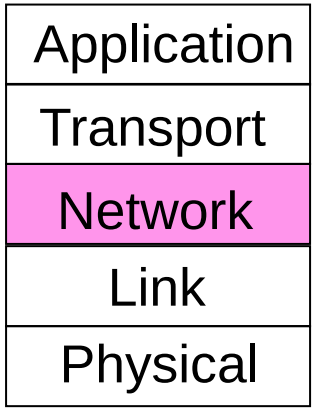
Capítulo 4

La Capa de Red – Generalidades



Capa de Red (CR)

- **Propósito de la capa de red:**
  - Llevar paquetes de un host de origen a uno de destino siguiendo una ruta conveniente

- **Asuntos de los que se encarga la capa de red:**
  - Almacenamiento y reenvío
  - Enrutamiento
  - Control de congestión
  - Conectar redes de distintas tecnologías
  - Fragmentación

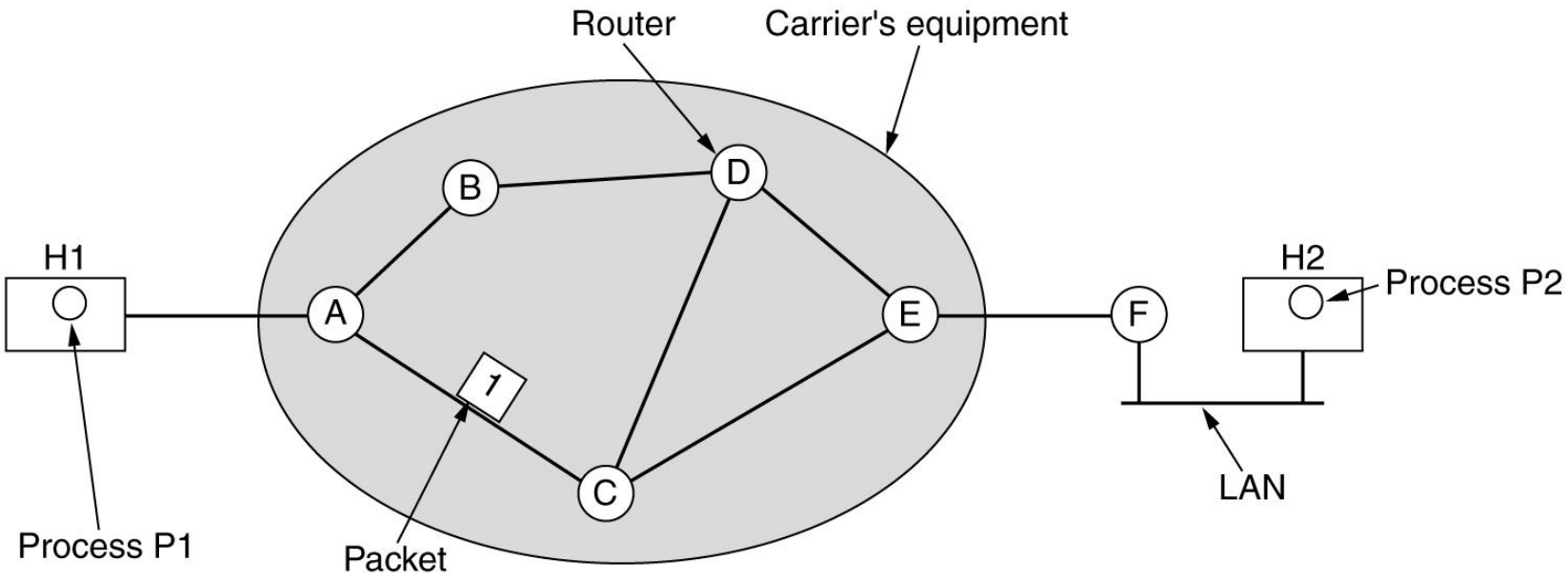
Capa de Red (CR)

- **¿Por qué estudiamos la capa de red?**
  - Para entender cómo están organizadas las redes.
  - Para entender cómo se interconectan redes de distintas tecnologías.
  - Para aprender algunos conceptos fundamentales, p.ej. proveedores de servicios de red, enrutadores, etc.
  - Para entender la necesidad de los enrutadores y cómo funcionan.
  - Para entender cómo se hacen asignaciones de direcciones de red a máquinas en una red local, a instituciones varias.
    - El por qué y cómo se lo hace.
  - Para entender algunos problemas fundamentales y algoritmos alternativos para su solución.
    - Enrutamiento, control de congestión, fragmentación.

Agenda de la primera parte de capa de red

- **Aprenderan:**
  1. **Cómo es el hardware subyacente a la CR**
  2. Enfoques usados para envío de paquetes entre dos hosts.
  3. Enrutamiento Jerárquico.
  4. Cómo es la arquitectura de un enrutador
  5. Conceptos básicos para algoritmos de enrutamiento
  6. Algoritmo de enrutamiento de caminos más cortos.

