

Tema 2

Sistemas de Conmutación de Circuitos y Paquetes

Sistemas de Conmutación

Grado en Ingeniería de Tecnologías de Telecomunicación
3^{er} Curso

Profesores: **Pablo Ameigeiras Gutiérrez, Pablo Muñoz Luengo**

Departamento de Teoría de la Señal, Telemática y Comunicaciones
E.T.S. Ingenierías Informática y Telecomunicación – Universidad de Granada
C/ Periodista Daniel Saucedo Aranda, s/n - 18071 – Granada (Spain)

Teléfono: +34-958 242306 - Fax: +34-958 243032 - Email: pameigeiras@ugr.es



© pameigeiras



Motivación y Objetivos

Sistemas de Conmutación - Grado en Ing. de Tec. de Telecomunicación - UGR
© Pablo Ameigeiras Gutiérrez

2

Sistemas de Conmutación

V1.2



Motivación

- La telefonía y la conmutación de circuitos tienen una historia de más de 100 años
- La telefonía es hoy en día un servicio esencial en nuestra sociedad ya que permite transferir la señal de voz entre usuarios (en casas, oficinas, tiendas, etc.) geográficamente distantes (incluso en distintos países)
- Internet representa la mayor plataforma para el intercambio de información
- Los conmutadores de paquetes y routers juegan un papel clave en las redes de datos ya que encaminan la información
- Los diseños de conmutadores de paquetes y routers deben adecuarse al entorno de despliegue, permitiendo gran capacidad de conmutación en entornos de gran cantidad de tráfico o bajo coste en entornos de poco tráfico agregado



Motivación y Objetivos

Objetivos

- Los objetivos del presente tema son los siguientes:
 1. Diferenciar entre conmutación de **circuitos** y de **paquetes**
 2. Aprender la estructura de un **conmutador de circuitos**
 3. Aprender las estructuras de conmutación **espaciales** y **temporales** en conmutadores de circuitos
 4. Diferenciar entre conmutación de paquetes **orientada a conexión** y **no orientada a conexión**
 5. Conocer la **estructura básica** de un conmutador de paquetes
 6. Conocer la **redes de interconexión principales** en conmutadores de paquetes
 7. Conocer las **arquitecturas principales de routers IP**
 8. Conocer las **redes definidas por software**



Índice de Contenidos

1. Introducción
2. Conmutación de Circuitos
 1. Conmutación Espacial
 2. Conmutación Temporal
 3. Conmutación Espacial y Temporal
3. Conmutación de Paquetes
 1. Arquitectura de Conmutadores
 1. Arquitectura por División Espacial
 2. Arquitectura de Memoria Compartida
 2. Arquitectura de Routers IP
 3. Redes definidas por software



Bibliografía

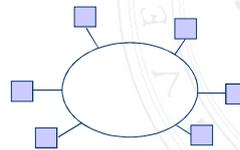
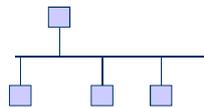
- **J. Bellamy:** *Digital Telephony*, John Wiley & Sons, 2000
- **J.E. Flood:** *Telecommunications Switching, Traffic and Networks*, Pearson Education, 1999
- **M. Schwartz:** *Redes de Telecomunicaciones. Protocolos, Modelado y Análisis*, Addison-Wesley, 1994
- **J. Chao y B. Liu:** *High Performance Switches and Routers*, John Wiley & Sons, 2007
- **J. Chao:** *Broadband Packet Switching Technologies*, John Wiley & Sons, 2001
- **J. Martínez y V. Casares:** *Conmutadores de Paquetes, Arquitectura y Prestaciones*, Editorial Universidad Politécnica de Valencia, 2001
- **William Stallings, et al.:** *Foundations of modern networking: SDN, NFV, QoE, IoT, and Cloud*, Pearson Education, 2016



1 Introducción

■ Tipos de Redes

- **Redes de Difusión:** existe un único medio de transmisión que comparten todas las estaciones, de manera que la información transmitida por una de ellas está accesible para todas las demás

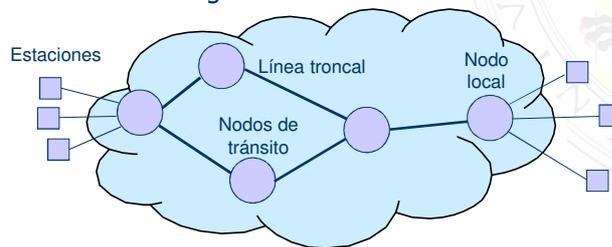


- **Redes Dedicadas** o Punto a Punto: se usa cada enlace para comunicar únicamente dos estaciones



1 Introducción

- **Redes Conmutadas:** consisten en un conjunto de líneas de transmisión y nodos, donde la información se transfiere encaminándola desde la estación origen a la destino mediante la conmutación en los nodos intermedios
- Se define la **conmutación** en un nodo a la conexión, física o lógica, de un camino de entrada con un camino de salida del nodo con el fin de transferir la información que llegue por el primer camino al segundo



1 Introducción

Tipos de Conmutación:

- **Conmutación de Circuitos:** se establece un camino de comunicación entre el terminal origen y el destino de manera que dicho camino se reserva de manera exclusiva para esta comunicación
 - El camino puede ser un camino físico o bien una ranura temporal (time slot)
 - Ejemplo: la red telefónica
 - Una vez establecida la conexión, la principal fuente de retardo de la comunicación es el tiempo de transmisión y propagación siendo despreciable el retardo introducido por cada nodo de la ruta
 - Esta técnica de conmutación se desarrolló originalmente para tráfico de voz
 - No es apropiada para la transmisión de información a velocidad variable (pej. tráfico de datos en ráfagas) ya que los recursos están reservados durante todo el tiempo que dure la comunicación



1 Introducción

- **Conmutación de Paquetes:** La información se fragmenta (por ejemplo en paquetes de 1500 octetos) que se entregan a la red de manera que el camino seguido por dichos paquetes puede ser compartido por varias comunicaciones
- Cada paquete contiene una parte de datos (información) más cierta información de control (cabecera)
- La red utiliza la información de control (cabecera) de cada paquete para encaminarlo a través de los distintos nodos
- En cada nodo, los paquetes se reciben, se almacenan temporalmente, se encaminan hacia el siguiente nodo y se envían
 - No existe una reserva estática de recursos
 - Los paquetes en cola se transmiten lo más rápido posible (a la máxima velocidad de la línea)
 - Hacen falta buffers para almacenar los paquetes en cada nodo



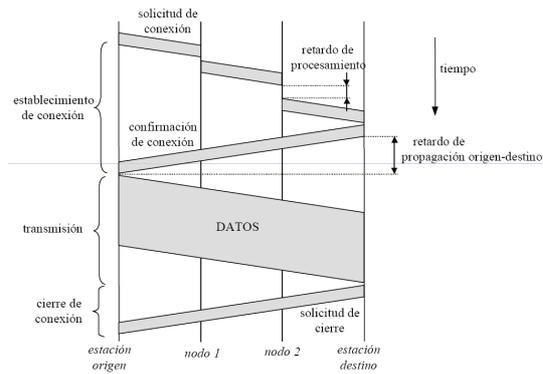
1 Introducción

- **Conmutación de Paquetes:**
 - Cada estación se conecta a su nodo con su propia velocidad
 - Se puede hacer uso de prioridades
 - Cuando la red está sobrecargada, los paquetes se pueden seguir aceptando (aunque aumenta el retardo en la transmisión)
 - Ejemplo: Internet



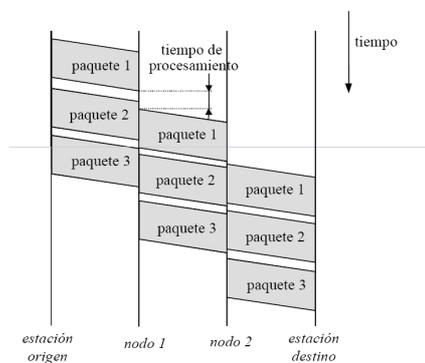
1 Introducción

■ Conmutación de Circuitos



1 Introducción

■ Conmutación de Paquetes





2 Conmutación de Circuitos



Conmutación de Circuitos:

