



Tema 3

Redes de Área de Sistema

Nicolás Calvo Cruz
Dpto. de Arquitectura y Tecnología de los Computadores
@ncalvocruz
ncalvocruz@ugr.es

Motivación

- Redes de **comunicación en computadores** paralelos.
- Hoy en día se están **sustituyendo los buses por redes** con conexiones **punto a punto** a todos los niveles:
 - Interno al chip
 - A nivel de tarjeta y placa
 - A nivel de chasis o caja
 - LAN y Router IP
- Conocer los algoritmos de **encaminamiento** y la **infraestructura** permite mejorar las **prestaciones**.

Objetivos

- Distinguir entre redes de **altas prestaciones** y **redes estándar**.
- Conocer la estructura general de un **conmutador**.
- Estudiar las **topologías** y nomenclaturas de las **redes de altas prestaciones**.
- Estudiar los **algoritmos de encaminamiento**.



Índice

1. Clasificación Sistemas de Comunicación
2. Propiedades
- 3. Diseñar una Red**
4. Prestaciones
5. Enrutamiento
6. Técnicas de conmutación
7. Ejemplo



3. Diseñar una red

Qué tener en cuenta para diseñar una red SAN

3. Diseñar una red

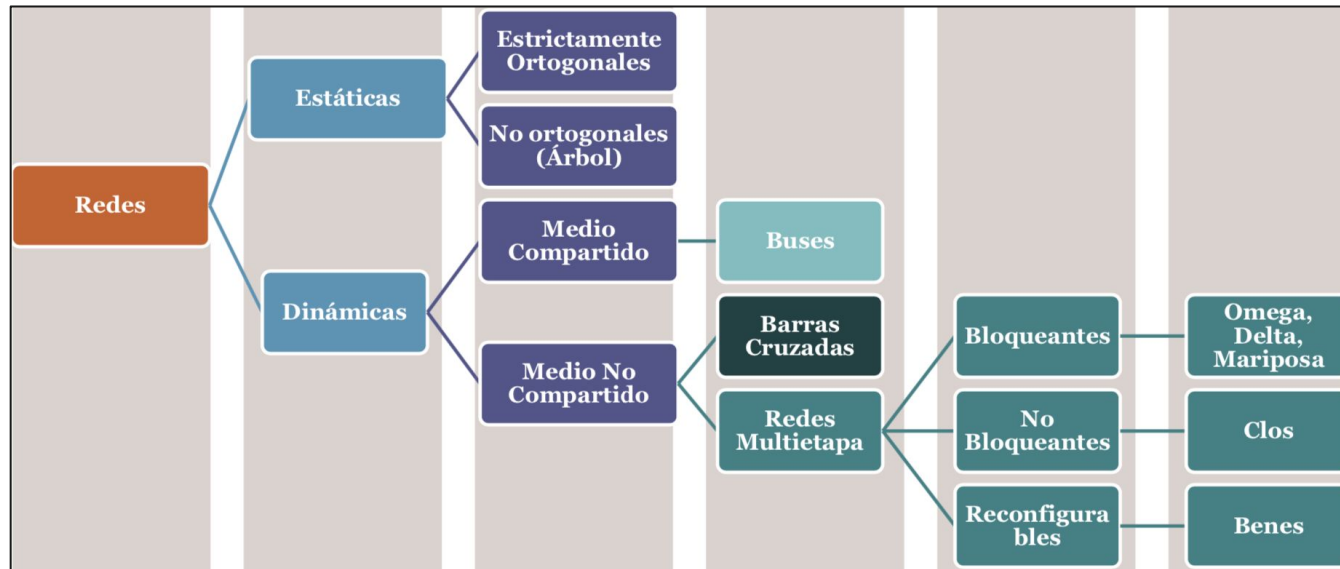


Hay **cuatro puntos clave** al diseñar una red que marcan su funcionamiento y prestaciones:

1. **Topología**
2. Estrategia de **conmutación**
3. Mecanismos de control de **flujo**
4. Algoritmo de **encaminamiento**

3. Diseñar una red: Topología

- La topología determinará muchas de las prestaciones de la red.