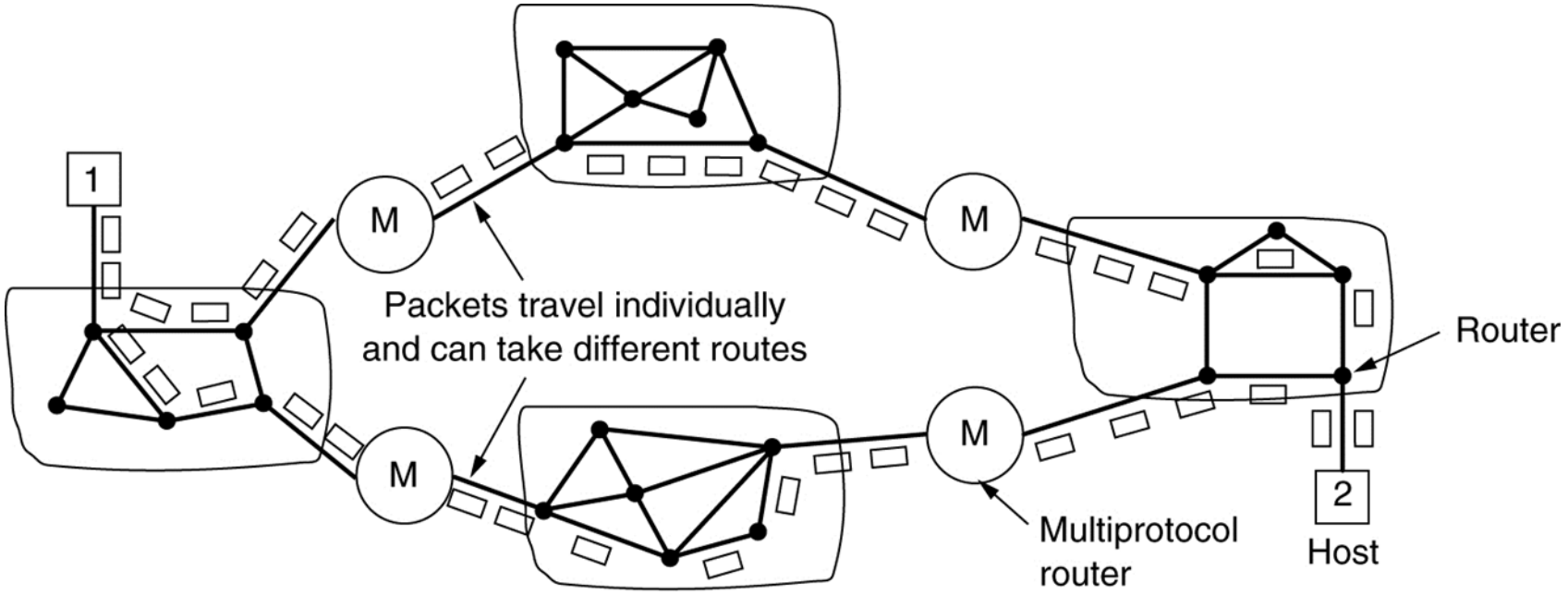
Capa de Red (CR)



Hardware subyacente de la CR:

- Subred: formada por enrutadores interconectados
- Hosts o LANs conectadas a subred

Capa de Red (CR)



¿Cómo es el hardware subyacente de la CR?

- Varias subredes de distinta tecnología unidas entre sí usando **puertas de enlace**

No puede pasar un paquete tal cual de una red a otra, porque:

- Formatos de paquetes difieren de una red a otra
- Tamaños máximos de paquetes difieren de una red a otra

Aprenderemos

- **Agenda:**
  1. Cómo es el hardware subyacente a la CR
  2. **Enfoques usados para envío de paquetes entre dos hosts**
     - **Para entender dos maneras existentes de hacer el enrutamiento en las subredes.**
  3. Enrutamiento Jerárquico.
  4. Cómo es la arquitectura de un enrutador
  5. Conceptos básicos para algoritmos de enrutamiento
  6. Algoritmos de enrutamiento de caminos más cortos

Envío de paquetes entre dos hosts

- **Enfoques para mandar un *conjunto de paquetes* desde un host de origen a un host de destino.**
- Hay dos bandos en relación a cómo se debe hacer esto:
  - Usar una ruta fija para mandar todos los paquetes (**servicio orientado a la conexión**).
  - La ruta puede cambiar, por lo que distintos paquetes pueden seguir distintos caminos (**servicio no orientado a la conexión**).

