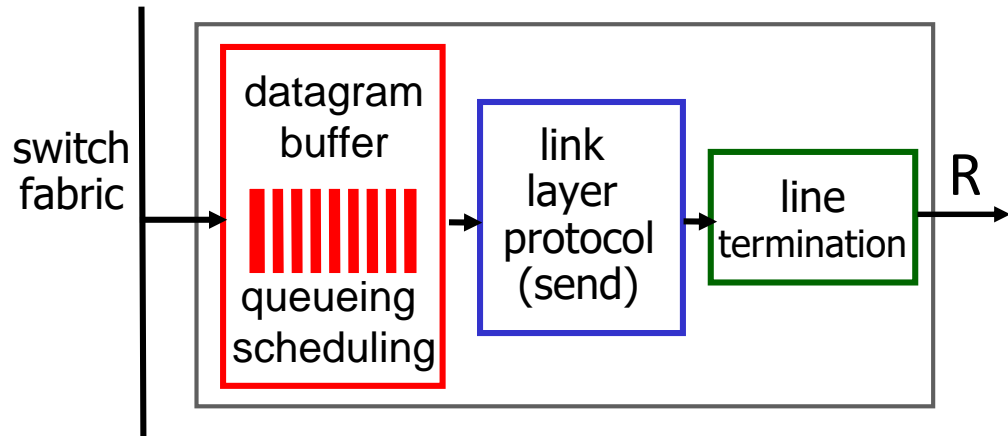
Three different senders - indicated by colors - send packets. The packets arrive and queue for service.

If the emission rate is higher than the transmission rate (both are slotted for better visualisation) a queue overflow will happen and according to the chosen method different packets will be dropped.

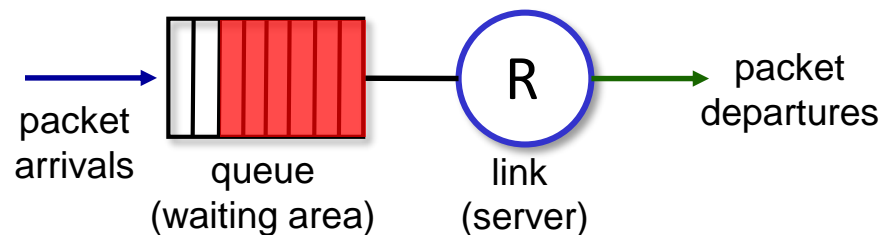


¿Cuál regla de colas maneja la aplicación interactiva?

Buffer Management



Abstraction: queue



buffer management:

- **drop**: qué paquete agregar/ descartar cuando los búferes estén llenos
- **tail drop**: descartar pkt de llegada
 - **priority**: descartar/remover basado en prioridad
- **marking**: qué paquetes marcar para señalar la congestión (ECN, RED)

Packet Scheduling: FCFS

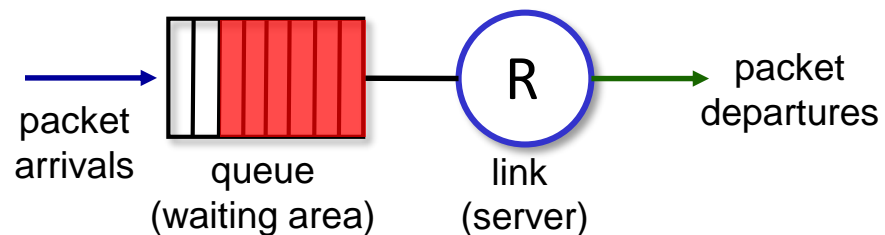
packet scheduling: Decide qué pkt es el siguiente en enviar al enlace

- first come, first served
- priority
- round robin
- weighted fair queueing

FCFS: paquetes transmitidos en orden de llegada al puerto de salida

- aka: First-in-first-out (FIFO)

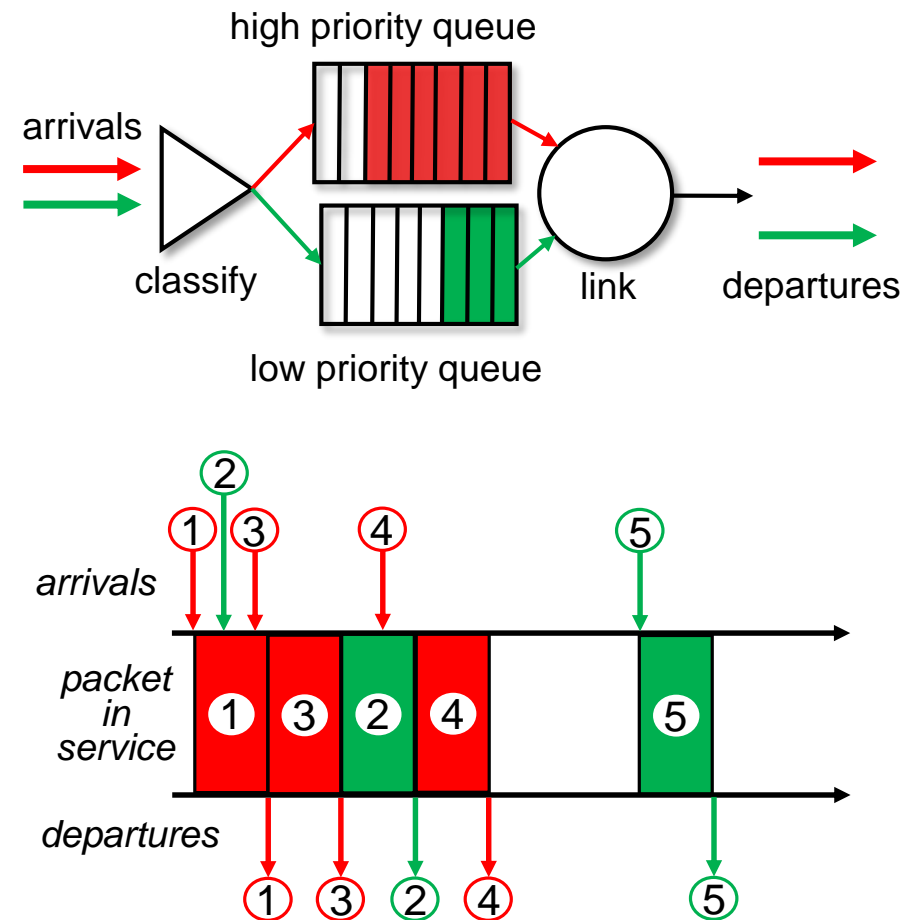
Abstraction: queue



Reglas de Scheduling: prioridad

Priority scheduling:

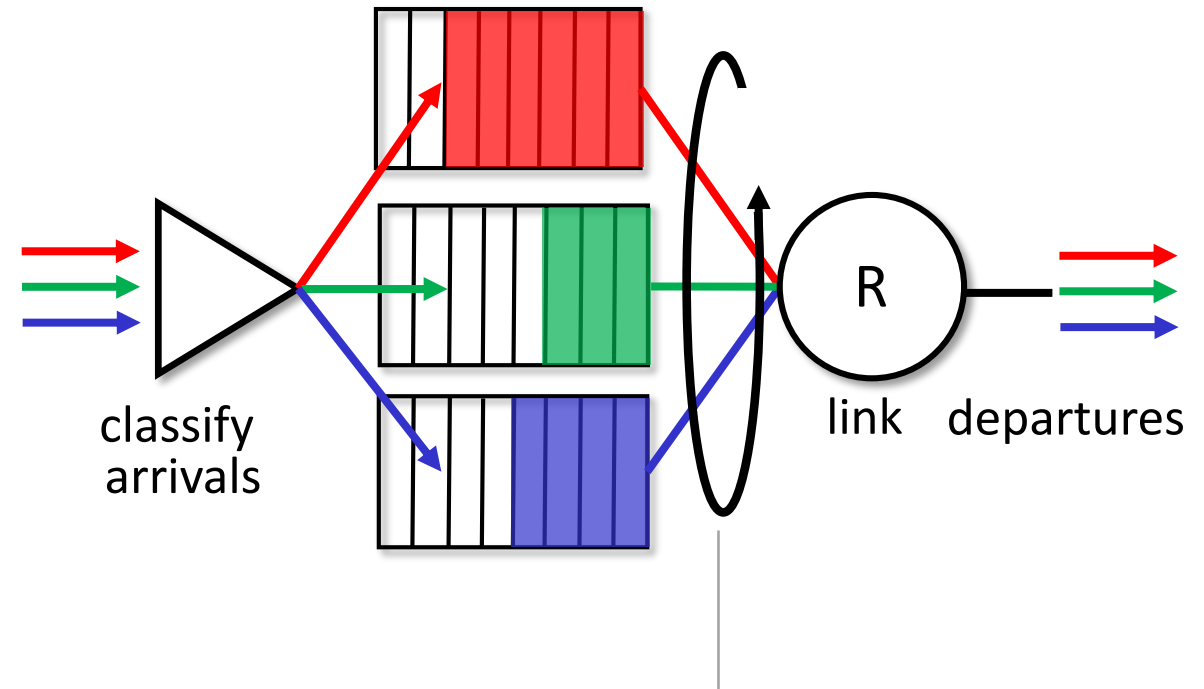
- tráfico de llegada clasificado, en cola por clase
- cualquier campo de encabezado se puede utilizar para la clasificación
- enviar paquetes desde la cola de mayor prioridad que tiene paquetes almacenados en búfer
- FCFS dentro de la clase de prioridad (First Come, First Served)



Reglas de Scheduling: round robin

Round Robin (RR) scheduling:

- tráfico de llegada clasificado, en cola por clase
- cualquier campo de encabezado se puede utilizar para la clasificación
- servidor cíclicamente, escanea repetidamente las colas de clases, enviando un paquete completo de cada clase (si está disponible) a su vez



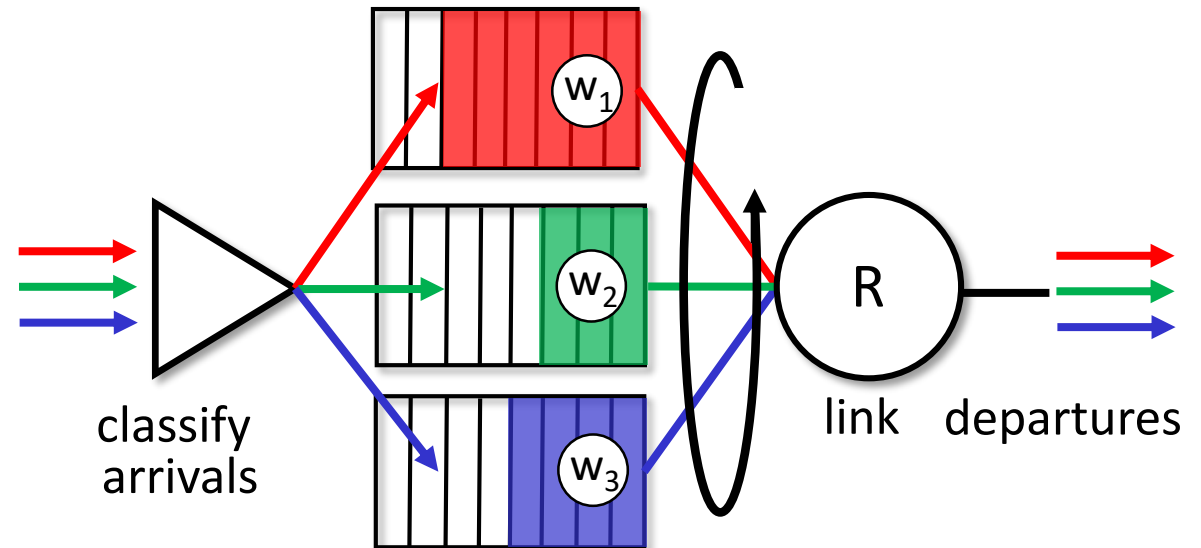
Reglas de Scheduling: weighted fair queueing

Weighted Fair Queuing (WFQ):

- Round Robin generalizado
- Cada clase, i , tiene un peso, w_i , y recibe una cantidad ponderada de servicio en cada ciclo:

$$\frac{w_i}{\sum_j w_j}$$

- garantiza un ancho de banda mínimo (por clase de tráfico)





HANDS-ON

- Proyecto Parte 1 – Poster de divulgación

PROYECTO DE CURSO – PARTE 1

- **Proyecto Parte 1: 4%**
- Proyecto parte 2: 16%
- **Tema:** Control de Congestión
- **Diseño de poster de divulgación**
para presentación de estado de arte
por grupos
- Describir generalidades del CCA
- Estudiar y describir estado del arte
de 10 referencias de journal
recientes (i.e. IEEE, ScienceDirect,
Taylor and Francis) del CCA asignado

Grupo 1	AIMD
Grupo 2	CWND
Grupo 3	SLOW START
Grupo 4	TCP TAHOE
Grupo 5	TCP RENO
Grupo 6	TCP Vegas
Grupo 7	TCP Cubic
Grupo 8	Agile-TCP
Grupo 9	Elastic-TCP
Grupo 10	ECN





Capítulo 4: Capa de red

Capa de red: descripción general

plano de datos

plano de control

¿Qué hay dentro de un enrutador?

