

Tema 01

curso 2018/19

Introducción a las Redes de Ordenadores e Internet

Antonio J. Estepa Alonso

Departamento de
Ingeniería Telemática



Índice del Tema 01

1.1 Introducción a las Redes de Ordenadores

1.2 Interconexión en las redes de Ordenadores, conceptos básicos de enlaces e introducción a Internet

1.3 Definiciones básicas y Modelos de capas OSI

1.4 Multiplexión Estadística

1.3.1 Repaso de la multiplexión estadística

1.3.2 Retardo y pérdidas en las colas de los conmutadores de paquetes



Aplicaciones NO Distribuidas (el curso pasado)

- ❑ Todo ocurre dentro de una sola máquina
 - ❖ Nuestros programas sólo se comunicaban con los dispositivos de I/O (p.e. teclado, pantalla, o disco duro) ..
- ❑ Ejemplo: aplicación que realiza el servicio de sumar 2 números

PROCESO 120

```
#include<stdio.h>
main(){
  int a,b,suma;

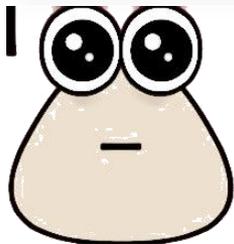
  do{
    scanf("%d %d",&a,&b);
    suma = a+b;
    printf("La suma es %d",suma);
  }while (1);
```

```
#include<stdio.h>
main(){
  int a,b,suma;

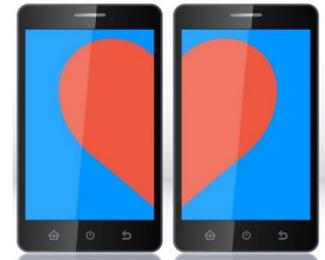
  do{
    scanf("%d %d",&a,&b);
    suma = hazlasuma(a,b);
    printf("La suma es %d",suma);
  }while (1);

  int hazlasuma(int a,int b)
  {
    return (a+b);
  }
```

○
más
modular



Aplicaciones
Tristes ...
solitarias ...



Este curso: Aplicaciones Distribuidas

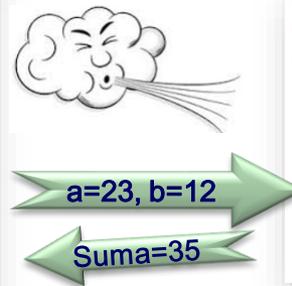
- Un programa se comunica con otro(s) programa(s) que pueden residir en otros equipos (remotos) para completar un servicio o tarea.
 - ❖ requiere de **funciones/métodos** que proporcionen un **servicio de envío/recepción de mensajes entre procesos de aplicación**.

PROCESO 120

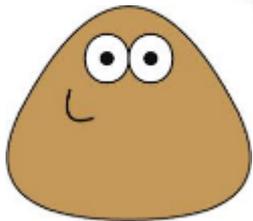
```
#include<stdio.h>
#include<red.h>
main(){
  int a,b,suma;
  do{
    scanf("%d %d",&a,&b);
    ENVIA(ORDENADOR B:PROCESO 14,a,b);
    RECIBE(suma);
    printf("La suma es %d",suma);
  }while (1);
```

PROCESO 14

```
#include<stdio.h>
#include<red.h>
main(){
  int a,b,suma;
  do{
    RECIBE(a,b);
    suma=a+b;
    ENVIA(ORDENADOR A:PROCESO 120, suma);
  }while (1);
```



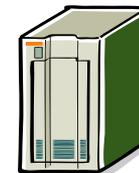
¿Pou lo nota?



ORDENADOR A



¿Conoces alguna app distribuida? ¿cuáles eran los ordenadores?

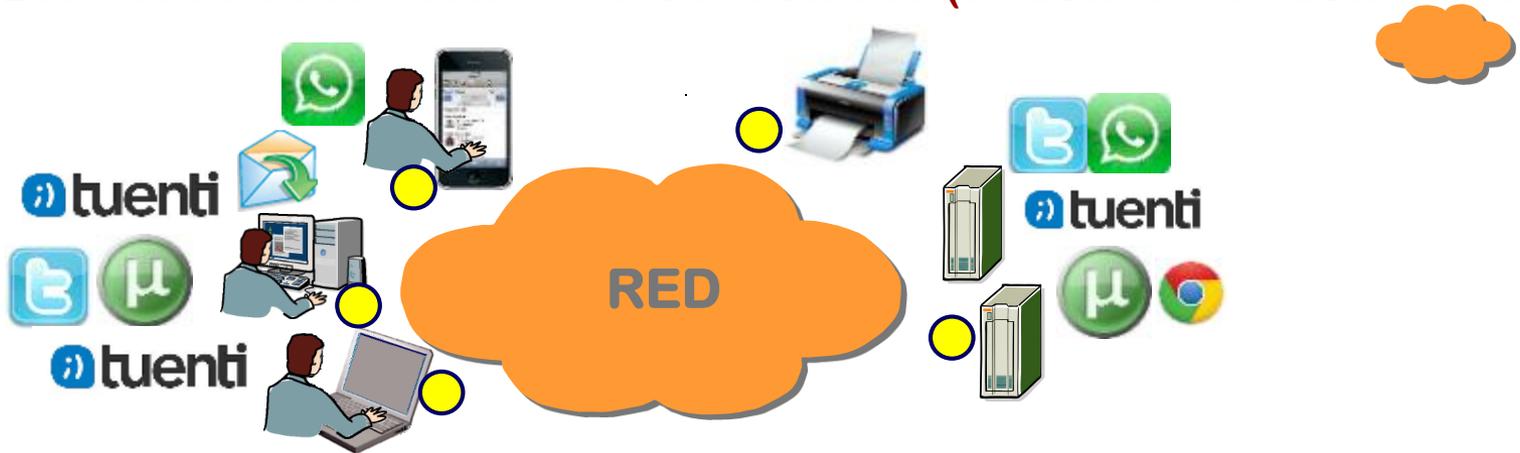


ORDENADOR B



¿Qué es una red de Ordenadores?

- ❑ Una red de ordenadores es un conjunto de ordenadores o equipos informáticos conectados entre sí que pueden intercambiar información. (definición formal D.R.A.E.L.)
 - ❖ Equipos informáticos (nodos): PC, tablet, impresora, etc. ●
 - ❖ Aplicaciones distribuidas → intercambio de información
 - ❖ Los nodos deben estar interconectados (directa o indirectamente)



- ❑ ¿Qué es la red?
 - ❖ Sistema de interconexión que posibilita la comunicación entre aplicaciones que residen en diferentes ordenadores



Índice del Tema 01

1.1 Introducción a las Redes de Ordenadores

1.2 Interconexión en las redes de Ordenadores, conceptos básicos de enlaces e introducción a Internet

1.3 Multiplexión Estadística

1.3.1 Repaso de la multiplexión estadística

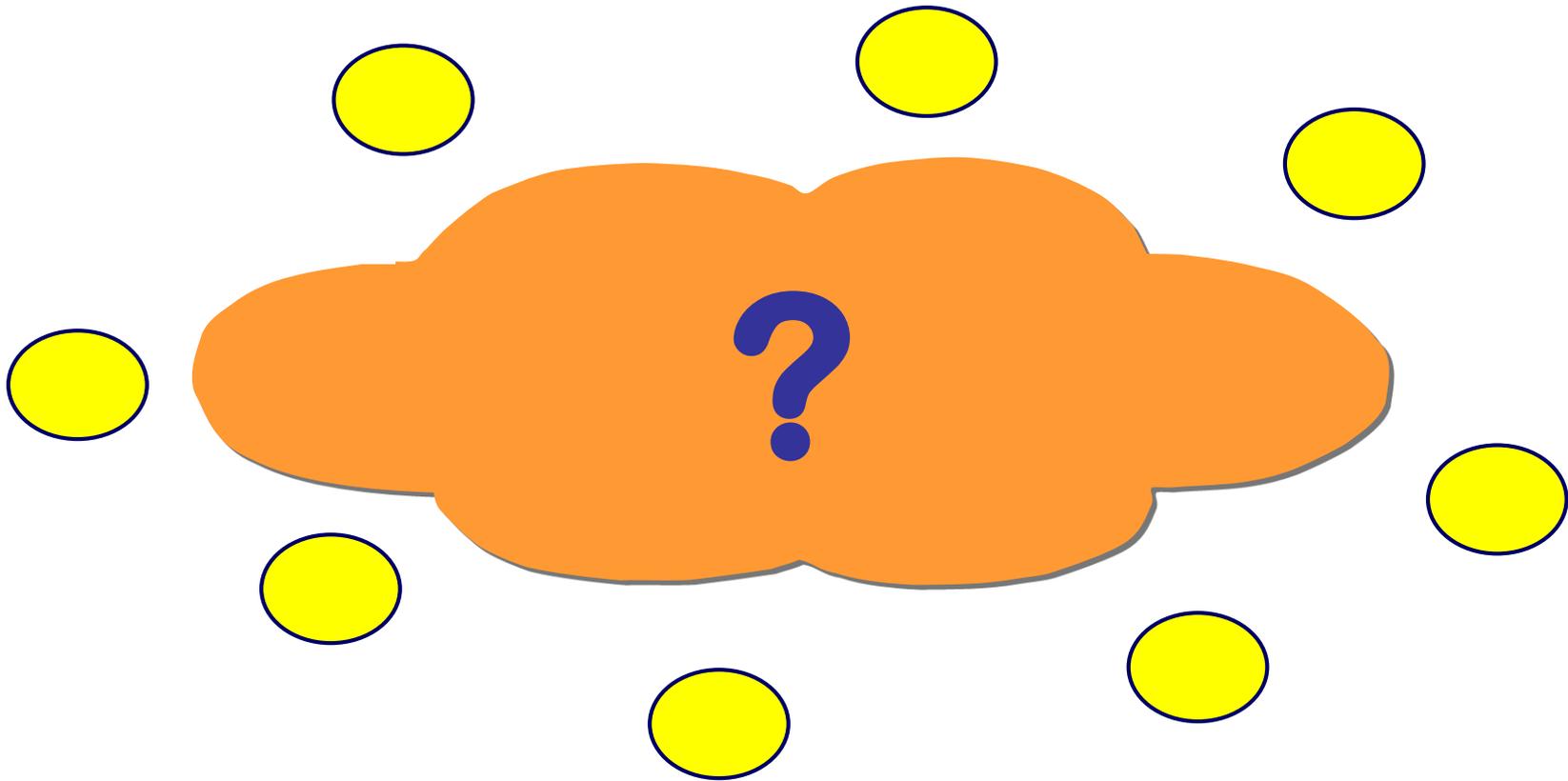
1.3.2 Retardo y pérdidas en las colas de los conmutadores de paquetes

1.4 Definiciones básicas y Modelos de capas OSI



Red: sistema de interconexión

- ❑ Sistema: Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto (D.R.A.E.)

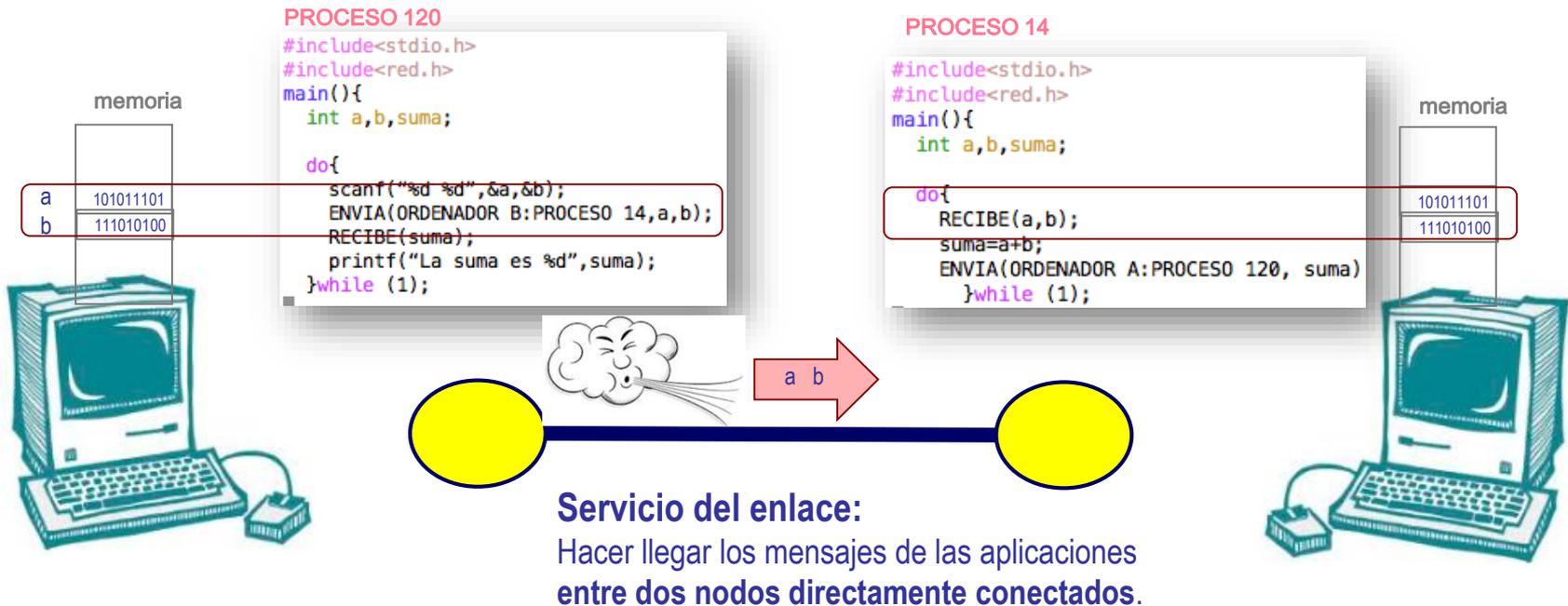


- ❑ ¿de qué formas podríamos interconectar los nodos?

4 posibles Formas de Interconexión: forma 1

1. Conectar directamente dos nodos mediante un enlace

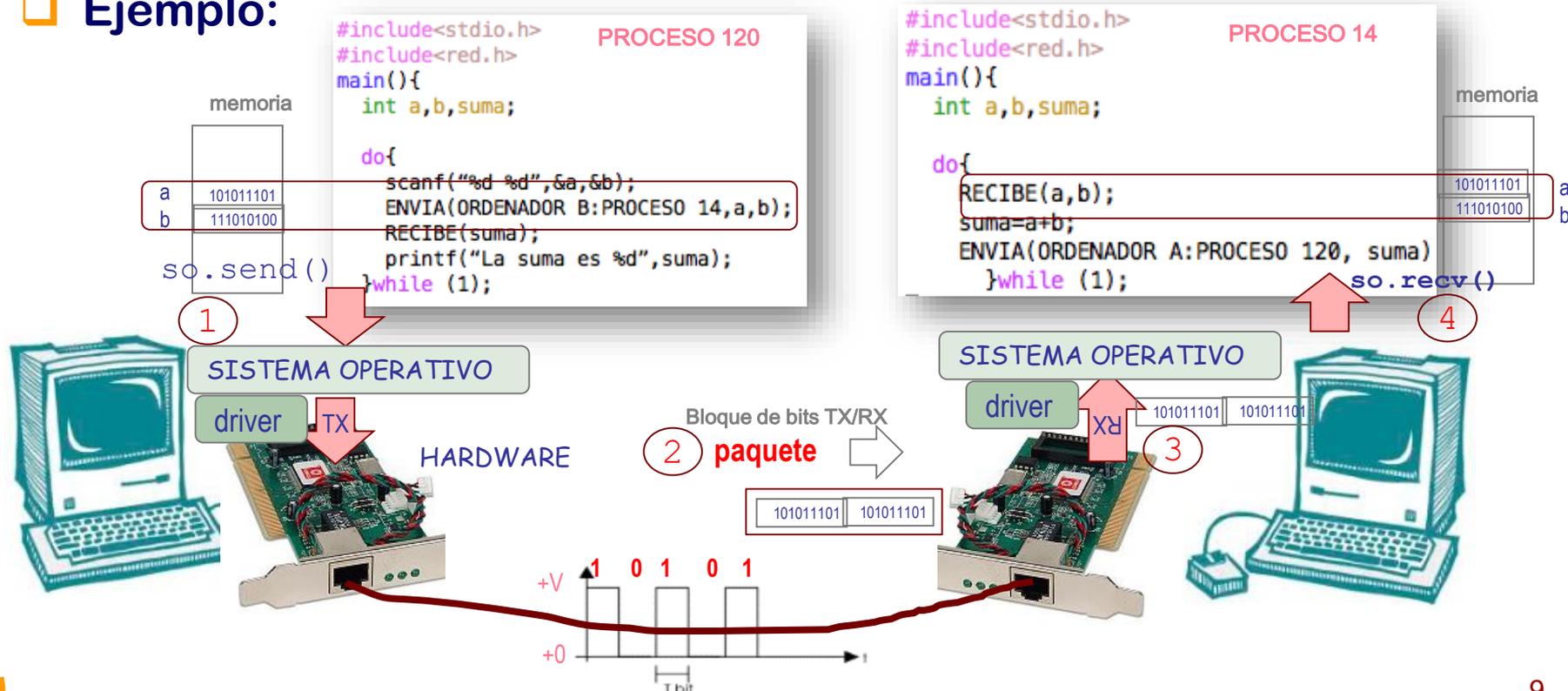
- ❖ Los procesos envían / reciben datos a través de la red (enlace)
- ❖ Equipo emisor:
 - Al ejecutar la función **ENVIA ()** , escribe bytes en el enlace
- ❖ Equipo receptor:
 - Al ejecutar la función **RECIBE ()** , lee bytes en el enlace



Funcionamiento Básico de un enlace punto a punto

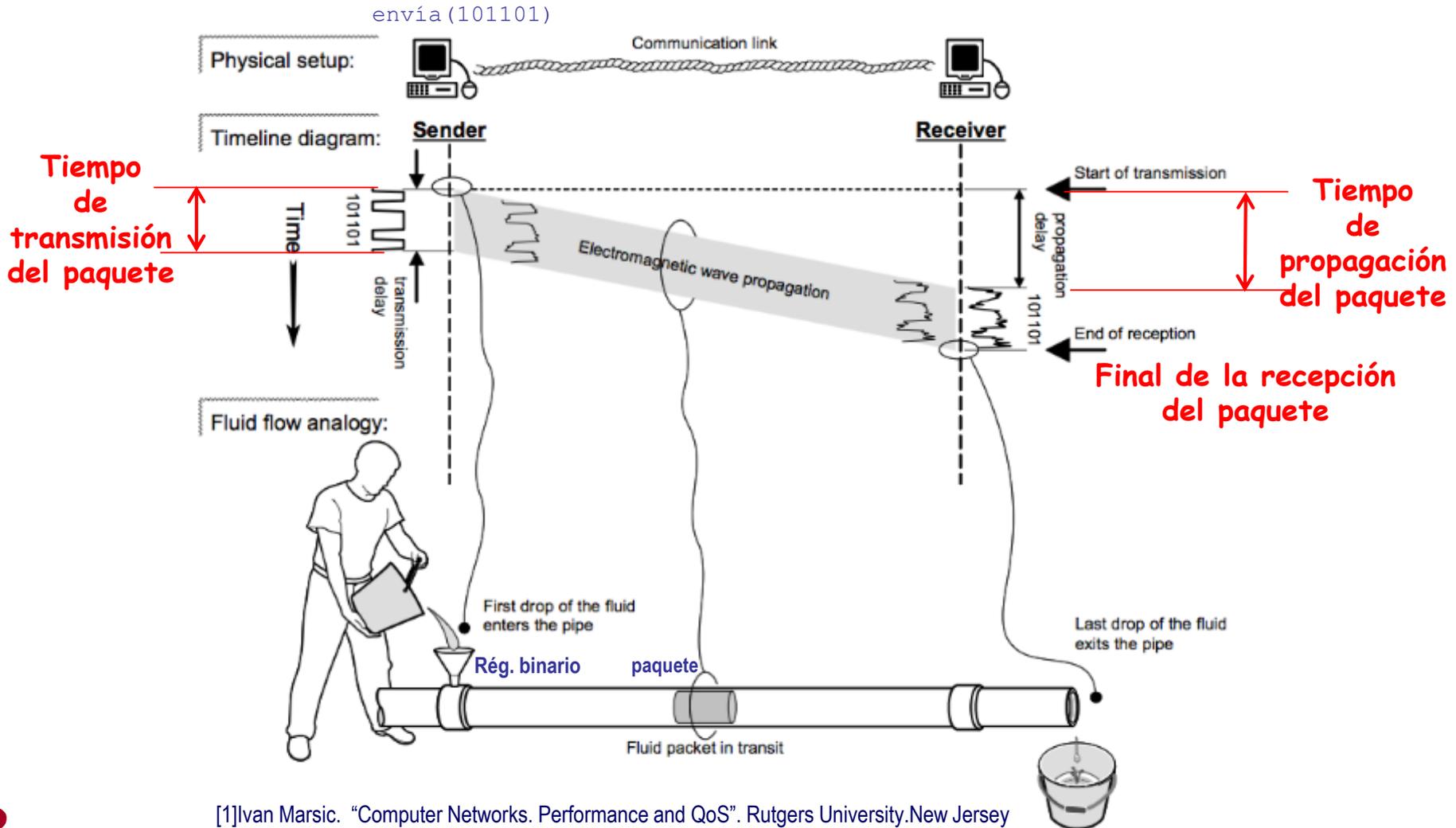
- ❑ Tarjeta de red (Network Interface Card, NIC): periférico que envía/recibe datos digitales a través de un enlace
 - ❖ Transmite (TX) / Recibe (RX) señales (radio)eléctricas que representan bits (código línea)
 - ❖ Régimen binario: velocidad a la que la NIC escribe/lee bits (bit/seg)