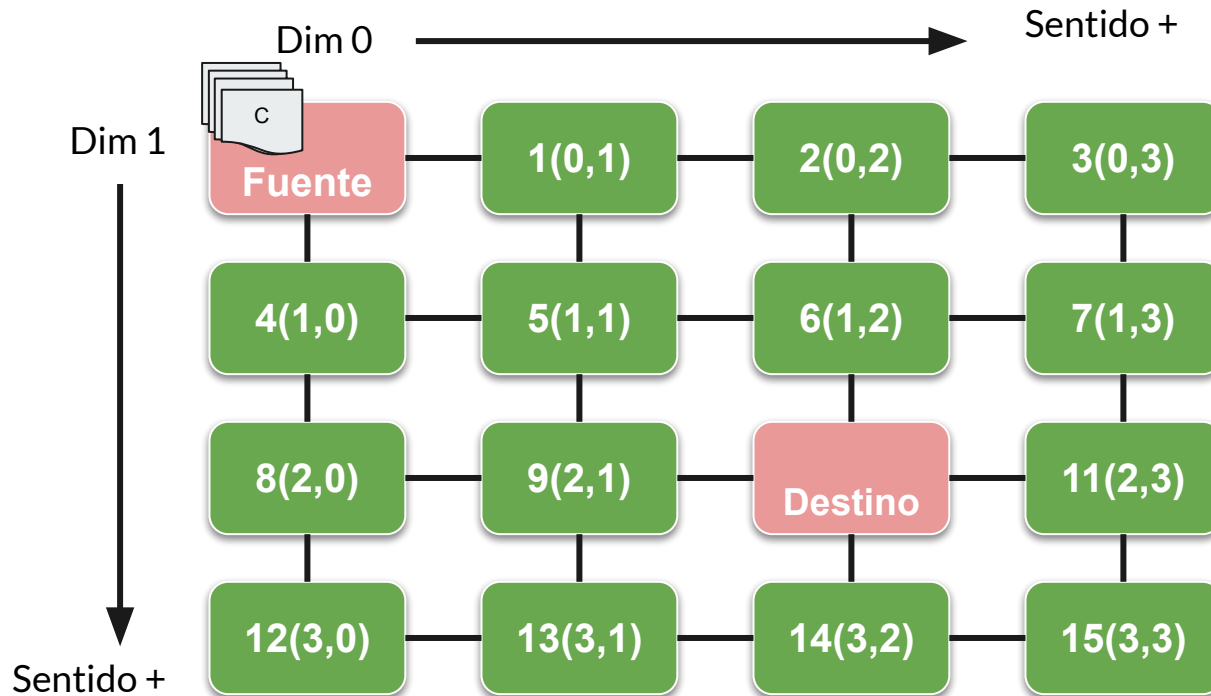


3. Diseñar una red: Estrategia de conmutación



- Determina **cómo viaja la información** de un nodo fuente a otro destino:
 - Estableciendo una conexión dedicada (**Conmutación de circuitos**)
 - Agrupada saltando de conmutador en conmutador (**Almacenamiento y Reenvío**)
 - Estirándose a lo largo del camino (**Vermiforme**)
 - Variantes/combinaciones de las anteriores (**Virtual Cut-Through**)

3. Diseñar una red: Estrategia de conmutación



3. Diseñar una red: Control de flujo



- **Determina para una unidad de información** (de cualquier nivel):
 - **Cuándo se mueve** entre los diferentes almacenamientos del sistema de comunicación.
 - **Cómo y cuándo se asignan los recursos de esos almacenamientos** para continuar el transporte de los datos en la red.
- **Según el nivel** en el que estemos, **las unidades de información que deben avanzar varían.**

3. Diseñar una red: Control de flujo



- Las unidades de información para las que se establece un control de flujo son:
 - **Phit** (*Physical Transport Unit*):
 - Unidad **más pequeña (nivel más bajo)** en la red cuya transferencia sólo **se acepta si hay espacio** en el destino.
 - Es la cantidad de **información transferible en un solo ciclo entre dos elementos** de la red.
 - Se transportan **en orden** entre nodos.
 - La velocidad de transferencia en este nivel se mide en **phits/seg**.

3. Diseñar una red: Control de flujo



- Las unidades de información para las que se establece un control de flujo son:
 - Flit (*Flow Control Unit*):
 - Mínima cantidad de información que se transfiere entre elementos del sistema de comunicación a nivel de control de flujo.
 - Incluye uno o varios phits.
 - Un flit no se enruta: **todos los flits de un paquete** siguen la **misma ruta** (que definirá el paquete, que contiene la dirección de destino).
 - Los **flits de un paquete** han de transportarse **en orden**.