- Para el establecimiento de una comunicación de conmutación de circuitos se requieren tres fases:
 - Establecimiento de la conexión:** se establece un circuito (secuencia de enlaces conectados entre nodos entre origen y destino). Esta fase implica un retardo ya que se ha de encontrar el camino
 - Transferencia de datos:** después del establecimiento del circuito se transmite la información (analógica o digital) a través del camino establecido. Los nodos operan de forma transparente para la comunicación transmitida.
 - Liberación de la conexión:** la conexión finaliza por orden de la estación de origen o destino. Los recursos dedicados a esta comunicación se liberan
- Una vez establecido el circuito en la red, la transferencia es transparente para los usuarios



2 Modulación por Impulsos Codificados



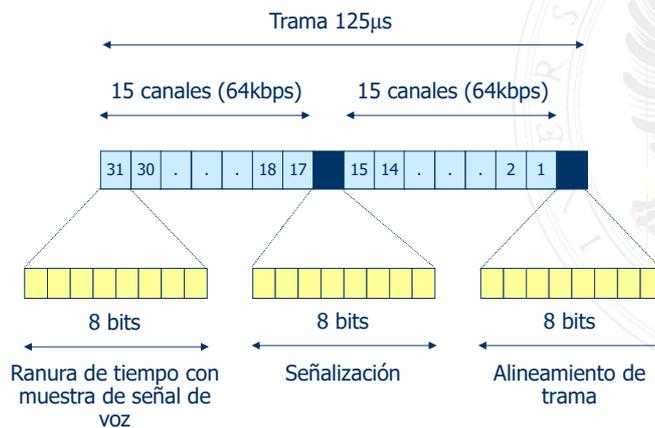
Modulación por Impulsos Codificados

- El método más común en telefonía fija para la representación digital de la señal de voz (analógica) es el denominado MIC \equiv Modulación por Impulsos Codificados (o PCM \equiv Pulse Code Modulation)
- Un canal telefónico ocupa el ancho de banda 300-3400 Hz, y típicamente se aplica una frecuencia de muestreo de 8 kHz
- Cada muestra de la señal de voz se codifica con 8 bits, con lo que un canal típico de voz PCM requiere una capacidad de transmisión de 64kbps
- El estándar europeo combina 30 canales de voz, más dos de señalización en cada trama de 125 μ s resultando en una tasa de bit total de 2.048Mbps
- El estándar americano (denominado T1) combina 24 canales de voz de 64kbps por cada trama de 125 μ s



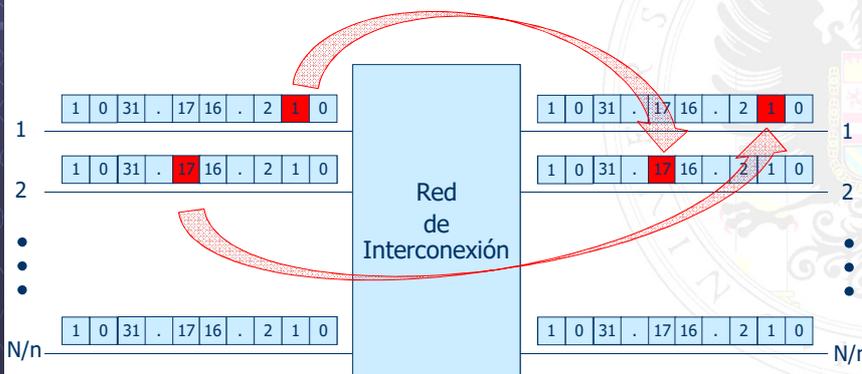
2 Trama MIC

Formato de Trama MIC de 30 Canales



2 Conmutación Espacial y Temporal

Conmutación Espacial y Temporal:

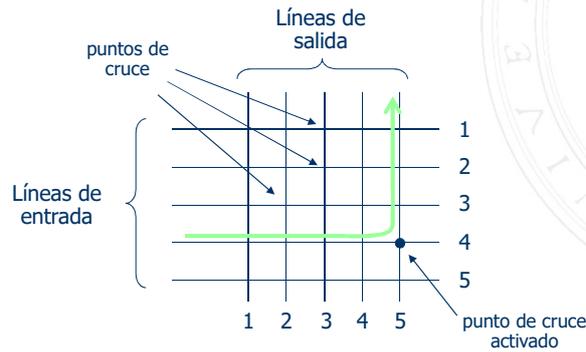




2.2 Conmutación Espacial

Conmutación Espacial:

- Las conexiones del conmutador entre las líneas de entrada y de salida se pueden llevar a cabo por medio de puntos de cruce en una matriz de conmutación de tamaño $N \times N$



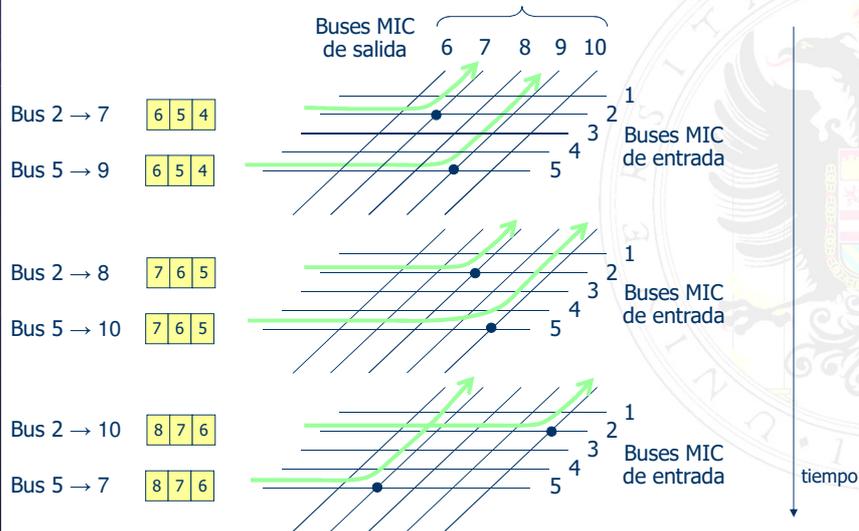
2.2 Conmutación Espacial

- Se deben habilitar los circuitos (puertas lógicas en conmutador digital o contactos metálicos en conmutador analógico) en cada uno de los puntos de cruce donde se intersecan las líneas de entrada y salida estableciéndose así la conexión
- En la conmutación espacial de señales digitales, un punto de cruce es compartido por distintas comunicaciones en distintas ranuras temporales
- Cada una de las líneas de entrada y de salida al conmutador es un sistema MIC
- La matriz de conmutación se va reestructurando en cada ranura temporal para habilitar los puntos de cruce en los distintos instantes de tiempo



2.2 Conmutación de Circuitos Digital

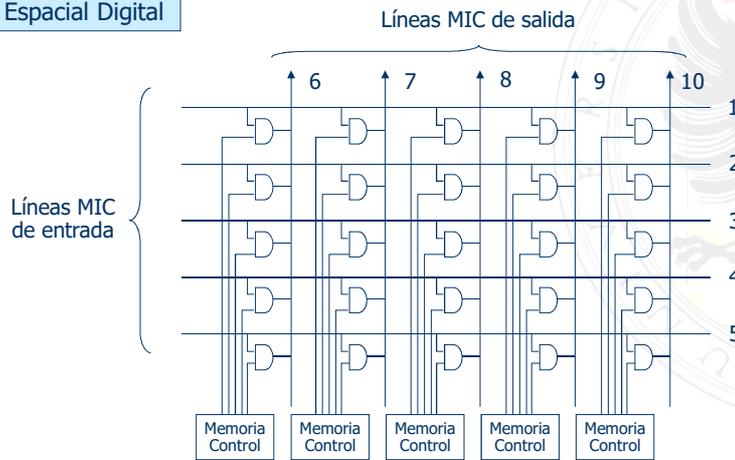
Sistemas de Conmutación - Grado en Ing. de Telecomunicación - UGR
© Pablo Ameigiras Gutiérrez



2.2 Conmutación de Circuitos Digital

Sistemas de Conmutación - Grado en Ing. de Telecomunicación - UGR
© Pablo Ameigiras Gutiérrez