❑ Ejemplo:



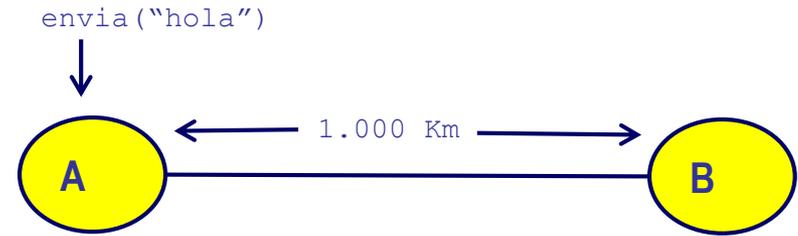
Transmisión física por un enlace: conceptos clave

□ Tiempos de transmisión y propagación (Analogía de Fluidos)



Ejemplo 1

- ❑ El nodo A tiene una NIC con régimen binario de 10 kbit/s
- ❑ La app. que se ejecuta en el nodo A quiere transmitir el mensaje “hola”
 - ❖ ¿tiempo de transmisión del paquete?
 - ❖ ¿cuánto tiempo transcurre desde que A comienza a transmitir el primer bit hasta que el último bit llega a B?
 - (vel. propagación señal de 10^5 Km/seg)
 - ❖ ¿cuál será el rég. binario de la NIC del nodo B?
 - ❖ Recalcular si el mensaje tuviese 4KiB



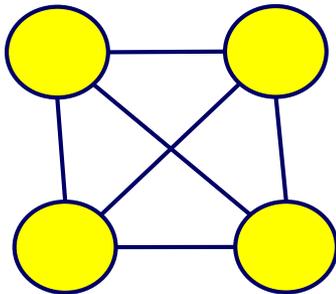
Ojo con las unidades

Unidades de información (del byte)			
Sistema Internacional (decimal)		ISO/IEC 80000-13 (binario)	
Múltiplo (símbolo)	SI	Múltiplo (símbolo)	ISO/IEC
kilobyte (kB)	10^3	kibibyte (KiB)	2^{10}
megabyte (MB)	10^6	mebibyte (MiB)	2^{20}
gigabyte (GB)	10^9	gibibyte (GiB)	2^{30}

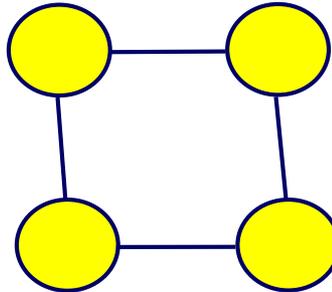


Limitaciones de los enlaces punto-a-punto

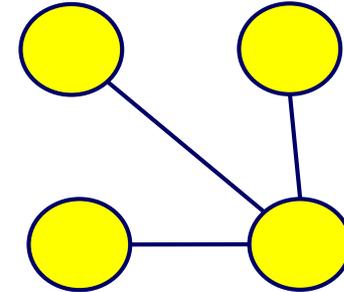
- ❑ DISTANCIA
- ❑ COSTOSO si se desea interconexión directa entre todos!
 - ❖ SÓLO INTERCONECTAN DOS PUNTOS !!



Topología
mallada



Topología
anillo



Topología
estrella

4 posibles formas de interconexión: forma 2

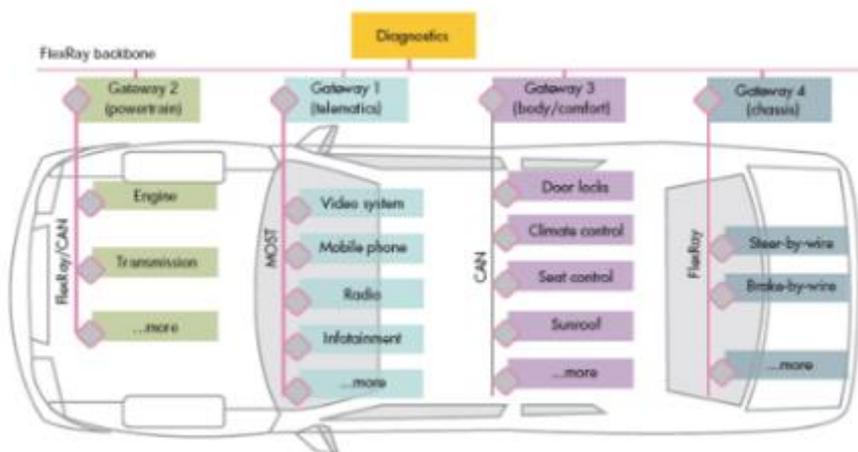
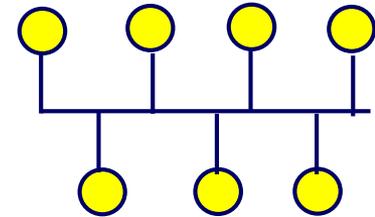
2. Conectar directamente múltiples nodos (enlace acceso múltiple)

❖ Medio físico compartido (cable o radio)

- En régimen binario del enlace se “reparte”
 - Repartición prefijada vs normas de uso

❖ Uso típico:

- Redes cableadas industriales (ahorra coste)
- Redes Inalámbricas
- (obsoleto) en redes domésticas o de oficina (hub)

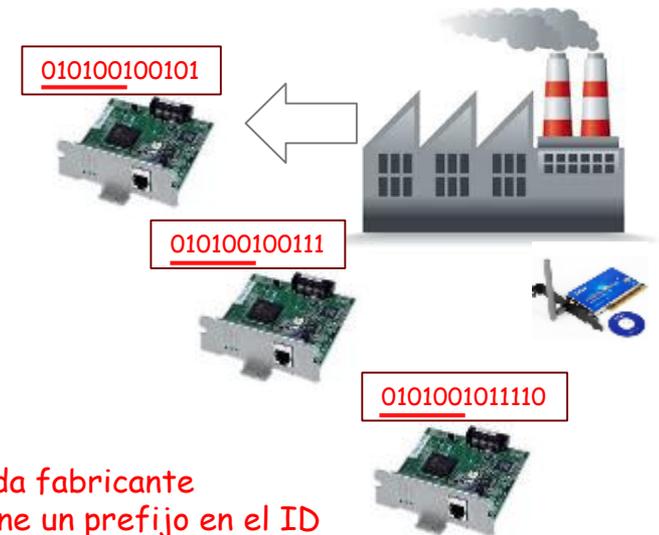
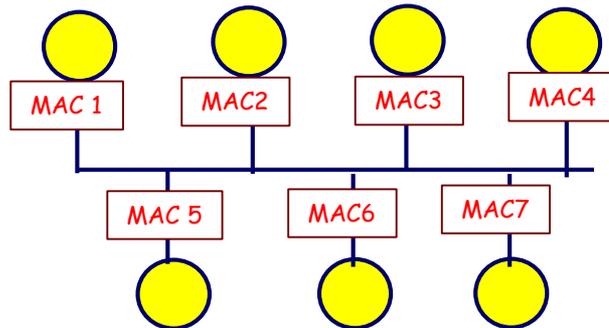


4 posibles formas de interconexión: forma 2

2. Enlaces de acceso múltiple

❖ Requieren

- Uso “ordenado” del enlace (posibilidad de colisión)
 - P.ej: turnos, acceso aleatorio, reserva, etc..
- Identificar del nodo destino (y origen) de cada transmisión
 - Cada NIC trae grabada de fábrica una dirección física única
 - Varios estándares: p.ej. IEEE 802.11, IEEE 802.3 → direcciones MAC (6bytes)



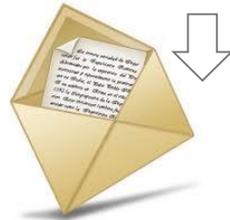
Cada fabricante
tiene un prefijo en el ID

<https://standards.ieee.org/develop/regauth/oui/oui.txt>

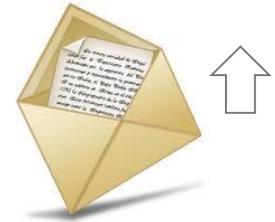
Concepto de Encapsulación / Desencapsulación

- Es **clave** en las redes de datos

Encapsular



Desencapsular



Ejemplo en C

```
char datos[10]="Pepito";
char datosEncapsulados[20];

sprintf(datosEncapsulados,"Saludar %s",datos);

envia(NodoB, Procesos12, datosEncapsulados);
```

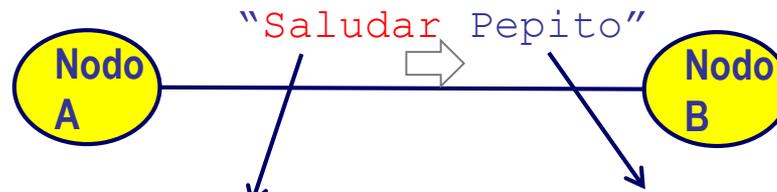
```
char cabecera[10];
char datos[10];
char mensajeencapsulado[20];

recibe(&mensajeencapsulado);
scanf(mensajeencapsulado,"%s %s",cabecera,datos);
if(!strcmp(cabecera,"Saludar"))
    printf("Hola %s",datos);
if(!strcmp(cabecera,"Despedir"))
    printf("Adios %s",datos);
```

...de otra forma ...

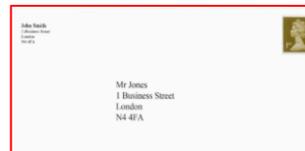
```
char str[80];
strcpy(str, "Saludar");
strcat(str, "Pepito");

envia(NodoB,procesos12, str);
```



información de control
(cabecera)

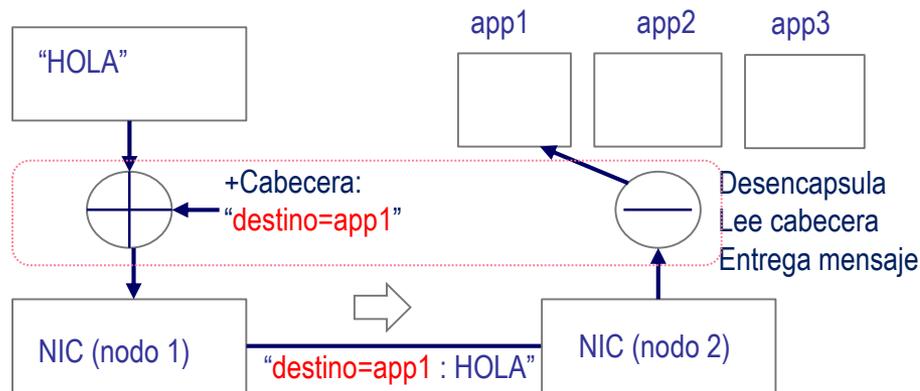
mensaje
(datos)



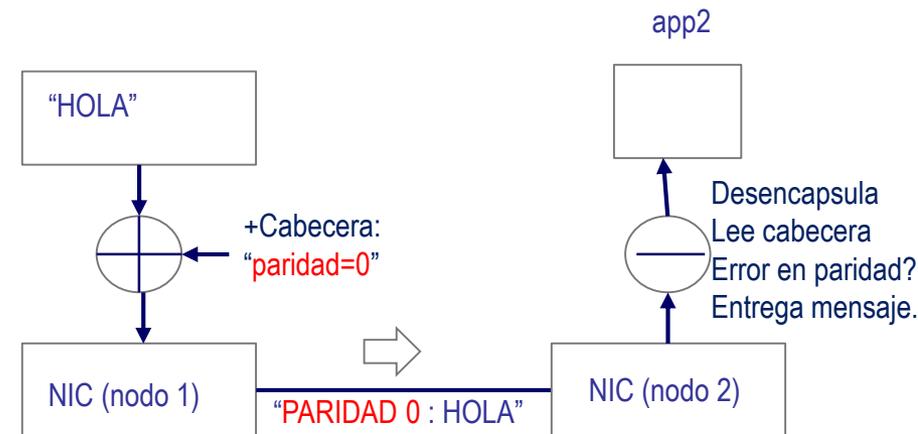
Utilidad de la encapsulación / desencapsulación

- ❑ Origen → añade cabecera (encapsula)
- ❑ Destino → lee cabecera (desencapsula) y actúa en consecuencia
- ❑ Según la información que pongamos en la cabecera se consiguen cosas útiles... **IDEA IMPORTANTE** (=la rueda en telemática)!

Ejemplo 2: desmultiplexión



Ejemplo 1: comprobación integridad



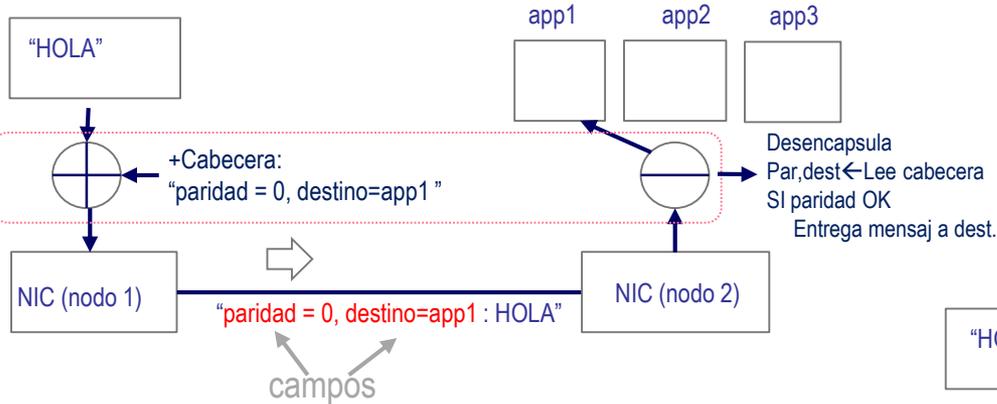
Protocolo : define las cabeceras y las acciones a realizar

↓ : llamada a la función/método de encapsulación (p.ej. Encapsula("HOLA","app1"))

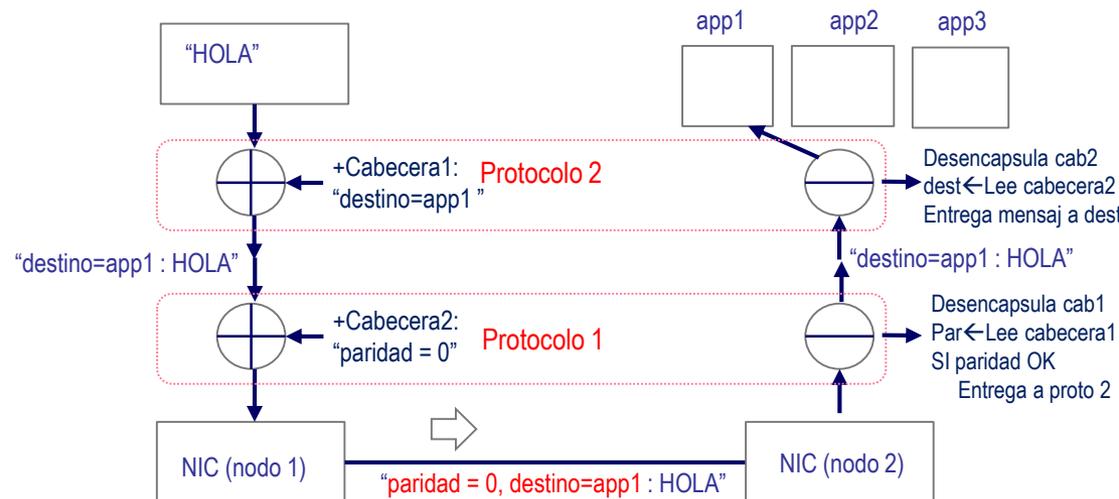


Pregunta: ¿podríamos hacer Ejm1 y Ejm2 a la vez?

- ❑ **Opción 1:** un protocolo cuya cabecera tuviese 2 campos



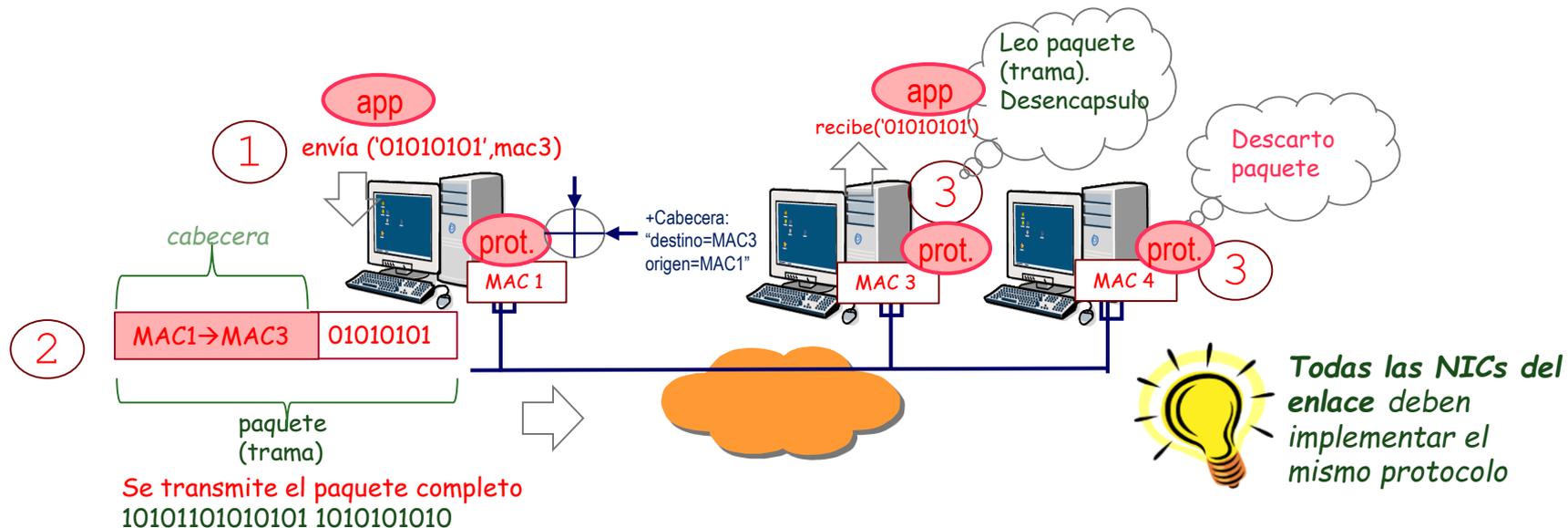
- ❑ **Opción 2:** los dos protocolos anteriores uno sobre otro (dependencia entre ellos)



Ejemplo de funcionamiento básico de un enlace de acceso múltiple (muy simplificado)

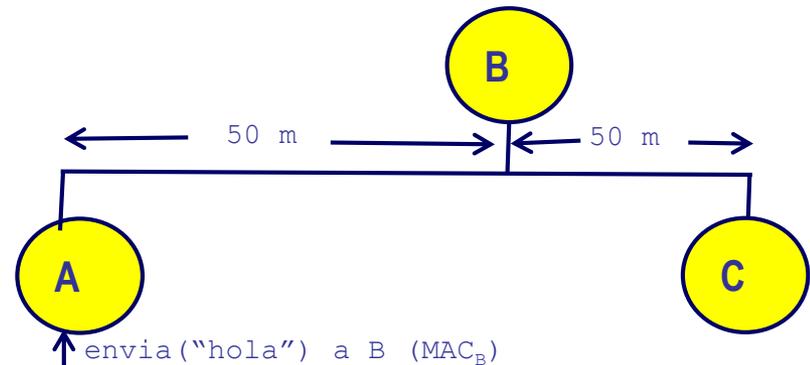
Programa implementado en las NICs (firmware): protocolo de enlace

- ❖ 1. Antes de enviar un mensaje, añade la dirección física destino y origen (encapsula) y TX el mensaje encapsulado por el enlace (genera paquete)
 - Arbitra un método de acceso ordenado y retransmisión ante colisión
- ❖ 3. Al recibir RX un nuevo paquete, lee la cabecera (desencapsula)
 - Si la dirección destino coincide con la dirección de la propia NIC, (saca la carta del sobre) y entrega los datos a la app
 - Si no, lo descarta.



Ejemplo 2

- ❑ El nodo A tiene una NIC con régimen binario de 100 Kb/s y protocolo enlace con cabecera de 12 bytes
- ❑ La app. que se ejecuta en el nodo A quiere transmitir el mensaje “hola” al nodo B
 - ❖ ¿cuánto tiempo transcurre desde que A comienza a transmitir el primer bit hasta que el último bit llega a B?
 - (despreciar t. propagación)
 - ❖ Recalcular si A envía a C
- ❑ Si todos los nodos quieren siempre transmitir 100B por turnos, ¿qué tasa de bit tendría cada aplicación?