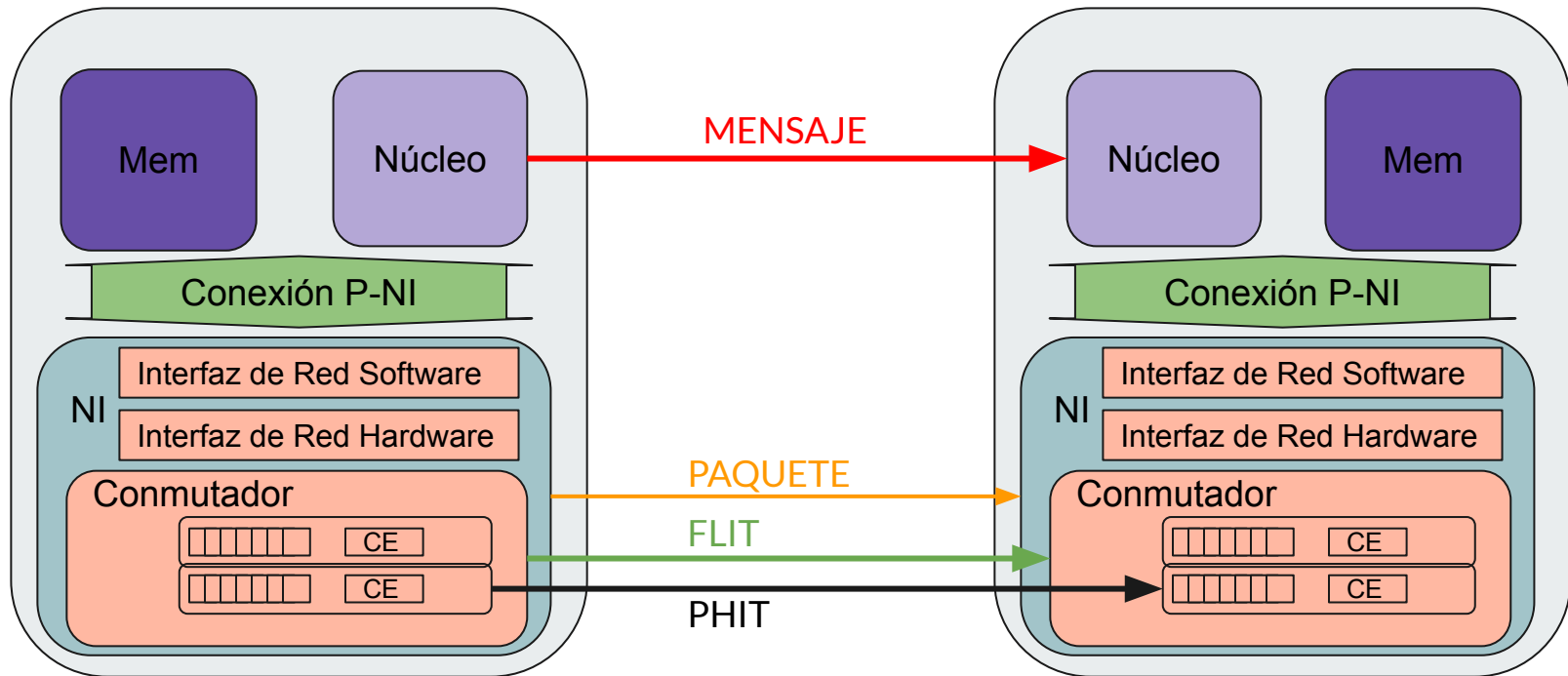
3. Diseñar una red: Control de flujo

- Las unidades de información para las que se establece un control de flujo son:
 - **Paquete:**
 - Unidad de información **siguiente al Flit**, formada mínimo por **1 de estos**.
 - Tiene por una **cabecera**, con la información de destino, una zona de **datos**, con el contenido del paquete y una zona **final** de comprobación.
 - Son las **unidades entre interfaces de red**, y **no** se envían en **orden**.

Cabecera	Datos	Fin
----------	-------	-----

- **Mensaje:** Conjunto de paquetes. Unidad de control de flujo a nivel de aplicación

3. Diseñar una red: Control de flujo



3. Diseñar una red: Control de flujo



Unidad	Componentes	Control de flujo
Mensaje	APP - APP	Control de flujo del programa
Paquete	NI - NI	Control de flujo de extremo a extremo
Flit	C - C	Control de flujo de conmutación
Phit	CE - CE	Control de flujo físico

3. Diseñar una red: Control de flujo

¿Qué **tareas** tiene exactamente el control de flujo?

1. **Garantizar** las **prestaciones** mínimas de la red
2. Asegurar **recepción** sin errores
3. **Garantizar** el **almacenamiento en destino** de las unidades
4. **Arbitrar** entre **unidades** que quieren acceder a la vez al mismo recurso
5. **Asegurar** que la **unidad de información** llega al destino **sin solaparse** con otras

Control de flujo de extremo a extremo (Paquete)

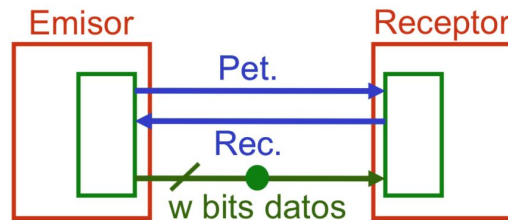
Control de flujo de conmutación (Flit)

Control de flujo físico (Phit)

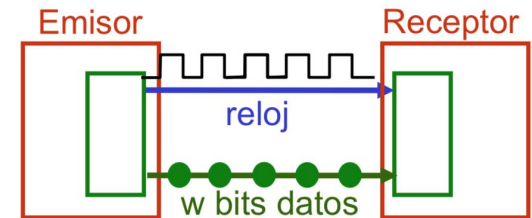
3. Diseñar una red: Control de flujo

Control de flujo físico

- Garantiza que los **phits** se transfieran por un enlace llegando al destino **sin solapamiento**. Hay 2 alternativas:
 - Síncrono
 - Asíncrono



Asíncrona
Líneas cortas

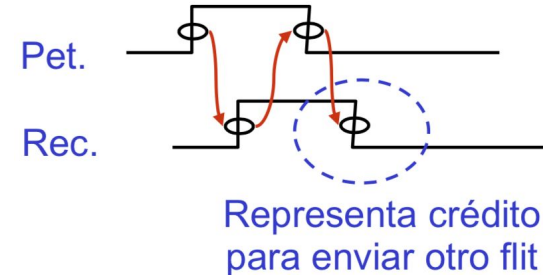


Síncrona
Líneas largas

3. Diseñar una red: Control de flujo

Control de flujo de conmutación

- Transfiere flits dentro de conmutadores: arbitra entre entradas que se quieren conectar simultáneamente a la misma salida.
- Transfiere flits entre conmutadores garantizando almacenamiento en el destino para todo el flit:
 - Con flit = phit, probable en líneas cortas: (a) con transferencia asíncrona se puede aprovechar la señal de reconocimiento para indicar disponibilidad de almacenamiento.

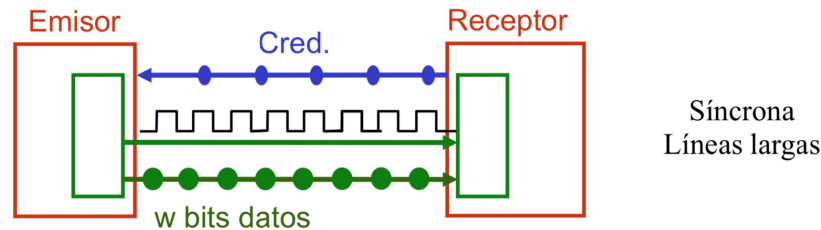


- (b) con transferencia síncrona se necesita una señal de control (rec) desde el receptor al emisor que informe si hay o hubo espacio para un flit

3. Diseñar una red: Control de flujo

Control de flujo de conmutación

- Con flit \neq phit, probable en líneas largas: como la transferencia es síncrona **se necesita información de control desde el receptor al emisor que informe de que hay espacio para almacenar un flit (crédito).**