puertos de entrada, conmutación, puertos de salida

gestión de búfer, scheduling

IP: el Protocolo de Internet

formato de datagrama

Direccionamiento

Traducción de Direcciones de RedIPv6

Reenvío generalizado, SDN

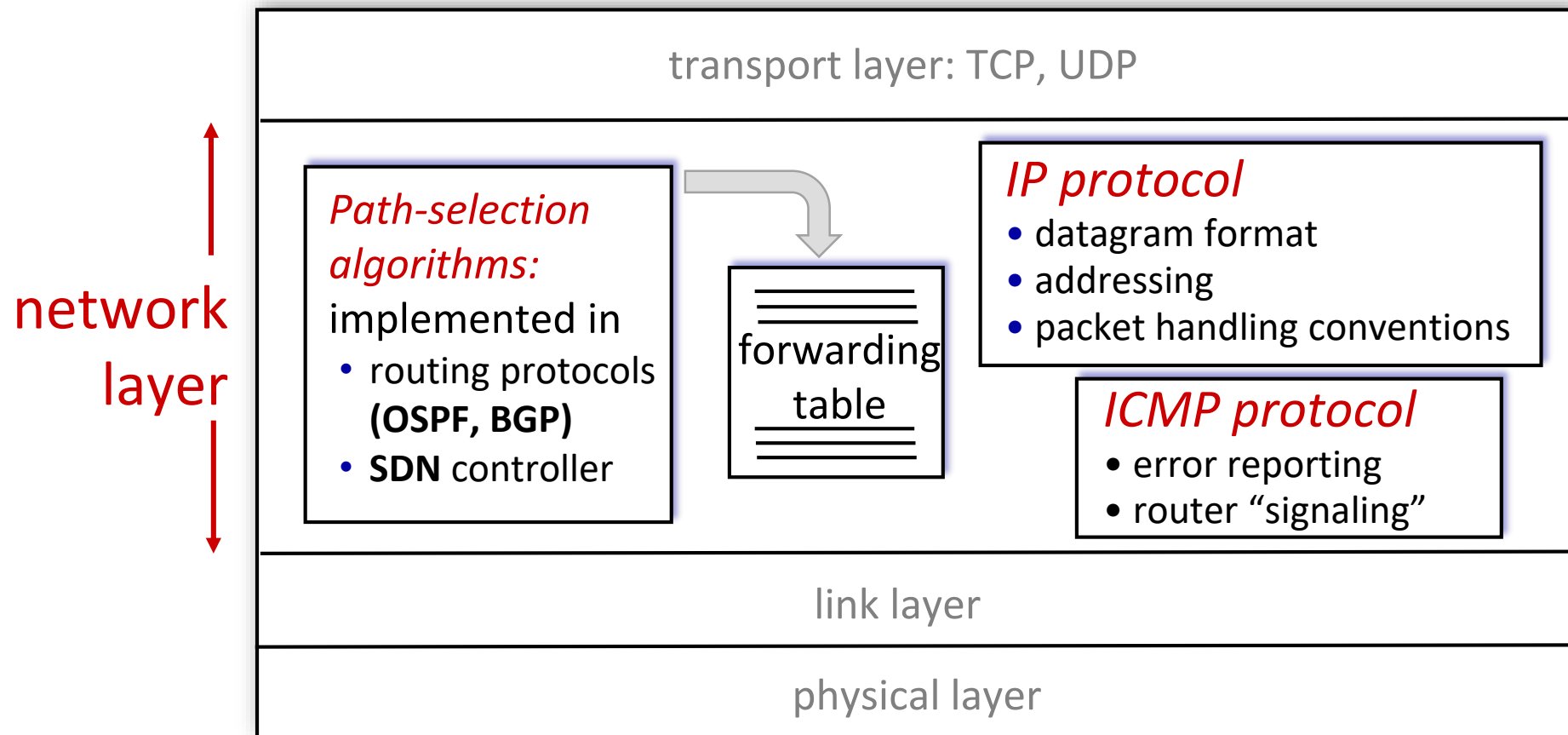
Match+action

OpenFlow: match+action en operación

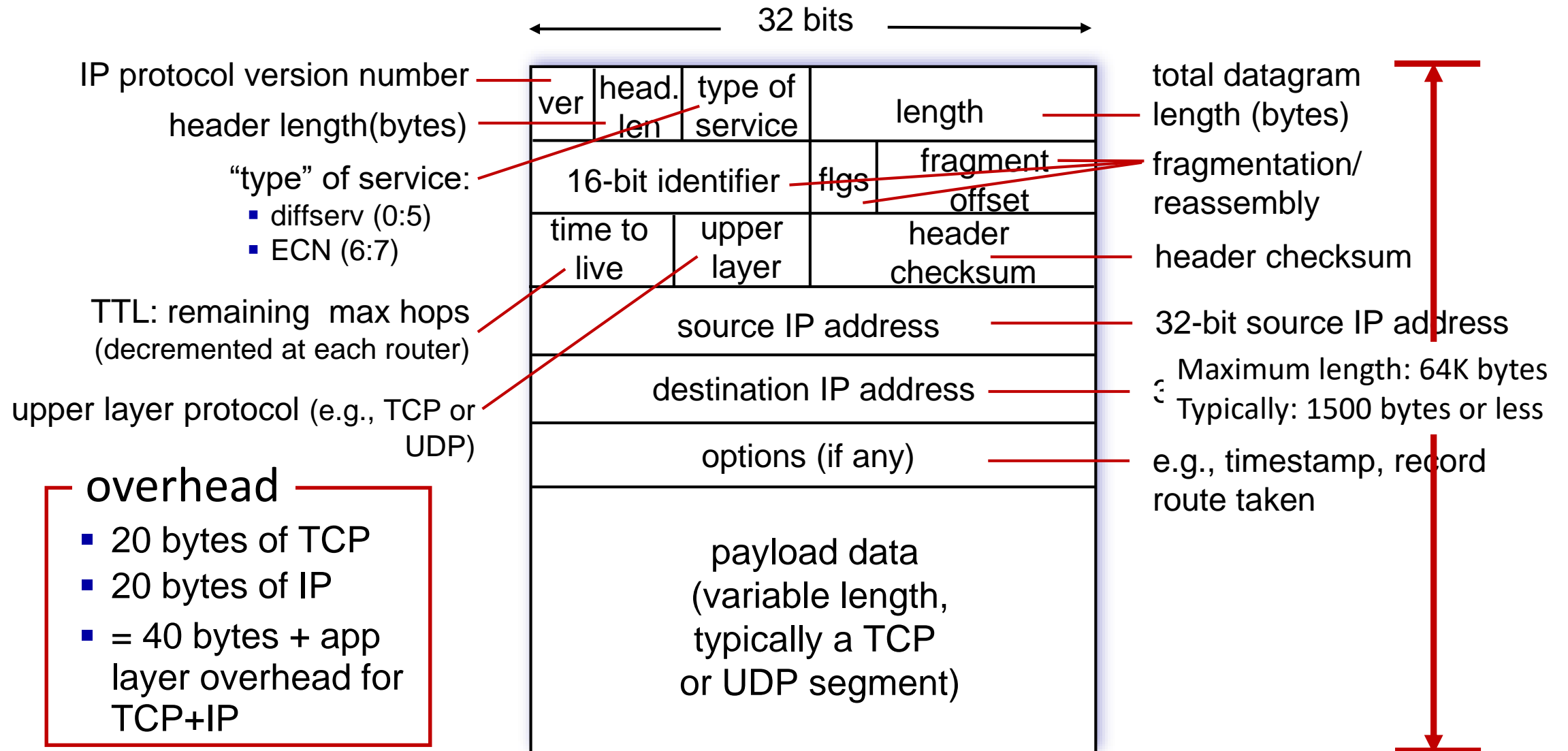
Middleboxes

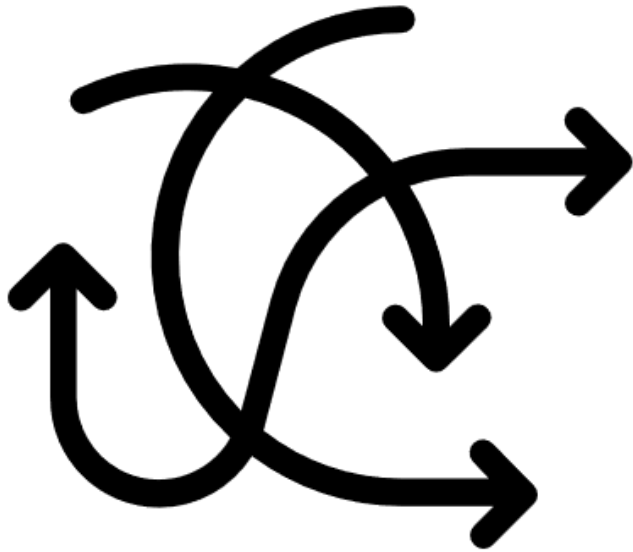
Capa de red: Internet

Host y router: funciones en la capa de red



IP Datagram (format)



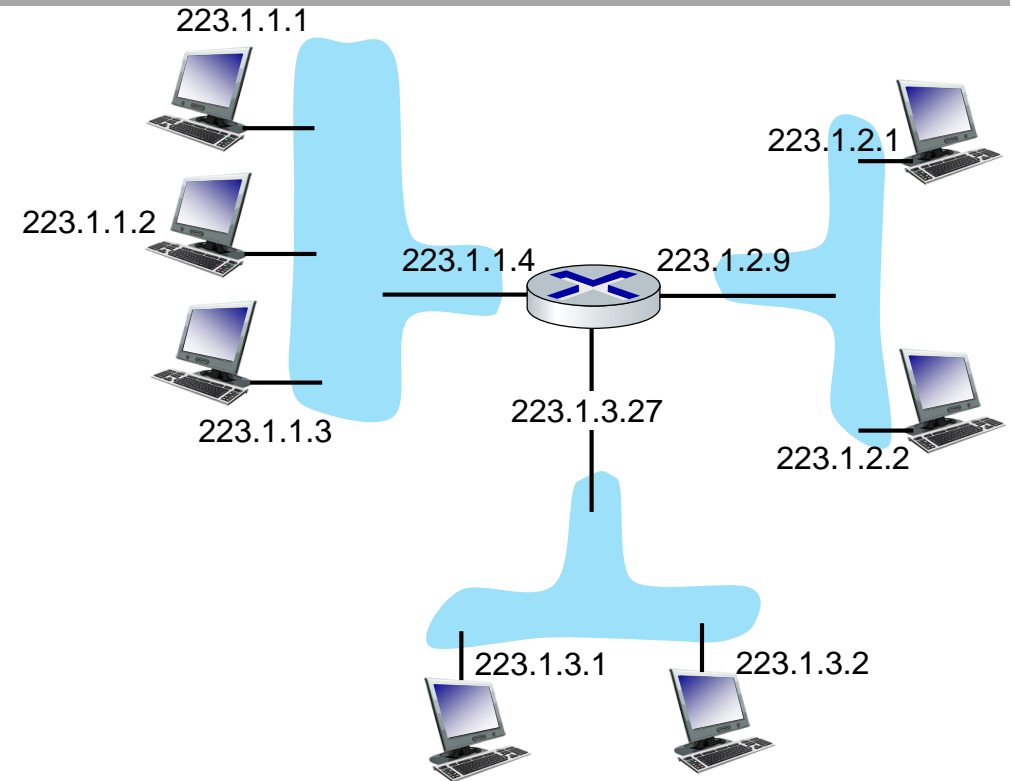


IP Datagram (formato)

- Número de versión (formato de paquete IPv4)
- Longitud de encabezado (en bytes)
- Tipo de servicio (TOS): e.g., real time vs non-real time
- Longitud del datagrama (en bytes): usualmente inferiores a 1500 bytes
- Identifier, flags, fragmentation offset: fragmentación
- Time to live (TTL): se decrementa en 1 cada vez que un paquete pasa por un enrutador (64 por defecto)
- Protocolo: interacción con capa de red (6:TCP, 17:UDP)
- Header checksum: header only
- Direcciones origen y destino
- Datos

IP addressing: introducción

- **IP address:** Identificador de 32 bits asociado con cada interfaz de host o enrutador
- **interface:** conexión entre host / enrutador y enlace físico
 - **los enrutadores** suelen tener múltiples interfaces
 - **el host** normalmente tiene una o dos interfaces (p. ej., Ethernet cableada, 802.11 inalámbrica)



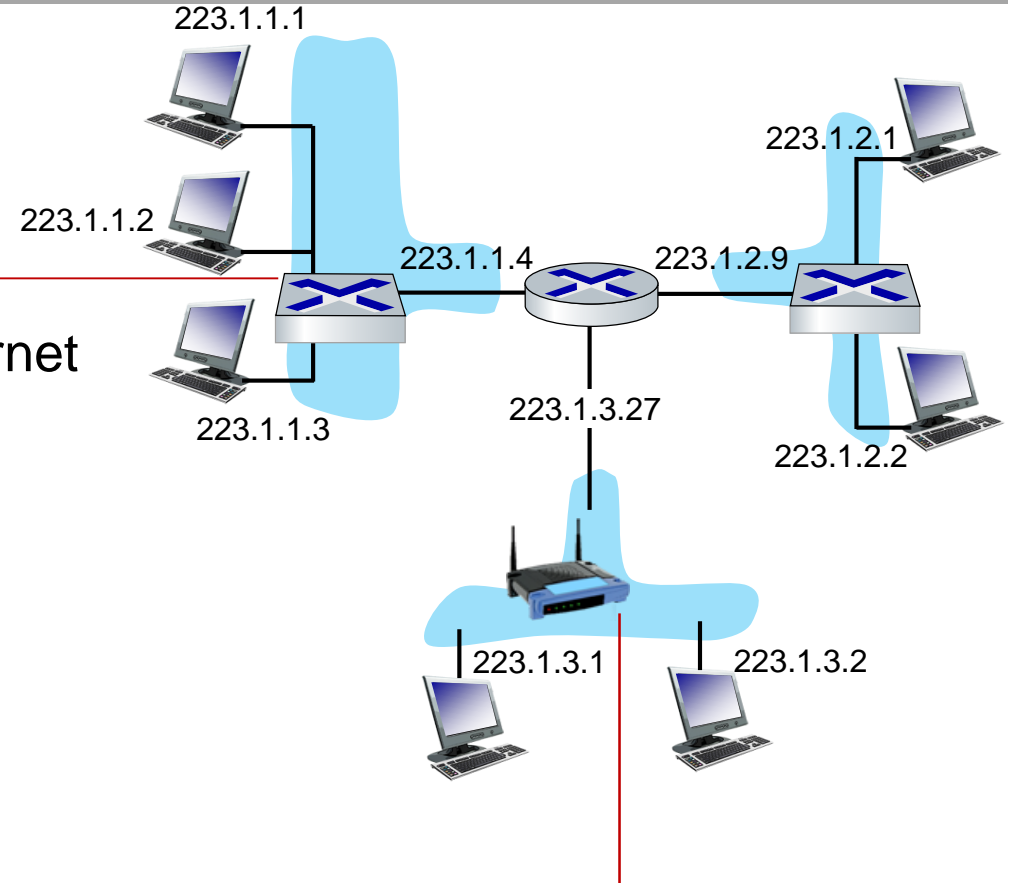
notación de dirección IP decimal con puntos:

$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_{1} \underbrace{00000001}_{1} \underbrace{00000001}_{1}$$

IP addressing: introducción

¿Cómo están
conectadas las
interfaces?