Limitaciones de los enlaces de acceso múltiple

❑ DISTANCIA

- ❖ Máxima del enlace (por atenuación y detección de colisión)

❑ NÚMERO MÁXIMO DE NODOS INTERCONECTADOS

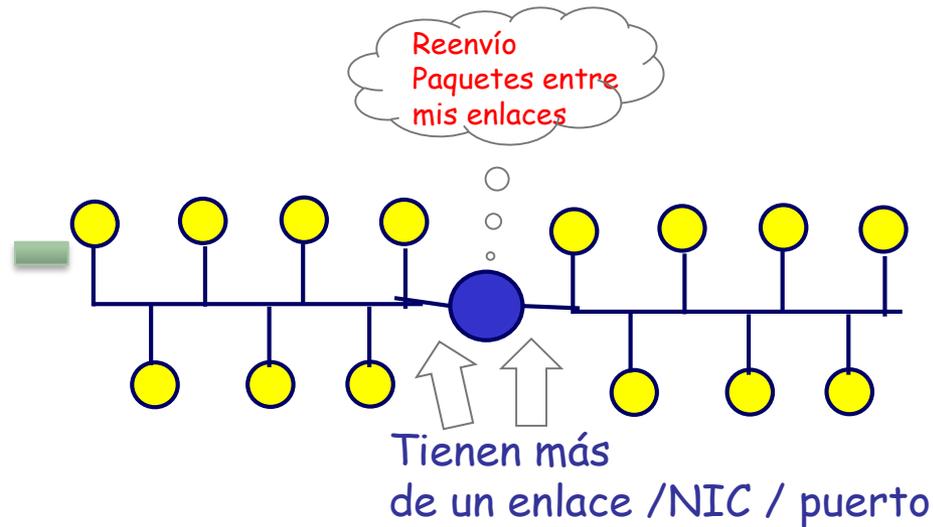
- ❖ Las prestaciones del servicio de enlace se degradan mientras mas nodos se interconecten.

Necesitamos otra forma de interconexión que mejore esto



4 posibles formas de interconexión: forma 3

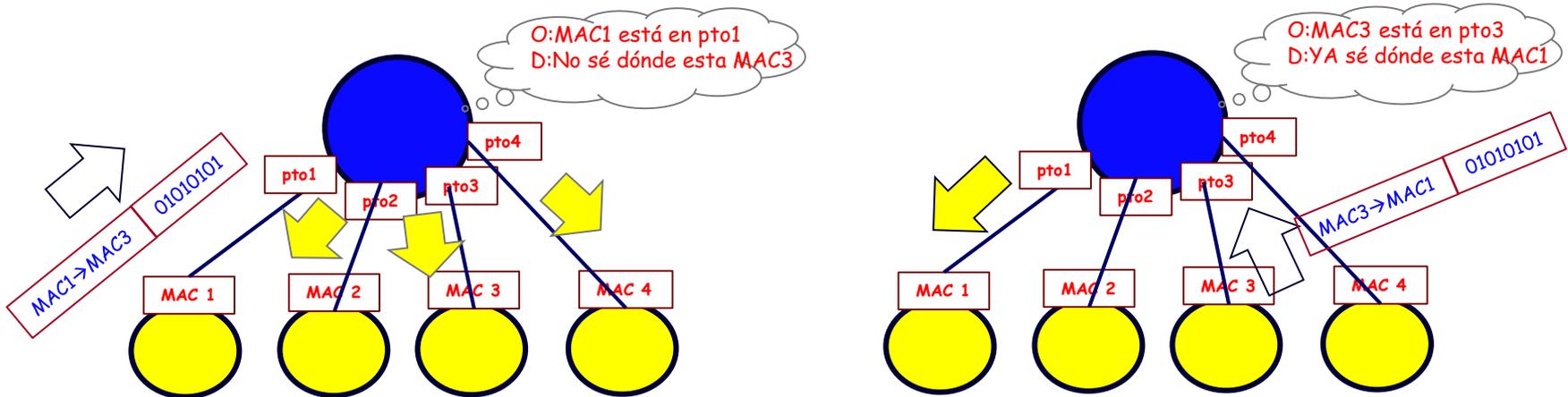
- ❑ Interconexión **INDIRECTA** de los nodos (conmutador de enlace)
 - ❖ Uso de **Nodos Especiales** dedicados sólo a la interconexión de enlaces (puentes)



- ❑ **Ventajas:**
 - ❖ aumenta el número de equipos interconectados
 - ❖ Aumenta la extensión geográfica de la red

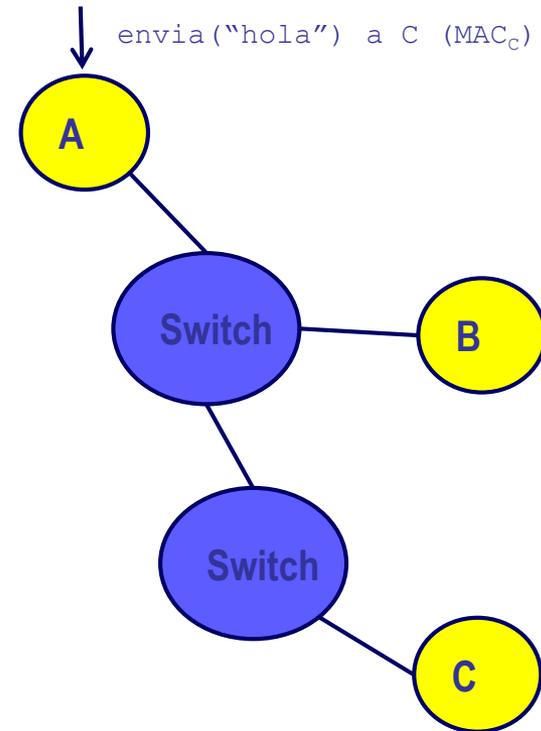
Conmutadores de nivel de enlace (conmutador, puente)

- ❑ Implementan una lógica de reenvío (“algoritmo de reenvío”)
 - ❖ Cada vez que recibe un paquete (completo), examina la cabecera del protocolo de enlace (direcciones MAC origen y destino)
 - Aprende y recuerda el puerto (enlace) al que pertenece cada nodo (mira la dirección MAC origen de la trama)
 - ❖ Reenvían el paquete por el puerto donde saben (han aprendido previamente) que está el destino. Si aún no lo han aprendido, lo reenvían por todos sus puertos.
- ❑ Son “invisibles” para el resto de nodos (aprenden solos)



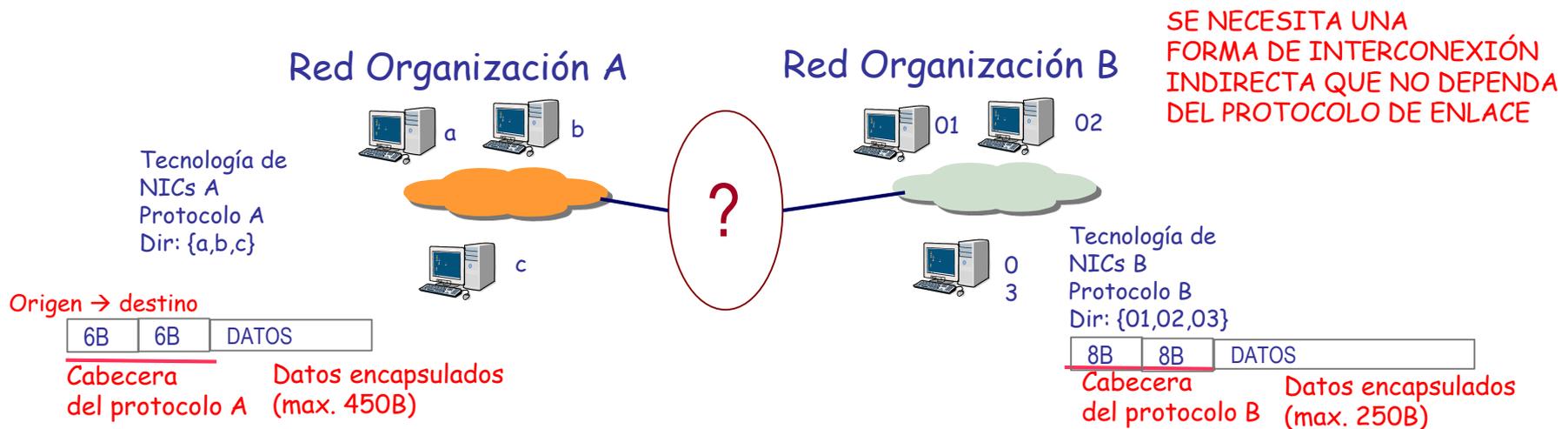
Ejemplo 3

- ❑ El nodo A tiene una NIC con régimen binario de 100b/s y protocolo enlace con cabecera de 12 bytes (MAC_o, MAC_d)
- ❑ La app. que se ejecuta en el nodo A quiere transmitir el mensaje “hola” al nodo C
- ❑ ¿cuánto tiempo tardaría en llegar el mensaje completo a C?
 - ❖ **suponga que los switches**
 - no han aprendido nada
 - el proceso de reenvío tarda 1 ms
- ❑ Recalcular $C \rightarrow A$ si la app que se ejecuta en C decide responder con un mensaje de 1MB.



Interconexión de redes heterogéneas

- ❑ Diferentes tipos de NICs implementan diferentes protocolos
 - ❖ Algunos protocolos (IEEE802.3, IEEE802.11, HDLC, PPP, ATM, ..)
- ❑ Si 2 organizaciones usan NICs con protocolos diferentes ¿podrían interconectar sus redes directamente?
 - ❖ ¿valdría un enlace? ¿valdría un conmutador de enlaces?
- ❑ ¿Qué problemas presentaría la interconexión de redes con tecnologías de enlace/protocolos diferentes?



4 posibles formas de interconexión: forma 4 (inter-red)

❑ Idea: nodo especial de interconexión de redes (router, gateway, encaminador) 

❖ Tiene tantos enlaces como redes interconecta

- Cada enlace pertenece a una red (NICs heterogén.)

❖ Reenvía paquetes desde una red hacia otra red

❖ No es transparente

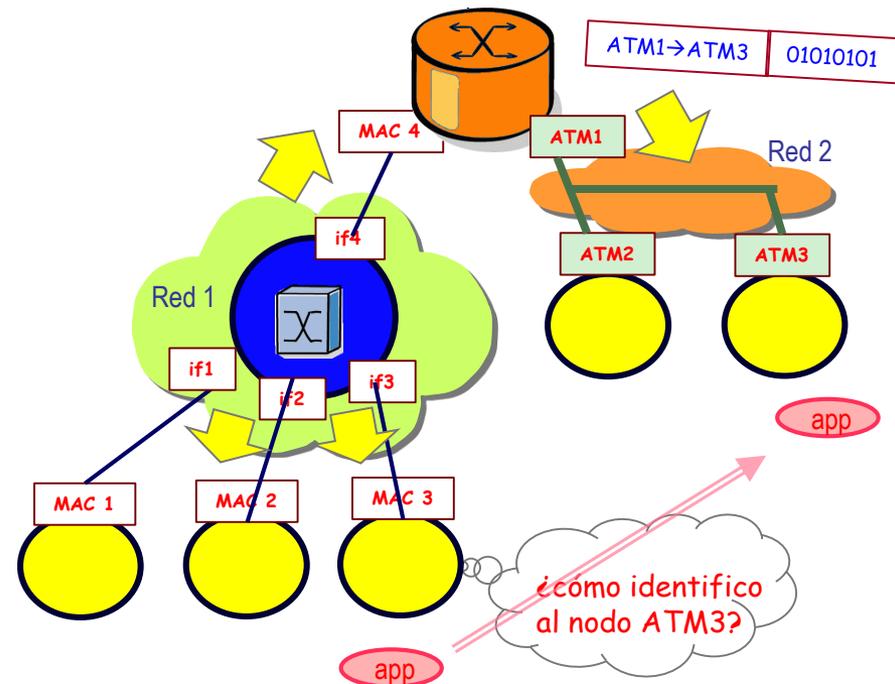
❑ Inter-red: red resultante de la interconexión de redes independientes

- ❖ Alternativamente: red computada de sub-redes
- ❖ Interconexión indirecta

Servicio de red (inter-red):

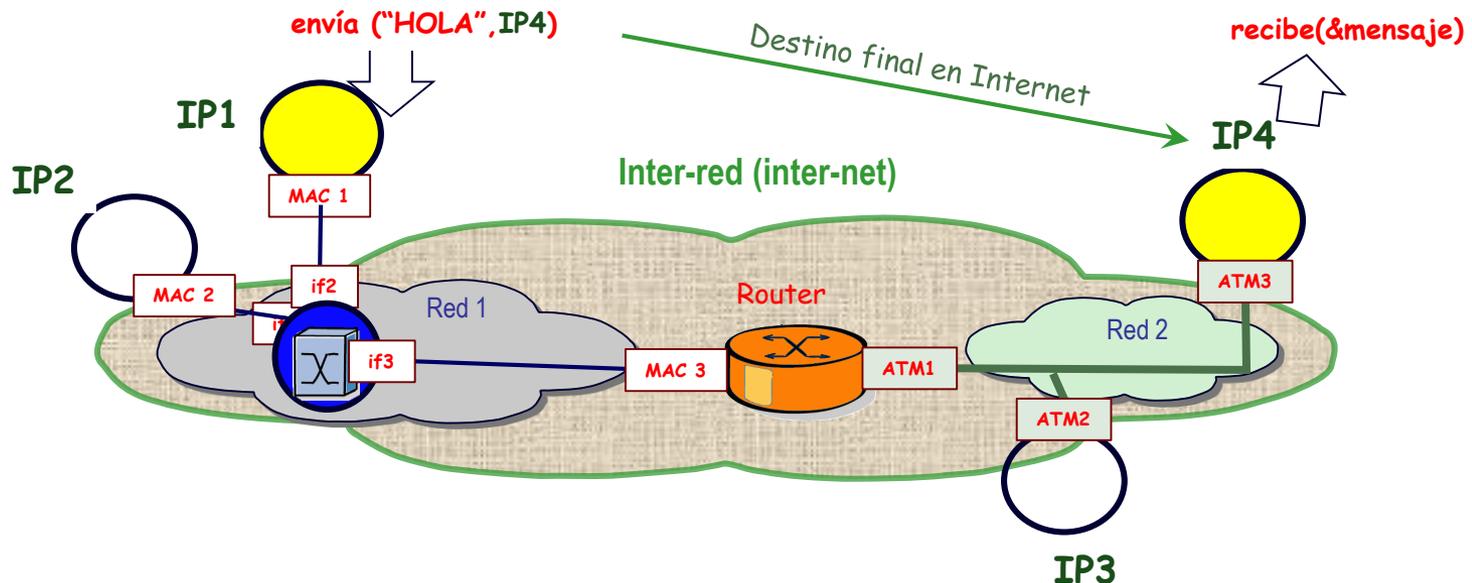
Hacer llegar los mensajes de las aplicaciones entre nodos de la inter-red (aunque estén en redes diferentes)

Inter-red = Red1 + Red 2



Inter-net: a nueva red (lógica) ... nuevas direcciones

- ❑ Necesidad de identificar nodos en la inter-red de forma homogénea: dirección IP (4 bytes en IPv4)
 - ❖ Las aplicaciones usan direcciones IP como destino / origen de su comunicación
 - ❖ Cada NIC en un nodo debería tener
 - una **dirección física** que identifica a una NIC en el enlace (acceso múltiple)
 - y otra **lógica (IP)**, que identifica al nodo en la inter-red

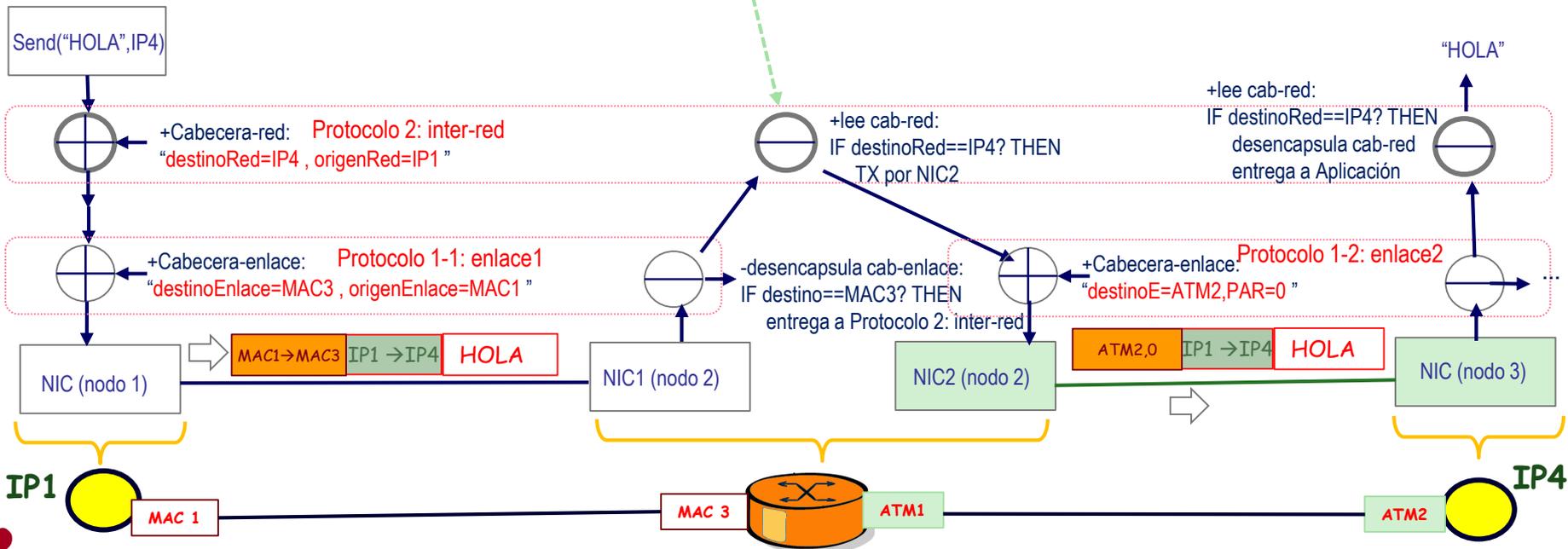


Inter-net: a nueva red (lógica) ... nueva lógica de reenvío

❑ Protocolo para el reenvío entre redes (Internet Protocol)

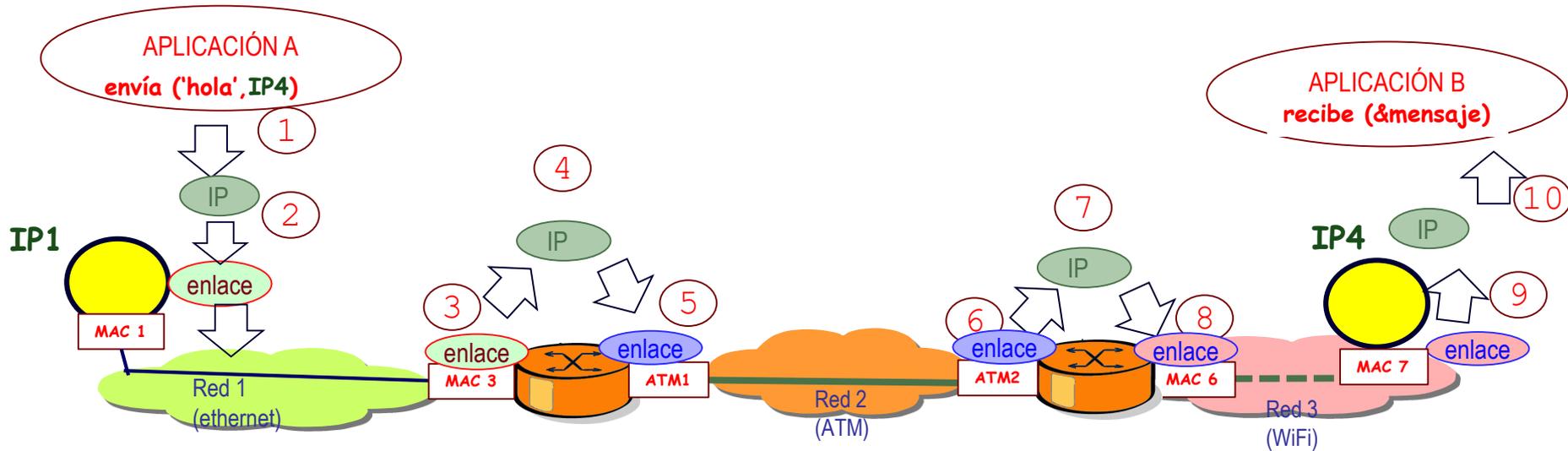
- ❖ Implementado por software en nodos y routers <https://github.com/torvalds/linux/tree/master/net/ipv4>
- ❖ Entre el protocolo de enlace y la app
 - Encapsula mensaje APP, a su vez, usa el servicio de enlace
 - Gestionado por el sistema operativo
- ❖ Implementa la lógica de reenvío entre redes (diferente a la lógica de las NICs)