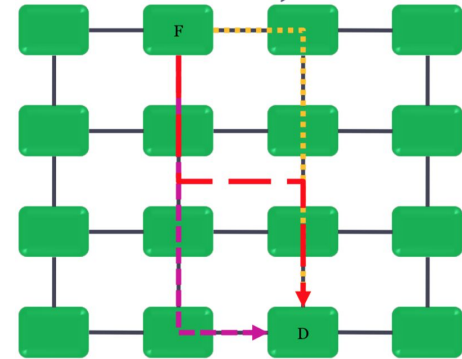


- El emisor puede tener un **contador de créditos asociado al canal**. Este se decrementa en cada envío y se incrementa con cada crédito recibido.
- **Si se multiplexan las señales de control con datos** (enlaces estrechos), **cada señal de reconocimiento puede representar un crédito** para n flits.
- Se pueden utilizar dos tipos de señales, una para parar el envío (STOP) y otra para activarla (GO)

3. Diseñar una red: Algoritmos de encaminamiento

- Debe realizar 2 funciones básicas:
 - **Función de Encaminar:** Determinar la ruta (o rutas) para hacer una operación de comunicación entre:
 - Una fuente y un destino (1-1)
 - Varias fuentes y un destino (M-1)
 - Varios destinos y una fuente (1-M).

Generará un conjunto de caminos candidatos.



- **Función de Selección:** Escoger entre los caminos encontrados cuál seguir para cada paquete.

3. Diseñar una red: Algoritmos de encaminamiento



- El algoritmo de encaminamiento (o enrutamiento) **es clave** porque:
 - Debe **balancear el tráfico**, incluso ante problemas. A mejor balanceo, mejores prestaciones.
 - Debe **evitar interbloqueos** entre comunicaciones siempre que se pueda.
 - El **camino** debe ser **lo más corto posible**, manteniendo prestaciones.
 - Debe ser adaptativo para **sobreponerse a fallos físicos de la red**.

3. Diseñar una red: Algoritmos de encaminamiento



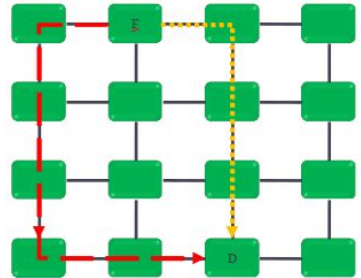
Clasificación:

- Según la selección de rutas:
 - **Deterministas:** Entre el conjunto de rutas posibles, se elige siempre la misma.
 - **No deterministas:** La selección del camino varía. Se pueden hacer 3 cosas:
 - No tomar información de la red (**Selección Aleatoria**)
 - Tomar información de parte de la red para decidir (**Parcialmente Adaptativo**)
 - Tomar información de toda la red para decidir (**Totalmente Adaptativo**)

3. Diseñar una red: Algoritmos de encaminamiento

Clasificación:

- Según la longitud del camino:
 - **Mínimo:** Entre las rutas posibles, se eligen entre las de menor distancia al destino (Algoritmo provechoso)
 - **No mínimo:** Se puede elegir una ruta no mínima (Algoritmo mal-enrutado)



3. Diseñar una red: Algoritmos de encaminamiento



Clasificación:

- Según la posibilidad de volver hacia atrás:
 - **Progresivo:** Un paquete no se enrutará por el mismo camino que ha llegado, siempre disminuyendo la distancia.
 - **Con retroceso** (*backtraking*): El camino por el que ha llegado también se tiene en cuenta en la función de selección.

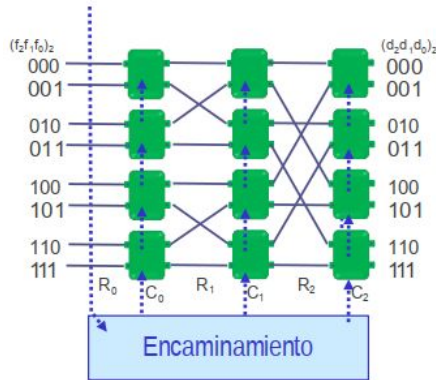
3. Diseñar una red: Algoritmos de encaminamiento



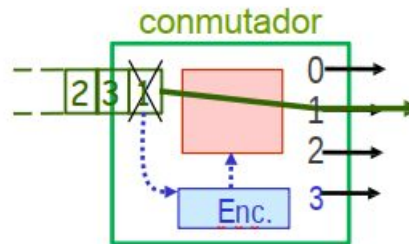
- Hay **otros criterios** de clasificación, **pero no incluyen** directamente la **función de selección**. Se pueden catalogar según:
 - Su **funcionalidad** (Si es para 1-1, 1-M o M-1)
 - **Dónde** se hace la función de selección (centralizado, distribuido, multifase o en fuente)
 - Su tipo de **implementación** (con tabla o sin tabla)
 - Los **canales seleccionados**, estableciendo siempre un conjunto de canales candidatos o sólo uno posible.

3. Diseñar una red: Algoritmos de encaminamiento

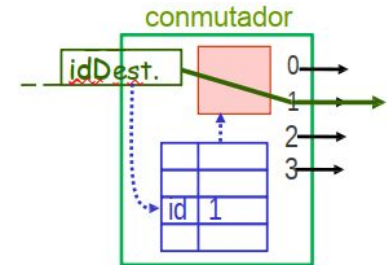
- Hay otros criterios de clasificación, pero no incluyen directamente la función de selección...



Enc. Centralizado: circuitería común genera las señales de control.



Enc. Fuente: los canales a utilizar se fijan en el fuente.



Enc. Distribuido-Tablas: cada conmutador obtiene el canal de salida a utilizar. Ej.: consultando una tabla de encaminamiento.