A: cableado
Interfaces Ethernet
conectadas por
conmutadores
Ethernet



A: interfaces WiFi inalámbricas
conectadas por estación base WiFi

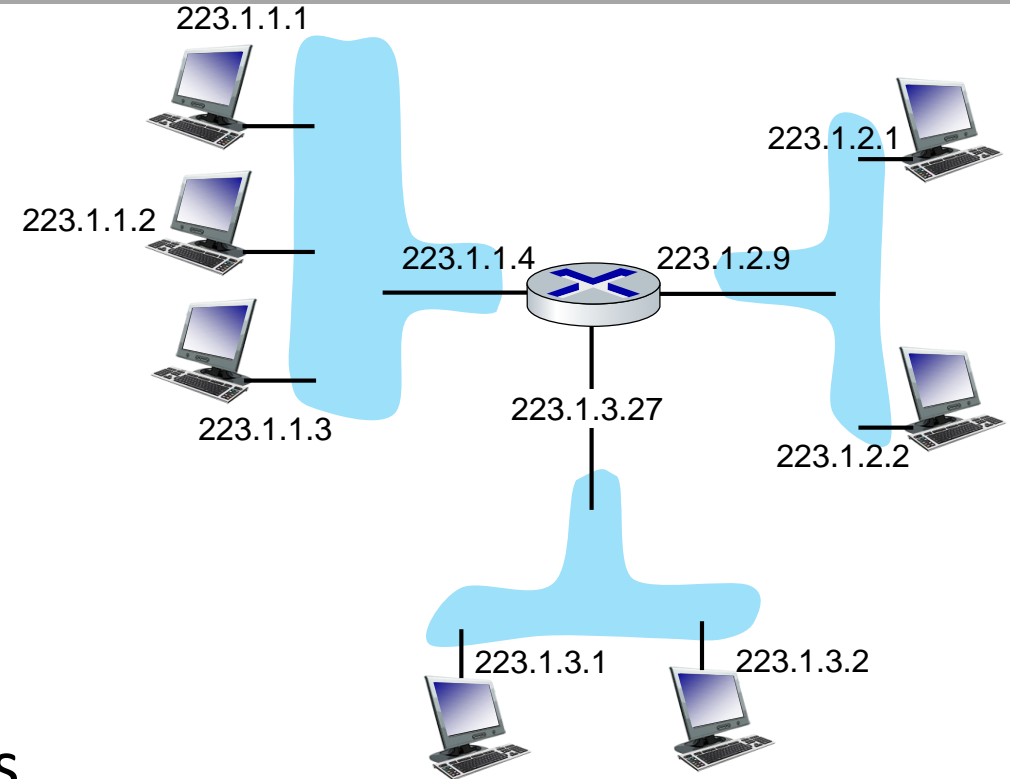
Subnets

■ ¿Qué es?

- interfaces de dispositivo que pueden llegar físicamente entre sí **sin pasar por un enrutador intermedio**

■ IP addresses (estructura):

- **subnet part:** los dispositivos en la misma subred tienen bits comunes de alto orden
- **host part:** bits restantes de orden inferior

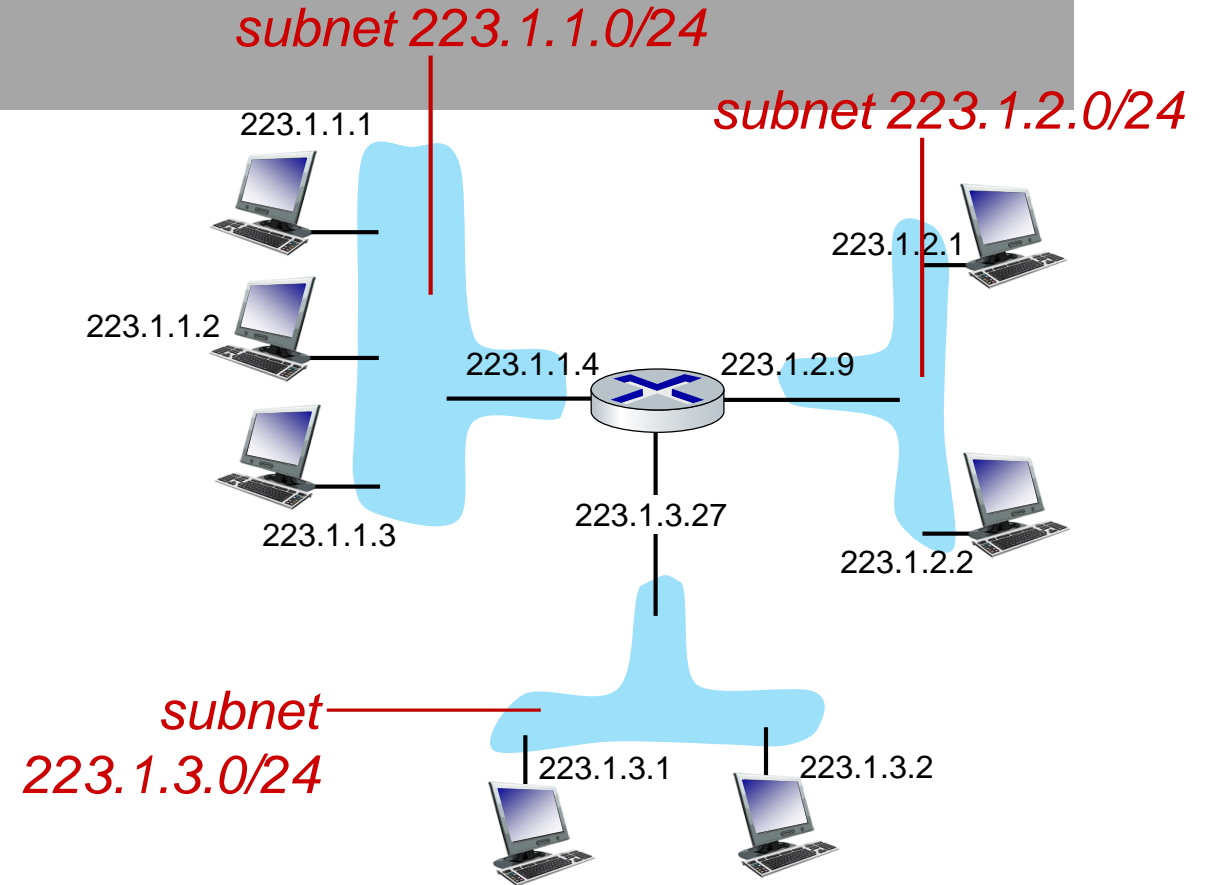


Red que integra 3 subredes (3 subnets)

Subnets

Definiendo subnets:

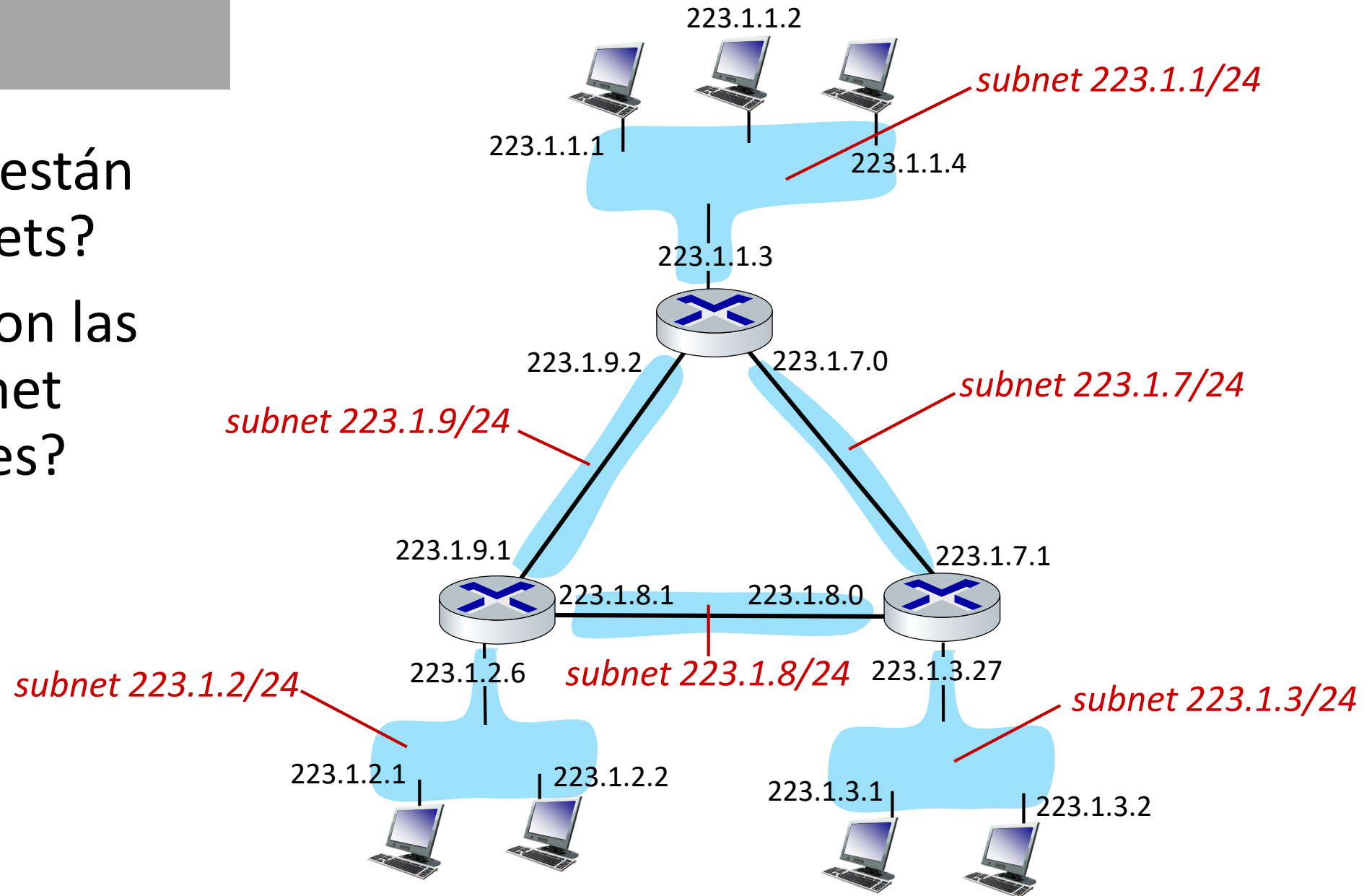
- separar cada interfaz de su host o enrutador, creando "islas" de redes aisladas
- cada red aislada se llama *subnet*



subnet mask: /24
(24 bits de orden superior: parte de subred de la dirección IP)

Subnets

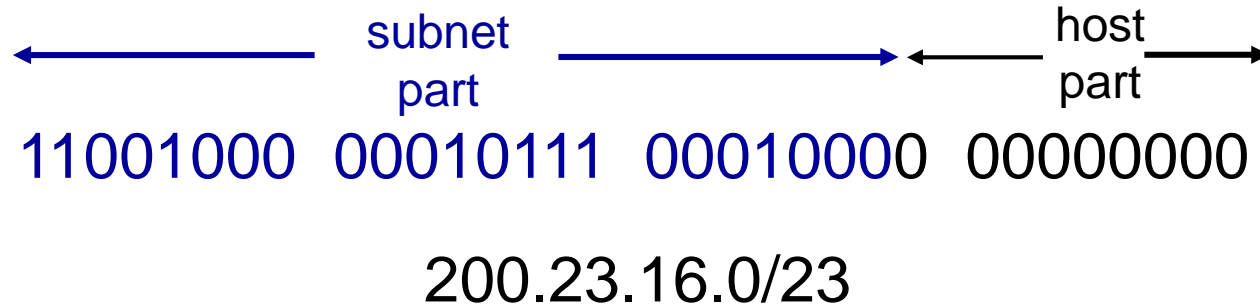
- ¿Dónde están las subnets?
- Cuáles son las /24 subnet addresses?



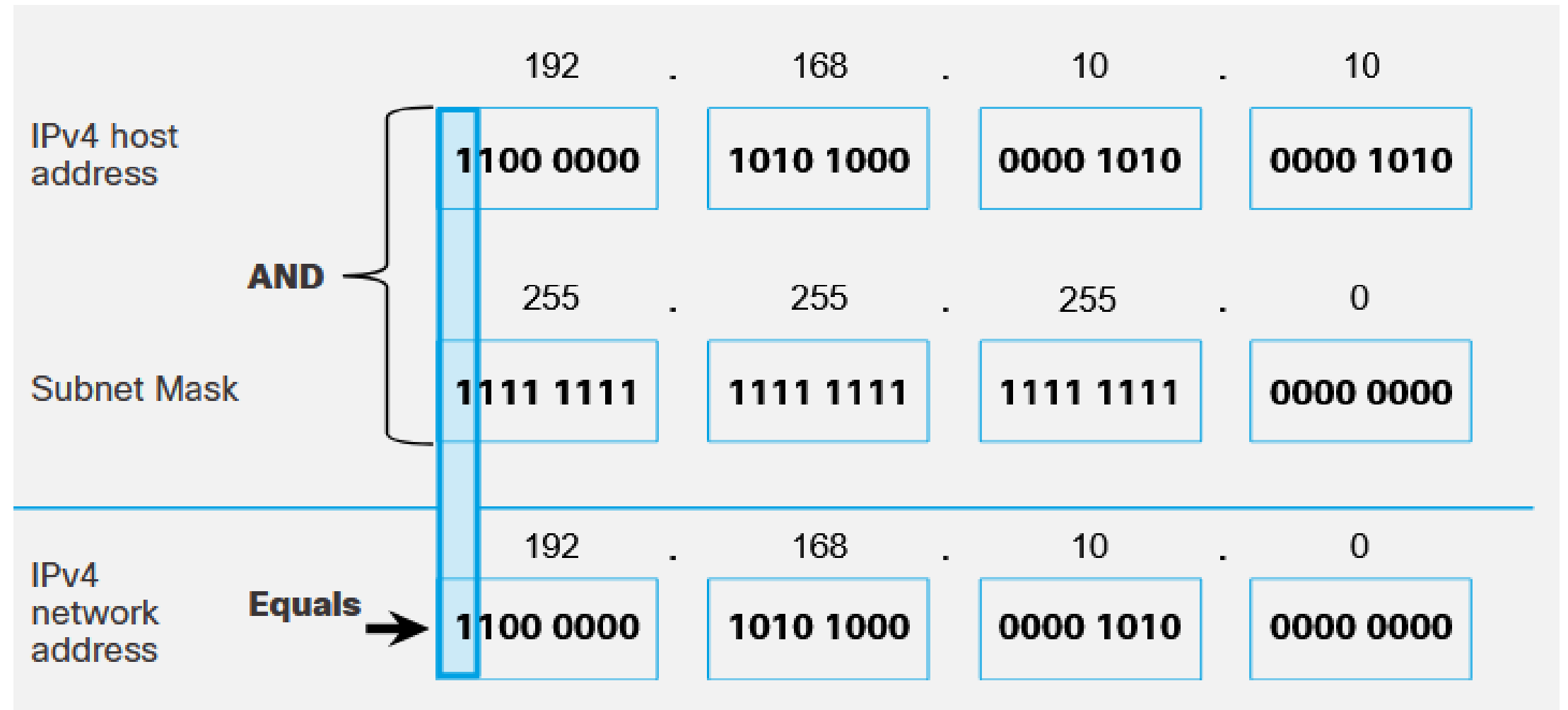
IP addressing: CIDR

CIDR: Classless InterDomain Routing (pronunciación en inglés “cider”)

- porción de subred de la dirección de longitud arbitraria
- formato de dirección: **a.b.c.d/x**, donde x es # bits en la porción de subred de la dirección



IP addressing: CIDR



Direcciones IP:

1. ¿Cómo obtiene un host la dirección IP dentro de su red (parte del host de la dirección)?
2. ¿Cómo obtiene una red una dirección IP por sí misma (parte de la dirección de red)?