EN ORIGEN: DOBLE ENCAPSULAMIENTO



Ejemplo 4

- Indique los pasos que seguiría el envío del mensaje “hola” desde la aplicación A a la B (encap. / desen)



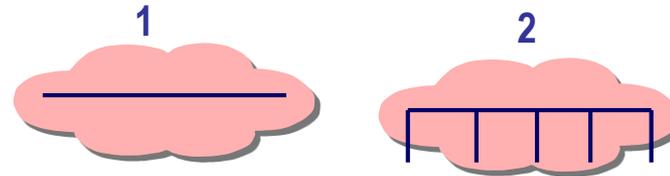
Resumen de las formas de Interconexión (redes)

❑ Red: sistema de interconexión que posibilita la comunicación

❑ Formas de Interconexión

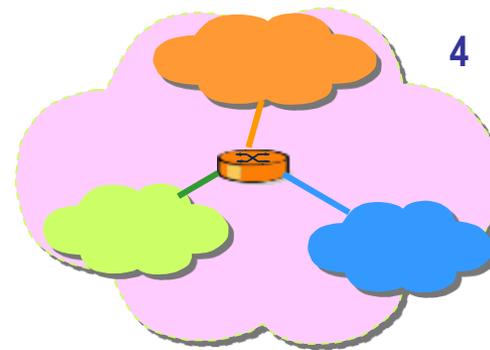
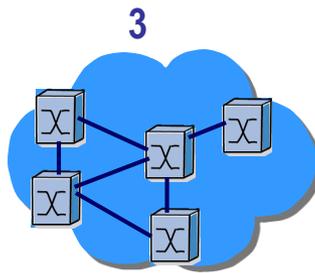
❖ Directa: uso de enlaces

- 1. punto-a-punto;
- 2. acceso múltiple



❖ Indirecta: uso de conmutadores de paquetes

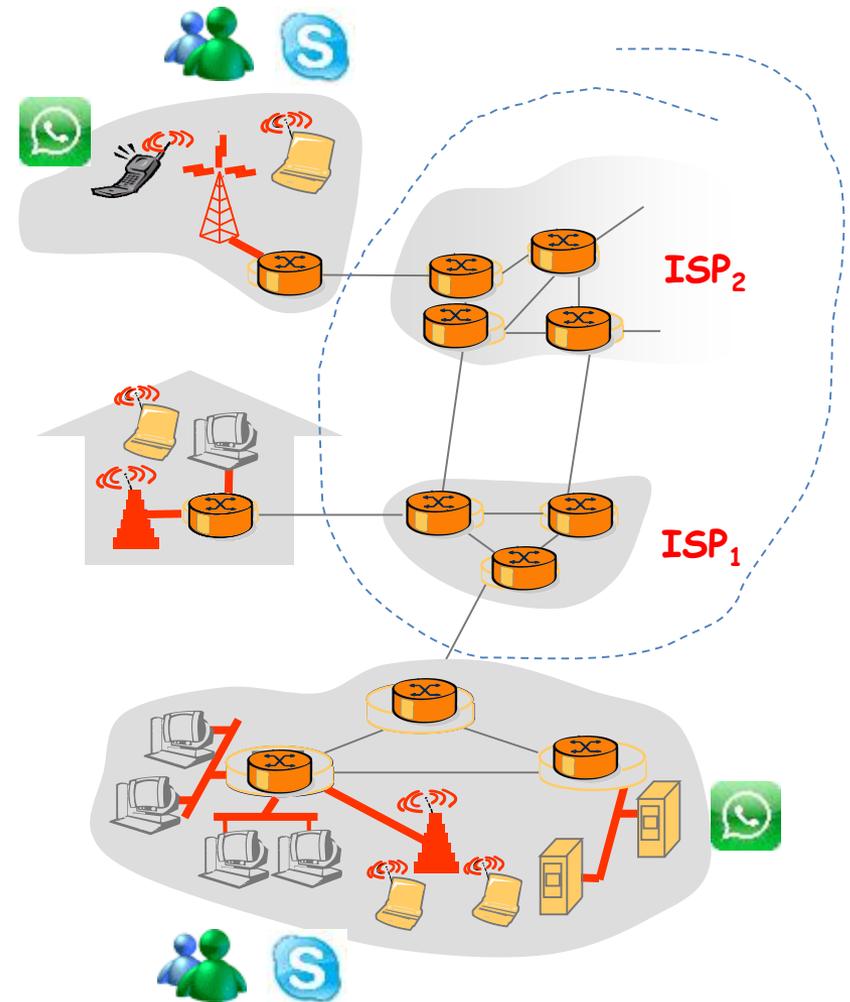
- 3. Redes Conmutadas (LAN): switches
- 4. Conjunto de redes conmutadas inter-net: routers



❑ Las 4 formas no son excluyentes

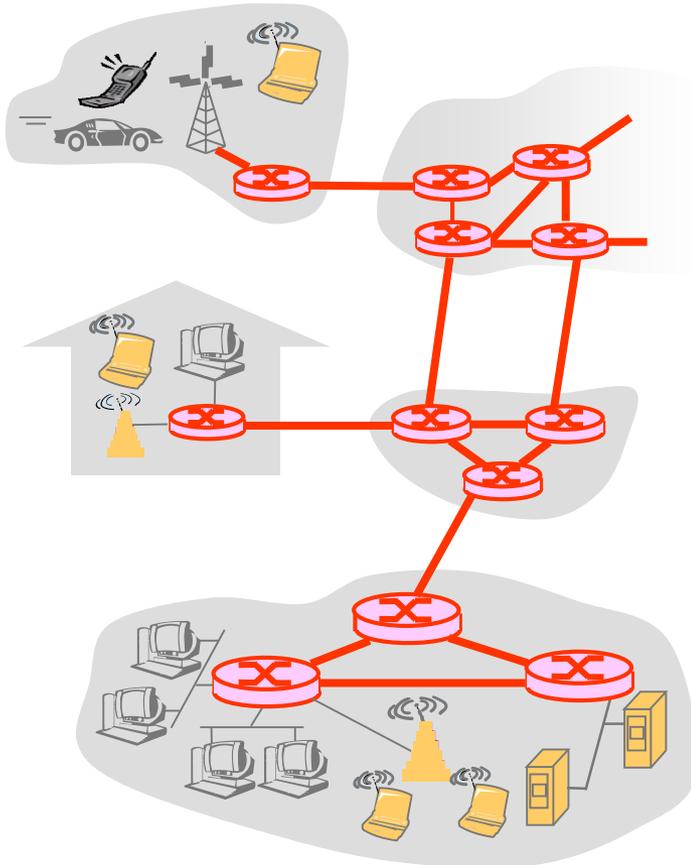
¿qué estructura tiene Internet? (visión de sistema)

- ❑ Internet es una inter-red con extensión mundial.
- ❑ Podemos distinguir entre
 - ❖ Borde de la inter-red : residen los hosts (nodos) (equipos ejecutando apps, p.e. Email, whatsapp)
 - ❖ Redes de Acceso: proporcionan a los hosts conexión con resto de Internet (p.e. red conmut., WiFi, 3G)
 - ❖ Redes del núcleo →
 - Conjunto de redes interconectadas a través de todo el mundo
 - Proveedores del servicio de Internet (ISPs).

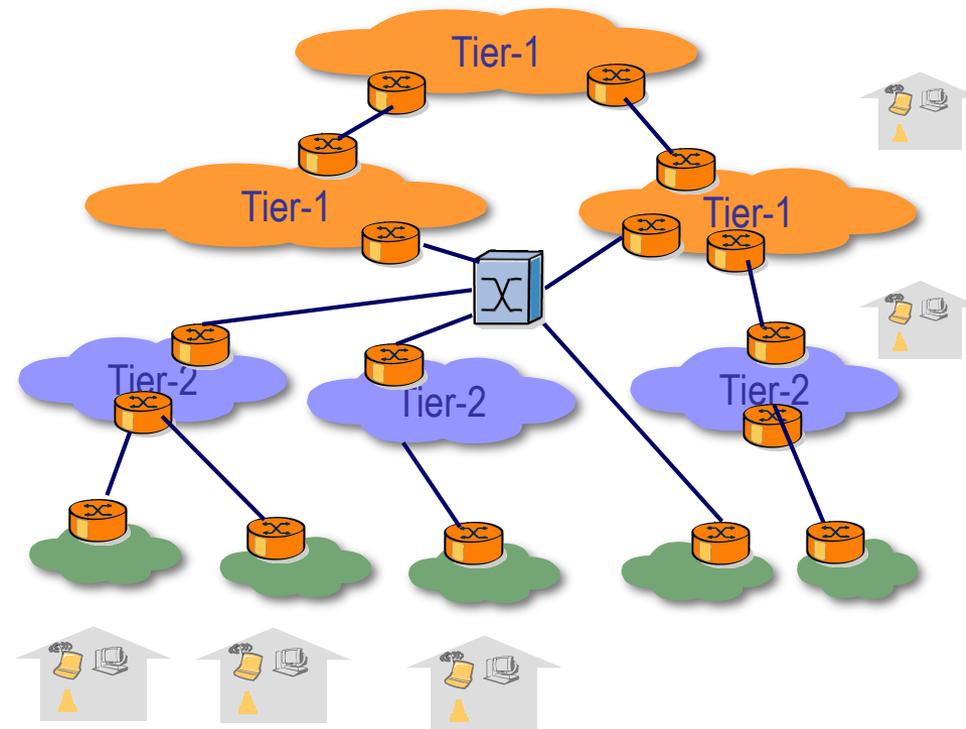


Estructura del Núcleo de Internet

- ❑ Enlaces comparten su régimen binario entre todos los usuarios de Internet
 - ❑ Mi ADSL sólo es el primer enlace !!



- ❑ ISPs de diferente tamaño. Jerarquía.
 - ❖ Los de menor tamaño se interconectan a través de los de un tamaño mayor (originalmente)
- ❑ Interconexión directa entre ISPs mediante puntos de Intercambio de tráfico (IXP)
 - ❖ Punto Neutro
 - ❖ Ver Espanix.



Organizaciones y Standares en el mundo de Internet

Organizaciones

https://en.wikipedia.org/wiki/Internet_organizations

Desarrollo y normativa

- IAB (Internet Architecture Board)
- IANA (Internet Assigned Numbers Authority)
- ICANN (Internet Corporation for Assigned Names and Numbers)
- IESG (Internet Engineering Steering Group)
- IETF (Internet Engineering Task Force)
- IRTF (Internet Research Task Force)
- ISOC (Internet Society)
- NANOG (North American Network Operators' Group)
- NRO (Number Resource Organization)
- W3C (World Wide Web Consortium)

Reserva

- RIR (registros Internacionales Regionales): RIPE

Investigación

- Internet2 (Universidades Americanas)

Empresas

- Amazon, Ebay, Google, Facebook, Wikipedia, LinkedIn, Twitter, Youtube, etc. ,

Standards

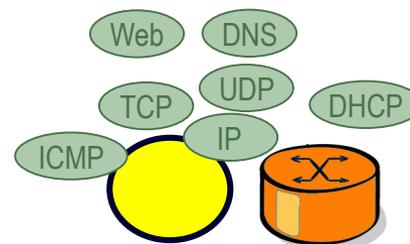
Request for Comments (RFCs)

<https://www.rfc-editor.org/rfc-index.html>

Definen protocolos, arquitecturas, aplicaciones, etc... (más de 7.000)

Gratuitos.

<https://www.rfc-editor.org/>



Índice del Tema 01

1.1 Introducción a las Redes de Ordenadores

1.2 Interconexión en las redes de Ordenadores y conceptos básicos de enlaces

1.3 Definiciones básicas y Modelo de capas OSI

1.4 Multiplexión Estadística

1.3.1 Repaso de la multiplexión estadística

1.3.2 Retardo y pérdidas en las colas de los conmutadores de paquetes



Definiciones básicas: Servicio y Protocolo

❑ **Servicio:** tarea que se realiza para un usuario

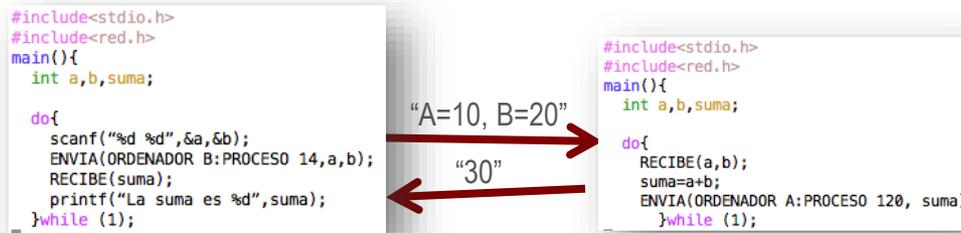
❖ Para prestar un servicio suele ser necesario que dos **entidades pares** colaboren

❑ Un **protocolo** define el formato, orden de los mensajes enviados y recibidos entre entidades pares y las acciones que se deben realizar al enviar o recibir cada mensaje

❖ OJO: un mismo servicio podría implementarse con protocolos diferentes

❑ **Ejemplos vistos:**

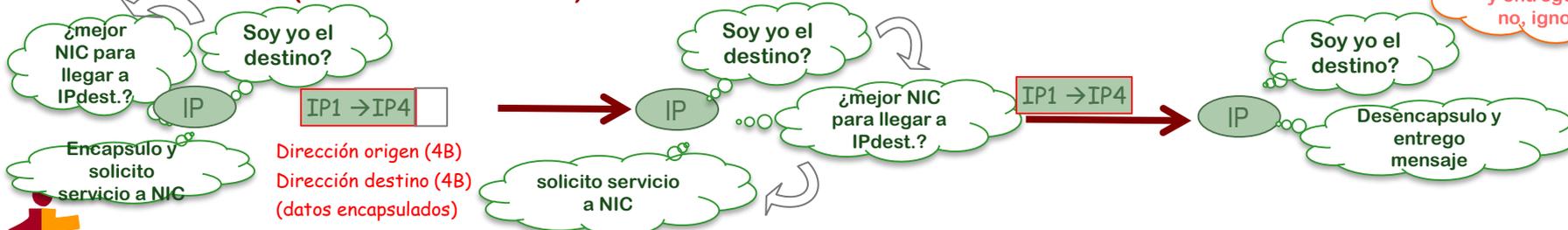
❖ **Aplicación (suma distribuida)**



❖ **Enlace (TX y RX a través de la NIC)**



❖ **Red (envío en Inter-red)**

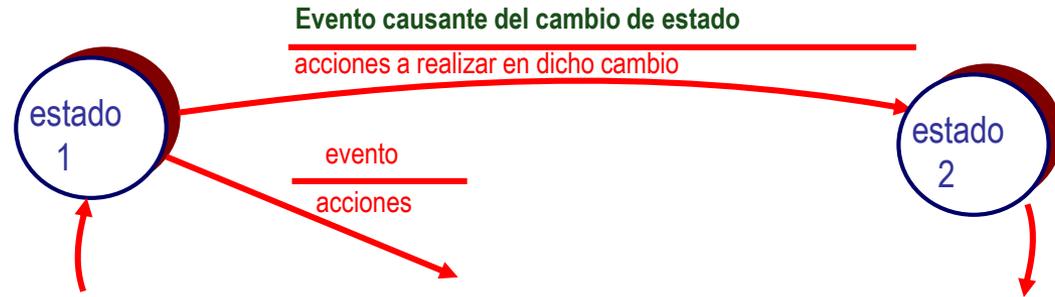


Especificación de un protocolo

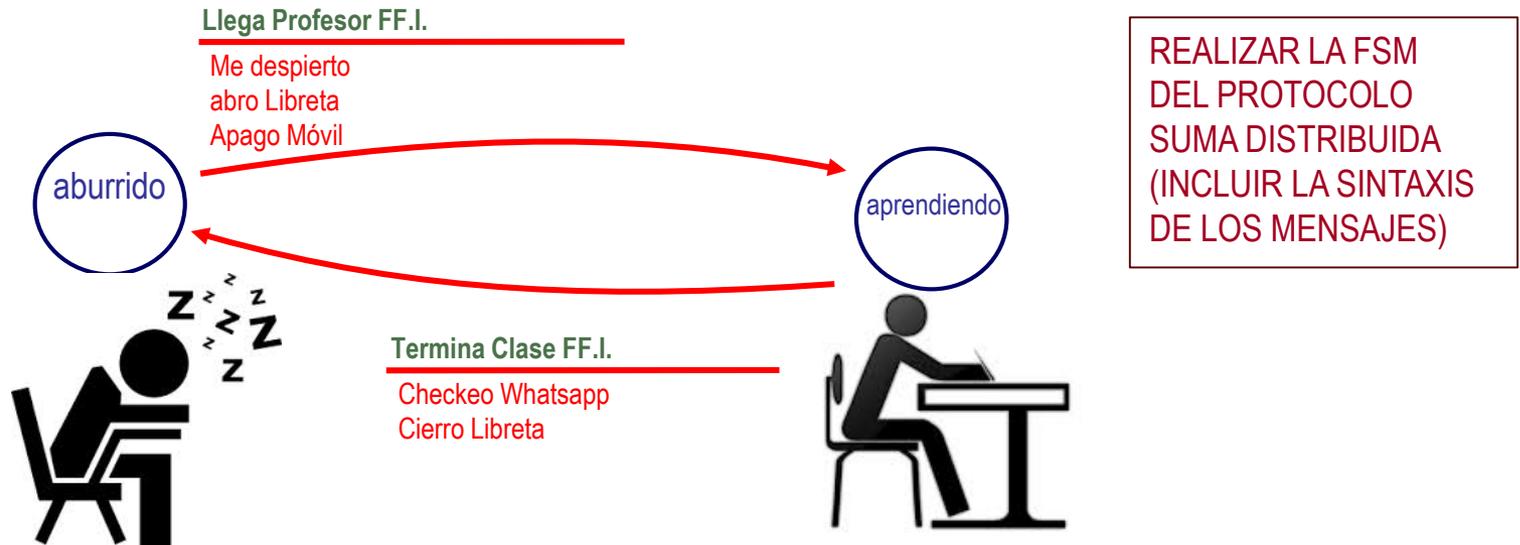
□ Un protocolo puede describirse de varias formas

- ❖ p.ej. con un diagrama de estados finitos (FSM) para cada extremo

estado: cuando se esta en un “estado” el siguiente evento determina el siguiente estado



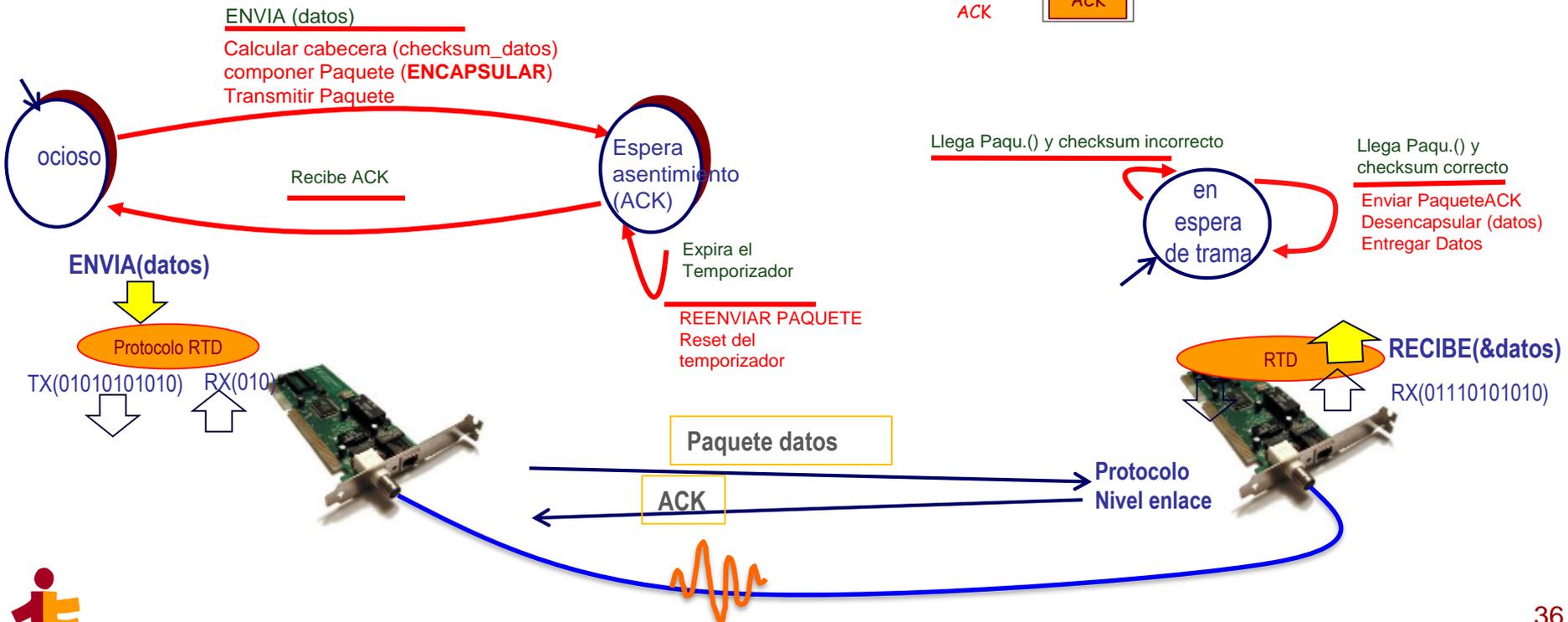
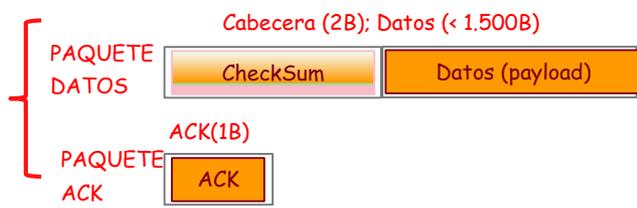
- ❖ Eventos y acciones en protocolo Fundamentos de Internet:



Ejemplo: Especificación de un Protocolo con FSM

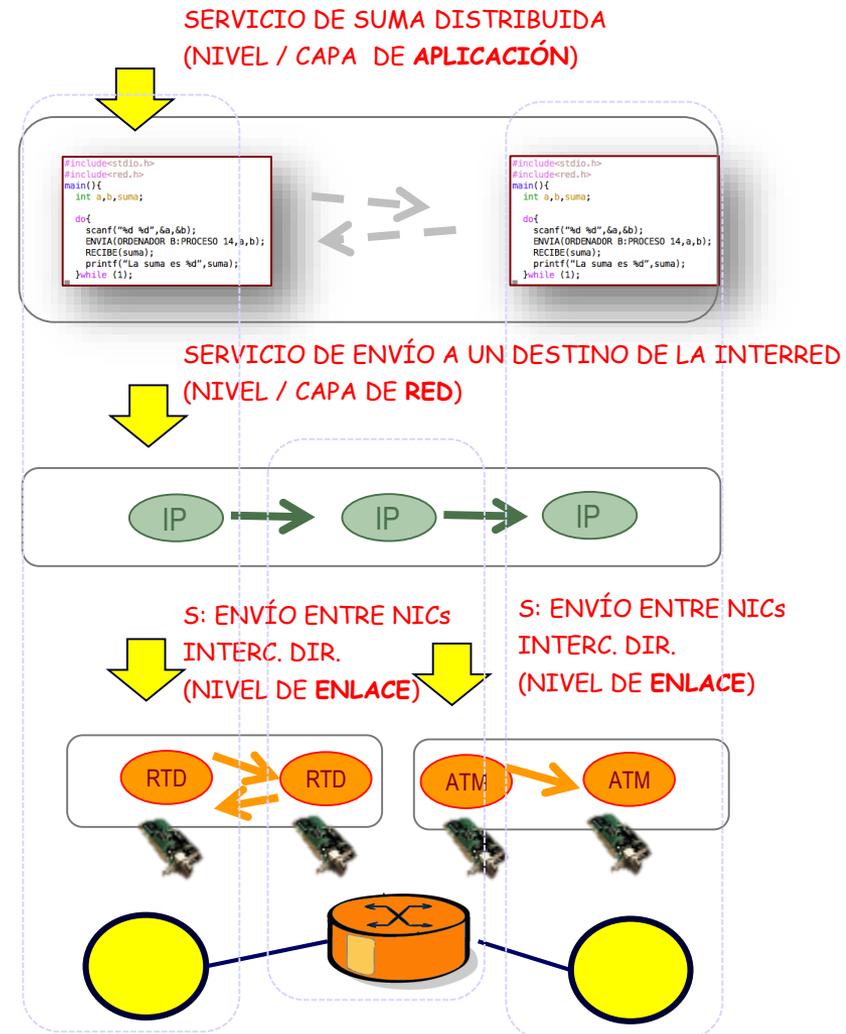
❑ Servicio: transferencia fiable dentro de un enlace

- ❖ **Fiable:** quien envía el mensaje tiene constancia de que ha llegado OK.
 - **Uso de asentimiento (ACK)** (... en realidad es más complejo)
- ❖ **Implementado en NIC**
- ❖ **Formato de mensajes del protocolo**



Modelo de Capas,

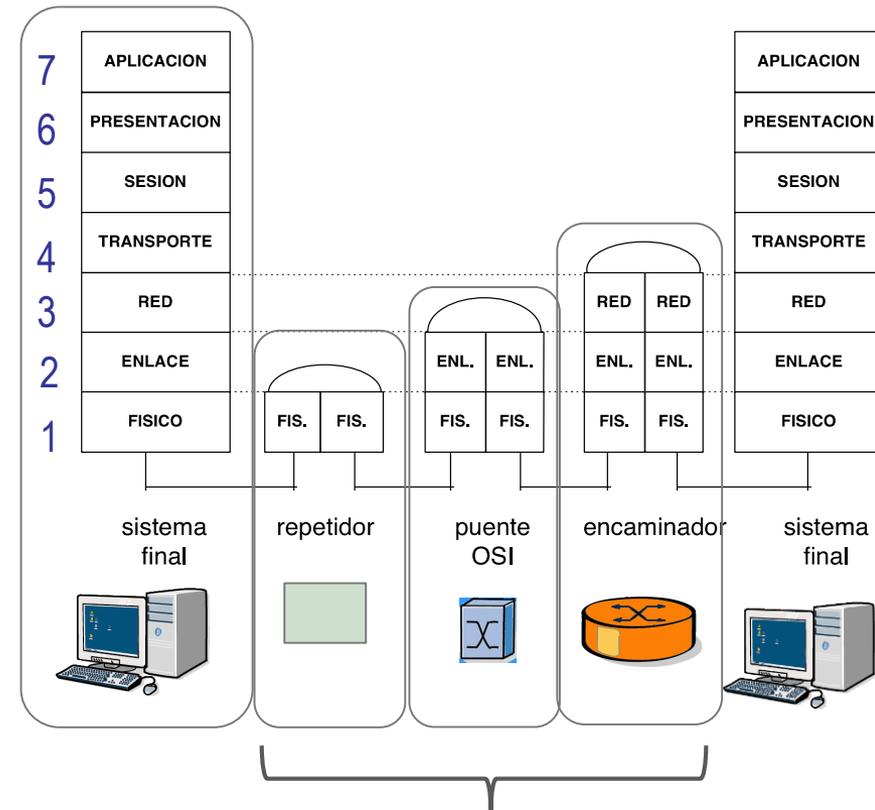
- ❑ La comunicación entre procesos de una aplicación distribuida requiere resolver múltiples problemas.
- ❑ Modelo de capas: solución modular que requiere descomponer un servicio final (comunicación distribuida) en varios servicios específicos inter-dependientes
 - ❖ El servicio de un nivel dependerá del servicio de nivel inferior.
 - ❖ Cada servicio tendrá su propia forma de solicitarlo y un protocolo que permita la colaboración de entidades pares.
 - ❖ Cada servicio se puede clasificar como perteneciente a un nivel o capa en función de las tareas que realice.



Modelo OSI (Open Systems Interconnection)

- ❑ Es un modelo de referencia a través de una arquitectura basada en 7 capas/niveles.
 - ❖ Cada sistema de comunicaciones se estructura en niveles o capas.
 - ❖ Las entidades pares de cada capa cooperan para ofrecer un servicio
 - ❖ Cada capa ofrece un servicio a su capa superior
- ❑ Desarrollado en 1984 por la Organización Internacional de Normalización (ISO) como standard de referencia para el intercambio de datos entre procesos de aplicación
 - ❖ Standard ISO/IEC 7498 / ITU-X.200

ARQUITECTURA DE CAPAS DEL MODELO OSI



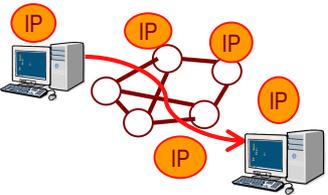
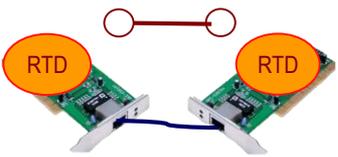
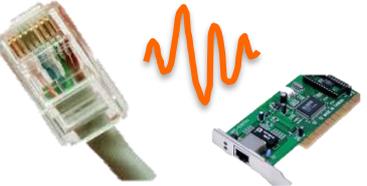
DISPOSITIVOS DE INTERCONEXIÓN SEGÚN EL MODELO OSI



Capas OSI (1-3), niveles inferiores

□ Breve descripción

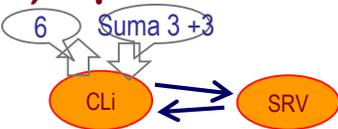
EJEMPLOS DE PROTOCOLOS QUE PERTENECEN A CADA CAPA:
[https://en.wikipedia.org/wiki/List_of_network_protocols_\(OSI_model\)](https://en.wikipedia.org/wiki/List_of_network_protocols_(OSI_model))

Capa (nivel)	Objetivo	Problemas a resolver
(3) Red 	Hace llegar la información recibida desde un origen hasta su destino, atravesando para ello tanto sistemas intermedios como sub-redes, escogiendo para ello la ruta apropiada. (servicio orientado a conexión o no)	<ul style="list-style-type: none">• Direccionamiento de red• Encaminamiento (búsqueda de rutas de origen a destino)• Reenvío de los paquetes (en cada salto)• Posiblemente: fragmentación y reensamblado e informes de errores
(2) Enlace 	Ofrecer un servicio de transporte fiable de datos entre dos entidades conectadas directamente por un enlace. (servicio orientado a conexión o no)	<ul style="list-style-type: none">• Direccionamiento físico• Topología de la red• Acceso al medio compartido• Detección y control de errores• Entrega ordenada de tramas• Control de Flujo (evitar desborde)
(1) Física 	Define las características eléctricas y físicas de los dispositivos (p.e. tarjetas de red) conectados a un medio de transmisión, así como la forma en la que se transmite la información a través del medio	<ul style="list-style-type: none">• Definir medios físicos• Definir caract. Mecánicas y eléctricas• Definir características funcionales del interfaz• Transmitir flujo de bits por el medio (modulación, rég. Binario)



Capas OSI (4-7) – niveles superiores

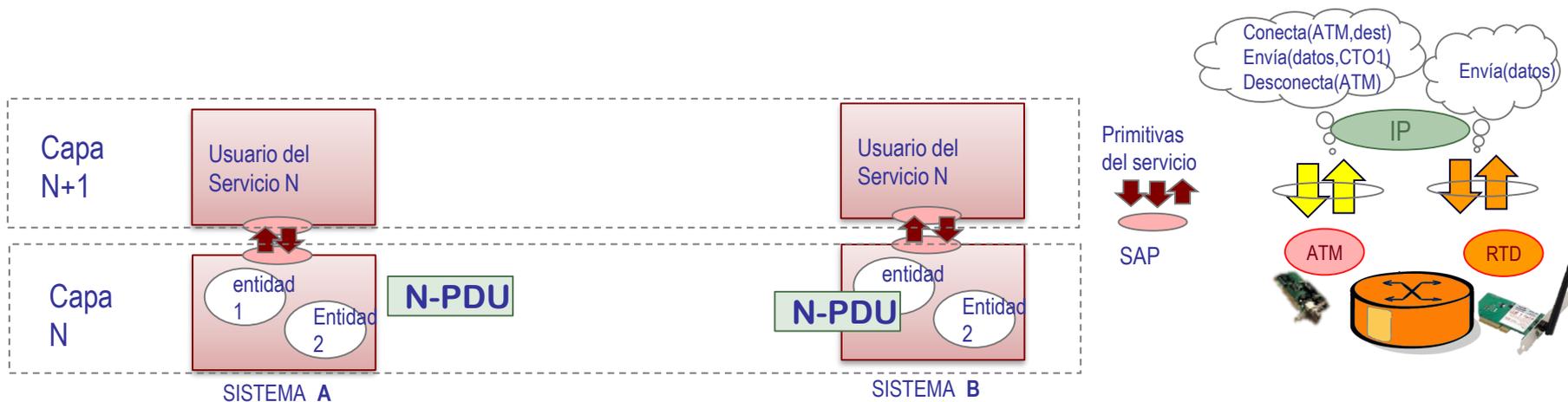
- Los niveles 4-7 sólo se implementan en los sistemas finales

Capa	Objetivo	Problemas a resolver
(7) Aplicación 	<p>Ofrece un servicio especializado a su usuario (personas u otros procesos de aplicación) También permite acceder al resto de servicios de las otras capas</p>	<p>Identificación de entidades pares, determinación de los protocolos de aplicación ofrecen, tareas especializadas del servicio</p>
(6) Presentación 	<p>Ofrecer un servicio que permita tener la misma interpretación de los datos al proceso origen y al destino (cada proceso de la aplicación distribuida podría tener su propia forma de representar / transmitir los datos según S.O., leng prog,..)</p>	<p>Conversión de las estructuras de representación de datos de ambos extremos de la comunicación</p> <p>Cifrado y Compresión</p>
(5) Sesión 	<p>Proporcionar un servicio de control de diálogo entre las aplicaciones (no todas las aplicaciones necesitan este servicio)</p>	<p>Control de sesiones, control del diálogo (full/half duplex, sincronización), agrupamiento de datos, recuperación ante errores mediante puntos de comprobación (checking points).</p>
(4) Transporte 	<p>Proporcionar un servicio de transferencia fiable y ordenada de datos entre sistemas finales independientemente de la calidad proporcionada por la red</p>	<ul style="list-style-type: none"> • Detección y Control de errores • Control de flujo • Segmentación y concatenación



Solicitud de Servicios según el modelo OSI

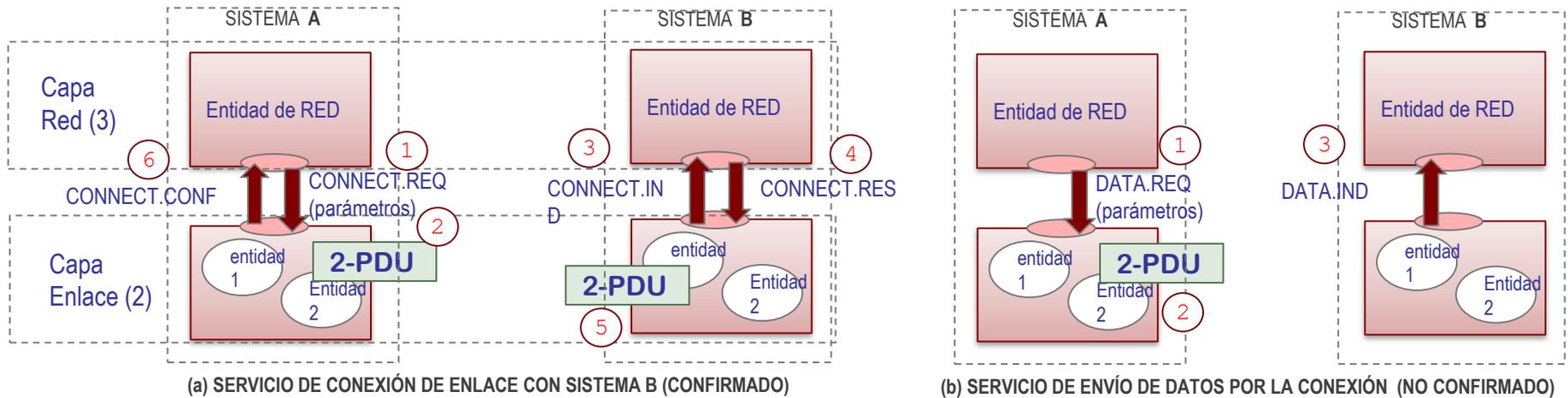
- ❑ Cada capa utiliza los servicios de la capa inmediatamente inferior.
- ❑ Solicitud de Servicios:
 - ❖ Puntos de acceso al servicio (SAP, Service Access Point) define el Interface de uso de un servicio (i.e. cómo solicitarlo por parte de la capa superior) *SAP is a conceptual location at which one OSI layer can request the services of another OSI layer*
 - Se compone de “primitivas del servicio” (mensajes con datos estructurados) que hay que usar para lograr una función concreta
 - P.ej. **SAP1-conectar.req(<dirección-destino>, <prioridad>)**
- ❑ Las entidades de una capa intercambian mensajes (unidad de datos del protocolo - PDU) para ofrecer el servicio



Servicios en el modelo OSI

Los servicios ofrecidos pueden ser confirmados o no.

- Los servicios confirmados utilizan 4 clases de primitivas del servicio (REQ, RES, IND, CONF). Los no confirmados sólo REQ, IND.

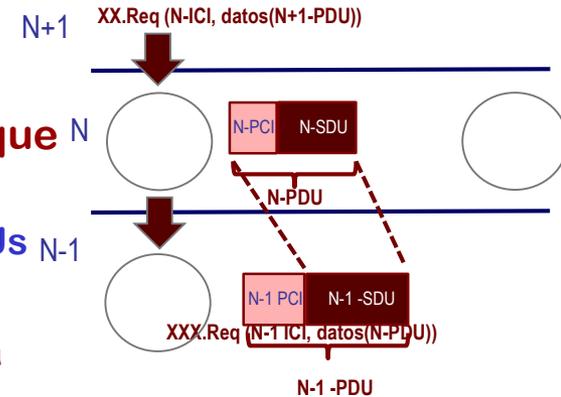


Unidad de Datos del Protocolo (PDU)

□ **N-PDU** (unidad de datos del protocolo) : información intercambiada por dos entidades de la capa N.

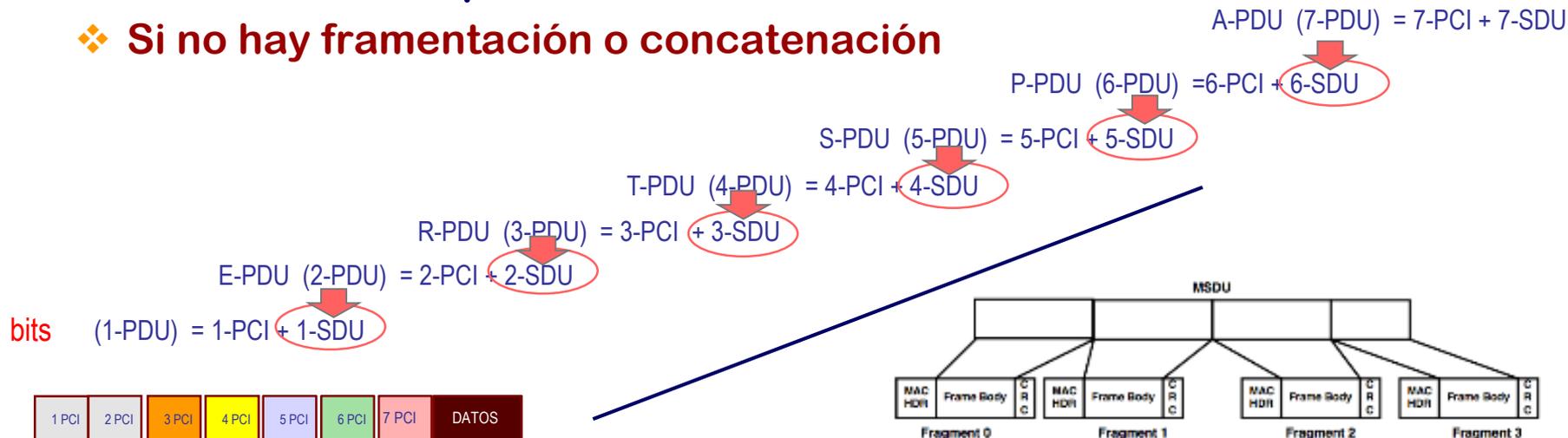
□ **N-PDU** compuesta por:

- ❖ **N-SDU** (Unidad de datos del servicio). Son los datos que le ha pasado la entidad (N+1) a la entidad (N)
 - En caso de fragmentación la SDU genera varias N-PDUs
- ❖ **N-PCI** (Información de control del protocolo) Información intercambiada entre entidades (N) para coordinar su operación conjunta.