Conmutador Espacial Digital





2.2 Conmutación de Circuitos Digital

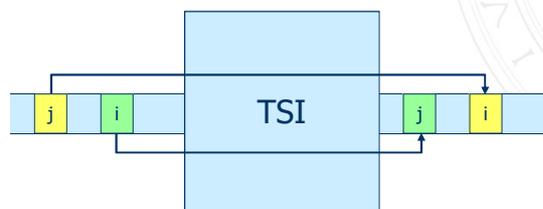
- Para activar el punto de cruce que permite la conexión entre las líneas de entrada y de salida deseadas, se lee la posición de la memoria de control correspondiente a la ranura temporal actual
- La lectura de las posiciones de la memoria de control se realiza de manera cíclica siendo la duración del ciclo precisamente la duración de la trama MIC
- La conexión de vuelta se puede llevar a cabo en la misma matriz de conmutación reorganizando las líneas de entrada y salida (así se evita añadir una matriz de conmutación para cada sentido de la comunicación)
- Un conmutador digital sólo permite establecer conexiones entre abonados en las mismas ranuras de tiempo de entrada y salida



2.3 Conmutación Temporal

Conmutación Temporal

- Consiste en cambiar el orden de las ranuras temporales entre la entrada y la salida de una misma línea MIC
- Los dispositivos de este tipo se denominan TSI (Time Slot Interchange)





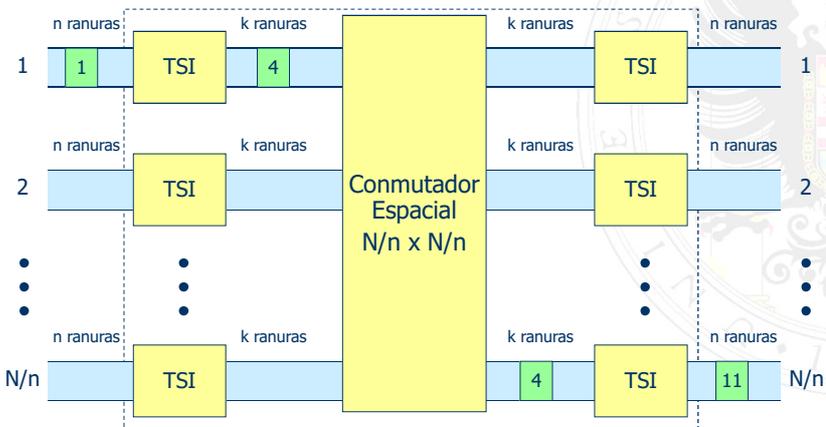
2.4 Conmutación Espacial y Temporal

- Conmutación Espacial y Temporal
 - Un conmutador compuesto únicamente por una etapa temporal tiene una capacidad de conmutación limitada
 - Un conmutador compuesto únicamente por una etapa espacial sólo permite establecer conexiones entre distintas líneas de una misma ranura
 - Los sistemas de conmutación grandes (gran número de líneas) incorporan en su redes etapas de espacio y tiempo en tandem
 - Existen distintos tipos de arquitecturas de conmutadores en las que varían tanto el número de etapas como la organización de dichas etapas



2.4 Conmutación TST

Red de Conmutación TST





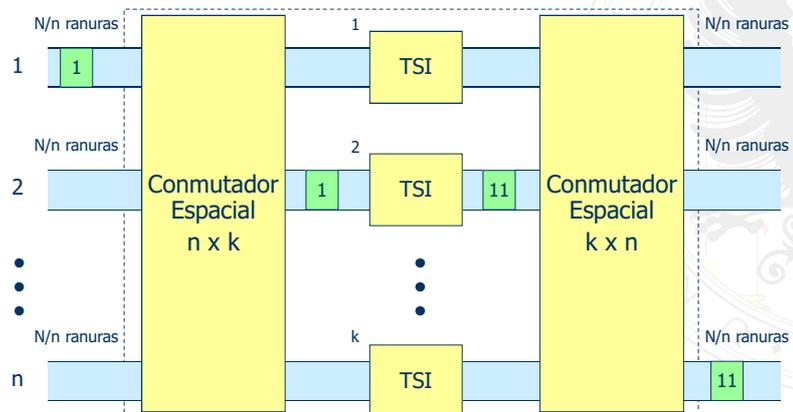
2.4 Conmutación TST

- La etapa espacial de la red de conmutación TST es la encargada de llevar a cabo el intercambio entre las N/n líneas MIC de entrada y de salida
- La disposición de los puntos de cruce de la etapa de conmutación espacial cambia en cada una de las k ranuras temporales
- Para hacer la conmutación en la etapa espacial hay que escoger una ranura libre de entre las k existentes que permita el encaminamiento entre las N/n líneas MIC de entrada y salida deseadas
- Si dicha ranura libre no existe, se produce bloqueo entre las dos etapas temporales
- Además, si $k < n$ también se puede producir bloqueo en la etapa temporal de entrada
- Si $k \geq 2n-1$ se obtiene una red de conmutación sin bloqueo



2.4 Conmutación STS

Red de Conmutación STS





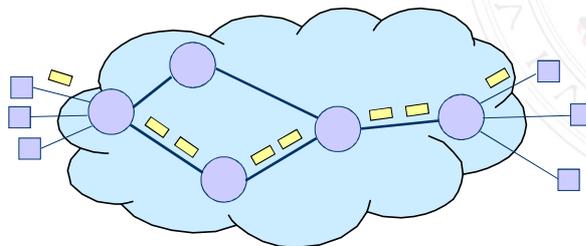
2.4 Conmutación STS

- La etapa temporal de la red de conexión STS es la encargada de llevar a cabo el intercambio entre las ranuras temporales de entrada y de salida
- Para establecer la conexión entre la entrada y la salida es necesario seleccionar un TSI de entre los k disponibles que tenga libres las ranuras temporales de entrada y salida
- Se produce bloqueo si no se encuentra un TSI que tenga libres dichas ranuras temporales de entrada y salida
- De nuevo si $k \geq 2n-1$ se obtiene una red de conmutación sin bloqueo



3 Conmutación de Paquetes: Circuito Virtual

- **Conmutación de Paquetes: Circuito Virtual**
 - En el servicio orientado a conexión, al principio de la comunicación, se fija una ruta, estableciendo una conexión lógica entre origen y destino
 - Se establece una ruta fija a través de los nodos intermedios para todos los paquetes que se intercambian durante la duración de la conexión





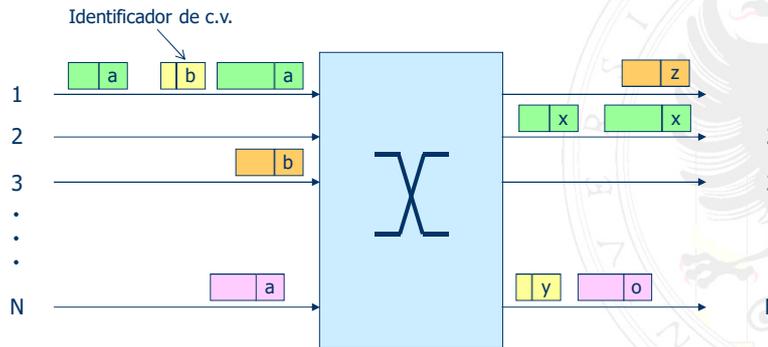
3 Conmutación de Paquetes: Circuito Virtual

- No obstante, los datos van multiplexados con otros pertenecientes a otros c.v. compartiendo por tanto las vías de comunicación
- En cada nodo intermedio se crea una entrada a la **tabla de circuitos virtuales** para indicar la ruta de la conexión establecida
- Cada paquete puede usar una pequeña cabecera que indique únicamente a qué c.v. pertenece (en vez de tener que incluir la dirección completa del destino)
- Los nodos procesan cada uno de los paquetes acorde a la información que se almacenó cuando se estableció la conexión
- No es necesaria la toma de decisiones de encaminamiento para cada paquete
- Hacen falta un conjunto de paquetes tanto para establecer la conexión como para liberarla
- Se pueden proporcionar mejoras para garantizar la fiabilidad de la comunicación. La entrega de paquetes en secuencia y libre de errores se garantiza, y además es típico el control de congestión para evitar el excesivo retardo de espera de las colas de los nodos
- Las redes más típicas que operan en modo de circuito virtual son MPLS y ATM