¿Cómo obtiene el host la dirección IP?

- Codificada por sysadmin en archivo de configuración(e.g., /etc/rc.config in UNIX)
- **DHCP**: **D**ynamic **H**ost **C**onfiguration **P**rotocol: obtener la dirección de forma dinámica desde un servidor “plug-and-play”

DHCP: Dynamic Host Configuration Protocol

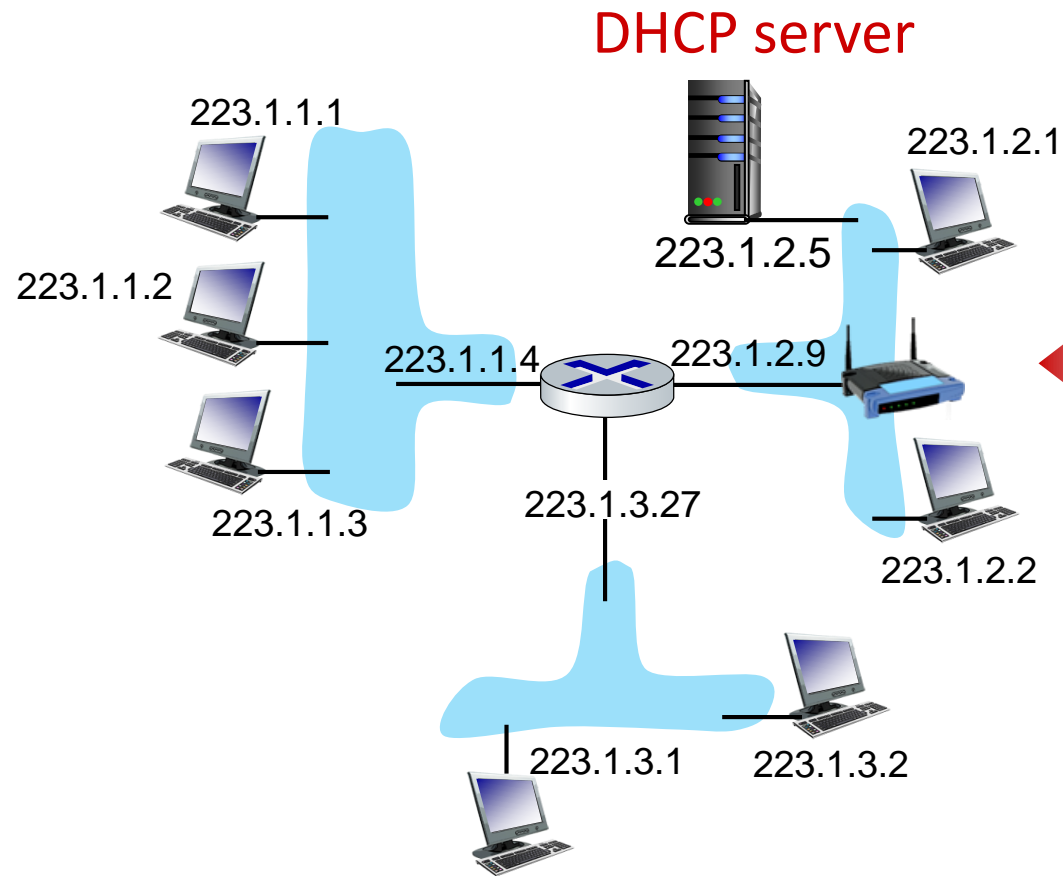
Objetivo: el host obtiene dinámicamente la dirección IP del servidor de red cuando se "une" a la red

- puede renovar su contrato de arrendamiento en la dirección en uso
- permite la reutilización de direcciones (solo mantiene la dirección mientras está conectado/encendido)
- soporte para usuarios móviles que se unen/abandonan la red

DHCP:

- host transmite **DHCP discover** msg [opcional]
- Servidor DHCP responde **DHCP offer** msg [opcional]
- host solicita dirección IP: **DHCP request** msg
- Servidor DHCP envía dirección: **DHCP ack** msg

DHCP cliente-servidor (escenario)



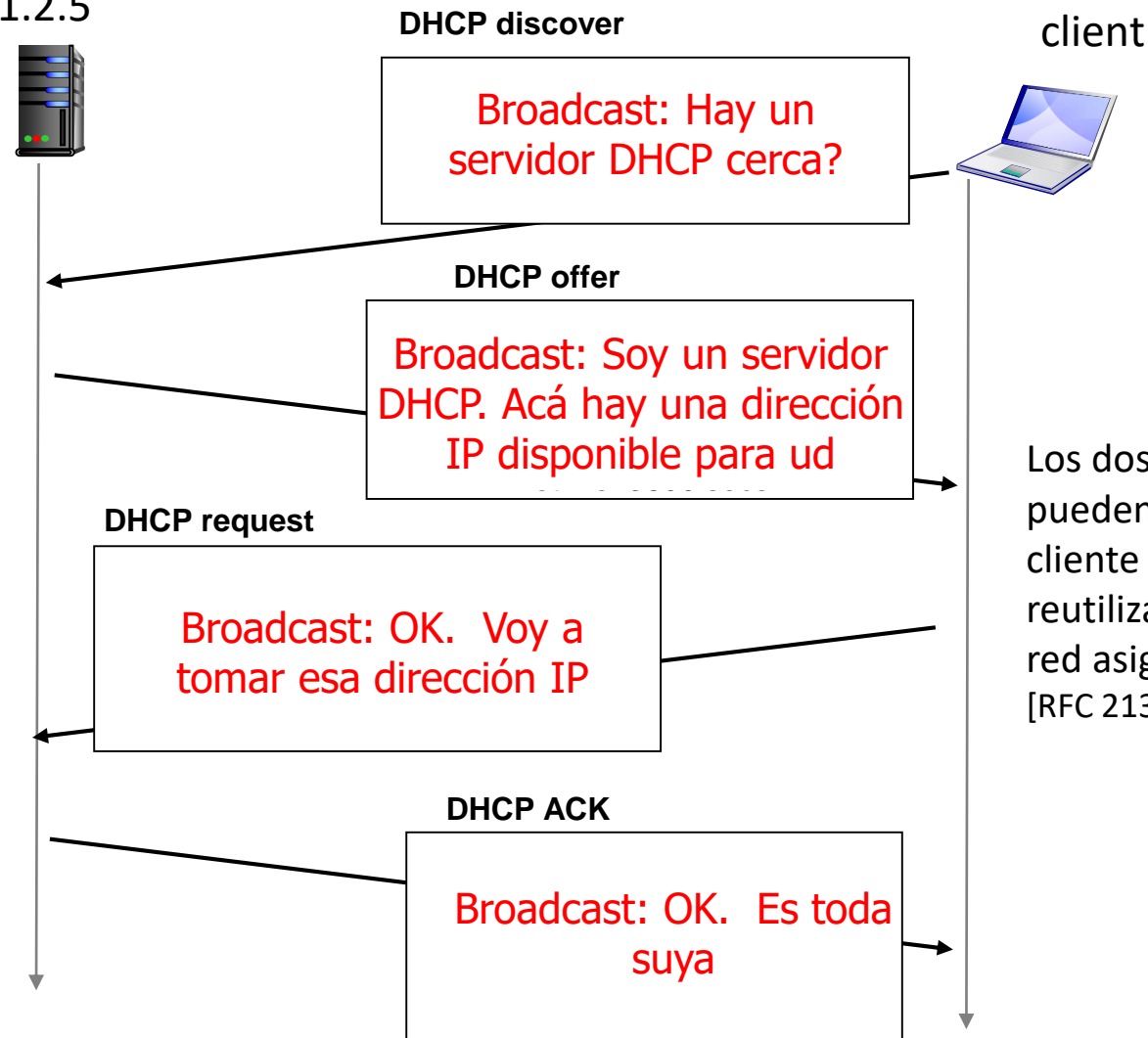
Por lo general, el servidor DHCP estará ubicado en el enrutador, sirviendo a todas las subredes a las que está conectado el enrutador



las necesidades del **cliente DHCP** que llegan en dirección a esta red

DHCP cliente-servidor (escenario)

DHCP server: 223.1.2.5



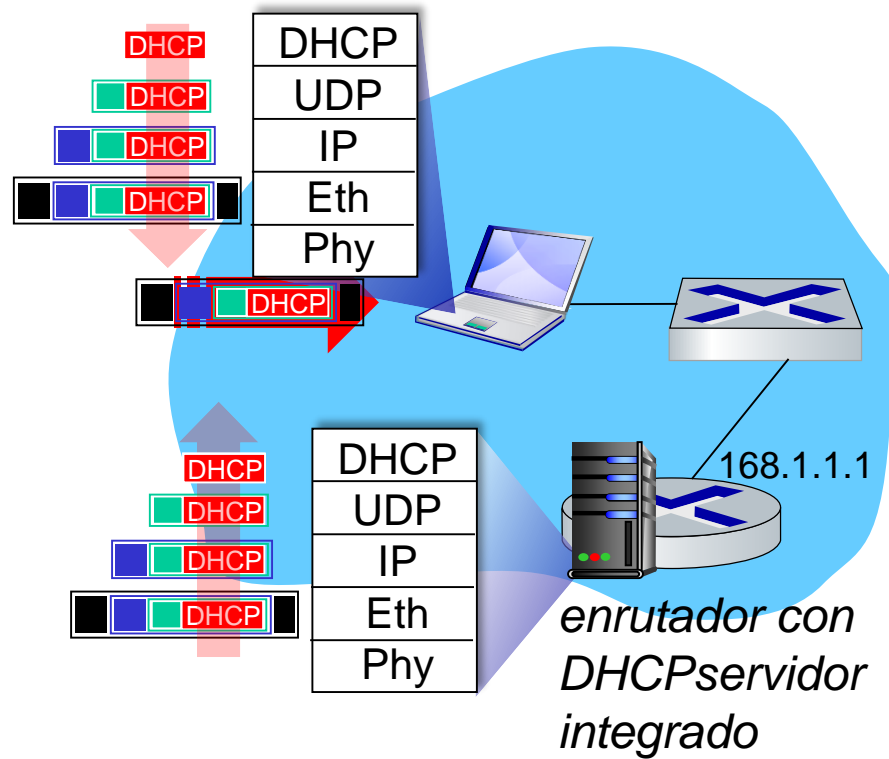
Los dos pasos anteriores se pueden omitir "si un cliente recuerda y desea reutilizar una dirección de red asignada previamente" [RFC 2131]

DHCP: más que direcciones IP

DHCP puede devolver más que solo una dirección IP asignada en la subred:

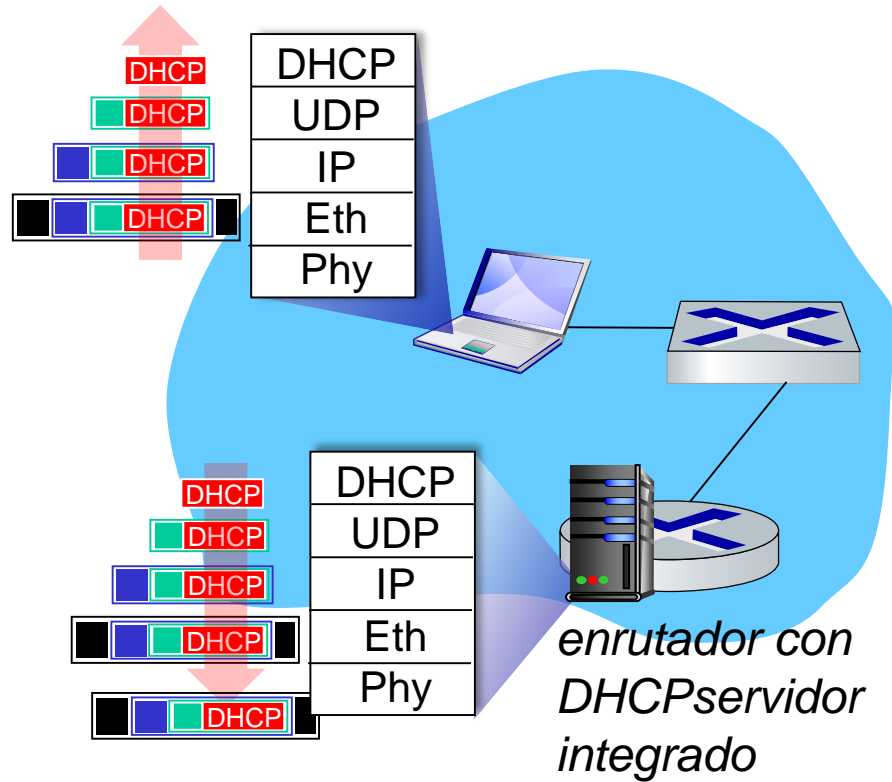
- dirección del enrutador de primer salto para el cliente
- nombre y dirección IP del servidor DNS
- máscara de red (que indica la parte de la dirección de la red frente al host)

DHCP: ejemplo



- La computadora portátil conectada usará DHCP para obtener la dirección IP, la dirección del enrutador de primer salto, la dirección del servidor DNS.
- Mensaje DHCP REQUEST encapsulado en UDP, encapsulado en IP, encapsulado en Ethernet
- Transmisión de trama Ethernet (dest: FFFFFFFFFFFFFFFF) en LAN, recibida en el enrutador que ejecuta el servidor DHCP
- Ethernet demux'ed ---> IP demux'ed, UDP demux'ed ---> DHCP

DHCP: ejemplo



- El servidor DHCP formula DHCP ACK que contiene la dirección IP del cliente, la dirección IP del enrutador de primer salto para el cliente, el nombre y la dirección IP del servidor DNS
- Respuesta del servidor DHCP encapsulado reenviado al cliente, demux hasta DHCP en el cliente
- el cliente ahora conoce su dirección IP, el nombre y la dirección IP del servidor DNS, la dirección IP de su enrutador de primer salto

Direcciones IP

¿Cómo obtiene la red parte de la subred de la dirección IP?