Envío de paquetes entre dos hosts

1. **Servicio no orientado a la conexión:**
   - Alentado por la comunidad de internet
   - Los paquetes se enrutan de manera independiente.
     - La ruta a usar entre los hosts va a ***cambiar cada cierto tiempo***.
     - Cada paquete debe llevar una dirección de destino completa.
   - **Nomenclatura usada**:
     - Paquetes = **datagramas**
     - Subredes = **subredes de datagramas**

Envío de paquetes entre dos hosts

1. **Servicio no orientado a la conexión (cont):**
   - **Diseño de la tabla de un enrutador.**
   - **Suponemos**:
     - Existe un procedimiento que dada la dirección del host de destino me retorna dirección del enrutador de destino (al cual está conectado host de destino).
       - Después veremos como se puede hacer esto.
     - Enrutador de destino ***sabe cómo entregar*** paquete a host de destino (por más que el host de destino esté en una LAN).

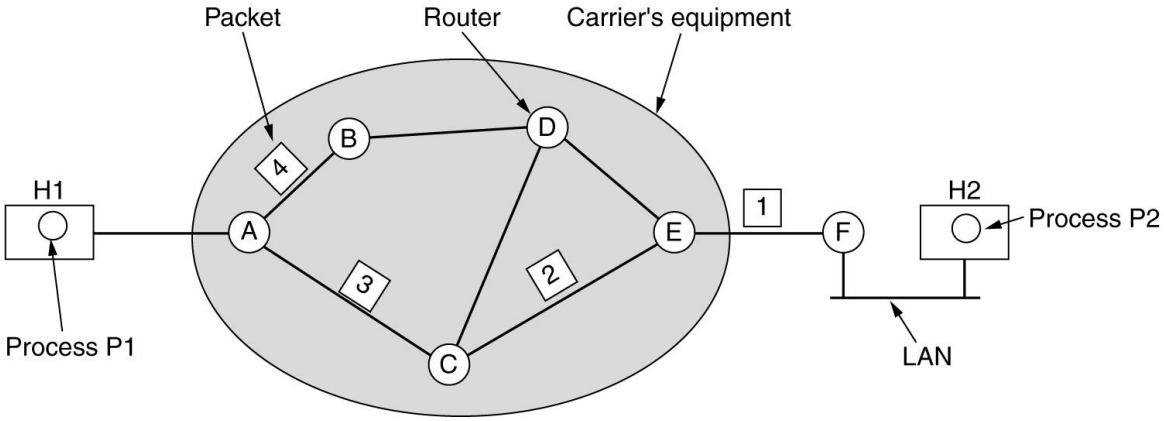
Envío de paquetes entre dos hosts

1. **Servicio no orientado a la conexión (cont):**
   - **Tabla del enrutador**
     - La tabla de enrutamiento solo necesita entradas para los enrutadores de la subred.
     - **entrada de tabla** de enrutador formada por filas:

       <enrutador de destino, línea de salida>

       - La línea de salida es la dirección de un enrutador.

Servicio no orientado a la conexión



A's table

Dest.	Line (initially)
A	–
B	B
C	C
D	B
E	C
F	C

later

Dest.	Line
A	–
B	B
C	C
D	B
E	B
F	B

C's table

Dest.	Line
A	A
B	A
C	–
D	D
E	E
F	E

E's table

Dest.	Line
A	C
B	D
C	C
D	D
E	–
F	F

tabla de reenvío
enrutadores destino líneas de salida
.....

Cuando llega un paquete a un enrutador:

1. Se lo almacena y se comprueba que llegó bien
2. Se determina el enrutador de destino asociado al host de destino (se usa la tabla de reenvio)
3. Se usa fila de ese enrutador de destino para reenviar el paquete por línea de salida de esa fila.

Servicio no orientado a la conexión

- Podemos pensar que **dirección de un host** es un número con dos partes:
  - <dirección de red, número de máquina>
  - La **dirección de red** sirve para identificar una red (p.ej. una red local).
  - El **número de máquina** sirve para identificar una máquina dentro de la red.
  - P.ej: direcciones de 8 bits, red de 4 máquinas viene dada por las direcciones: 11010000, 11010001, 11010010 y 11010011. (dirección de red es 110100)
  - IP respeta esta convención pero con direcciones de 32 bits.

Servicio no orientado a la conexión

- Podemos pensar que los enrutadores que ***están conectados a hosts de una misma red*** (puede ser por medio de un conmutador), que también forman parte de esa red.
  - O sea tienen el mismo valor en la parte de dirección de red (que las máquinas de esa red).
  - P.ej. asumimos que en el ejemplo anterior 11010011 es dirección de un enrutador conectado a red local con máquinas para las demás direcciones que corresponden a hosts.
- Todo host de destino va a tener un enrutador con igual dirección de red y hay que usar ese enrutador de destino para llegar al host de destino.