Índice

1. Clasificación Sistemas de Comunicación
2. Propiedades
3. Diseñar una Red
- 4. Prestaciones**
5. Enrutamiento
6. Técnicas de conmutación
7. Ejemplo



4. Prestaciones en una red

Medidas de rendimiento en una red SAN

4. Prestaciones



- Las **prestaciones** de una red se miden **según distintas variables** cuyo significado depende también de cómo y a qué nivel se midan:
 - **Coste:** Cantidad de **dinero** que costará la red.
 - **Latencia:** Indica cuánto **tiempo** tardará un **mensaje pequeño** ($L=T(0)$).
 - **Productividad:** **Bits por unidad de tiempo** transportables desde una fuente a un destino.
 - **Tolerancia a fallos**
- Hay **otras**, como la escalabilidad, calidad del servicio... **que vimos en las propiedades de las redes.**

4. Prestaciones: Tipos de prestaciones



- Las prestaciones se pueden clasificar en 2 grupos:
 - **Prestaciones extremo-a-extremo:** Se miden las prestaciones del envío de mensajes **entres dos nodos de la red** cualesquiera, X e Y.
 - **Globales:** Se miden a nivel de toda la red con múltiples transferencias simultáneas. La **medición** se hace **inyectando tráfico** en la red siguiendo alguna función de distribución que permita representar una cierta situación real.

4. Prestaciones: Prestaciones Extremo-a-Extremo

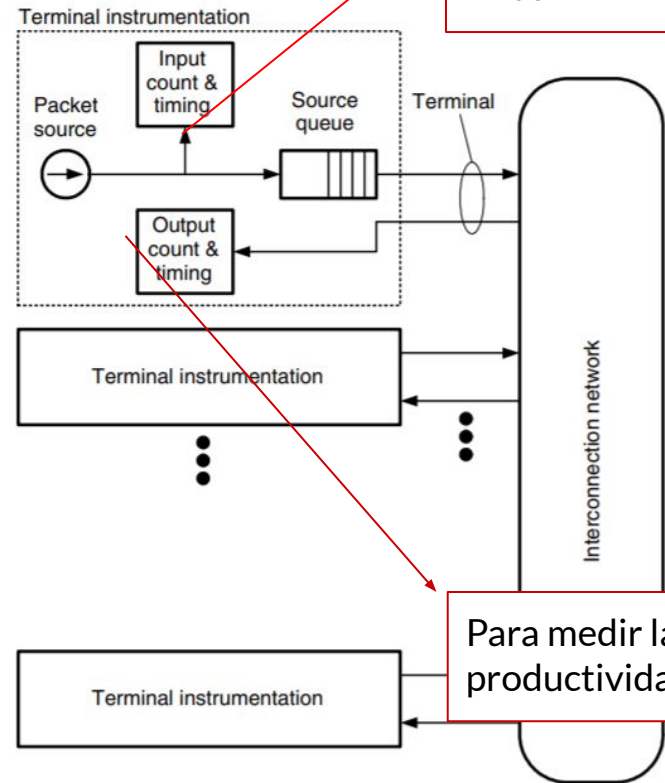
- Tienen en cuenta software y hardware:
 - Ancho de banda teórico o capacidad del canal (Mb/s)
 - Latencia total o tiempo de comunicación, T .
 - Latencia, $L=T(0)$: tiempo que supone la transferencia de un mensaje de tamaño pequeño (0, 1 byte, 8 bytes).
 - Productividad o ancho de banda efectivo, P . (MB/s). $P(m)=m/T(m)$
 - Ancho de banda asintótico, B : productividad máxima.
(Ejemplo Micro-benchmark: test ping-pong)
- Aproximación del tiempo de comunicación a partir de la latencia y el ancho de banda asintótico: $T(m) = L + m / B$

4. Prestaciones: Prestaciones Globales

El sistema de medida debe poder generar tráfico según una distribución deseada:

- Uniforme: Mensajes a cualquier salida de la red con la misma probabilidad.
- Normal: A cualquier salida pero más probablemente a aquellas más cercanas.
- No uniforme: La probabilidad de enviar a un destino depende de la frecuencia con la que se escoge.

Distintos tamaños y regularidad concreta.



Para inyectar tráfico incluyendo los tiempos en colas

Para medir la productividad

4. Prestaciones: Prestaciones Globales



Se toman en distintas situaciones:

- **Sin tráfico previo** para ver cómo evoluciona la red al ir generando el tráfico de forma automática.
- **En equilibrio** (*steady-state*). Una red está en equilibrio cuando todos sus almacenamientos intermedios (colas) no aumentan ni disminuyen rápido. Estas medidas se hacen en 3 fases: **warm-up, measurement y drain**.

4. Prestaciones: Prestaciones Globales

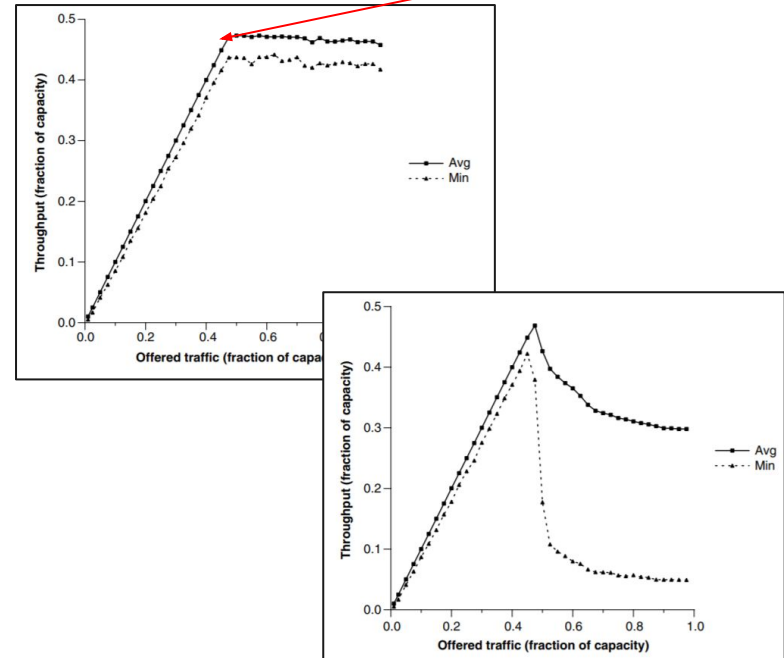
- (...) Estas medidas se hacen en **3 fases**:
 - Warm-up: Se inyecta el tráfico necesario para poner la red en equilibrio.
 - Measurement: Se ejecutan tantos ciclos de inyección de paquetes como se necesite. Estos van etiquetados con marca de tiempo.
 - Drain: Se van eliminando paquetes hasta comprobar que se han recibido todos (midiendo su tiempo).