3 Conmutación de Paquetes: Circuito Virtual

Operación de un Conmutador de Circuito Virtual



I	VCI	O	VCI
1	a	2	x
1	b	n	y
3	b	1	z
n	a	n	o

Tabla de Circuitos Virtuales

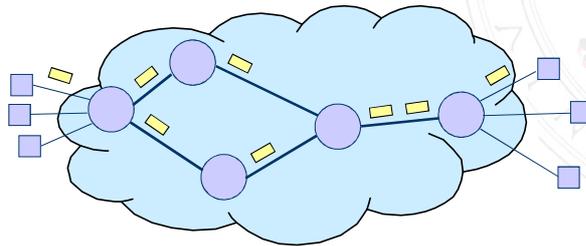




3 **Conmutación de Paquetes: Datagrama**

Conmutación de Paquetes: Datagrama

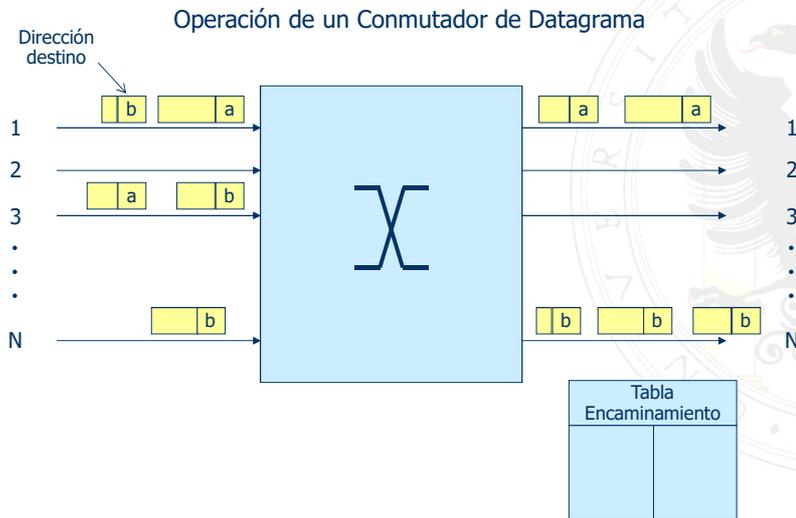
- En el servicio no orientado a conexión, cada paquete viaja de manera independiente entre el origen y el destino
 - No se establece conexión
 - Los paquetes llevan la dirección de las estaciones origen y destino



3 **Conmutación de Paquetes: Datagrama**

- Los nodos intermedios encaminan los paquetes en función no sólo de la dirección de destino, sino también en función de información del estado del tráfico en nodos adyacentes, fallos de líneas, etc.
- Los paquetes pueden llegar desordenados al destino
- Tampoco se garantiza que lleguen todos los paquetes
- Dependiendo del tipo de red, tanto la ordenación de los paquetes como la recuperación de los paquetes perdidos la pueden llevar a cabo bien el nodo de salida o bien la estación final
- En esta técnica los paquetes se denominan datagramas

3 **Commutación de Paquetes: Datagrama**



3.1 **Arquitectura de Conmutadores Ethernet**

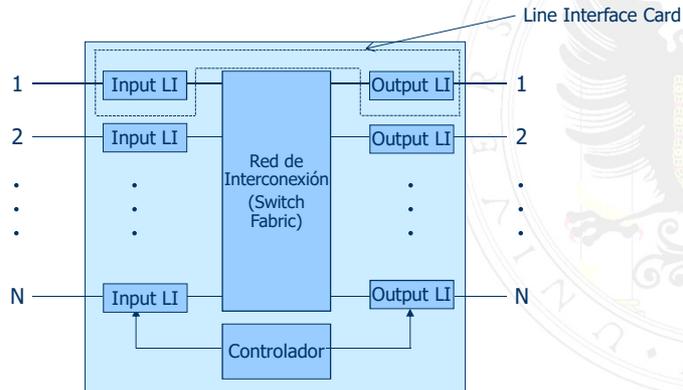
Arquitectura de Conmutadores (Ethernet)

- Un conmutador Ethernet debe reenviar una trama Ethernet en función de:
 - El campo *Destination Address* de la trama
 - La tabla de direcciones MAC: tiene una correspondencia entre la dirección MAC y el puerto de salida por el que reenviar la trama
- En Ethernet, la trama tiene un tamaño variable, en función del tamaño del paquete de nivel superior que se encapsula en el campo de datos
- Un conmutador Ethernet está compuesto de varios bloques funcionales:
 - ◆ Interfaz de línea de entrada
 - ◆ Interfaz de línea de salida
 - ◆ Red de interconexión
 - ◆ Controlador



3.1 *Arquitectura de Conmutadores*

Arquitectura de Conmutador



3.1 *Arquitectura de Conmutadores*

■ Interfaz de línea de entrada

- Capa PHY: decodificación de la señal y conversión en flujo de bits
- Las funciones de PHY cambian según la tecnología de nivel físico (fibra óptica/par trenzado, tasa de datos, etc).
- Capa MAC: detección y reconstrucción de trama Ethernet, y verificación del campo de control de errores (FCS)
- Proceso de búsqueda (Lookup) de la dirección MAC en una tabla de reenvío para hallar el puerto de destino
- El proceso de búsqueda ha de operar a la velocidad de la línea \Rightarrow uso de tablas *CAM* (Content Addressable Memory) las cuales son direccionadas en función del contenido en vez de la dirección de memoria
- En caso de multi-cast, la trama ha de ser reenviada por múltiples puertos
- Proceso de aprendizaje de dirección MAC origen
- Control de flujo en recepción



3.1 *Arquitectura de Conmutadores*

■ Interfaz de línea de salida

- Capa PHY: codificación del flujo de bits en señal eléctrica/óptica
- Capa MAC: si el medio es compartido (semi-duplex) ⇒ procedimiento de acceso múltiple (p.ej. CSMA/CD)
- Si no es compartido (full-duplex) ⇒ no hay acceso múltiple
- Si la trama ha cambiado (p.ej. VLAN) es necesario recalcular el campo de control de errores (FCS)
- Control de flujo en transmisión



3.1 *Arquitectura de Conmutadores*

■ Red de interconexión (o matriz de conmutación)

- proporciona la interconexión necesaria para la conmutación entre las interfaces de línea de entrada y salida
- almacena temporalmente las tramas cuando existe conflicto en el acceso a un puerto de salida
- descarta selectivamente
- realiza múltiples copias de tramas si fuera necesario

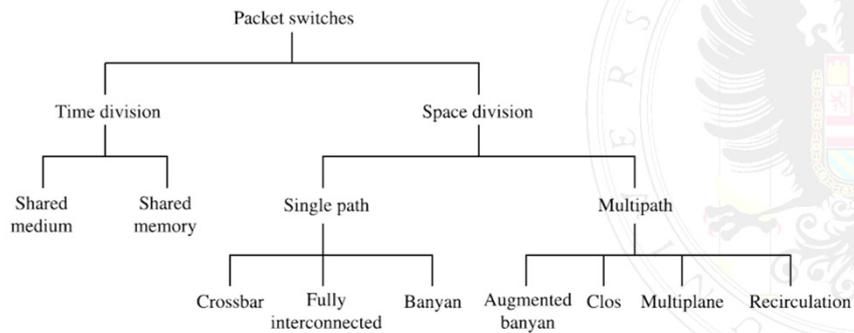
■ Controlador

- Gestión del switch (implementación de telnet, SNMP, interfaz de usuario, protocolo TCP/IP para las conexiones de gestión)
- Diagnostico interno y mantenimiento (inicialización del conmutador, rutinas de diagnostico, posibilidad de actualizar la ROM, etc.)
- Ejecución de protocolos del plano de control (p.ej. STP, Link Aggregation Control Protocol)



3.1 *Arquitectura de Conmutadores*

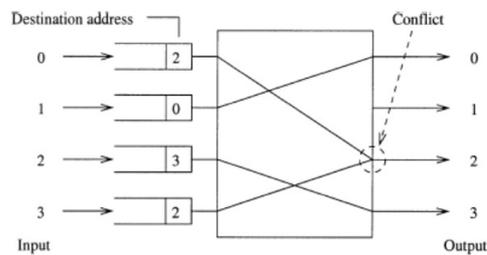
Clasificación de Arquitecturas de Redes de Interconexión