Obtiene una parte asignada del espacio de direcciones de su proveedor ISP

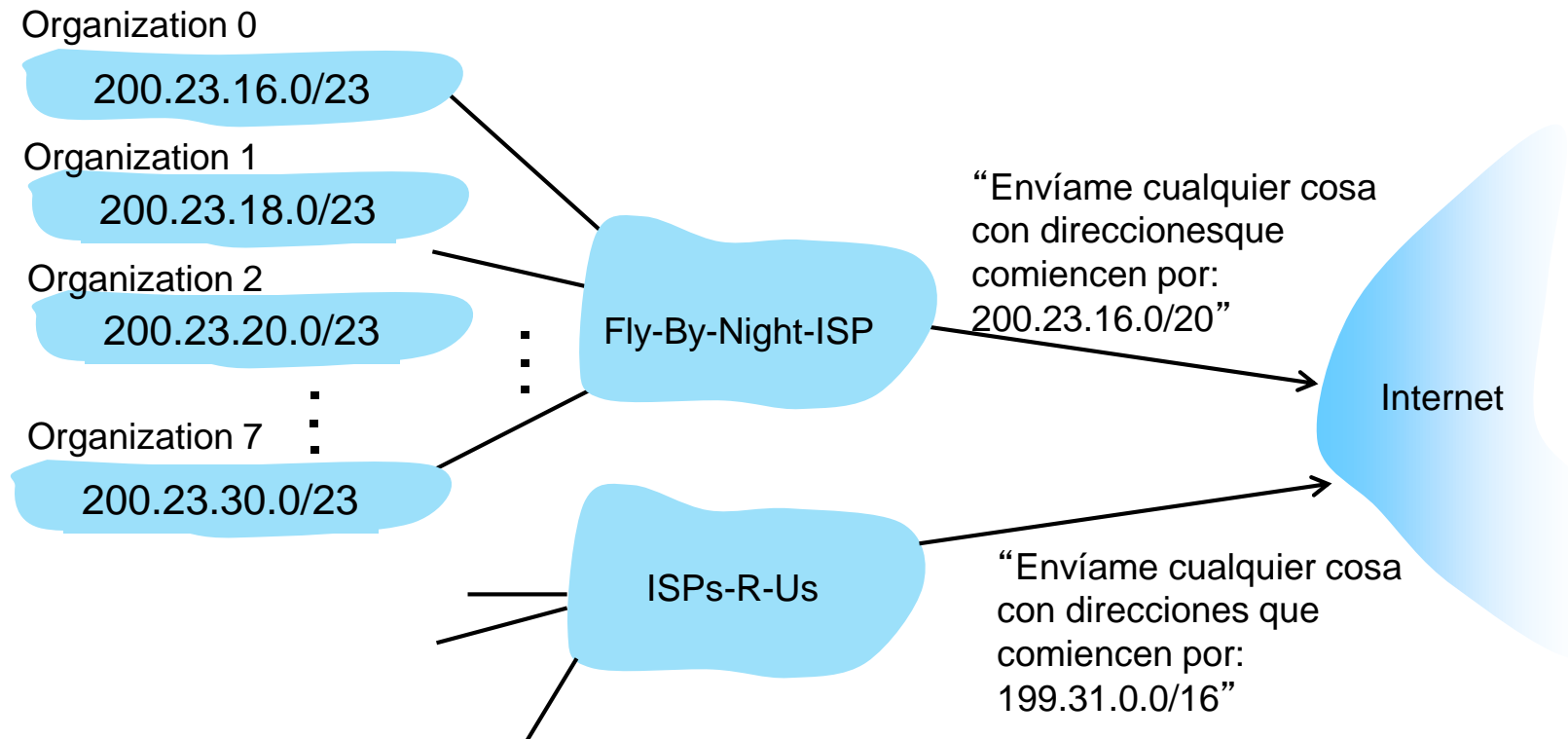
ISP's block 11001000 00010111 00010000 00000000 200.23.16.0/20

ISP luego puede asignar su espacio de direcciones en 8 bloques:

Organization 0	<u>11001000 00010111 00010000</u>	00000000	200.23.16.0/23
Organization 1	<u>11001000 00010111 00010010</u>	00000000	200.23.18.0/23
Organization 2	<u>11001000 00010111 00010100</u>	00000000	200.23.20.0/23
...
Organization 7	<u>11001000 00010111 00011110</u>	00000000	200.23.30.0/23

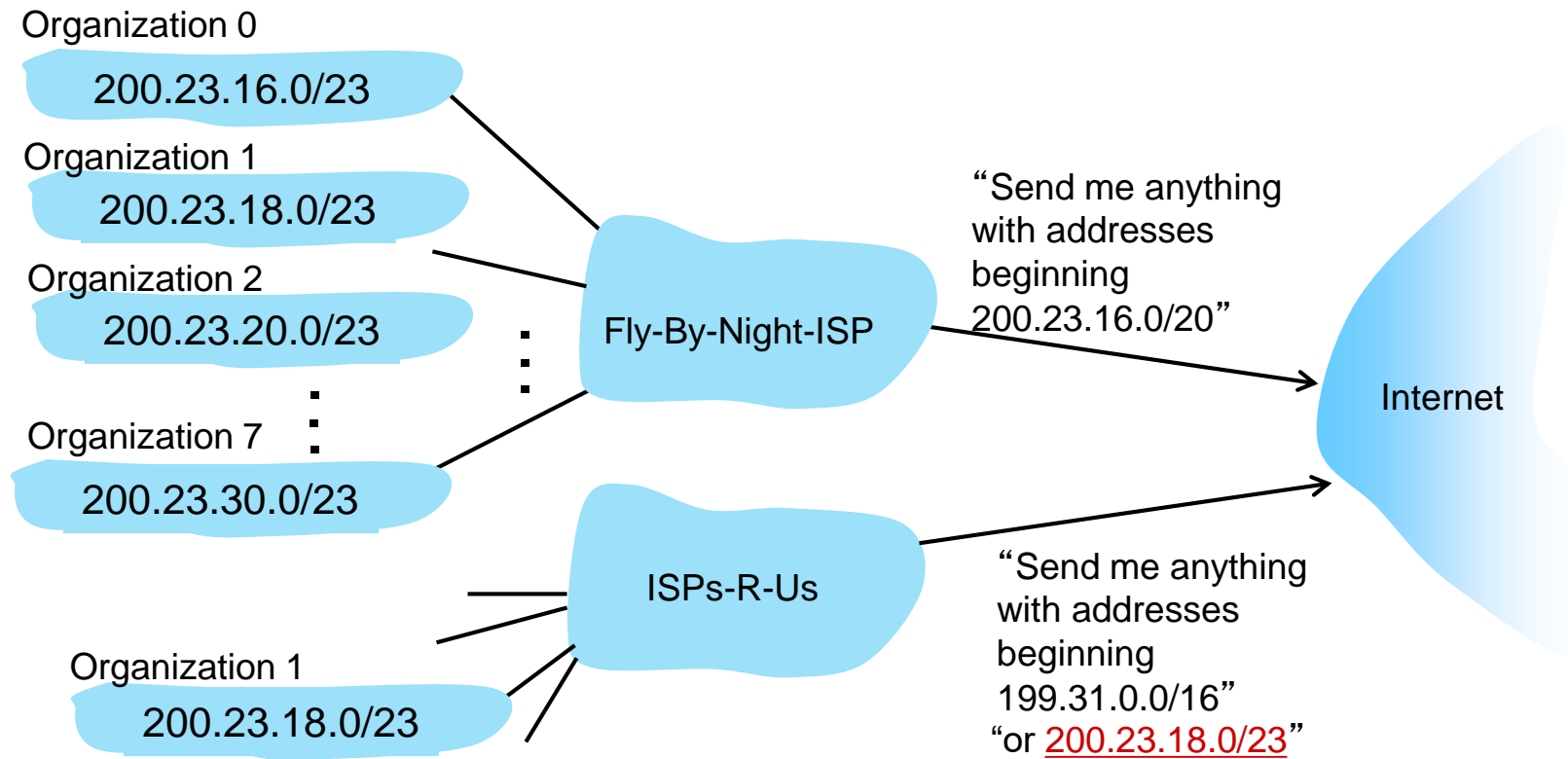
Hierarchical addressing: route aggregation

El direccionamiento jerárquico permite una publicación eficiente de la información de enrutamiento:



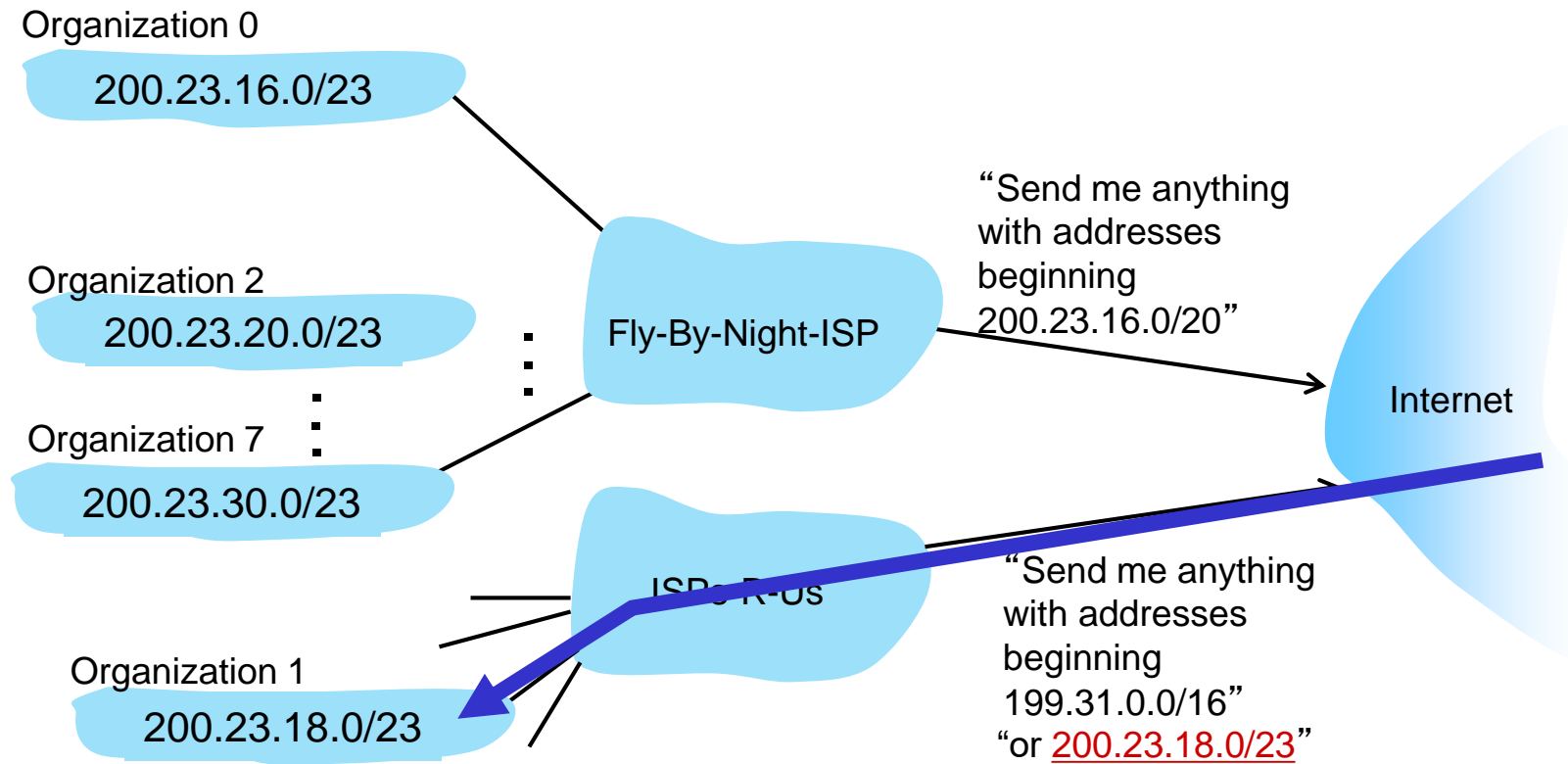
Hierarchical addressing: rutas más específicas

- Organization 1 se cambia de Fly-By-Night-ISP a ISPs-R-Us
- ISPs-R-Us ahora anuncia una ruta más específica a Organization 1



Hierarchical addressing: rutas más específicas

- Organization 1 se cambia de Fly-By-Night-ISP a ISPs-R-Us
- ISPs-R-Us ahora anuncia una ruta más específica a Organization 1



Direcciones IP

¿Cómo obtiene un ISP un bloque de direcciones?

ICANN: Internet Corporation for Assigned Names and Numbers
<http://www.icann.org/>

- asigna direcciones IP, a través de 5 registros regionales (RR) (que luego pueden asignar a los registros locales)
- gestiona root zone del DNS, incluida la delegación de la gestión de TLD individuales (.com, .edu,...)

¿Hay suficientes direcciones IP de 32 bits?