Las mediciones de la latencia consideran todos los paquetes que llegan a destino en las dos últimas fases, pero no los enviados en las fases warm-up ni drain.

4. Prestaciones: Productividad (Throughput)

- Se mide contando **todos los paquetes generados** desde cualquier entrada a cualquier salida de la red en un periodo de tiempo.
- También se llama “**Tráfico aceptado o productividad aceptada**” en contraposición al “**Tráfico ofrecido**” que es la tasa de paquetes inyectados.

Punto de saturación, representa el ancho de banda global para un patrón de comunicación dado.



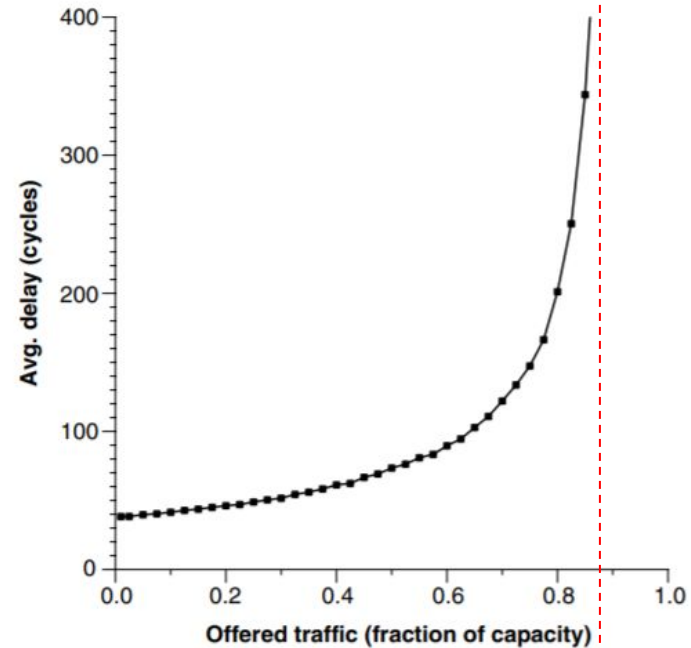
4. Prestaciones: Latencia

- **Latencia total de comunicación:** Tiempo total empleado en el envío de un mensaje de tamaño M desde que un procesador da la orden hasta que el procesador de destino acaba el procesamiento de lo recibido.
- **Latencia de transporte:** Tiempo que pasa desde que se inyecta en la red el primer bit del paquete hasta que el último bit llega al interfaz de destino. **Debe incluirse el tiempo de encaminamiento** (tiempo desde que la cabecera se inyecta en la red hasta que llega al destino).
- **Latencia de red observada** es el tiempo que transcurre desde que un nodo se desentiende del envío, hasta que el destino se da por enterado.

4. Prestaciones: Latencia

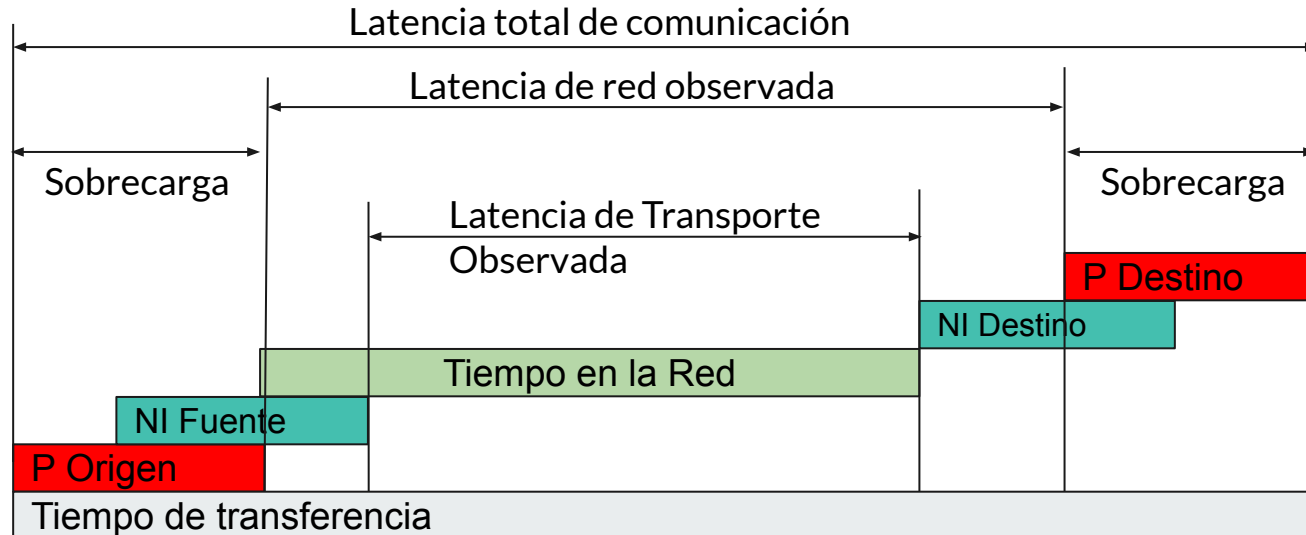
- La latencia o tiempo medio transcurrido en una comunicación en el sistema bajo un patrón de comunicación dado, se puede ver en función del tráfico ofrecido.
- **A más tráfico más latencia**, hasta que se alcanza el punto de saturación

Punto de saturación, representa el ancho de banda global para un patrón de comunicación dado.