3.1.1 *Conceptos Básicos de Conmutadores de Paquetes*

Contención de un puerto de salida:

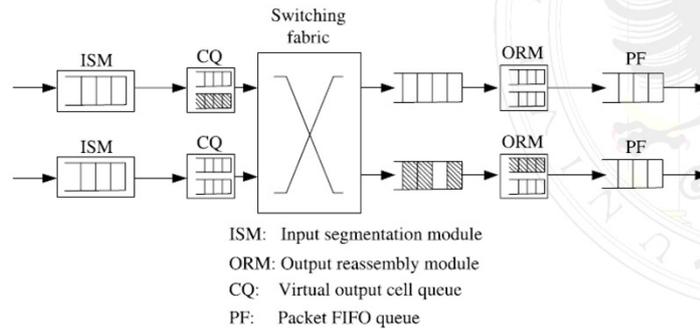
- Sucede cuando múltiples células que acceden al conmutador por diferentes puertos de entrada tienen como destino el mismo puerto de salida



3.1.1 Conceptos Básicos de Conmutadores de Paquetes

Comutación en modo de celda o modo paquete:

- Las longitudes variables de las tramas (o paquetes) complica el diseño de la Red de Interconexión.
- Un método tradicional consiste en segmentar las tramas (o paquetes) en celdas de tamaño fijo



3.1.1 Conceptos Básicos de Conmutadores de Paquetes

Principales indicadores de rendimiento (no comerciales):

- Throughput (o caudal):** número de bits que la Red de Interconexión reenvía con éxito por unidad de tiempo. El éxito del reenvío hace referencia al hecho de que el paquete alcance el puerto de salida deseado. El cálculo se realiza agregado a todas las líneas de salida. Se puede mostrar de forma porcentual.
- Retardo:** tiempo transcurrido desde que se recibe el último bit del paquete por un puerto de entrada hasta que es retransmitido el último bit del paquete por un puerto de salida. El retardo se produce, principalmente, cuando la célula es almacenada en una cola
- Probabilidad de pérdida interna:** número de paquetes perdidos dividido por el número de paquetes que acceden al conmutador



3.1.1 Conceptos Básicos de Conmutadores de Paquetes

Principales especificaciones técnicas (comerciales):

- **Bandwidth (o ancho de banda):** tasa máxima de bit de la Red de Interconexión.
Figura comercial (a veces su cálculo no es muy preciso).
Incluye la tasa de enlaces de entrada y salida
- **Latencia:** tiempo transcurrido desde que se recibe un paquete por un puerto de entrada hasta que es retransmitido por un puerto de salida bajo la suposición de que el puerto de salida está libre.
Es una medida del retardo interno del conmutador.
- **Base de datos de direcciones MAC:** Número máximo de direcciones que el conmutador puede almacenar



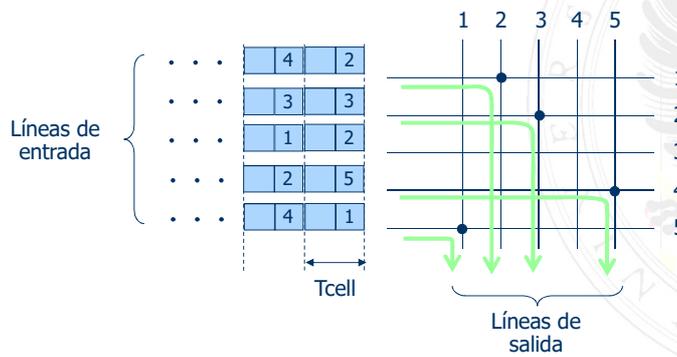
3.1.1 Arquitectura por División Espacial

Conmutadores con Arquitectura por División Espacial

- La interconexión por división espacial más sencilla consiste en una *red de interconexión espacial monoetapa sin memoria* compuesta de una matriz de barras cruzadas de $N \times N$ puntos de cruce
- Supongamos que a la Red de Interconexión llegan celdas de tamaño fijo y sincronizadas en el tiempo



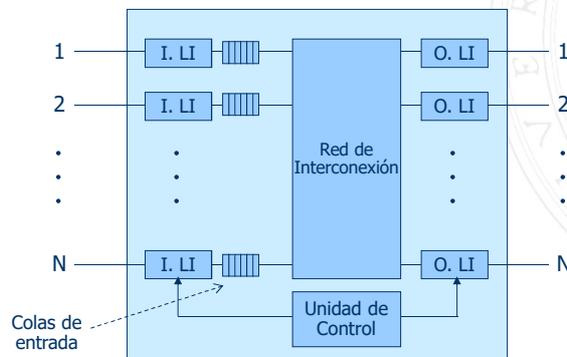
3.1.1 Arquitectura por División Espacial



3.1.1 Arquitectura por División Espacial

Red de Interconexión con memoria a la entrada

- Una red de interconexión espacial y memoria a la entrada se caracteriza por incluir colas en cada uno de los puertos de entrada





3.1.1 *Arquitectura por División Espacial*

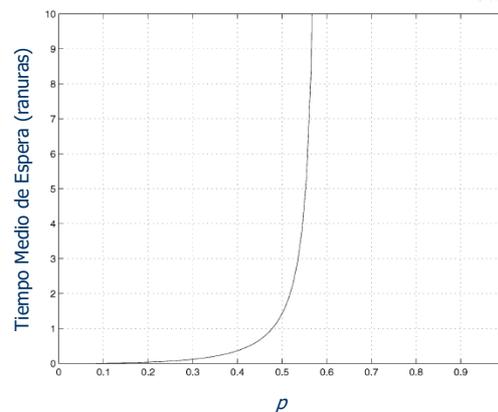
- La red de conexión monoetapa y memoria a la entrada sufre el conocido como **Head of Line (HOL) blocking**
- El bloqueo HOL sucede cuando hay células que no pueden alcanzar un puerto de salida libre porque otras células por delante de ellas en la cola no se pueden transmitir a través de la red de conexión
- Debido al bloqueo HOL, el throughput de esta arquitectura queda significativamente reducido

N	Throughput / N
1	1.0
2	0.75
3	0.6825
4	0.6553
5	0.6399
6	0.6302
7	0.6234
8	0.6184
∞	0.5858



3.1.1 *Arquitectura por División Espacial*

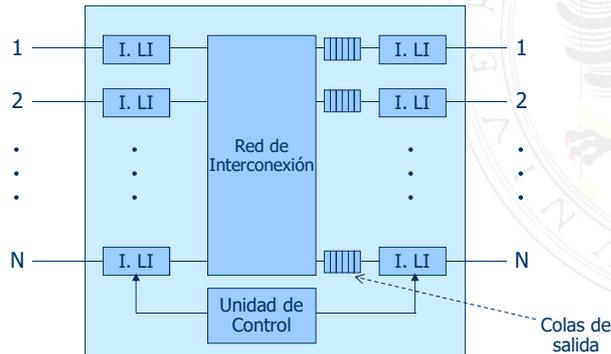
Tiempo Medio de Espera en Conmutador con Memorias FIFO a la Entrada para el Caso $N \rightarrow \infty$



3.1.1 Arquitectura por División Espacial

Red de Interconexión con memoria a la salida

- Una red de interconexión espacial y memoria a la salida consiste en introducir colas en los puertos de salida del conmutador



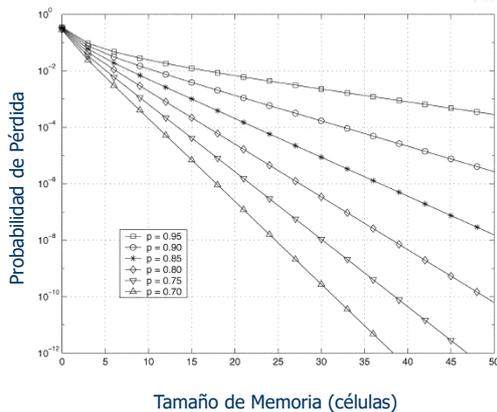
3.1.1 Arquitectura por División Espacial

- La red de interconexión monoetapa y memoria a la salida permite obtener un **throughput del 100%** en el conmutador
- En cambio, dicha red de interconexión **ha de operar N veces más rápido** ya que podría suceder que todas las células de entrada tuvieran que ser encaminadas por el mismo puerto de salida
- Además el buffer de salida ha de almacenar N células en cada instante de tiempo, con lo que la velocidad de la memoria impone un límite máximo en el tamaño del conmutador (N)
- Si el tamaño de las memorias de salida son finitas, se pueden producir pérdidas de células si éstas encuentran los buffers llenos al salir de la red de interconexión