- La ICANN asignó la última parte de las direcciones IPv4 a los RR en 2011
- NAT (siguiente) ayuda al agotamiento del espacio de direcciones IPv4
- IPv6 tiene un espacio de direcciones de 128 bits

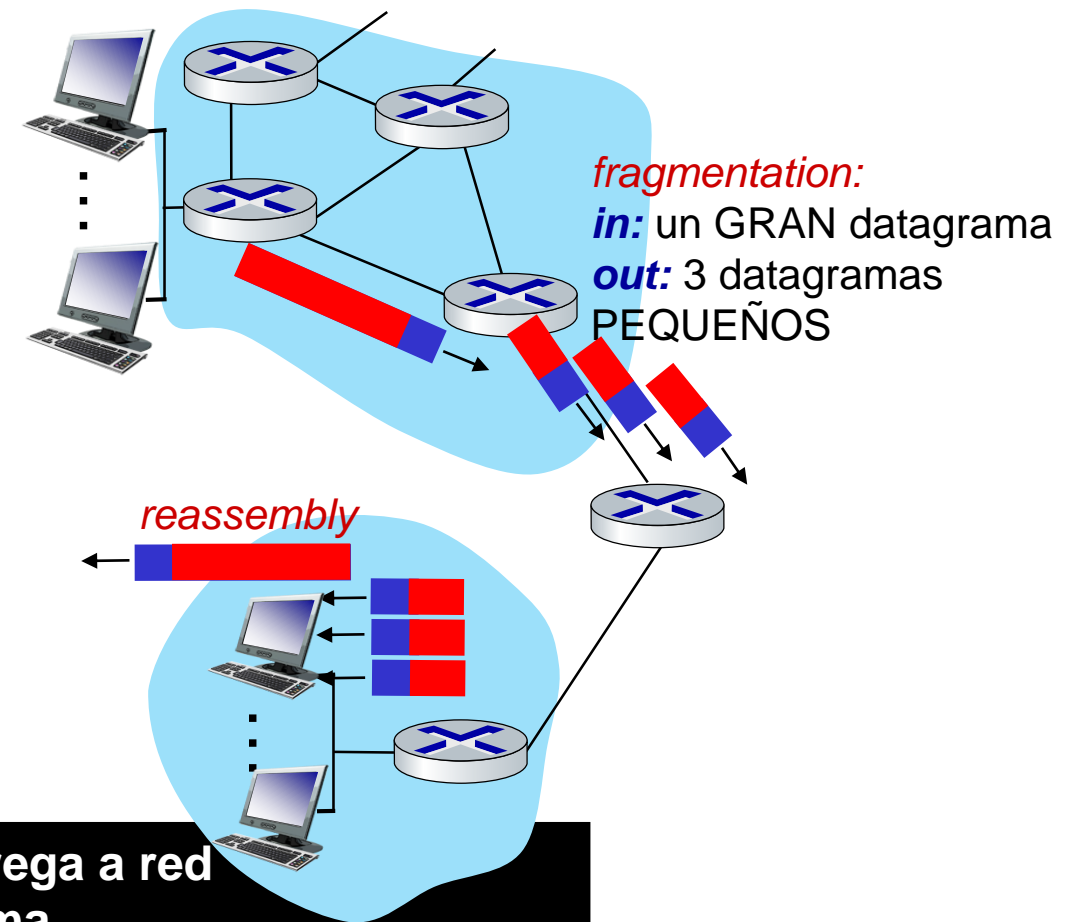
"¿Quién diablos sabía cuánto espacio de direcciones necesitábamos?" Vint Cerf (reflexionando sobre la decisión de hacer que la dirección IPv4 tenga una longitud de 32 bits)

IP fragmentation/reassembly

- Diferentes enlaces transmiten paquetes de diferentes tamaños
MTU: maximum transmission unit
- Ethernet: 1500 bytes - LTE: 1428 bytes - WiMax: 1400 bytes - BLEb(bluetooth low energy): 23
- Necesario fragmentar datagramas para enviarlos a través de los enlaces
- 3 campos: Identifier, flags, fragmentation offset
 - Identifier: identificador de cada datagrama de red (se usa para identificar todos los fragmentos de un mismo datagrama – nodo terminal)
 - Flags: si es el último fragmento del datagrama, flag es 0 (dfc es 1)
 - Offset: lugar en el que encaja el fragmento (medido en partes de 8 bytes de tamaño)

IP fragmentation/reassembly

- Los enlaces de red tienen MTU (tamaño máximo de transferencia): la trama de nivel de enlace más grande posible
- diferentes tipos de enlaces, diferentes MTU
- datagrama de IP grande dividido ("fragmentado") dentro de la red
- un datagrama se convierte en varios datagramas "Reensamblado" solo en el destino
- Bits de encabezado de IP utilizados para identificar y ordenar fragmentos relacionados



Nodo terminal reconstruye datagrama original y entrega a red
Si se pierde algún fragmento se descarta el datagrama
Enrutadores no reconstruyen datagrama, solo los fragmentan de ser necesario

IP fragmentation/reassembly

Ejemplo:

- Un datagrama de 4000 bytes llega al enrutador y se debe enviar a través de un enlace con MTU de 1500 bytes
- 20 bytes de encabezado, 3980 bytes de payload (datos)
- Se deben armar 3 paquetes (datagramas IP)
 - Paquete 1: 1480 bytes (byte 0 al 1479 - parte 0 a 184), offset: 0
 - Paquete 2: 1480 bytes (byte 1480 al 2959 - parte 185 a 369), offset: 185
 - Paquete 3: 1480 bytes (byte 2959 al 3979 - parte 370 a 498), offset: 370

IP fragmentation/reassembly

Ejemplo:

- 4000 byte datagram
- MTU = 1500 bytes

	length	ID	fragflag	offset	
	=4000	=x	=0	=0	

*un datagrama grande se convierte
en varios datagramas más pequeños*

1480 bytes in
data field

offset =
 $1480/8$

	length	ID	fragflag	offset	
	=1500	=x	=1	=0	

	length	ID	fragflag	offset	
	=1500	=x	=1	=185	

	length	ID	fragflag	offset	
	=1040	=x	=0	=370	

IP fragmentation/reassembly

Ejemplo:

- 4000 byte datagram
- M...

	length =4000	ID =x	fragflag =0	offset =0	
--	-----------------	----------	----------------	--------------	--

Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that $185 \cdot 8 = 1,480$)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980-1,480-1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that $370 \cdot 8 = 2,960$)	flag = 0 (meaning this is the last fragment)

	=1040	=x	=0	=370	
--	-------	----	----	------	--



HANDS-ON

- 3_PPTA_IP Basic 1
- Wireshark Hands-On 4 – IP Basic y fragmentación

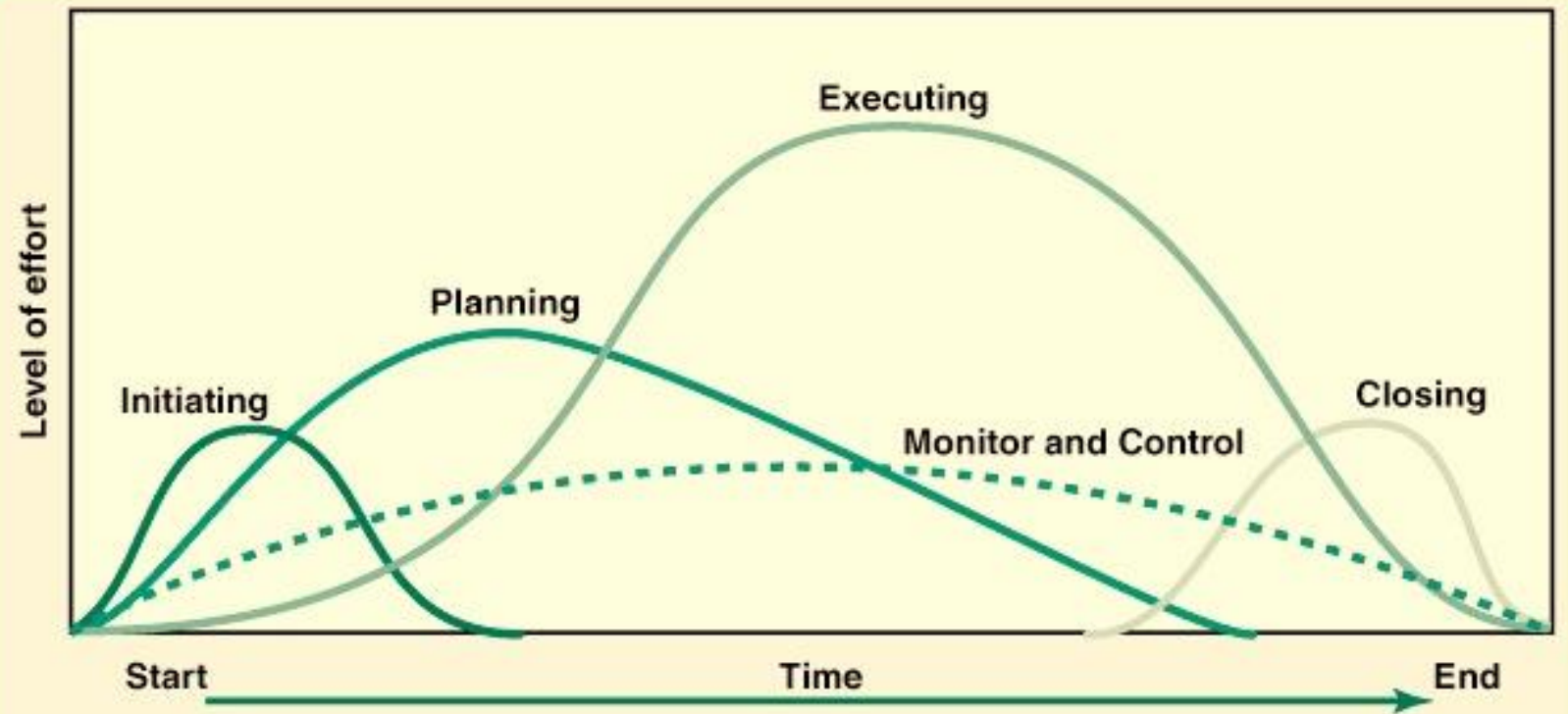
Metodología PMI para planeación:

- Aplicar metodologías PMI - Project Management Institute.
- Los certificados emitidos por PMI lideran proyectos en casi todos los países del mundo.
- El programa fue desarrollado para empresas que han adaptado muchos principios de gestión de proyectos, incluidos enfoques como Scrum, Kanban, Lead entre otros.
- La certificación de Project Management Professional (PMP) requiere:
 - Un título de 4 años junto con 4.500 horas de participación en la gestión de proyectos
 - O 7.500 horas sin un título de 4 años
 - 35 horas de educación en gestión de proyectos
 - Un examen de certificación riguroso.
 - Los PMP también deben ganar 60 unidades de desarrollo profesional cada tres años.

Metas de un profesional en gestión de proyectos:



PROJECT LIFE-CYCLE



Initiating

1. Definition of project scope
2. High-level budgets
3. Indicative time-frame
4. Project organisation
5. Establish project charter

Planning

1. Development of the project management plan
2. Detailed schedules
3. Detailed budgets
4. Resourcing
5. Risk analysis
6. Quality definition
7. Communication planning
8. Planning procurement

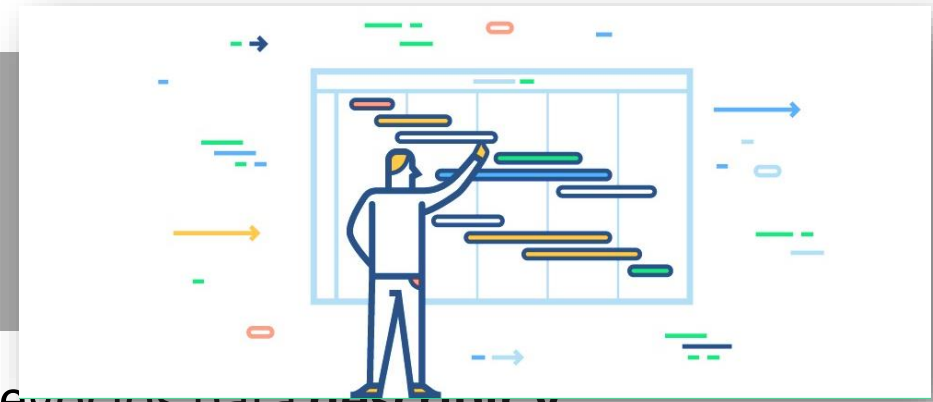
Executing

1. Change management
2. Quality control
3. Project reporting
4. Delivery of outcomes and outputs
5. Contract management
6. Team management
7. Stakeholder management

Closing

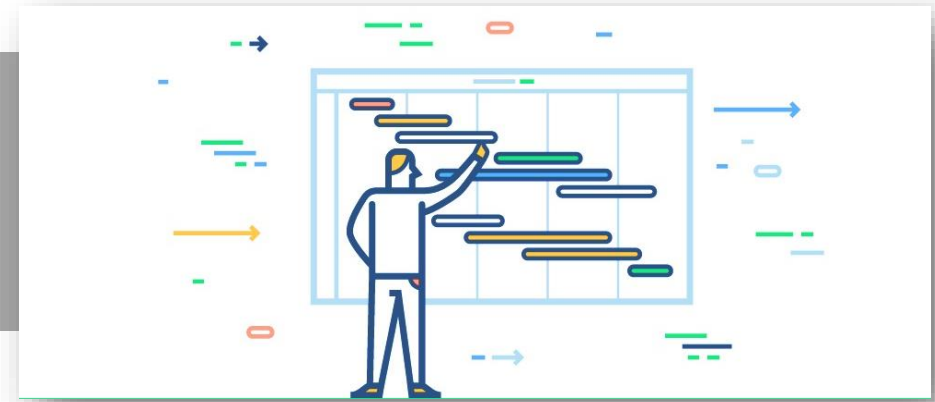
1. Financial closure
2. Final project reporting
3. Final handover to customer
4. Lessons learned
5. Release of resources