4. Prestaciones: Conceptos básicos

- Tipos de latencia, relación con la comunicación y sobrecarga:



4. Prestaciones: Latencia y ancho de banda



- Latencia media: latencia de todos los paquetes dividida entre el número de paquetes.
- Productividad global o productividad aceptada: Bits/s que llegan por todas las salidas dividido entre el número de salidas (bits por segundo y nodo).
- Productividad solicitada o aplicada: Bits/s generados por todas las entradas dividido entre el número de entradas (bits por segundo y nodo).
- Productividad máxima o ancho de banda asintótico: Ancho de banda que se puede aprovechar de forma efectiva para una distribución de destinos dada (punto de saturación de la red).

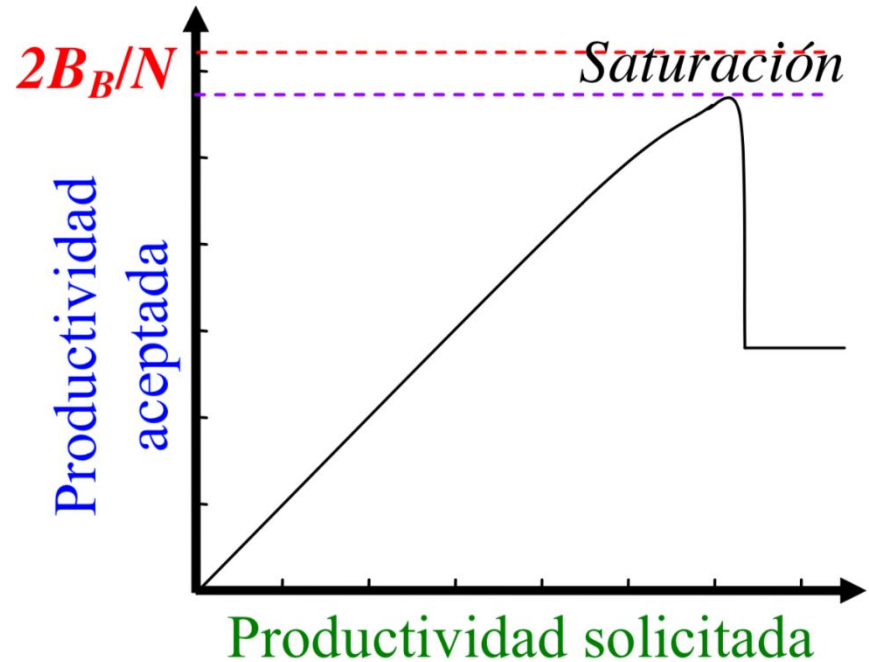
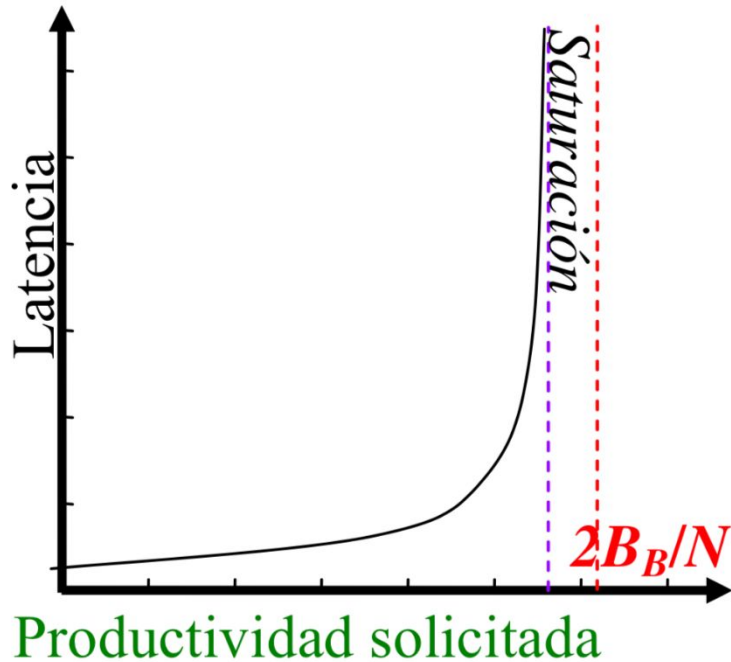
4. Prestaciones: Curiosidades de la latencia

- El **punto de saturación** de una red se puede aproximar por dos veces el ancho de banda de la bisección de la red (B_b) dividido por el número de nodos, (N):

$$\text{Punto de Saturación} = 2 \frac{B_b}{N}$$

- B_b es el ancho de banda existente en los canales que corta en el ancho de la bisección. Representa la **cantidad de datos que la red es capaz de enviar de una mitad a otra por unidad de tiempo.**

4. Prestaciones: Curiosidades de la latencia



4. Prestaciones: Latencia

- La latencia se calcula como:

$$\text{Latencia total}(T) = \text{Latencia de la cabecera}(T_h) + \text{Latencia del cuerpo}\left(\frac{L}{b}\right)$$

- Siendo T_h el tiempo necesario para que la cabecera del mensaje llegue a su destino, L la longitud del mensaje y b el ancho de banda del canal por el que se ha enrutado.
- La latencia de la cabecera está formada por los factores, T_r y T_w , que son el tiempo de enrutamiento y el tiempo de transporte, respectivamente.