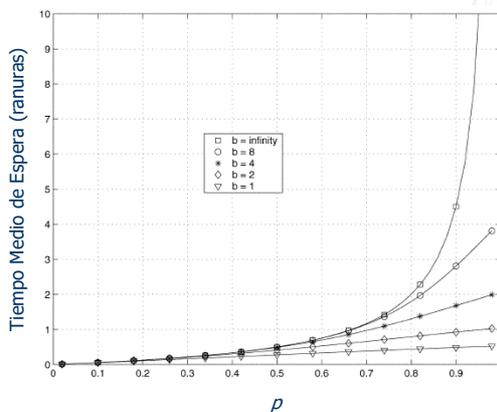
3.1.1 *Arquitectura por División Espacial*

Probabilidad de Pérdida de Células en Conmutador con Memoria a la Salida para el Caso $N \rightarrow \infty$



3.1.1 *Arquitectura por División Espacial*

Tiempo Medio de Espera en Conmutador con Memorias a la Salida para el Caso $N \rightarrow \infty$

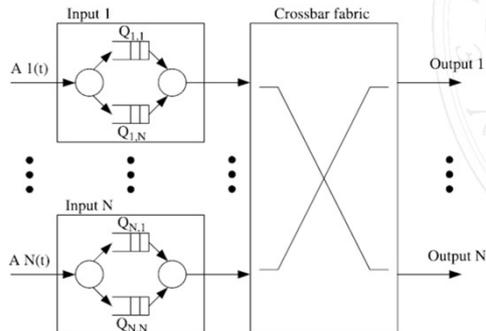




3.1.1 Arquitectura por División Espacial

Red de Interconexión con memorias virtuales de salida

- Una red de interconexión espacial y memorias virtuales de salida (VOQ) consiste en que en cada línea de entrada se mantiene una cola separada para cada salida



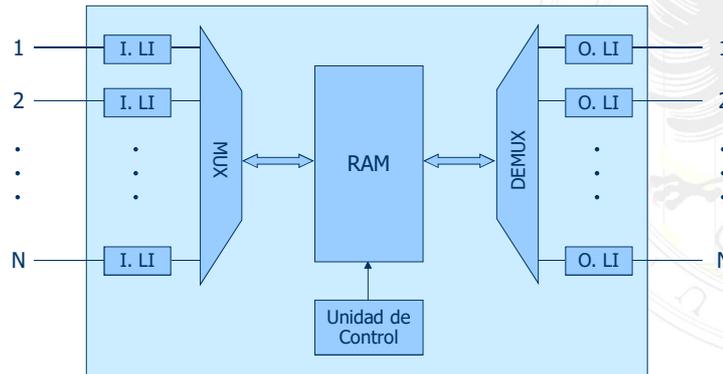
3.1.1 Arquitectura por División Espacial

- En vez de mantener una única cola FIFO para todas las celdas en una entrada, cada entrada mantiene una cola para cada salida
- Hay un total de N^2 colas FIFO a la entrada, y cada cola es denominada Virtual Output Queue (VOQ)
- Un planificador selecciona qué celdas de cada VOQ debe ser transmitida en cada ranura temporal
- Se **elimina el bloqueo HOL**
- El rendimiento de la Red de Interconexión depende del algoritmo de planificación de paquetes
- Con un algoritmo de planificación adecuado se puede alcanzar el **100% del throughput**



3.1.2 *Arquitectura de Memoria Compartida*

Conmutador con Arquitectura de Memoria Compartida



3.1.2 *Arquitectura de Memoria Compartida*

Conmutadores con Arquitectura de Memoria Compartida

- Se caracteriza porque todos los puertos de entrada y salida tienen acceso a una memoria común
- Las células en los puertos de entrada se multiplexan (TDM) en único flujo que alimenta la memoria compartida
- En cada ranura temporal, todos puertos de entrada almacenan sus células y todos los puertos de salida extraen sus células de la memoria
- Internamente, la memoria organiza las células en colas lógicas separadas, una para cada puerto de salida
- El flujo de células de salida se forma extrayendo las células al frente de las colas (HOL)
- Más tarde dicho flujo se demultiplexa y se transmite en cada una de las líneas de salida



3.1.2 *Arquitectura de Memoria Compartida*

- Desde un punto de vista de rendimiento, el conmutador con arquitectura de memoria compartida se comporta esencialmente como un conmutador con memoria a la salida y por tanto consigue el mismo throughput y retardo
- Debido al manejo de memoria centralizada, el tamaño del conmutador está limitado por el tiempo de lectura/escritura de la memoria en el que se ha de acceder a las N células de entrada y N células de salida (es decir, 2N accesos en cada ranura)

- Por tanto el tiempo de acceso a la memoria ha de cumplir:

$$t_{\text{acc}} \leq \frac{\text{cell_lengt}}{2 \cdot N \cdot \text{link_sn}}$$

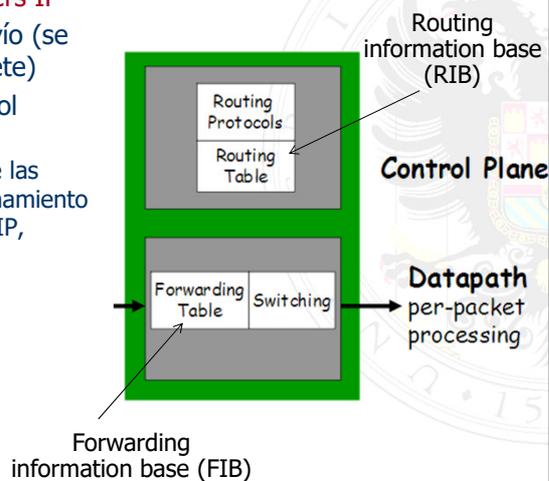
- Por ejemplo, para una duración de célula de 2.83 μseg y con un tiempo de acceso de 10ns $\Rightarrow N \leq 141$



3.2 *Arquitectura de Routers IP*

Funciones de los Routers IP

- Funciones de reenvío (se ejecutan por paquete)
- Funciones de control
 - Construcción y mantenimiento de las tablas de encaminamiento (p.ej. mediante RIP, OSPF,...).
 - Configuración
 - Gestión

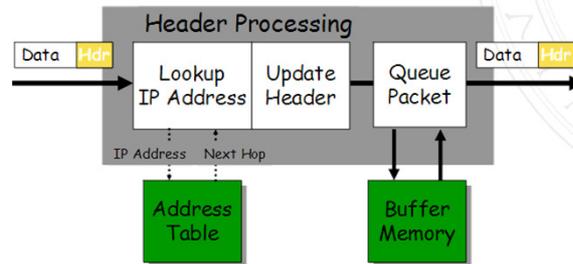




3.2 Arquitectura de Routers IP

Funciones de reenvío:

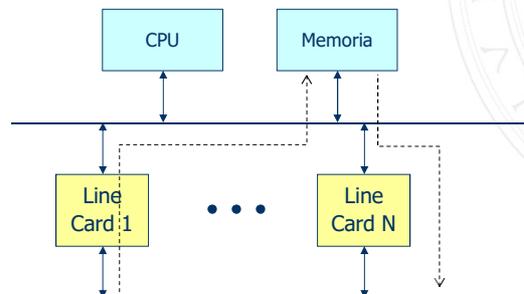
- Recepción del paquete en el puerto de entrada y validación
- Proceso de búsqueda (Lookup): de la dirección IP en una tabla de reenvío para hallar el puerto de destino
- Procesamiento de la cabecera del paquete: Decremento del campo TTL, actualización del checksum
- Reenvío: hacia el puerto de salida



3.2 Arquitectura de Routers IP

Arquitectura de Routers IP: Primera Generación

- La primera generación de routers IP estaba basada en implementaciones software en una única CPU de propósito general
- Fueron usados hasta finales de los años 80, y se usan hoy en día para los routers de menores prestaciones