Diagrama de Gantt



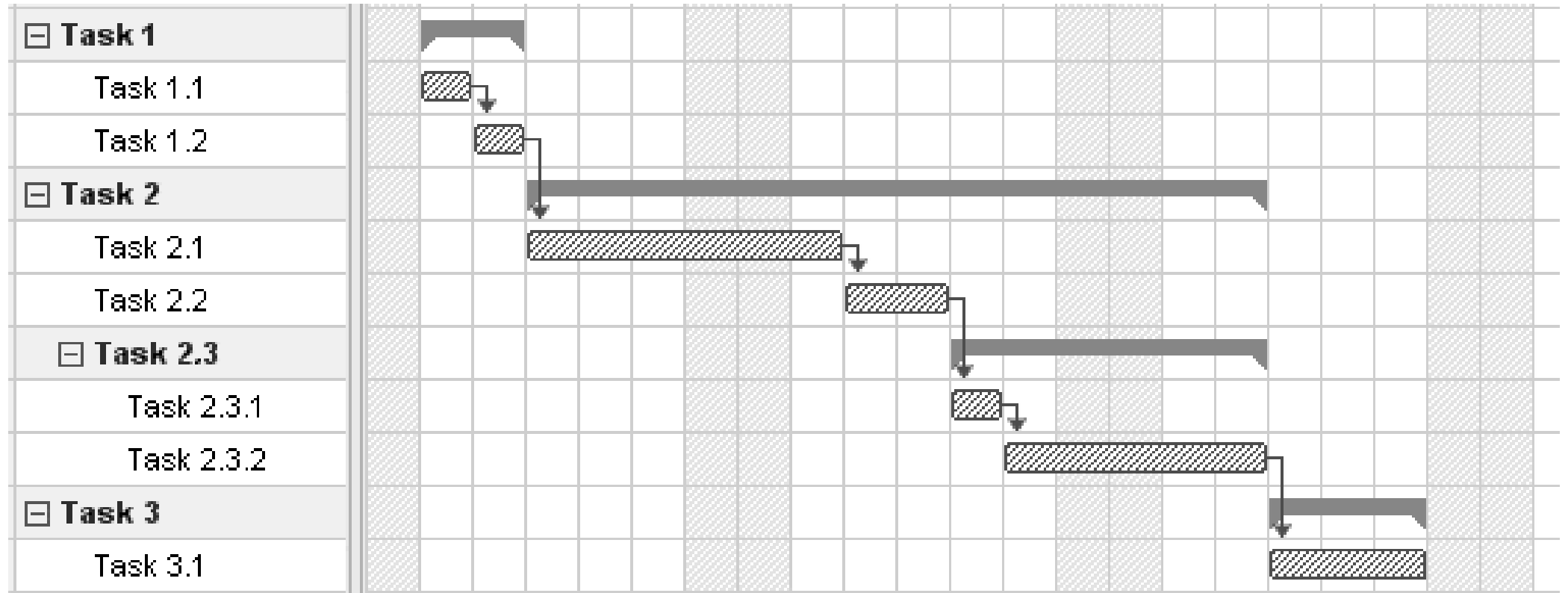
- Los diagramas de Gantt se utilizan ampliamente en los negocios para **describir y monitorear todo tipo de proyectos de acuerdo con las reglas de gestión de proyectos.**
- Existen herramientas informáticas como Microsoft Project, Primavera Project Planner, MindView, Ganttter, GanttProject, Excel, Trello, Gantt, PM, Teams, entre otros. Las diferentes aplicaciones de Gantt tienen diferentes características y capacidades.
- Lo primero que necesitará antes de configurar un diagrama de Gantt es **un plan detallado del proyecto.** Un plan de proyecto es una serie de tareas interdependientes que deben realizarse en un orden particular.
 - **Ejemplo:** Al mudarse a una nueva oficina, no puede comenzar a rediseñar el espacio de la oficina antes de que se firme el contrato de arrendamiento.

Diagrama de Gantt



- Los planes de proyecto tienen una **fecha de inicio específica**, que corresponde al inicio de la primera tarea, y una **fecha de finalización específica**, que corresponde al final de la última tarea.
- Una forma de crear un plan de proyecto es utilizar una **estructura de desglose del trabajo**, una técnica para dividir las tareas en subtareas y crear una jerarquía de tareas. Las aplicaciones de Gantt generalmente le permitirán reflejar la jerarquía del proyecto en la lista de tareas de Gantt a la izquierda del gráfico.

Diagrama de Gantt



Construir el Diagrama de Gantt



- Defina **la fecha de inicio, fecha de finalización y modo de programación**. El modo de programación más común es hacia adelante desde la fecha de inicio del proyecto. En este modo, el valor predeterminado es que las tareas comiencen lo antes posible, lo que significa que todo el proyecto finaliza en la fecha más temprana posible.
- Definir el **calendario del proyecto**. Esto establece el número de días laborables en la semana, el número de horas laborables en el día, etc.
- Ingrese o edite **nombres y duraciones de tareas**.
- Configure **una lista de recursos globales y asigne recursos a las tareas**. Aunque a menudo puede definir los recursos según los necesite, generalmente es más rápido comenzar configurando una lista global de recursos desde la cual puede seleccionar recursos para asignar a las diversas tareas del proyecto.

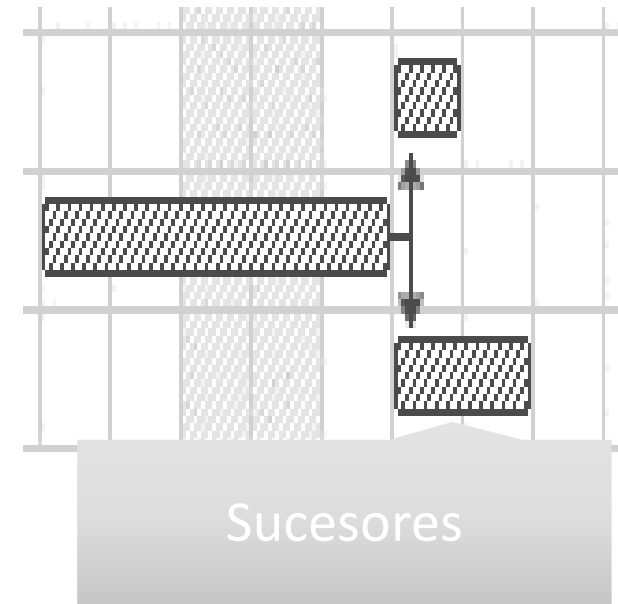
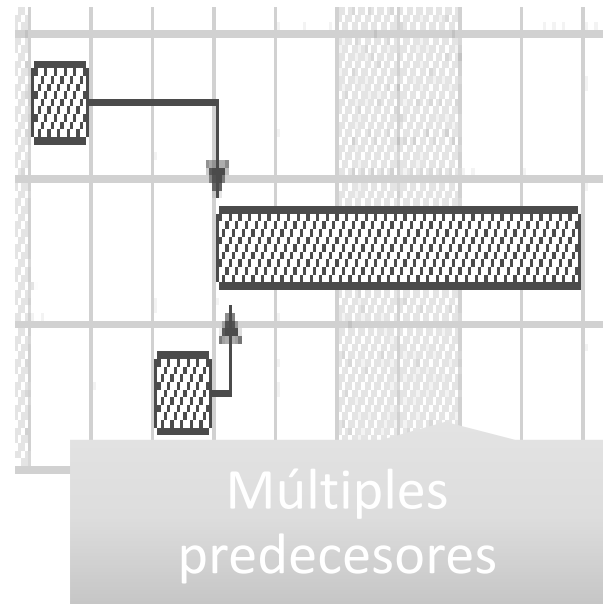
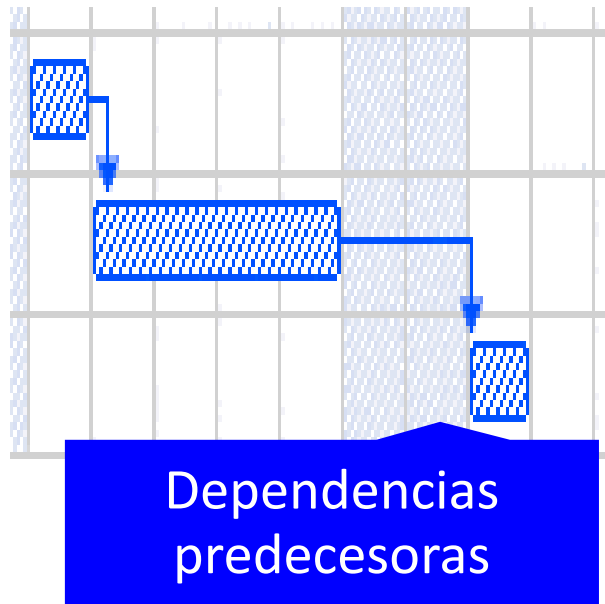
Construir el Diagrama de Gantt

- Cree enlaces para **especificar las dependencias entre las tareas del proyecto.**
- **Establezca restricciones en las tareas** según sea necesario.
- Haga **ajustes finales al plan del proyecto.**
- Una vez que el proyecto haya comenzado realmente, inspecciónelo a intervalos regulares para detectar posibles problemas o conflictos de programación y realice las correcciones necesarias



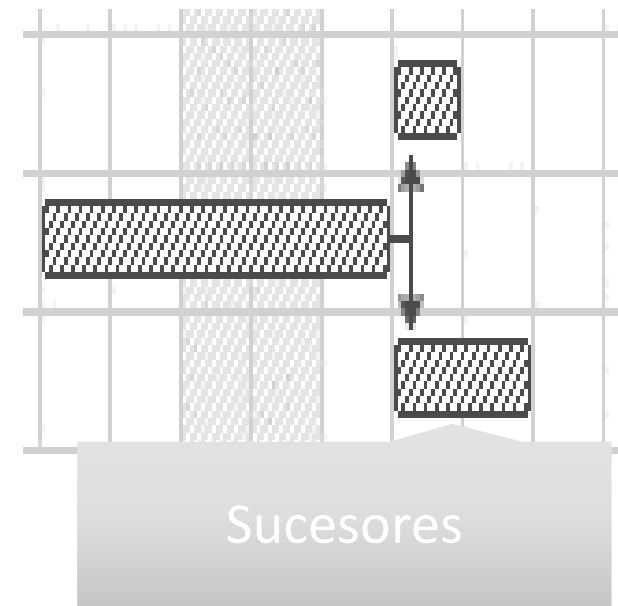
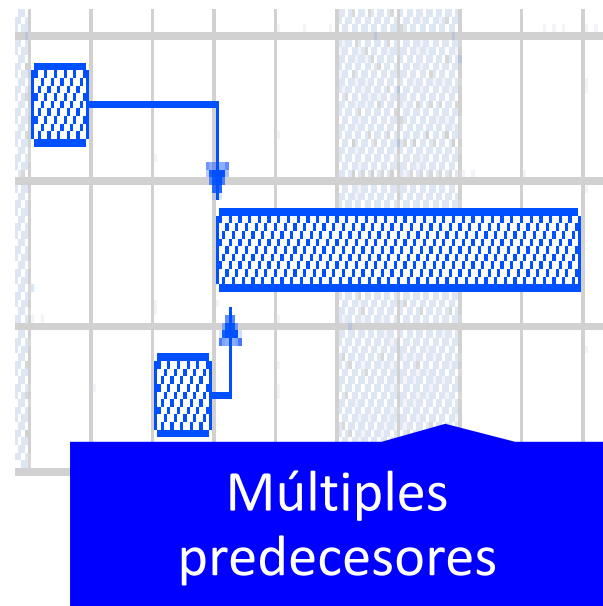
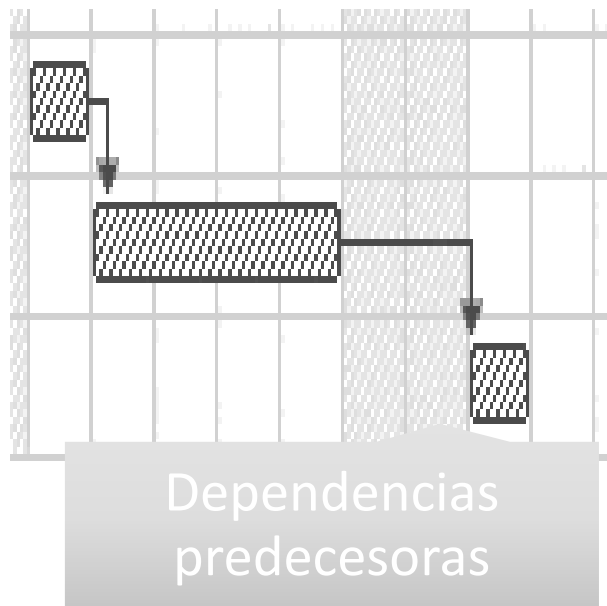
Vínculos de tareas

- Los planes de proyecto normalmente requieren que las tareas se realicen en un orden específico.
- De manera predeterminada, las tareas generalmente están vinculadas en una relación 'Finalizar para comenzar' (dependencia), lo que significa que la primera tarea que seleccione (la tarea predecesora) debe finalizar antes de que pueda comenzar la siguiente tarea que seleccione (la tarea sucesora).
- Esto generalmente se representa en el diagrama de Gantt mediante líneas con puntas de flecha que unen cada tarea a su sucesor. La punta de flecha indica la dirección del enlace: va del predecesor al sucesor.