4. Prestaciones: Latencia

- Todos los cálculos se suponen en ausencia de contención, es decir sin que el paquete tenga que pararse para continuar:

$$\text{Latencia de la cabecera } (T_h) = F(N, T_r, T_w)$$

- La función que marca la latencia de cabecera será una función en la que intervienen el número de nodos a atravesar, N , el tiempo de enrutamiento de la cabecera al atravesar todos los nodos, T_r , y el tiempo del transporte físico de los bits, T_w .

4. Prestaciones: Latencia

- Si tomamos un caso base en el que el mínimo número de saltos que hay que dar para llegar desde un origen a un destino dado es H_{\min} y el tiempo de enrutamiento en cada salto es t_r .
- Y si tomamos que el tiempo de transporte, T_w , como el número de enlaces a atravesar como mínimo, D_{\min} , dividido por la velocidad de envío en ese enlace, v , ...entonces podemos afirmar que el tiempo mínimo de latencia es T_0 :

$$\text{Latencia de enrutamiento } (T_r) = H_{\min} t_r$$

$$\text{Latencia de transporte } (T_w) = \frac{D_{\min}}{v}$$

$$T_0 = H_{\min} t_r + \frac{D_{\min}}{v} + \frac{L}{b}$$

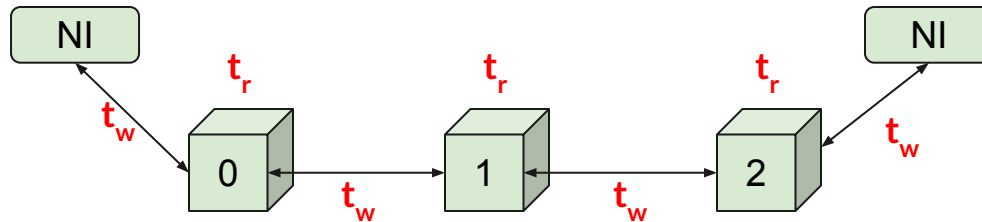
4. Prestaciones: Latencia



- T_0 es el tiempo mínimo en transferir un mensaje en nuestra red en ausencia de contención y con una carga mínima.
- Si introducimos un entorno más real, con contenciones, aparecerá otro término denominado T_c , que representara el tiempo que el paquete está retenido en los diferentes buffers de almacenamiento intermedio.
- H_{\min} , D_{\min} y b , son parámetros que vienen marcados por la topología.

4. Prestaciones: Latencia, Ejemplo

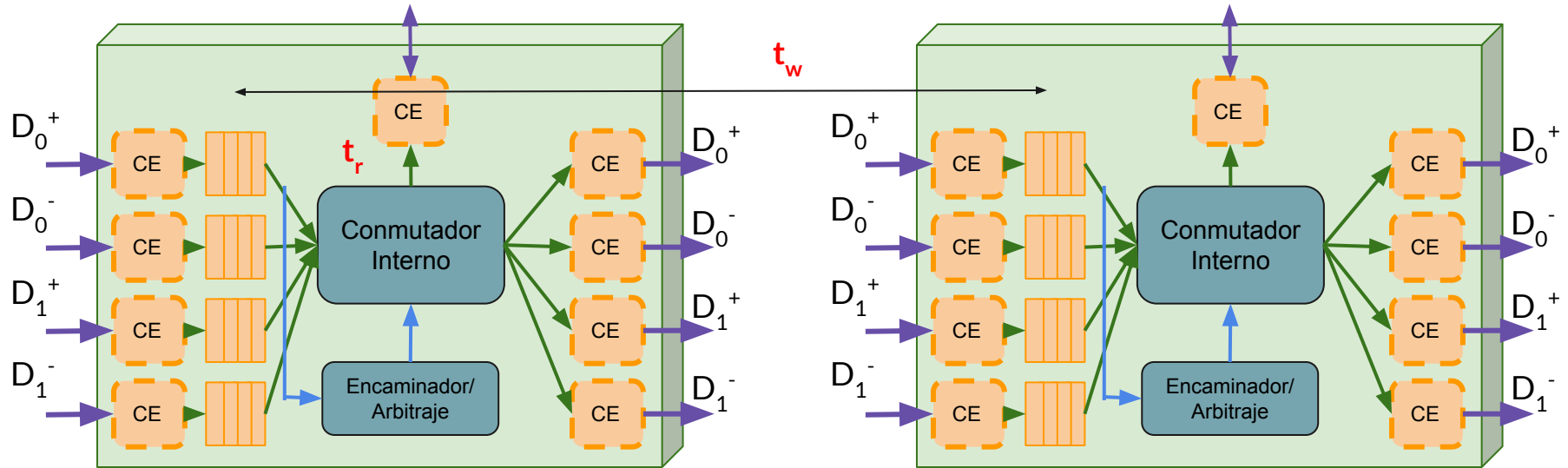
- 1 flit = 1phit = w bits.
- Cabecera = 1 flit
- Longitud del mensaje, L , $L/w = 3$ flits
- D: Número de parejas conmutador-enlace a atravesar desde el origen hasta el destino.
- Conmutadores sin buffer a la salida
- Entorno sin contenciones



Paquete con 1
cabecera y 3 flits

El tiempo mínimo de la cabecera será $4 \cdot t_w + 3 \cdot t_r$
La distancia entre el nodo 0 y el 2 será 2

4. Prestaciones: Latencia, Ejemplo





Gracias.