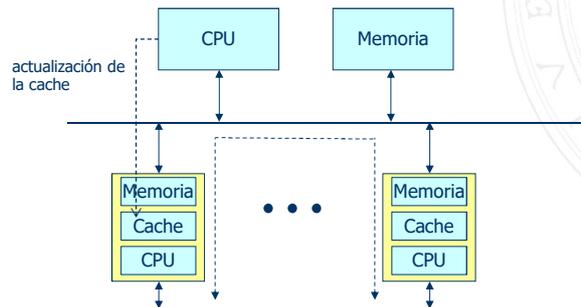




3.2 Arquitectura de Routers IP

Arquitectura de Routers IP: Segunda Generación

- En los routers de segunda generación (principio de años 90), la arquitectura anterior es mejorada mediante la distribución de las funciones de reenvío



3.2 Arquitectura de Routers IP

Arquitectura de Routers IP: Tercera Generación

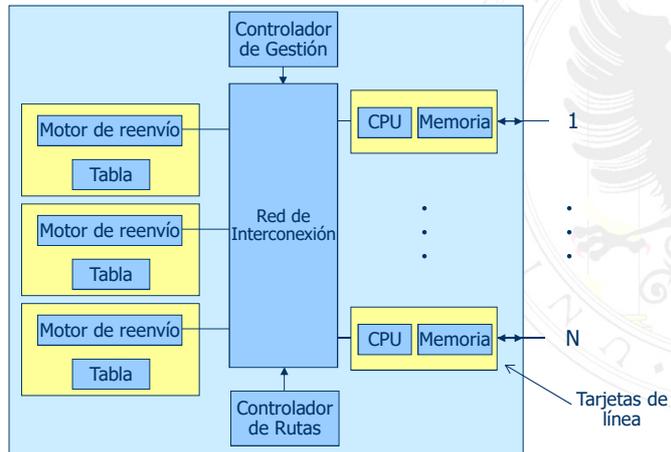
- Routers de grandes prestaciones (finales de los 90)
- Se sustituye el bus compartido por una red de interconexión (p.ej. de barras cruzadas)
- La red de interconexión incrementa el throughput en varios ordenes de magnitud \Rightarrow el procesamiento de la cabecera es el cuello de botella
- Se crean motores de reenvío (*forwarding engine*) dedicados exclusivamente a leer la cabecera, determinar el puerto de salida y actualizar la cabecera del datagrama
- Para determinar el puerto de salida, los motores de reenvío cuentan con caché de rutas
- Estas arquitecturas se clasifican en:
 - Centralizadas
 - Distribuidas



3.2 *Arquitectura de Routers IP*

Sistemas de Comunicación - Grado en Ing. de Tec. de Telecomunicación - UGR
© Pablo Ameigiras Gutiérrez

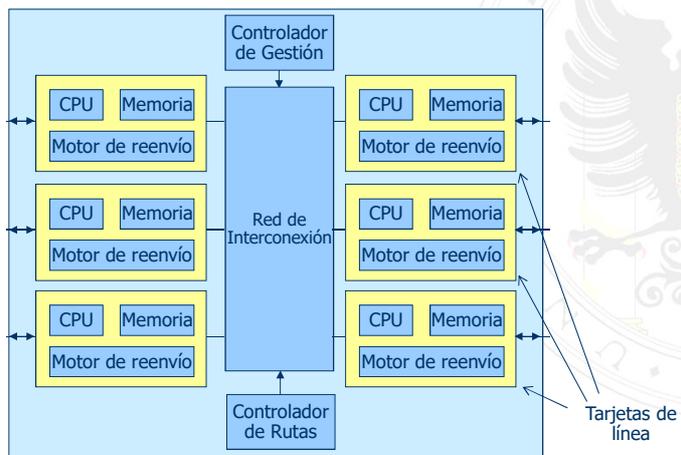
Arquitectura Centralizada



3.2 *Arquitectura de Routers IP*

Sistemas de Comunicación - Grado en Ing. de Tec. de Telecomunicación - UGR
© Pablo Ameigiras Gutiérrez

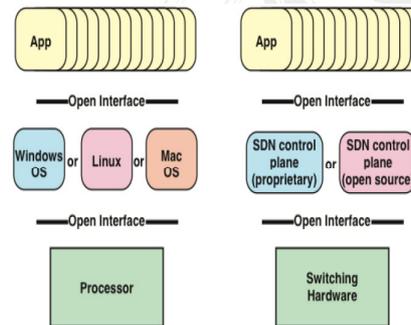
Arquitectura Distribuida



3.3 Motivación de SDN

Motivación de SDN

- Evolución de la tecnología de computadores: desde soluciones cerradas hacia entorno con interfaces abiertos que permite mover aplicaciones de la plataforma de un fabricante a otra
- Software Defined Networking (SDN)** pretende seguir una evolución similar en la tecnología de redes

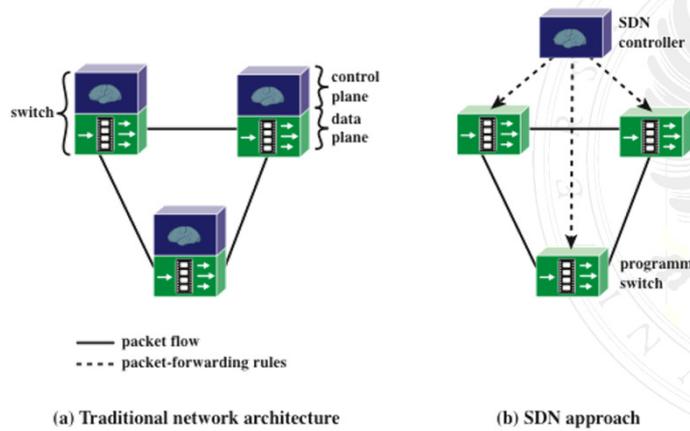


3.3 Fundamentos de SDN

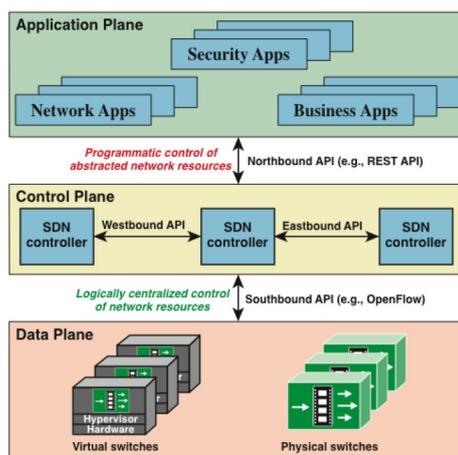
Fundamentos de SDN:

- El plano de control se separa del plano de datos. Los dispositivos del plano de datos (conmutadores/routers) pasan simplemente a hacer reenvío de paquetes
- El plano de control está implementado en uno o varios controladores centralizados
 - Si hay varios éstos están coordinados
- Se definen interfaces abiertos entre dispositivos en el plano de control (controladores) y en el plano de datos
- La red es programable mediante aplicaciones que se ejecutan sobre los controladores.
- Los controladores presentan una visión abstracta de los recursos de la red a las aplicaciones

3.3 Fundamentos de SDN



3.3 Arquitectura de SDN



SDN architecture



3.3 *Arquitectura de SDN*

- **Plano de Datos de SDN**
 - También conocido como capa de infraestructura o capa de recursos
 - En el plano de datos es donde los conmutadores llevan a cabo el transporte y procesamiento de los paquetes según las decisiones del plano de control SDN
- **Plano de Control de SDN**
 - Traduce las solicitudes de la capa de aplicación en comandos a los conmutadores del plano de datos con información sobre la topología y actividad del plano de datos
- **Plano de Aplicación de SDN**
 - Implementa aplicaciones y servicios que monitorizan y controlan los recursos de la red y definen su comportamiento



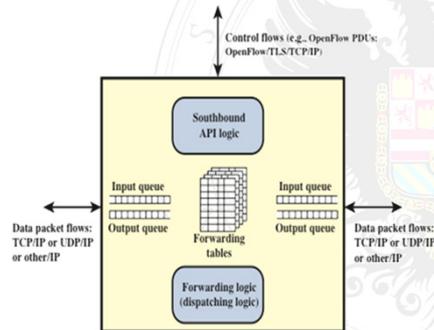
3.3 *Arquitectura de SDN*

- **Southbound API**
 - Cada conmutador implementa un modelo o abstracción del reenvío de paquetes que es uniforme y abierto a los controladores SDN
 - Dicho modelo está definido en términos de una API (Application Programming Interface) entre el plano de control y de datos (Southbound API)
 - Los controladores usan dicha API para controlar el conmutador en el plano de datos
- **Northbound API**
 - Los controladores también exponen una API que permite a los desarrolladores desplegar aplicaciones de red personalizadas
 - No existe todavía una API Northbound estandarizada (algunos fabricantes ofrecen una API REST)