Servicio no orientado a la conexión

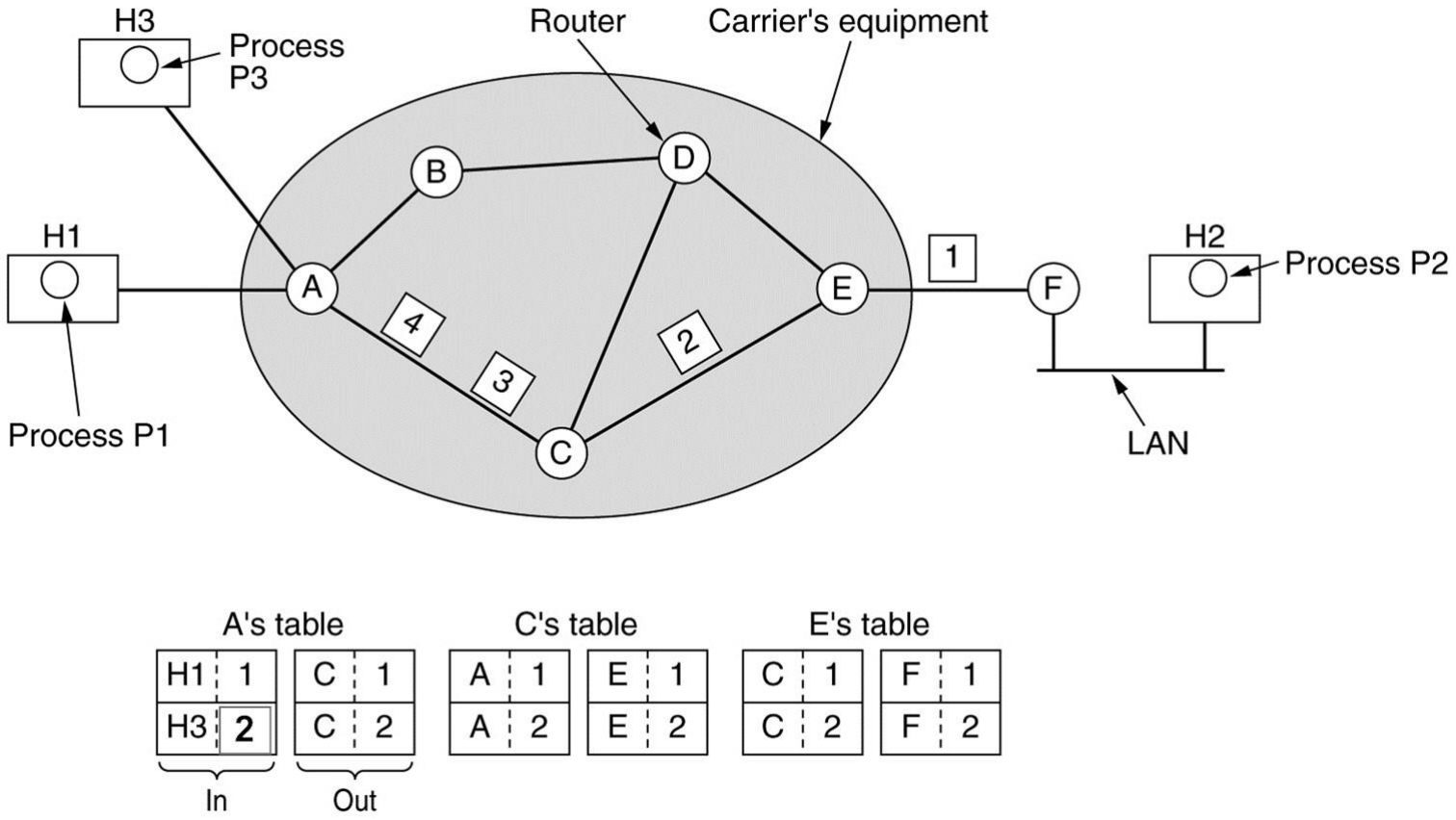
- Dada dirección de host de destino, para ***encontrar enrutador de destino apropiado***:
  - Buscar enrutador de destino cuya dirección concuerde con la mayor cantidad de bits desde la izquierda con la dirección de host de destino.
- P.ej. en el ejemplo anterior, suponer que tenemos enrutadores de destino de direcciones: 10010000, 11000001, 11110010 y 11010011. Suponemos que nos llega un paquete dirigido a host de destino 11010010.
  - El enrutador de destino que concuerda en más bits con ese host de destino es: 11010011 y ese es el que se va considerar para llegar al host de destino.

Envío de paquetes entre dos hosts

2. Servicio orientado a la conexión

- Alentado por las **compañías telefónicas** (p.ej. ATM)
- Todos los paquetes se mandan por la misma ruta.
- **Trabajo a realizar antes de mandar paquetes:**
  - Hay que ***configurar*** una ruta del host de origen al de destino.
  - Esto se llama crear una **conexión.**
  - **Circuito virtual (CV)** = conexión
- Cada paquete lleva un **identificador** que indica a cual CV pertenece.
- Cuando no se necesita enviar más paquetes se **libera la conexión.** Al hacer eso también se termina el CV.