□ **N-PDU** suele encapsular la N+1 PDU

❖ Si no hay fragmentación o concatenación

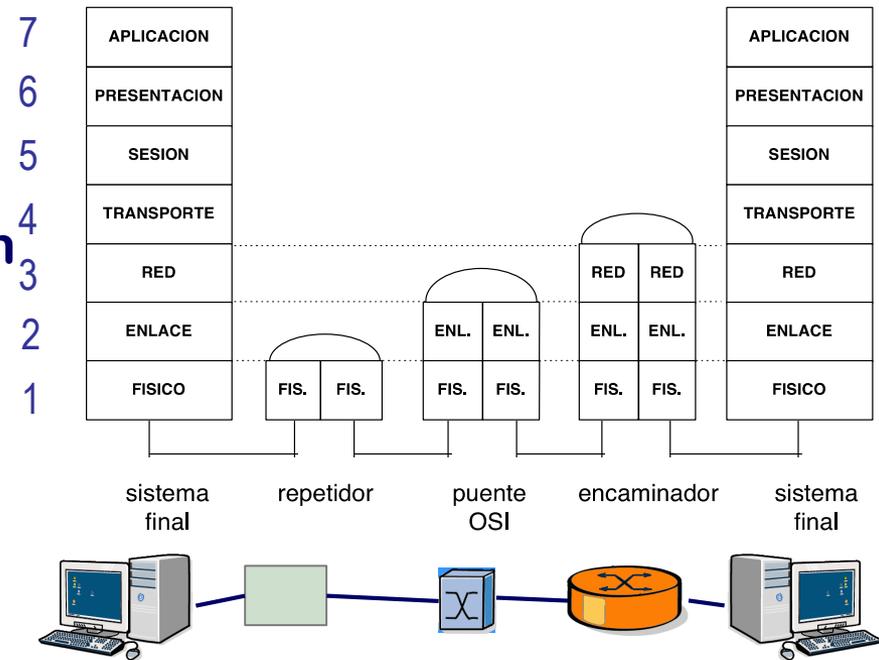


Arquitecturas de red

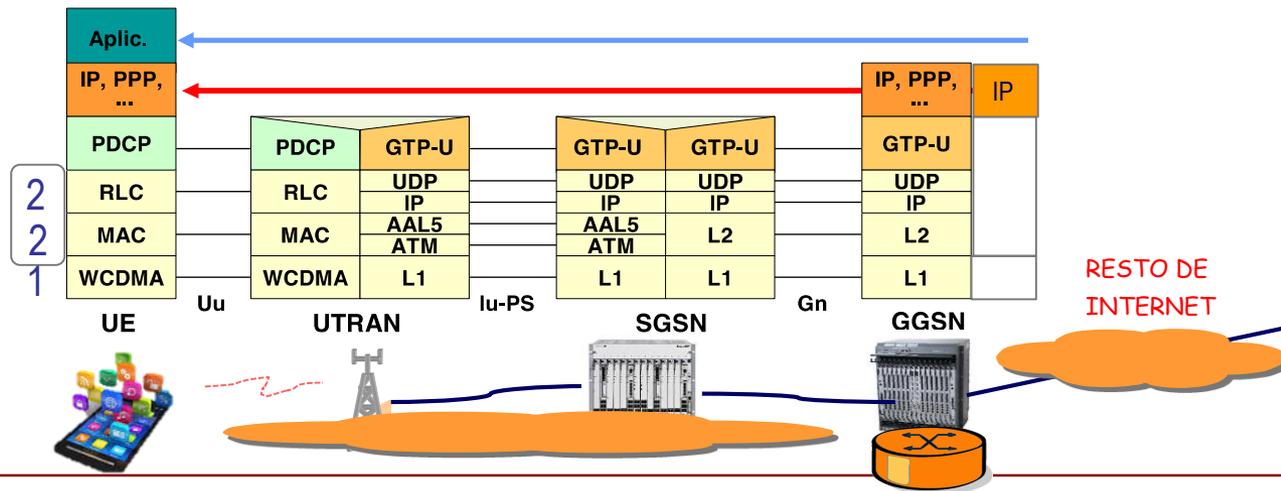
□ Torre /Pila de protocolos: el conjunto de protocolos implementados en un sistema ordenados según su posición en el modelo OSI.

□ Permiten definir la arquitectura de una red (entidades que se comunican y el nivel al que lo hacen)

□ P.ej:



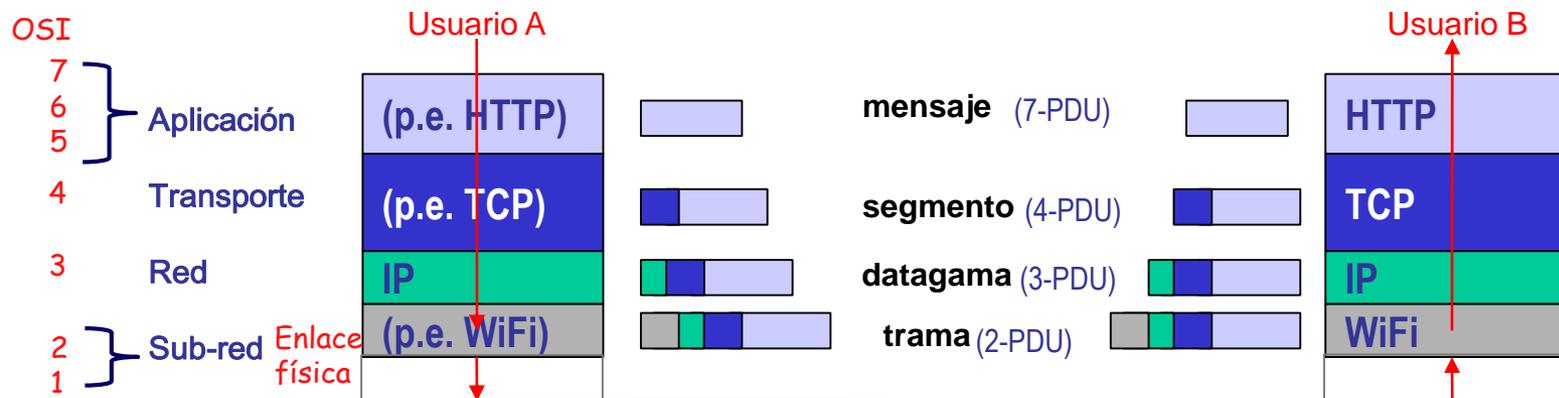
ARQUITECTURA DE ACCESO A INTERNET 3G (SUB-CAPAS)



El modelo de Internet vs el modelo OSI

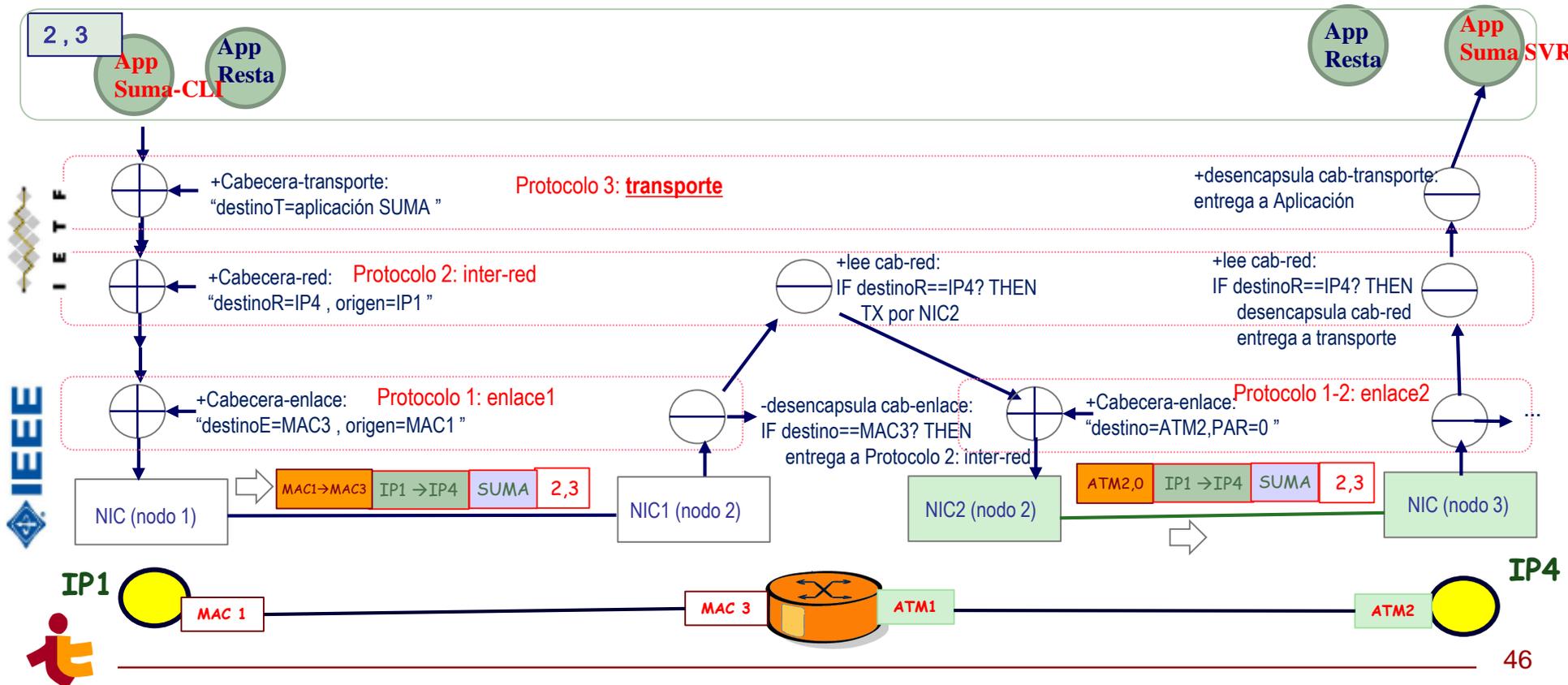
- ❑ El modelo de Internet no es realmente un modelo de capas independientes. Más bien, podemos verlo como un conjunto de protocolos inter-relacionados que pueden ser clasificados como pertenecientes a una capa OSI en función del servicio que prestan.
- ❑ Mapeo de capas OSI en el modelo de Internet

Las tareas OSI de sesión, presentación y aplicación serían implementadas en la capa de aplicación cuando fuesen necesarias integradas en la aplicación o con protocolos propios



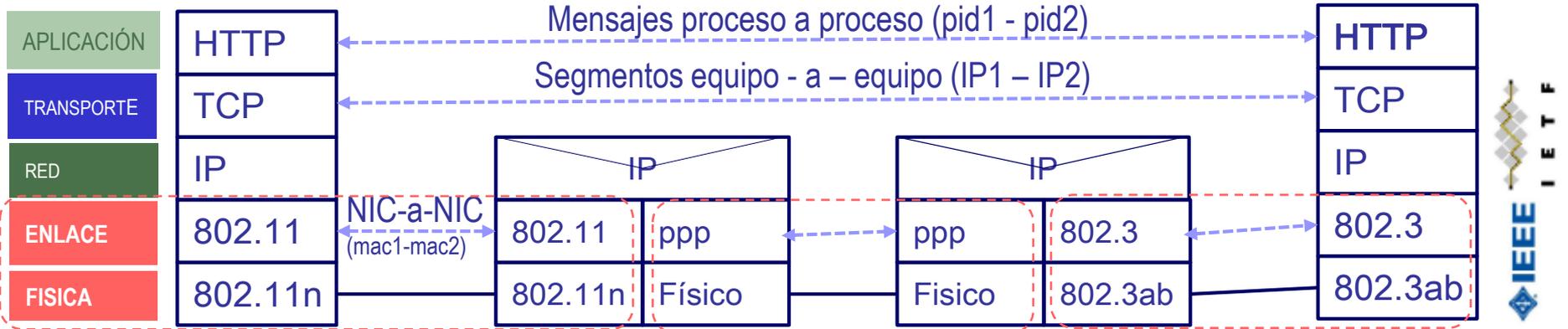
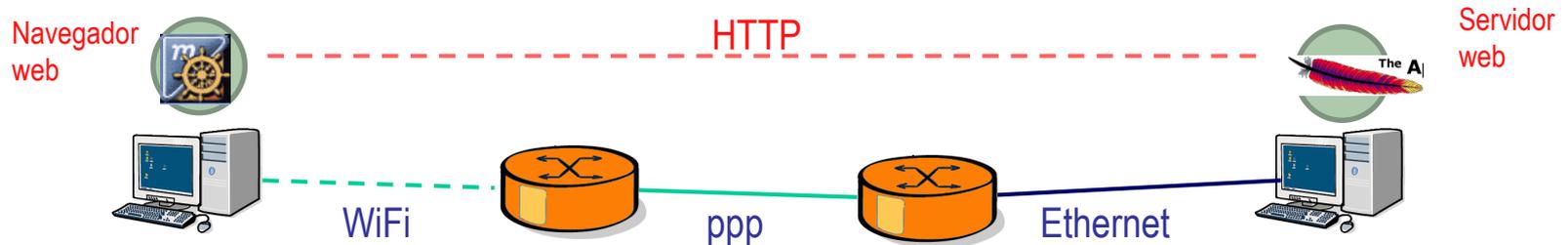
La capa de transporte en el modelo de Internet

- ❑ Identifica a los procesos Origen y Destino de la comunicación
 - ❖ Origen: encapsula (cabecera identificando al proceso destino)
 - ❖ Destino: lee cabecera, y entrega mensaje al proceso correspondiente
 - ❖ Existen varios protocolos (p.ej: TCP, UDP)
 - ❖ A su vez usa el servicio de inter-red



Ejemplo de arquitectura de protocolos en Internet

Red simple Internet: aplicación web

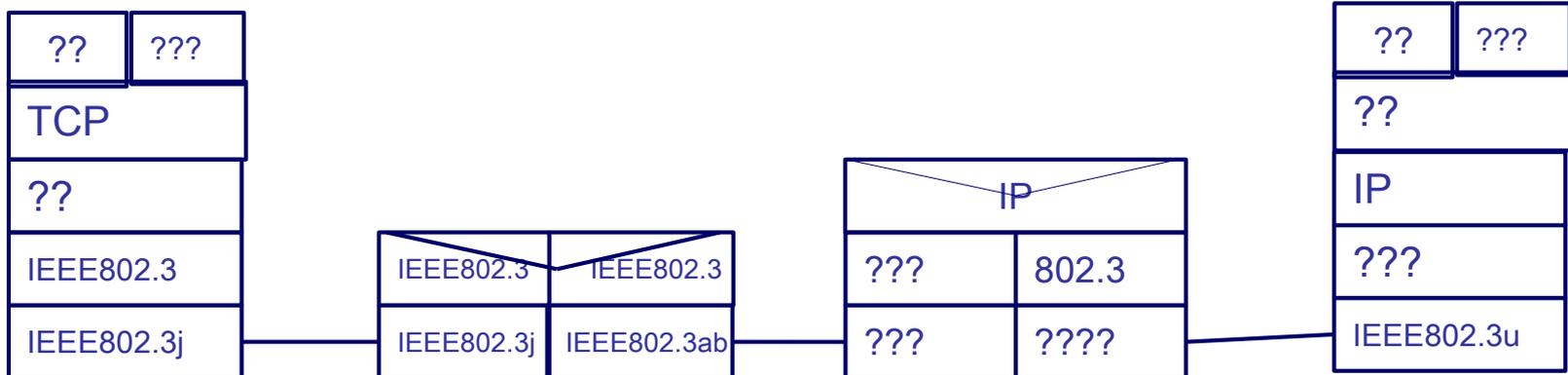
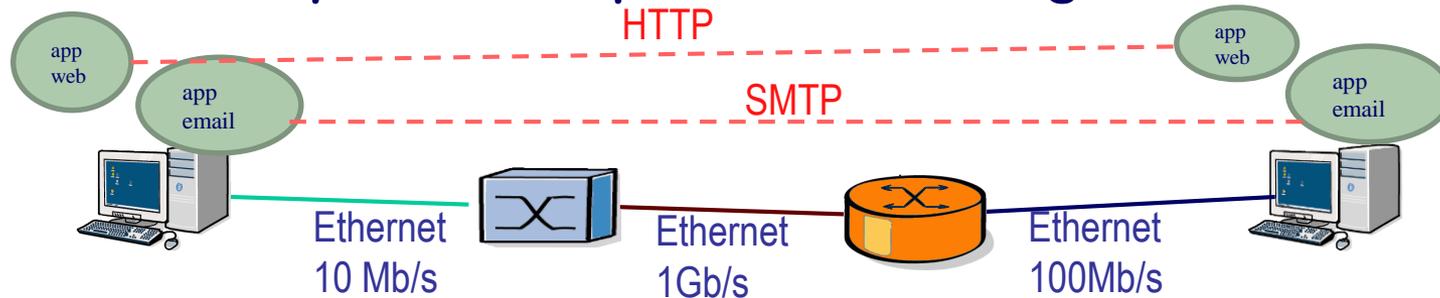


https://en.wikipedia.org/wiki/IEEE_802.3



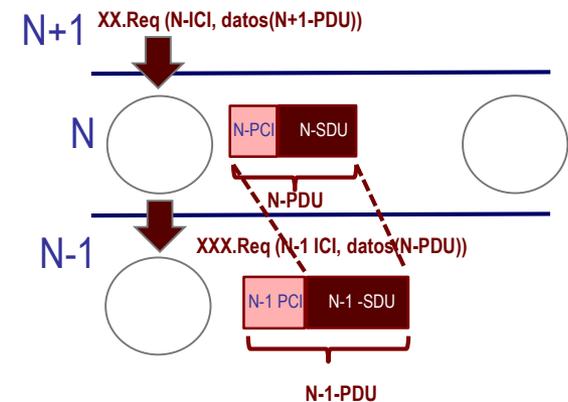
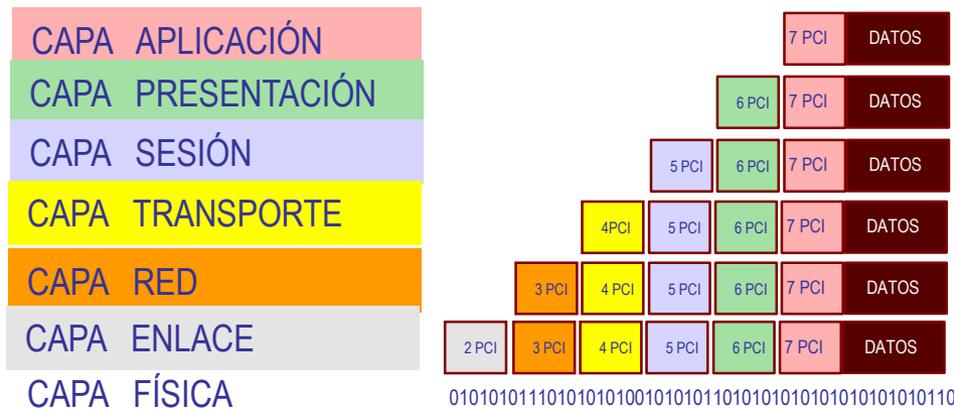
Ejercicio

¿Puedes completar la arquitectura de la figura?



Resumen de la sección 1.4 (semana 3)

- ❑ Los modelos de capas permiten descomponer el complejo funcionamiento de las redes en problemas independientes
- ❑ Modelo OSI : modelo o marco de referencia



- ❑ Cada nivel presta sus servicios al nivel superior y usa los servicios del nivel inferior.
- ❑ Modelo OSI define varios conceptos clave: PDU, primitivas, etc..
- ❑ Internet no cumple estrictamente el modelo OSI, pero puede verse como un conjunto de protocolos interrelacionados que pueden ser clasificados como pertenecientes a alguna capa OSI



Índice del Tema 01

1.1 Introducción a las Redes de Ordenadores

1.2 Interconexión en las redes de Ordenadores y conceptos básicos de enlaces

1.3 Definiciones básicas y Modelos de capas OSI

1.4 Multiplexión Estadística

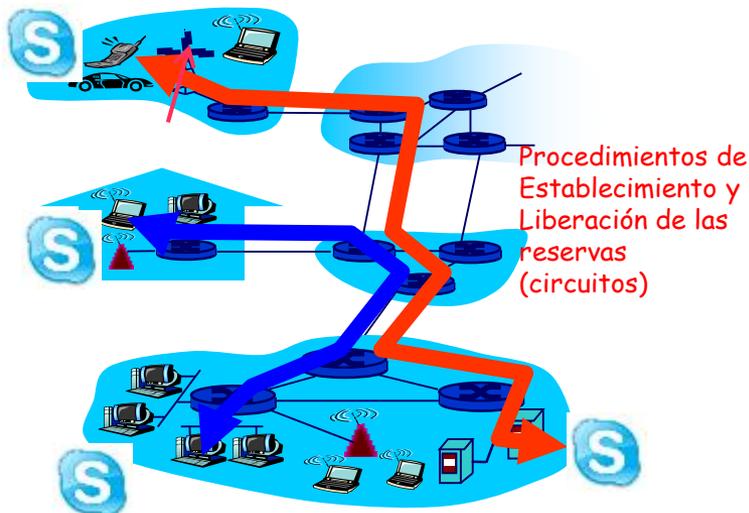
1.4.1 Repaso de la multiplexión estadística

1.4.2 Retardo y pérdidas en las colas de los conmutadores de paquetes



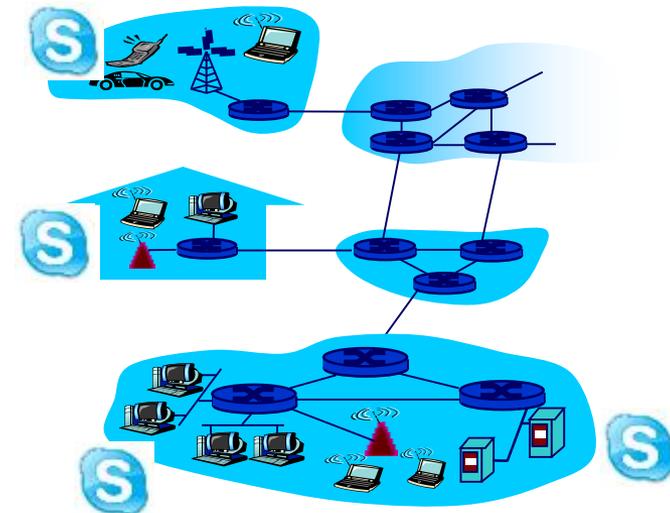
Dos formas de compartir el régimen binario