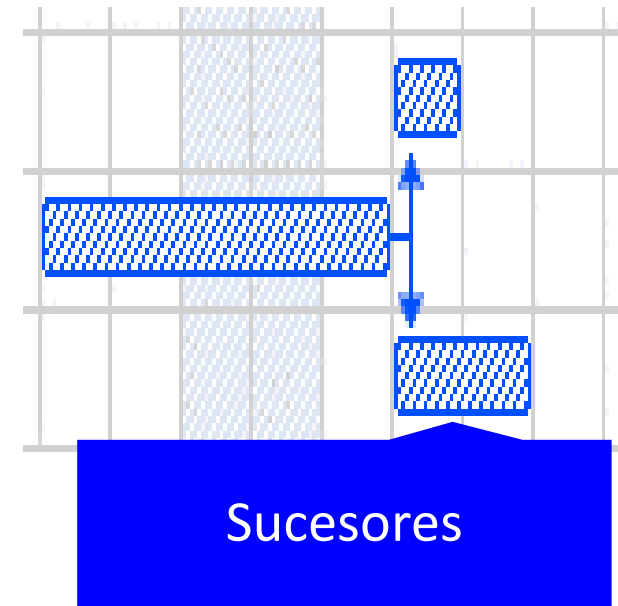
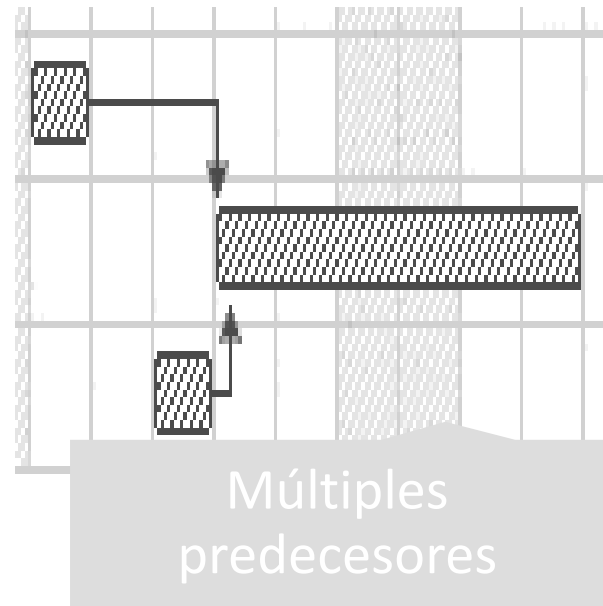
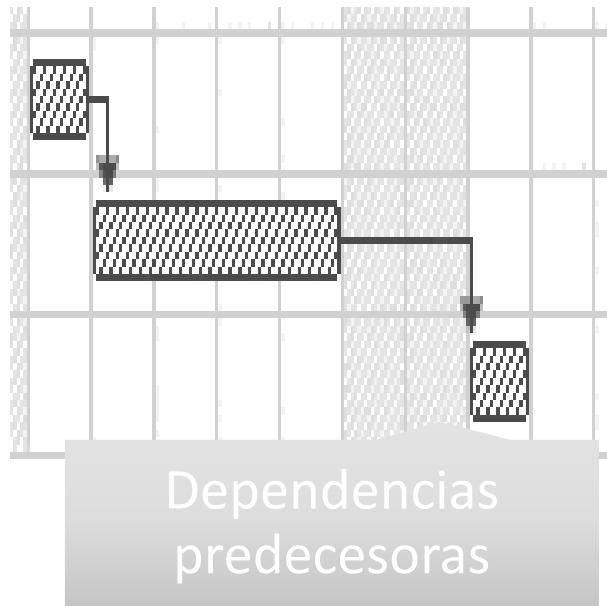
Vínculos de tareas

- Una tarea puede tener más de un predecesor. En este caso, su fecha de inicio está determinada por el enlace predecesor que le da la última fecha de inicio. A medida que las fechas y horas cambian durante el curso del proyecto, el enlace predecesor que determina la fecha de inicio de la tarea también puede cambiar.



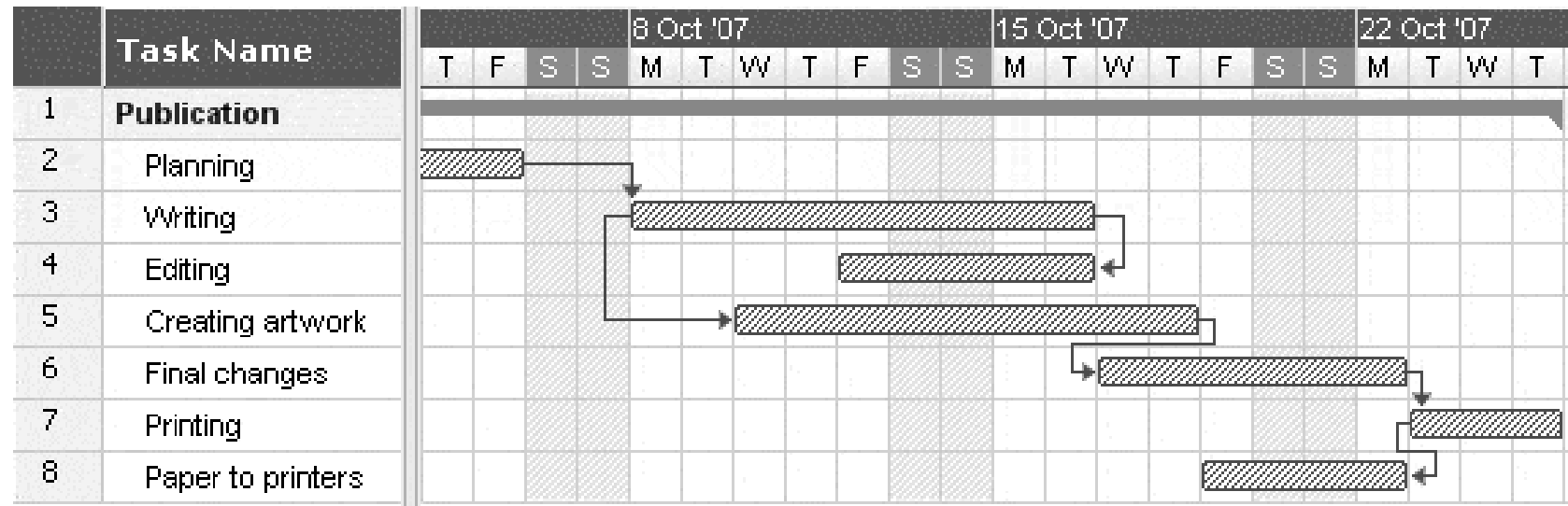
Vínculos de tareas

- Del mismo modo, una tarea puede tener varios sucesores. En este caso, la tarea determina la fecha de inicio de todas sus tareas sucesoras.



Tipos de dependencias

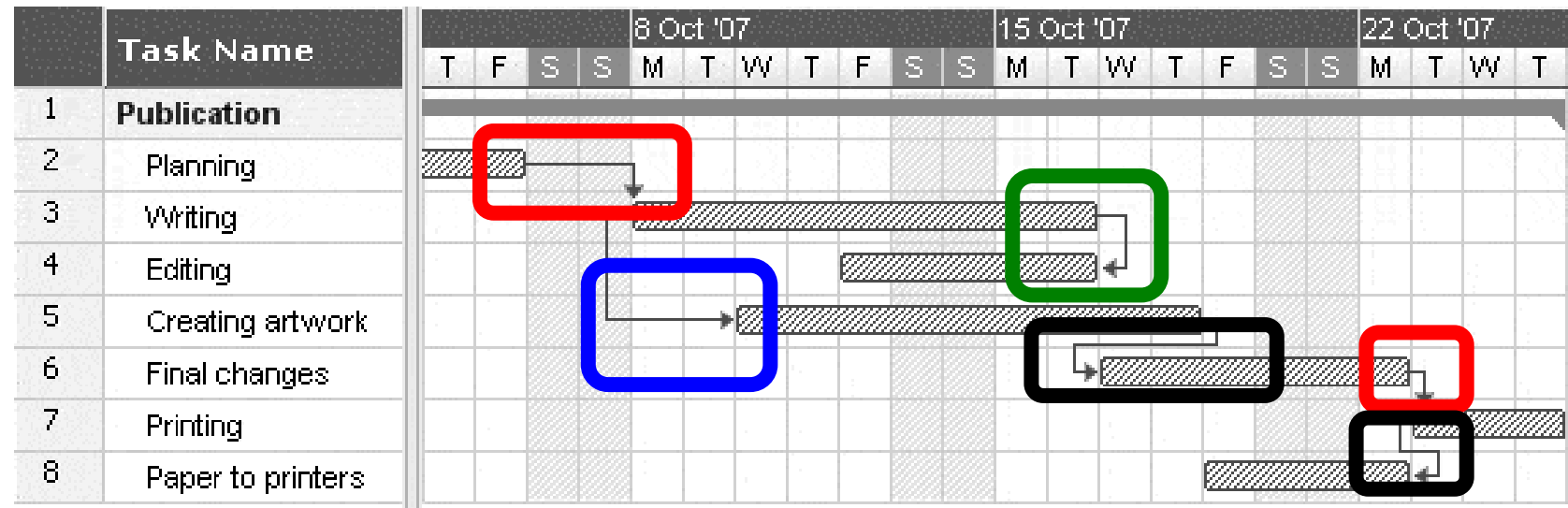
Diagrama de Gantt



- **Finish to Start (FS):** la tarea no puede iniciarse antes de que termine su predecesor, aunque puede comenzar más tarde. Este es el tipo más común de relación, y se describe anteriormente.
- **Start to Start (SS):** la tarea no puede comenzar hasta que se inicie el predecesor, aunque puede comenzar más tarde. Esto puede ser útil si tiene una tarea cuya fecha de inicio depende de la fecha de inicio de otra tarea.
- **Finish to Finish (FF):** la tarea no puede finalizar antes de que finalice el predecesor, aunque puede finalizar más tarde.
- **Start to Finish (SF):** la tarea no puede finalizar antes de que comience el predecesor, aunque puede finalizar más tarde. Esta relación de tareas rara vez se usa.

Tipos de dependencias

Diagrama de Gantt

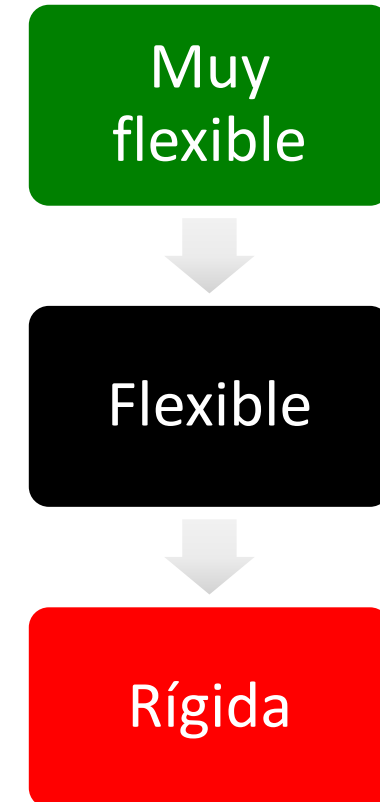


- **Finish to Start (FS):** la tarea no puede iniciarse antes de que termine su predecesor, aunque puede comenzar más tarde. Este es el tipo más común de relación, y se describe anteriormente.
- **Start to Start (SS):** la tarea no puede comenzar hasta que se inicie el predecesor, aunque puede comenzar más tarde. Esto puede ser útil si tiene una tarea cuya fecha de inicio depende de la fecha de inicio de otra tarea.
- **Finish to Finish (FF):** la tarea no puede finalizar antes de que finalice el predecesor, aunque puede finalizar más tarde.
- **Start to Finish (SF):** la tarea no puede finalizar antes de que comience el predecesor, aunque puede finalizar más tarde. Esta relación de tareas rara vez se usa.

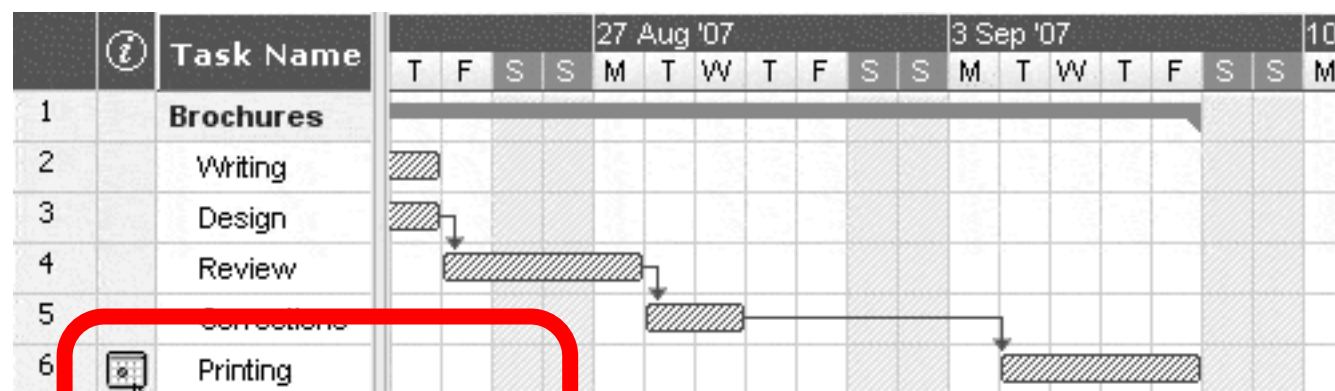
Restricciones de tareas

Diagrama de Gantt

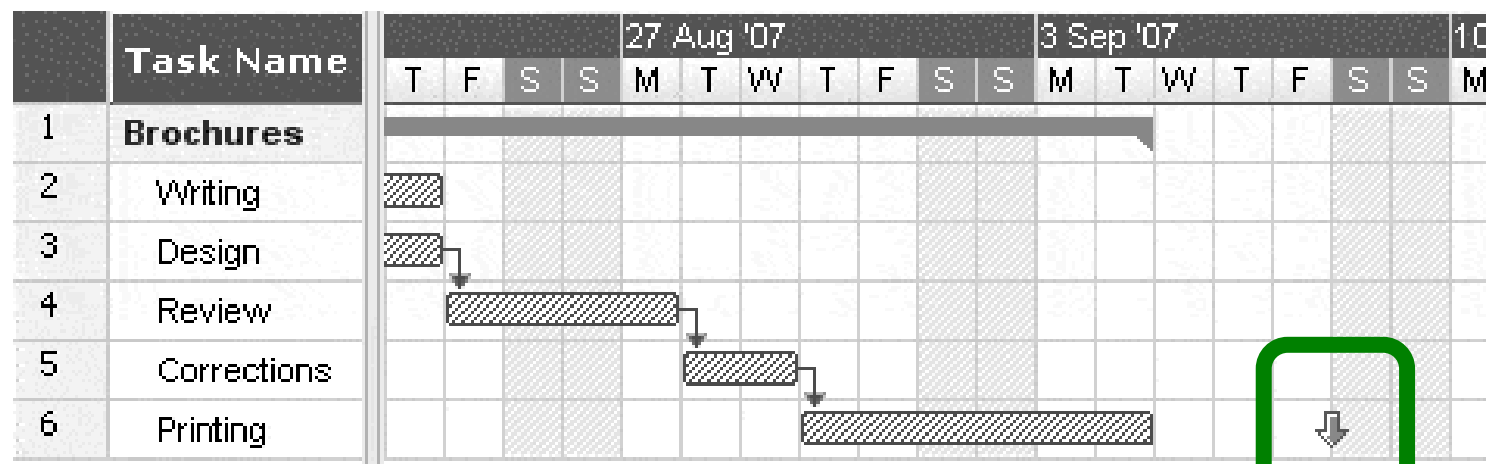
- **As Soon As Possible (ASAP)**
- **As Late As Possible (ALAP)**
- **Start No Earlier Than (SNET).**
- **Start No Later Than (SNLT)**
- **Finish No Earlier Than (FNET)**
- **Finish No Later Than (FNLT)**
- **Must Start On (MSO)**
- **Must Finish On (MFO)**



Visualización de restricciones y deadlines



A constraint is set on this task:
Must Finish On 07/09/2007



Visualización de ruta crítica de proyecto y retrasos