Servicio orientado a la conexión



- Se elige una **ruta** de la máquina de origen a la de destino
  - Esta ruta se almacena en **tablas** dentro de los enrutadores.

Comparación de Subredes de Circuitos Virtuales y Subredes de Datagramas

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Aprenderemos

- **Agenda:**
  1. Cómo es el hardware subyacente a la CR
  2. Enfoques para envío de paquetes entre dos hosts
  3. **Enrutamiento Jerárquico.**
     - **Para entender cómo organizar tablas de enrutamiento en redes muy grandes.**
  4. Cómo es la arquitectura de un enrutador
  5. Conceptos básicos para algoritmos de enrutamiento
  6. Algoritmo de enrutamiento de caminos más cortos.

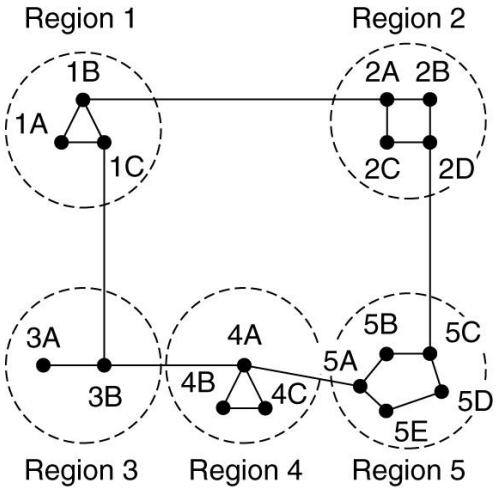
Enrutamiento Jerárquico

- **Cuando crece mucho el tamaño de las subredes**
  - También lo hacen las tablas de enrutamiento
- **Consecuencias de tener tablas de enrutamiento grandes:**
  - Estas tablas consumen memoria del enrutador, necesitan más tiempo de CPU para examinarlas.

Enrutamiento Jerárquico

- **Problema: ¿Cómo hacer para que las tablas de enrutamiento no crezcan demasiado cuando crece mucho el tamaño de la subred?**
- **Solución: enrutamiento jerárquico**
  - Los enrutadores se dividen en **regiones.**
  - Un enrutador sabe cómo enrutar paquetes a destinos en su región.
  - También sabe cómo enrutar a otras regiones.
  - Pero no sabe nada de la estructura interna de las regiones en las que no está.
- **Precio a pagar con enrutamiento jerárquico:** una *longitud de ruta mayor* (no se puede aspirar a encontrar la mejor ruta).
- **¿Cómo son las tablas para los enrutadores jerárquicos?**
- **¿Cómo nombrar los enrutadores y las regiones en una tabla?**

Enrutamiento Jerárquico



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Best choice to reach nodes in 5 except for 5C

- **Tablas de enrutamiento jerárquico**
  - Entradas para todos los enrutadores locales.
  - Entradas para las las demás regiones en las que no está el enrutador
- **Importante**: Observar cómo se nombran las regiones y los enrutadores.
- Cuando estudiemos internet veremos otra manera de asignar nombres.

Enrutamiento Jerárquico

- **Problema**: en las redes enormes, una jerarquía de dos niveles es insuficiente;
- **Solución**: agrupar las regiones en clústeres, los clústeres en zonas, las zonas en grupos, etc.