3.3 Plano de Datos de SDN

Funciones del Plano de Datos de SDN

- Función de soporte del control
- Función de reenvío de datos: reenvía los paquetes de los flujos* por el camino establecido según las reglas creadas por las aplicaciones
- Las reglas se implementan en las tablas de reenvío:
 - Deciden el siguiente salto
 - Pueden modificar la cabecera
 - Pueden descartar el paquete



Flujo*: secuencia de paquetes que atraviesan la red y que comparten ciertos valores de las cabeceras

3.3 Plano de Datos de SDN

Protocolos del Plano de Datos

- Los flujos de paquetes de datos están compuestos de datagramas IP
- La tabla de reenvío puede tener entradas basadas en cabeceras de protocolos de distintos niveles (p.ej. Ethernet, IP, TCP, UDP, otro transporte, o protocolos de aplicación).

OpenFlow

- OpenFlow representa la implementación más destacada de:
 - Protocolo (seguro) entre el controlador SDN y el conmutador
 - Permite añadir, actualizar y borrar **entradas de flujos** en las tablas de reenvío de forma proactiva o reactiva (en respuesta a paquetes)
 - Southbound API
 - Es decir, de la especificación de la estructura lógica de la funcionalidad del conmutador
 - Esta estructura lógica puede ser implementada en diferentes modos dependiendo del fabricante

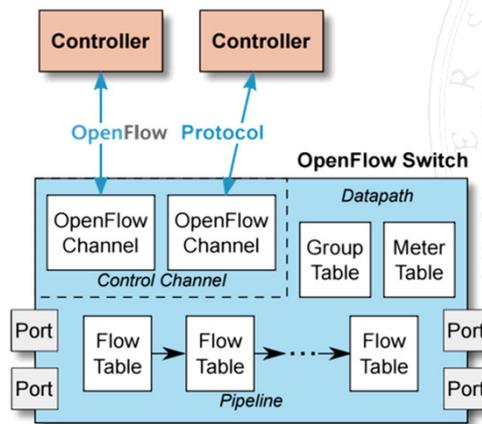


3.3 OpenFlow

Sistemas de Comunicación - Grado en Ing. de Tec. de Telecomunicación - UGR
© Pablo Ameigiras Gutiérrez



Modelo de Switch OpenFlow



3.3 OpenFlow

Sistemas de Comunicación - Grado en Ing. de Tec. de Telecomunicación - UGR
© Pablo Ameigiras Gutiérrez



Tabla de flujos

- Busca la coincidencia entre el paquete de entrada y un flujo determinado, y especifica las acciones a realizar en dicho paquete
- La tabla de flujos consiste en un número de filas llamadas **entradas** compuesta de varias componentes
- Se pueden hacer uso de múltiples tablas (con pipelining) con el objetivo de hacer un tratamiento diferenciado a los subflujos que componen un flujo

Match fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
--------------	----------	----------	--------------	----------	--------	-------

(a) Flow Table Entry Fields

Ingr port	Egr port	Ethr SA	Ethr DA	Ethr Type	IP prot	IPv4 SA	IPv4 DA	IPv6 SA	IPv6 DA	TCP Src	TCP Dest	UDP Src	UDP Dest
-----------	----------	---------	---------	-----------	---------	---------	---------	---------	---------	---------	----------	---------	----------

(b) Flow Table Match Fields (required fields)



3.3 OpenFlow



Instrucciones

- Se ejecutan cuando un paquete coincide con una entrada
- **Acciones**
 - Output: reenvía el paquete a un puerto específico, que puede ser físico, virtual o reservado (all, controller, table...)
 - Set-queue: selecciona la cola del puerto de salida para el paquete
 - Group: procesa el paquete a través de un grupo específico
 - Push-Tag/Pop-Tag: añade o elimina cabecera VLAN o MPLS
 - Set-Field: modifican los valores de los campos de la cabecera
 - Change-TTL: modifica el campo TTL del paquete (en cabecera IPv4, IPv6 o MPLS)
 - Drop: elimina el paquete
- **Conjuntos de acciones**
 - Representa una lista de acciones asociadas a un paquete y que se acumulan mientras que el paquete es procesado en cada tabla y que son ejecutadas cuando el paquete sale del pipeline



3.3 OpenFlow



Ejemplos de Entradas de una Tabla de Flujos

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop



3.3 Plano de Control de SDN

Plano de Control de SDN

- El controlador SDN puede ser visto como un Sistema Operativo de Red (NOS)
- Como en un SO convencional, un NOS proporciona a los desarrolladores servicios esenciales, APIs y una abstracción de los dispositivos de las capas más bajas
- Las funciones del NOS permiten a los desarrolladores gestionar la red y definir políticas sin tener en cuenta los detalles de los dispositivos, que pueden ser heterogéneos y dinámicos

Implementaciones destacadas de controladores SDN

OpenDayLight
ONOS

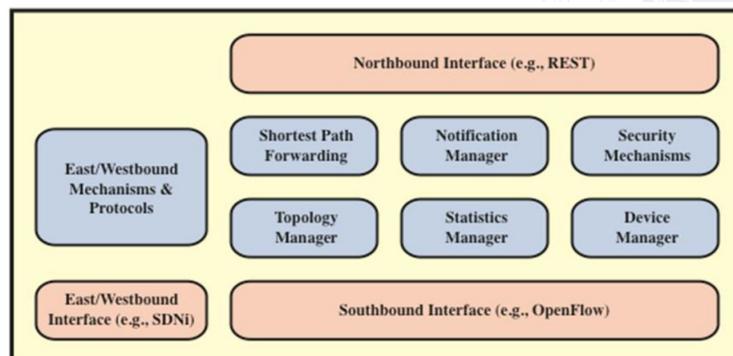
POX
Beacon

Floodlight
Ryu



3.3 Plano de Control de SDN

Funciones del Plano de Control SDN e Interfaces





3.3 Plano de Aplicación de SDN

Plano de Control de SDN

- Contiene las aplicaciones y servicios que monitorizan y controlan los recursos de red y definen su comportamiento
- Se pueden clasificar en dos tipos: aplicaciones y casos de uso

Aplicaciones SDN

Seguridad
QoS
Traffic Engineering
Balanceo de Carga
U-ACL

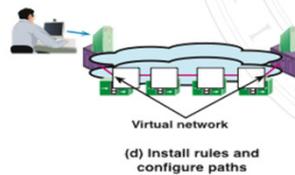
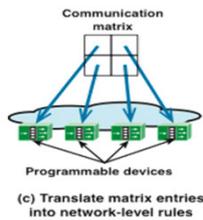
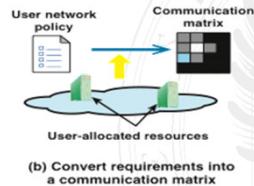
Casos de uso SDN

Cloud Computing
Information Content
Networking
Redes móviles
Network Function
Virtualization



3.3 Ejemplo de caso de uso de SDN

Cloud Networking sobre SDN





3.3 **El mercado de SDN**

Según el plano de datos/control ofrecido:

- **Bare Metal:** el fabricante ofrece solo el plano de datos (HW)
 - Menor coste, permite adaptar el SW (NOS) según las necesidades
 - El HW (ASIC) puede ser:
 - ◆ Propietario (custom, in-house)
 - ◆ Ensamblado a partir de chips (merchant silicon, off-the-shelf)
- **White Box:** bare metal + NOS (propietario o abierto)

Según el hardware utilizado:

- **Físico:** incluye HW especializado, mayor control y visibilidad de la red, aunque el número de entradas de flujo es limitado
- **Virtual (overlay):** SW que se ejecuta sobre un hipervisor o SO, tiene más memoria y CPU disponible (más entradas de flujos)
- **Híbrido:** incluye los dos tipos anteriores