- ❑ **Conmutación de Circuitos:** se dedica un *circuito* para cada comunicación durante toda su duración (como red telefónica)



- ❑ **Conmutación de Paquetes:** los datos se envían a la red en bloques – paquetes –) sin asignación de recursos en exclusiva (redes de datos)

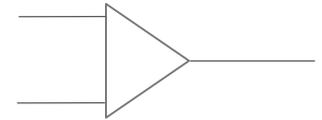
Cada paquete usa toda la capacidad del canal cuando le toca ser servido (i.e. transmitido)



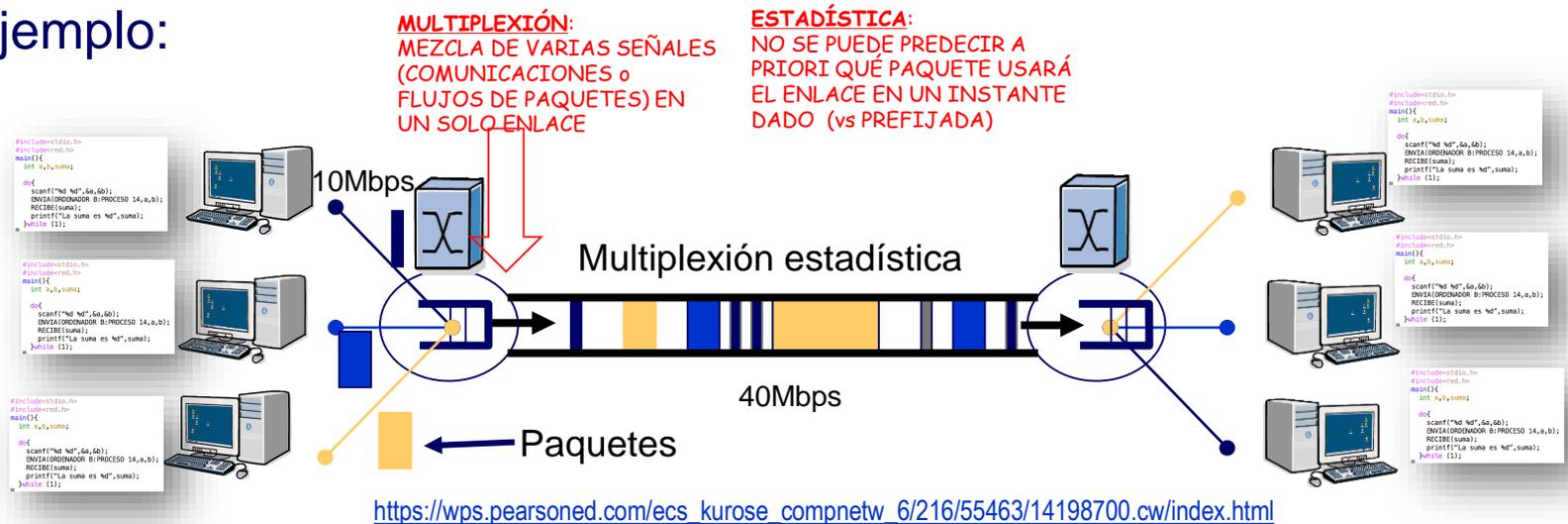
Multiplexión Estadística (STDM)

Visión física: Multiplexión Estadística

- ❖ Asignación no prefijada
- ❖ No hay reserva de capacidad (compartida bajo demanda)
- ❖ Paquete: bloque de datos a servir (tiene un tamaño máximo)



Ejemplo:

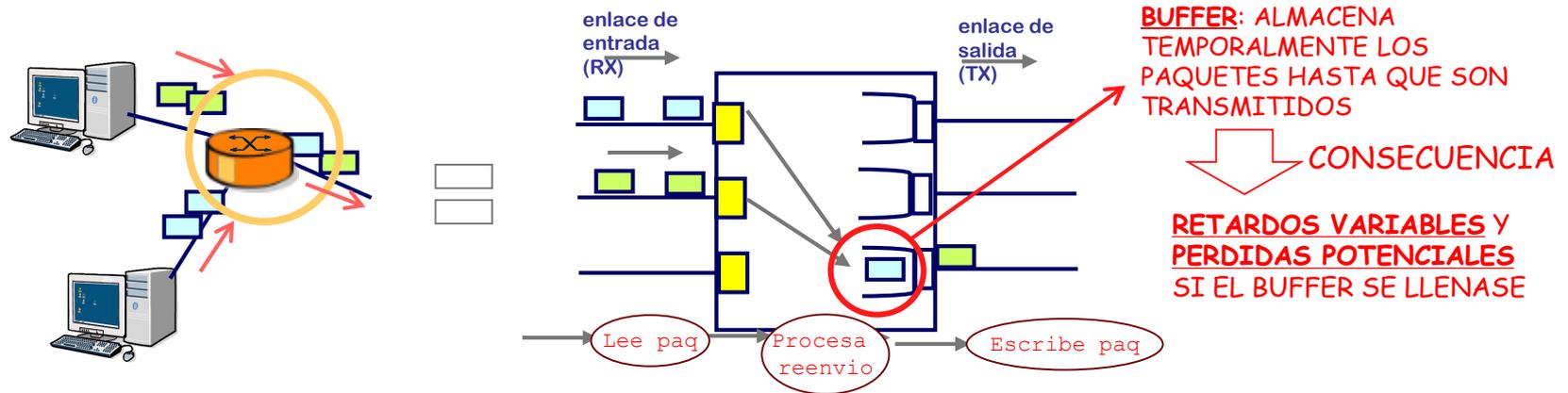


- ❖ Tiempo máximo de uso del enlace determinado por el tamaño máximo del paquete.

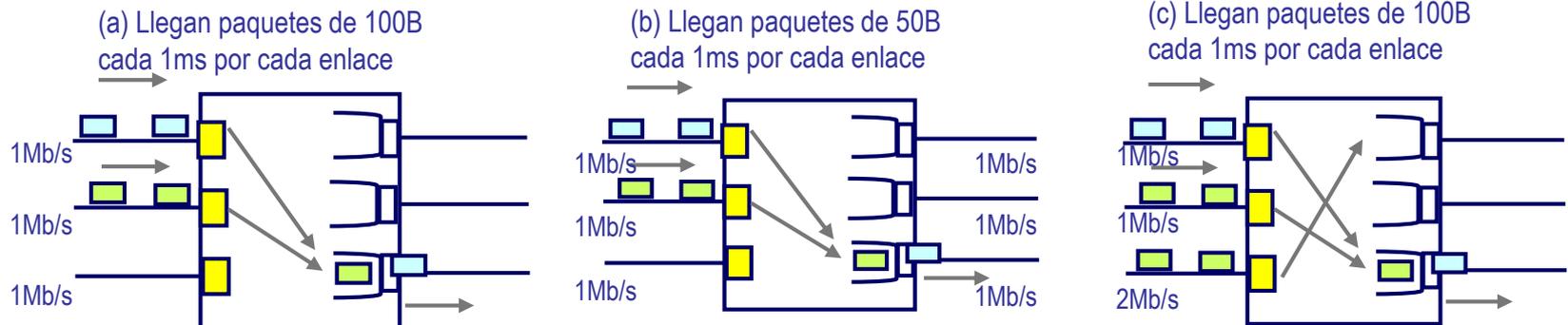


Conmutación de paquetes: necesidad de buffer y espera en cola

- ¿qué ocurre si dos paquetes que se han recibido “a la vez” quieren salir por el mismo enlace?
- Necesidad de un buffer de espera temporal



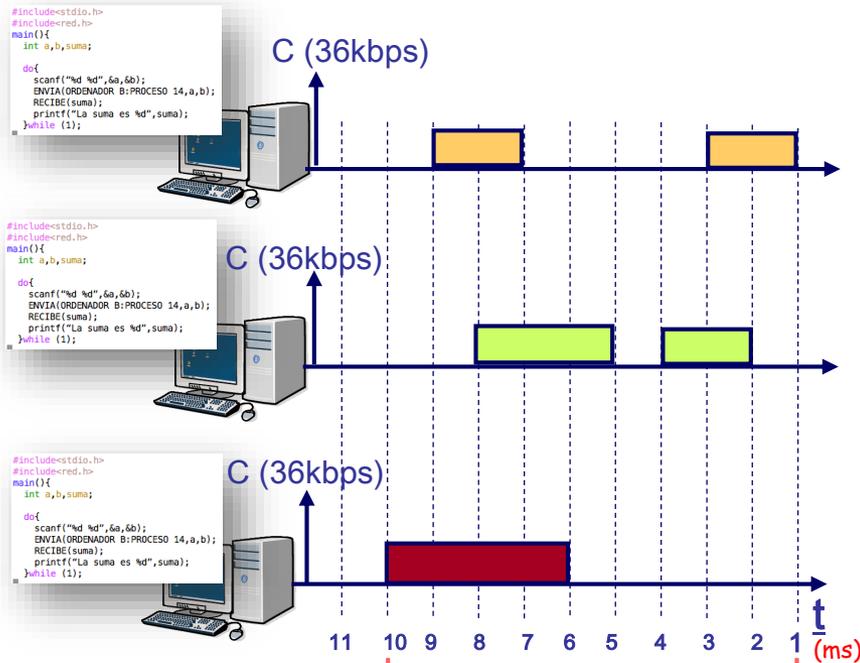
❖ Ejemplos:



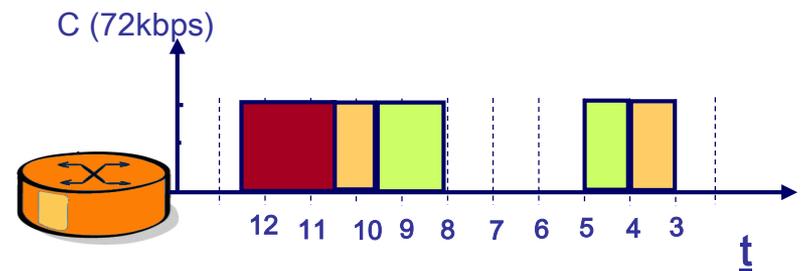
Ejercicio

□ Dibujar la ocupación del enlace de salida si las aplicaciones generasen un tráfico periódico (10 i.t.) como el indicado en la figura.

❖ un paquete no se PROCESA hasta que no se ha recibido completamente



Periodo del tráfico (10 intervalos de tiempo)



¿caudal (throughput) de cada usuario?
Ganancia de STDM con respecto a circuitos de 36kbs para cada comunicación?

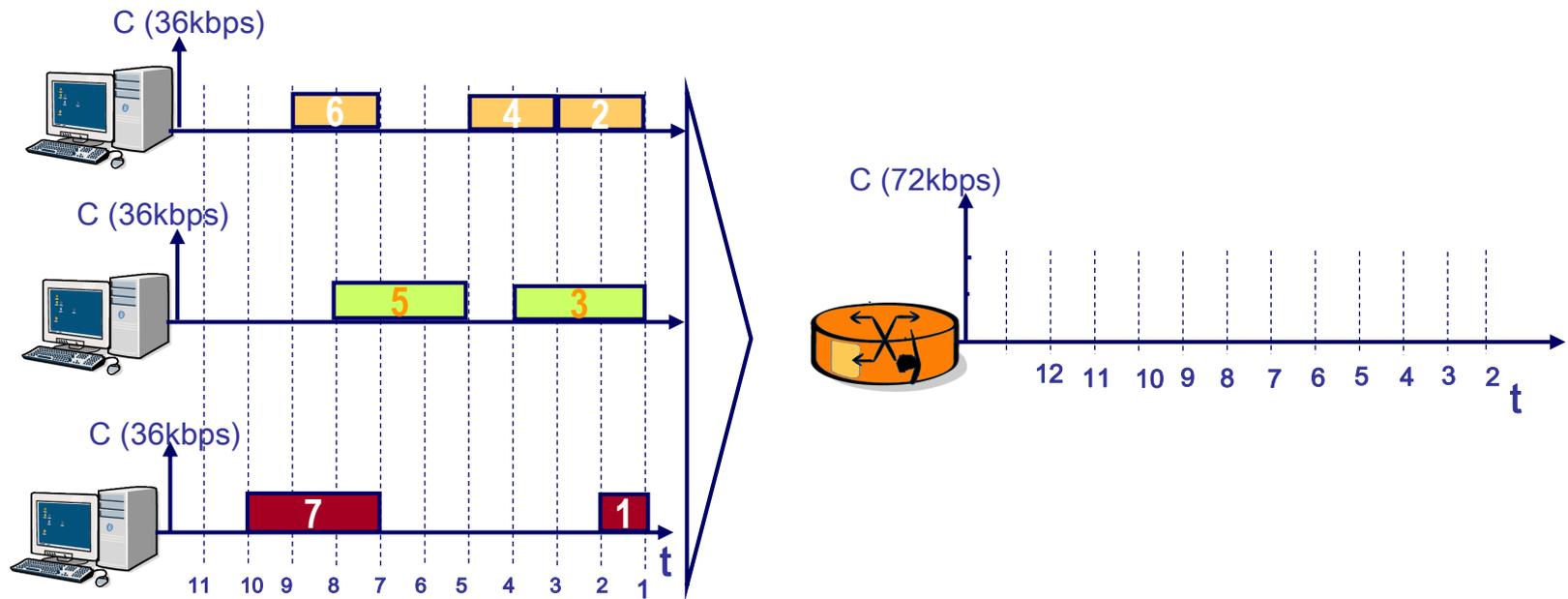
Caudal: (throughput): cantidad de información que atraviesa un sistema en un periodo de tiempo

Utilización: (ocupación) porcentaje de tiempo que un sistema (p.ej. enlace) está en uso.



Ejercicio (II)

□ Dibuje la ocupación del enlace de salida

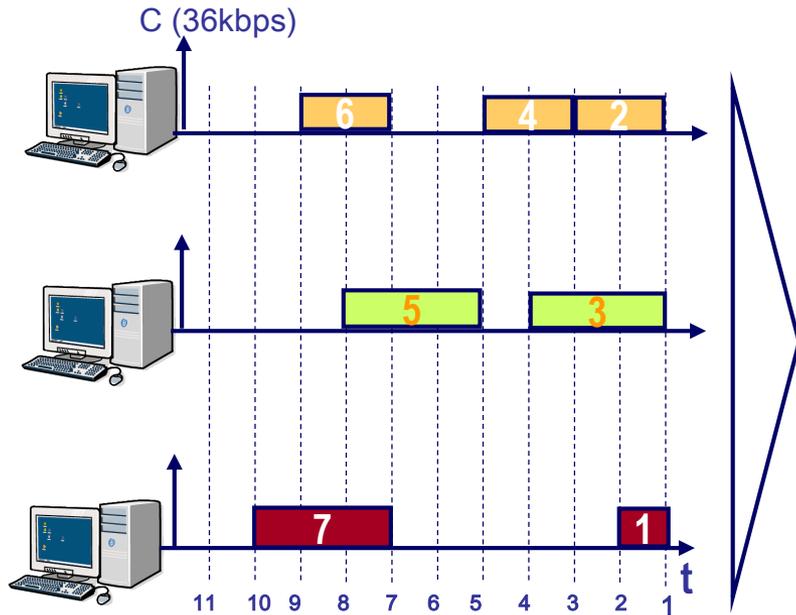


- ❖ ¿cuál es el tiempo medio de espera en cola que sufren los paquetes?
- ❖ Si el patrón de tráfico se repite periódicamente cada 10 i.t. ¿qué porcentaje de tiempo esta ocupado el enlace de salida?
- ❖ ¿qué tamaño de cola necesitamos para que no haya pérdidas?

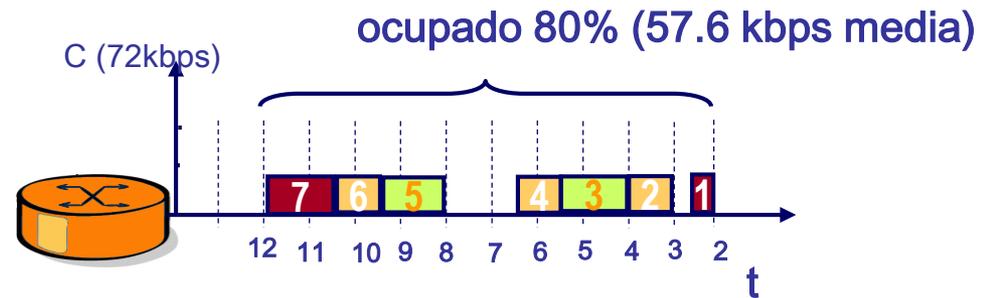


Solución

□ Solución



Tamaño buffer para almacenar el paquete 7

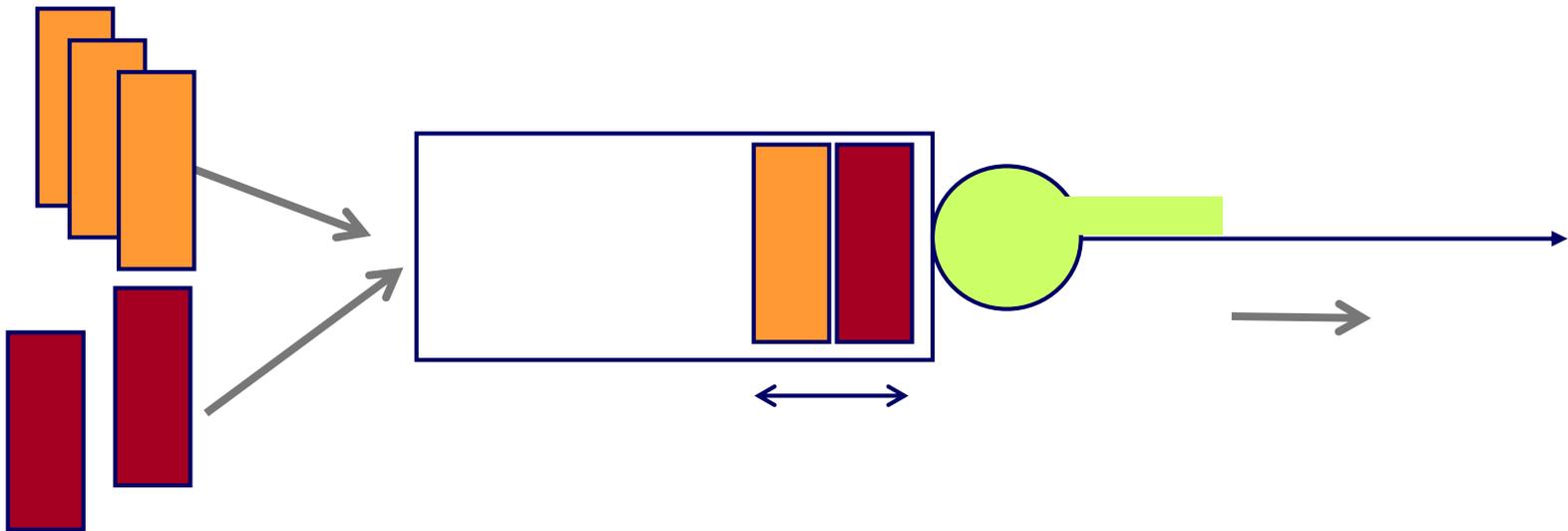


#PAQ	t_arriv	t_serv	t_out	W
1	2	2	2.5	0
2	3	3	4	0
3	4	4	5.5	0
4	5	5.5	6.5	0.5
5	8	8	9.5	0
6	9	9.5	10.5	0.5
7	10	10.5	12	0.5



Ventajas e inconvenientes de la Multiplexión Estadística

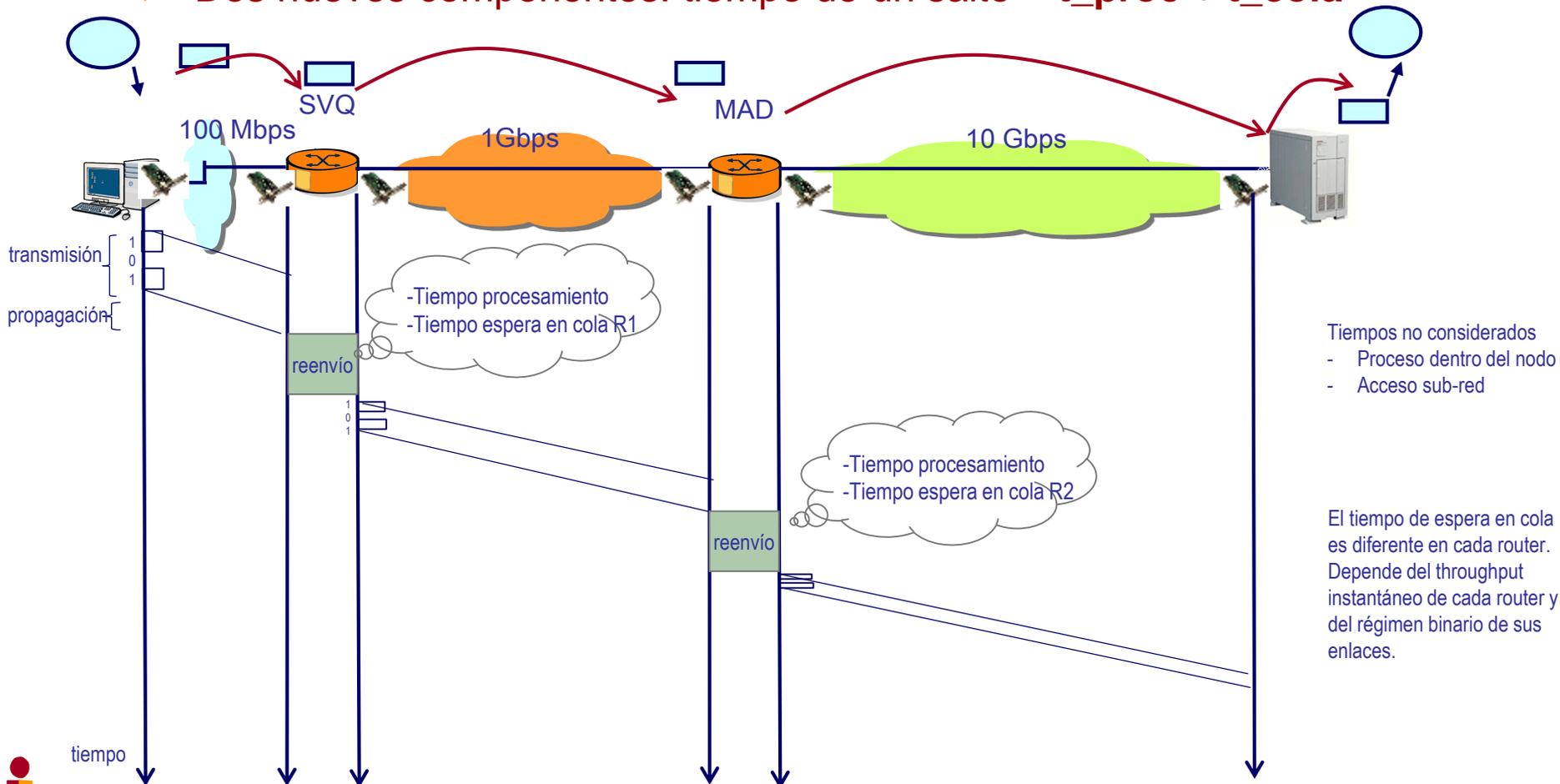
- ❑ + Mejor uso de los enlaces: (permite que mas usuarios usen la red)
 - ❖ ganancia frente a conmutación de circuitos
- ❑ - Congestión si hay tráfico excesivo (conmutadores de paquetes)
 - ❖ Posibles pérdidas de paquetes (cola finita) si entra más caudal del que puede salir por un enlace durante un cierto tiempo.
 - ❖ **Retrasos variables en las colas de los nodos (i.e. Routers).**
 - ❖ Si aumenta el tráfico empeoran las prestaciones (en red telefónica no)



¿Cuánto tardan los paquetes en llegar a su destino ?