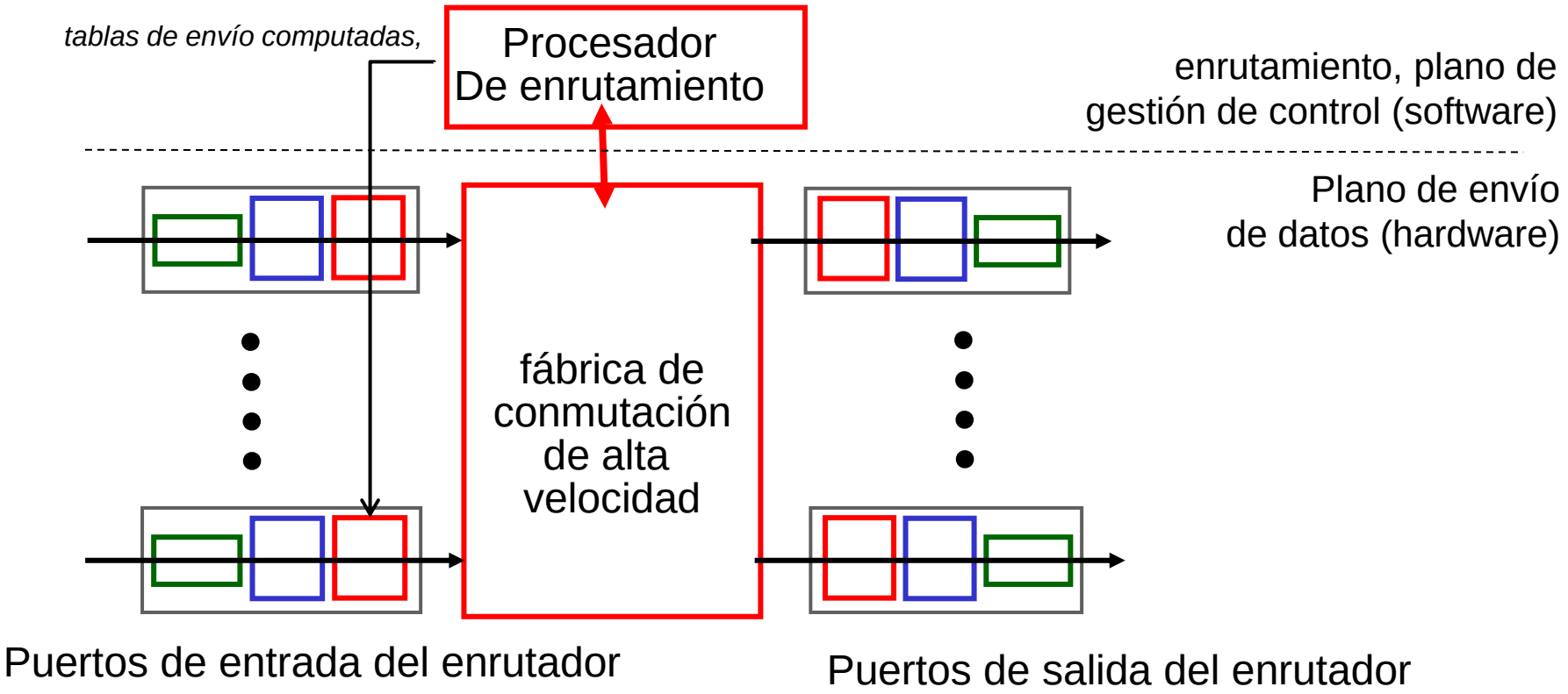
Aprenderemos

- **Agenda:**
  1. Cómo es el hardware subyacente a la CR
  2. Enfoques para envío de paquetes entre dos hosts
  3. Enrutamiento Jerárquico.
  4. **Cómo es la arquitectura de un enrutador**
     - **Para entender funcionamiento de enrutador.**
  5. Conceptos básicos para algoritmos de enrutamiento
  6. Algoritmo de enrutamiento de caminos más cottos

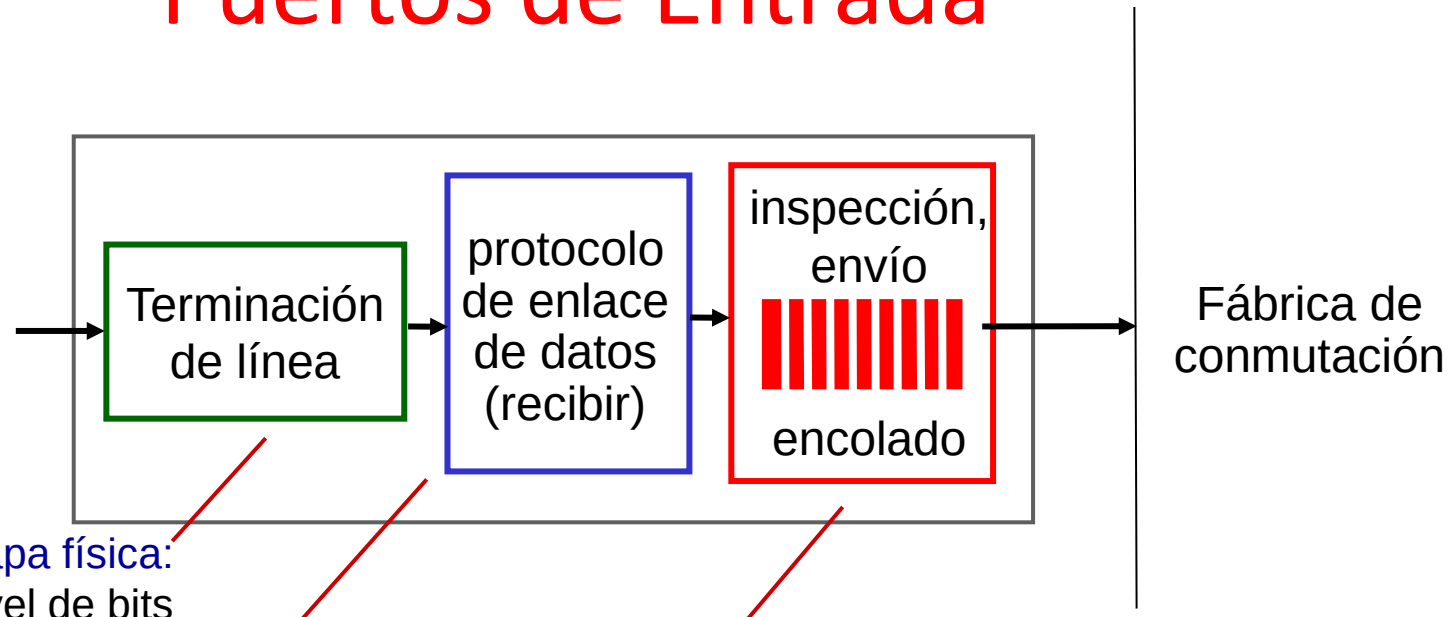
Arquitectura de un Enrutador

Funciones clave de un enrutador:

- Ejecutar algoritmos de enrutamiento/protocolos (RIP, OSPF, BGP)
- *Enviar* paquetes de enlaces de ingreso a enlaces de salida



Puertos de Entrada



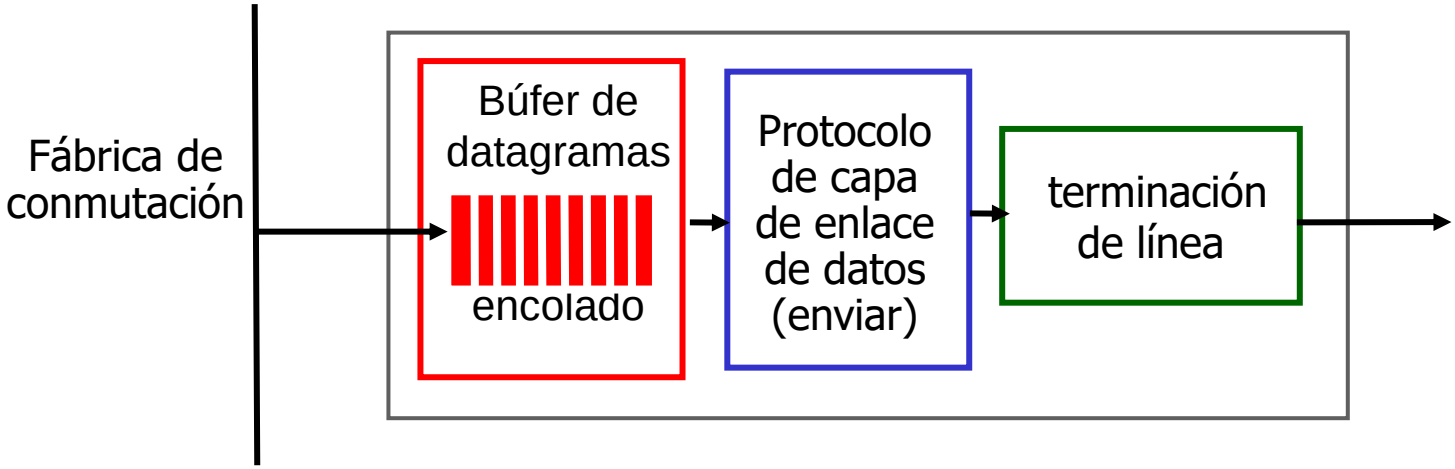
Capa física:
Recepción a nivel de bits

Capa de enlace de datos:
e.g., Ethernet

Conmutación descentralizada:

- Dado paquete, busca puerto de salida usando tabla del enrutador
- **meta**: procesamiento completo del input a la "velocidad de la línea".
- **encolado**: si los paquetes arriban más rápido que la tasa de envío en la fábrica de conmutación

Puertos de Salida



- *Encolado* requerido cuando los paquetes llegan de la fábrica de conmutación más rápido que la tasa de transmisión.

Paquetes pueden perderse debido a congestión, carencia de búferes

- *Disciplina de planificación* elige entre los paquetes encolados para transmisión.

Planificación por prioridades – ¿quién logra el mejor desempeño?

Aprenderemos

- **Agenda:**
  1. Cómo es el hardware subyacente a la CR
  2. Enfoques para envío de paquetes entre dos hosts
  3. Cómo es la arquitectura de un enrutador
  4. Enrutamiento Jerárquico.
  5. **Conceptos básicos para algoritmos de enrutamiento**
     - **Para entender el por qué de su necesidad y qué es lo que buscan.**
     - **Para entender cómo representar matemáticamente una subred.**
  6. Algoritmo de enrutamiento de caminos más cortos.

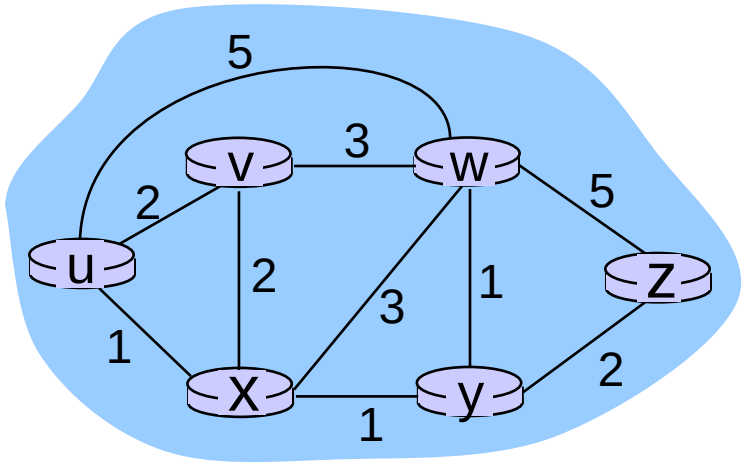
Algoritmos de enrutamiento

- **Meta: queremos evitar los siguientes efectos indeseados:**
  - Algunos enrutadores pueden quedar inactivos.
  - Los caminos pueden ser innecesariamente largos.
  - Se pueden sobrecargar algunas de las líneas de comunicación y los enrutadores asociados a ellas.
- **Causa de ellos**: la CR ***elige mal*** las rutas para enviar paquetes.
- **Problema: ¿Cómo escoger bien las rutas para enviar paquetes?**
- **Solución**: Usar **algoritmos de enrutamiento *eficientes.***
  - Un algoritmo de enrutamiento se ejecuta en los **enrutadores** de la subred;
  - es responsable de llenar y actualizar las tablas de enrutamiento

Abstracción de Grafo

- Antes de comenzar con algoritmos de enrutamiento vemos algunos conceptos.
- **¿Cómo representar una subred como un grafo?**

Abstracción de Grafo



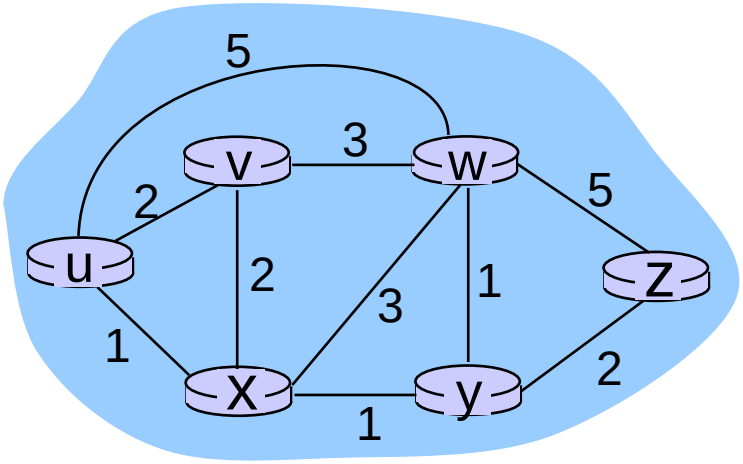
Grafo: G = (N,E)

N = conjunto de enrutadores = { u, v, w, x, y, z }

E = conjunto de enlaces ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Los arcos tienen **etiquetas** para el ***costo*** de atravesarlos

Abstracción de Grafo: Costos



c(x,x’) = costo de enlace (x,x’)
e.g., c(w,z) = 5

Los costos de los arcos podrían calcularse como función de varios parámetros:

- La distancia, ancho de banda, tráfico medio, costo de comunicación, longitud media de las colas, retardo medio, y otros factores.

Cálculo del costo de un camino:

costo del camino $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

Aprenderemos

- **Agenda:**
  1. Cómo es el hardware subyacente a la CR
  2. Enfoques para envío de paquetes entre dos hosts
  3. Cómo es la arquitectura y funcionamiento de un enrutador
  4. Enrutamiento jerárquico
  5. Conceptos básicos para algoritmos de enrutamiento
  6. **Algoritmo de enrutamiento de caminos más cortos**

Enrutamiento de caminos más cortos

- **Algoritmo de enrutamiento de caminos más cortos**:
  - Para elegir una ruta entre un par de enrutadores, encontrar en el grafo una de **las rutas más cortas** entre ellos.
- Algoritmos de cálculo de la ruta más corta entre dos nodos.
  - Uno de ellos es el algoritmo de **Dijkstra** (1959).
    - Dado grafo conexo con costos en los enlaces, y nodo *n* en el grafo, obtiene **árbol de caminos más cortos** desde *n* hacia todos los demás nodos.
    - El árbol de caminos más cortos se representa con un **mapeo** donde para cada nodo del grafo de la subred asigna su padre (en el árbol de caminos más cortos).
    - Repasar los detalles del algoritmo de Dijkstra visto en algoritmos 2.

Enrutamiento de caminos más cortos

- **Procedimiento para calcular tablas de reenvío en redes de datagramas usando algoritmo de Dijtstra.**
  1. Construir grafo de la subred con costos.
  2. Ingresar grafo de la subred con costos en los enrutadores.
  3. En cada enrutador construir tabla de enrutamiento; para eso:
     a. Ejecutar algoritmo de Dijkstra en el enrutador
     b. A partir de árbol de caminos más cortos con raíz en el enrutador obtenido generar la tabla de reenvío del enrutador.