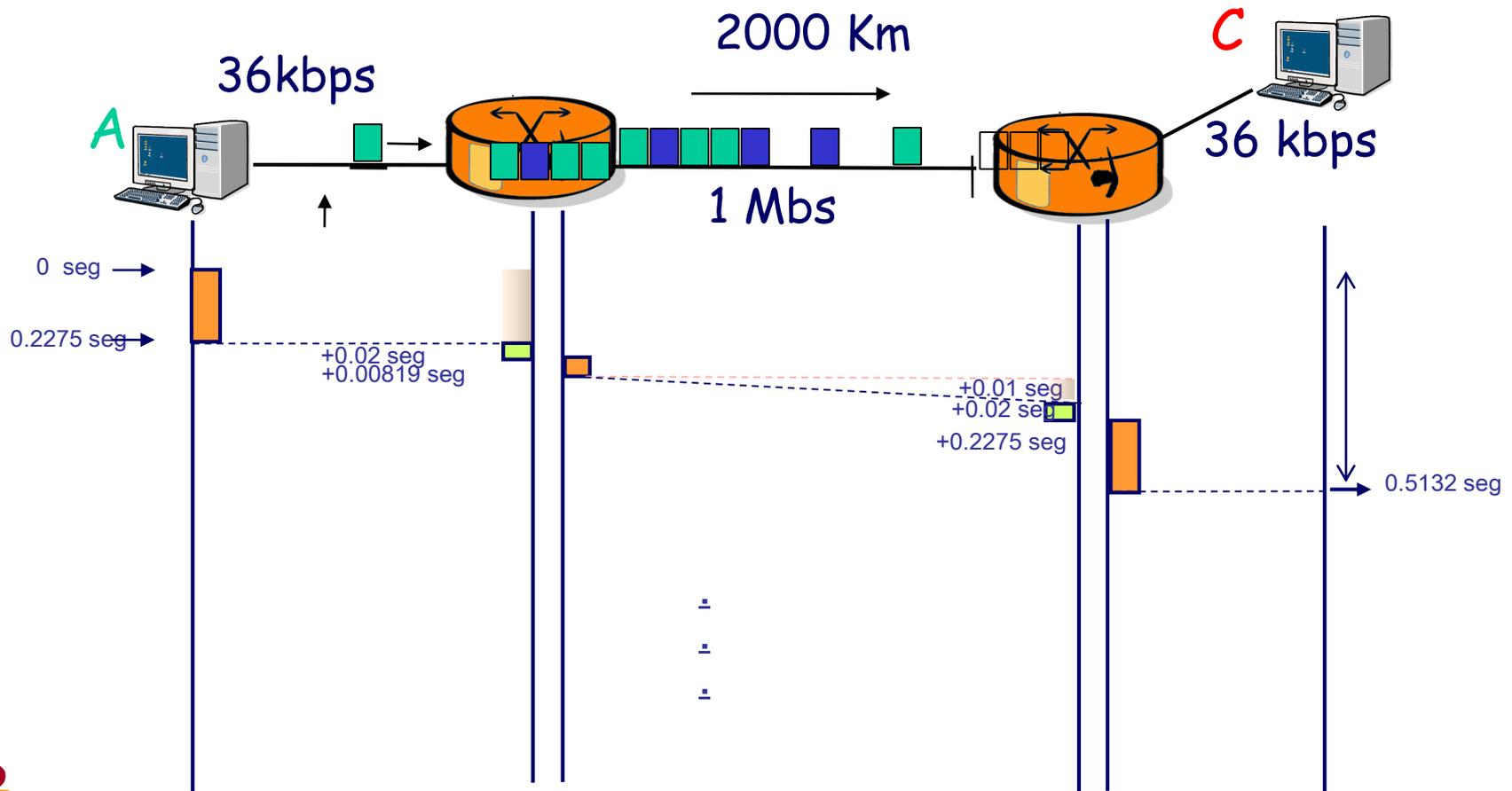
□ Diagrama Temporal → sumar todos los tiempos! (pero no sumes cosas que ocurren en paralelo)

❖ Dos nuevos componentes: tiempo de un salto = $t_{proc} + t_{cola}$



Ejemplo cálculo de retardo

- ❑ Considerar sólo el retardo de transmisión, de propagación ($2 \cdot 10^5 \text{ km/s}$) y de espera en cola (20ms en cada router)
 - ❖ Sólo considerar la propagación donde se indique la longitud del enlace



Tamaño máximo de los paquetes y fragmentación

Las NIC transmiten bloques con un número máximo de bits

❖ **Maximum Transfer Unit (MTU)** : tamaño máximo de la carga útil de una trama (**NO INCLUYE LA CABECERA DE ENLACE**)

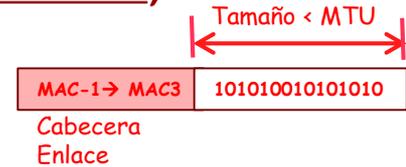
- dependen del standard de la NIC

❖ **Mientras más grandes,**

- mayor probabilidad de corrupción
- más tiempo se ocupa el canal (recurso compartido)

❖ **Mientras más pequeño**

- Mayor proporción de cabecera frente a carga útil



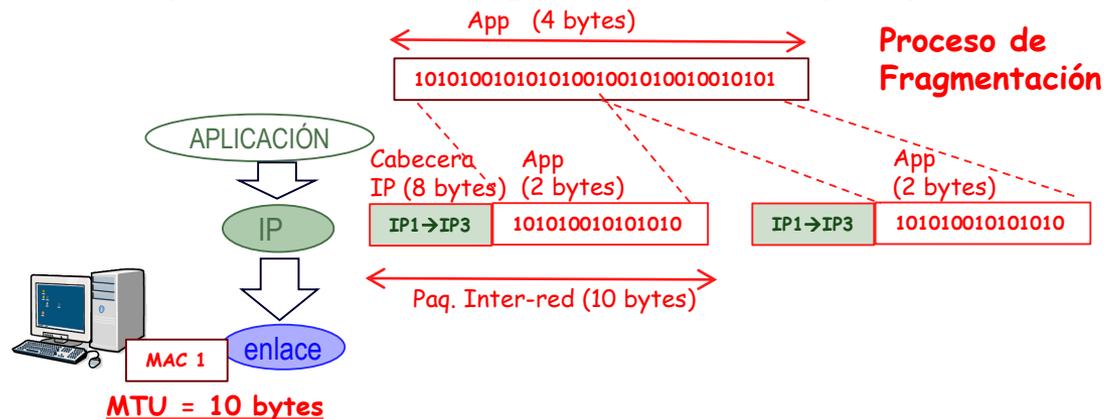
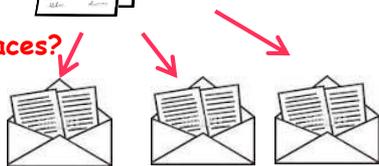
Un mensaje de aplicación podría dar lugar a varios paquetes

❖ ¿de qué depende?

Todos no caben en un solo sobre.

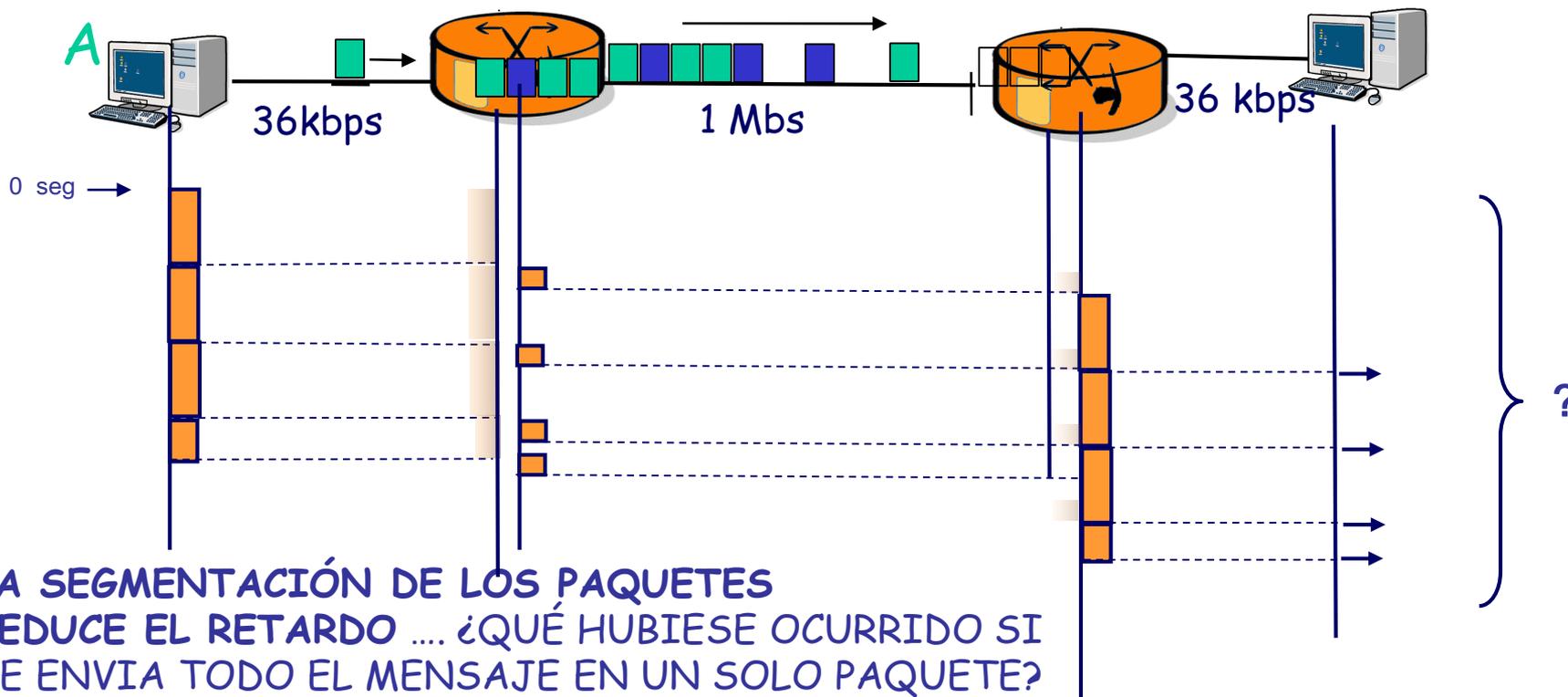


¿Qué haces?



Problema

- ❑ El equipo A envía un mensaje de aplicación de 3KB a C. ¿cuánto tarda el equipo C en recibir todo el mensaje?
 - ❖ Considerar **sólo** el retardo de transmisión, MTU = 1.000B, cabeceras de enlace e inter-red de 8 bytes.



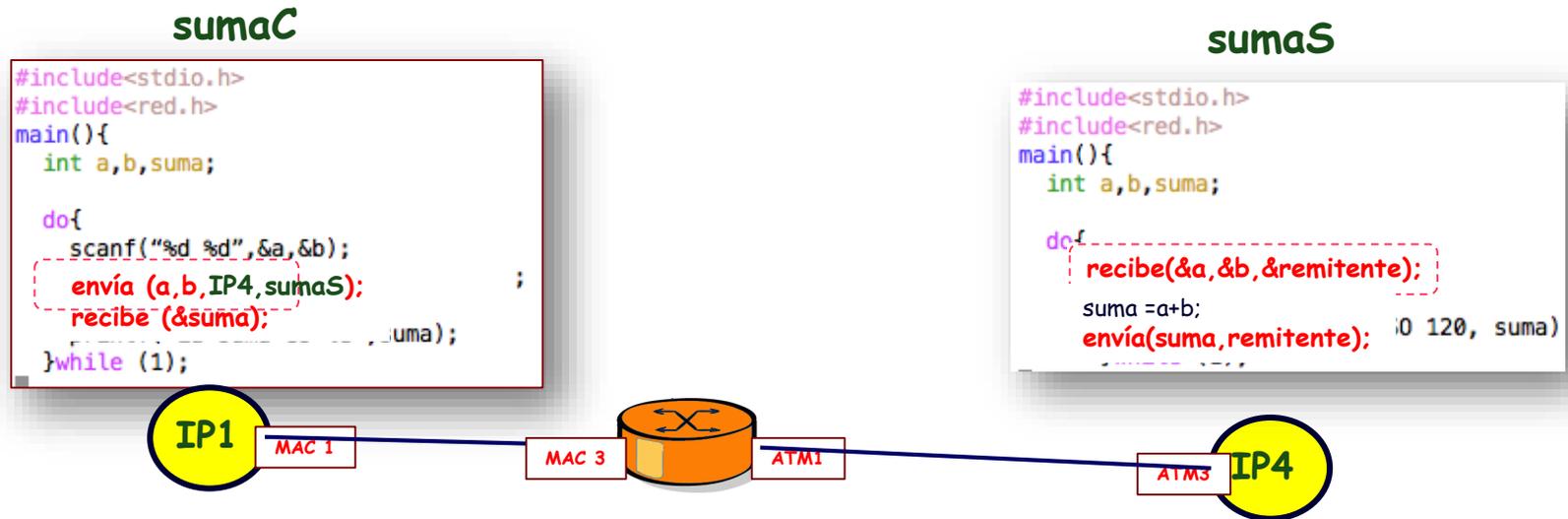
Resumen de la sección 1.4 (semana 3)

- ❑ Los recursos de conmutación de los nodos (switches o routers) deben ser compartidos por los usuarios de la red.
 - ❖ Las redes de datos actuales emplean la multiplexión estadística para la compartición de estos recursos.
- ❑ La multiplexión estadística reparte la capacidad de transmisión de un enlace “bajo demanda”.
 - ❖ Ello permite un uso mas eficiente que con reservas prefijadas (conmutación de circuitos)
 - ❖ Pero pueden aparecer retardos variables e incluso pérdidas de paquetes en el tránsito por un nodo muy congestionado.
- ❑ Los paquetes que envía una aplicación distribuida tardarán un tiempo variable en llegar a su destino según el camino seguido y el estado del tráfico.
- ❑ En Internet no se pueden dar garantías estrictas sobre el tiempo que tardarán los paquetes en llegar al destino ni tan siquiera sobre que no serán eliminados por desbordamiento de algún nodo.



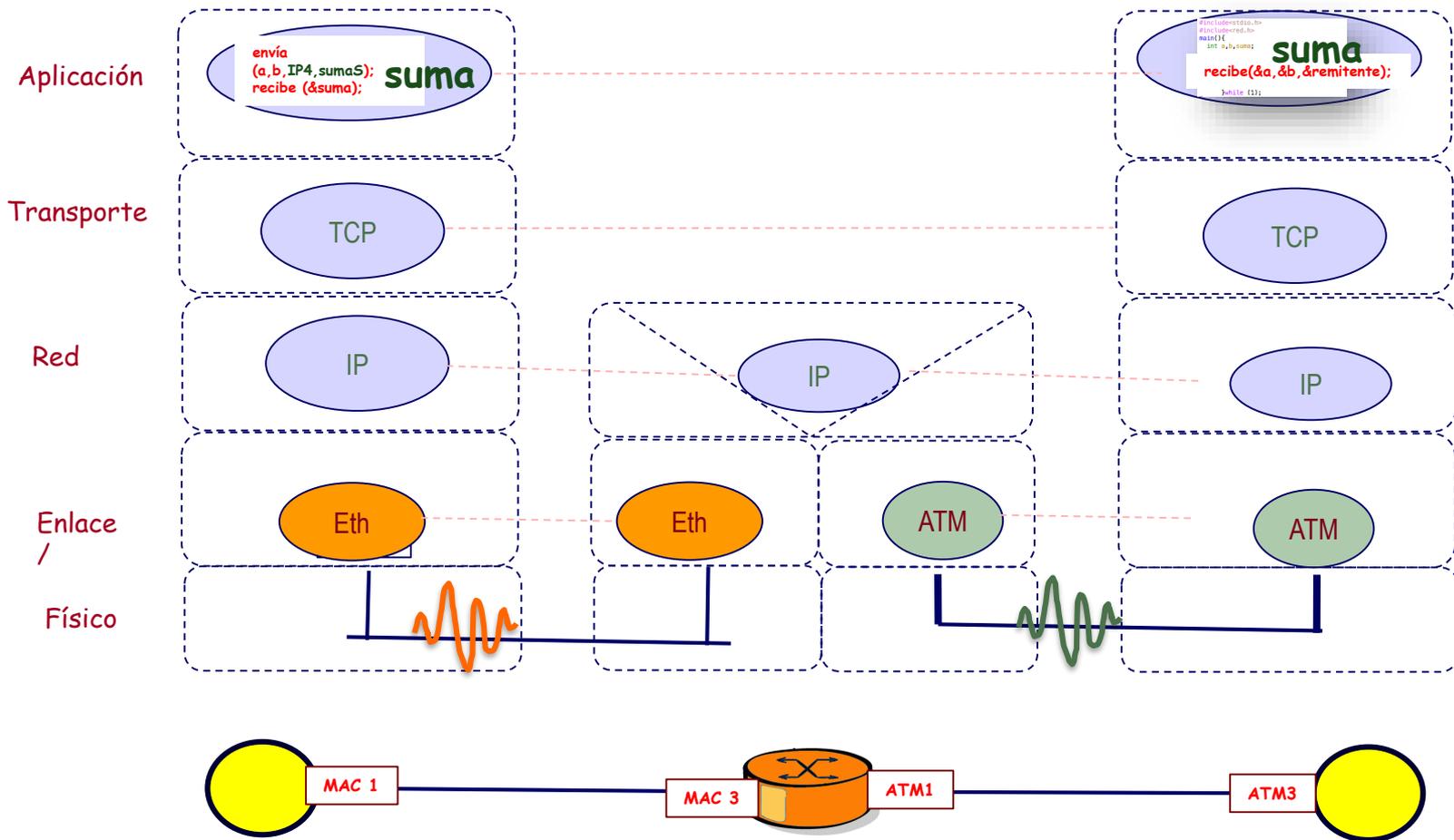
Ejercicio 1: modelo OSI y arquitectura de protocolos

- Dibujar la arquitectura de protocolos (suponiendo "Transporte" como protocolo de transporte.. Protocolos "Eth" y "ATM" en los enlaces 1y 2 respectivamente) identificando las entidades de cada capa
- Identifica Solicitudes de Servicio entre capas y los procesos de encapsulamiento y desencapsul.
- Dibuja las tramas que se envían por cada enlace
- Identifica qué cabecera lee/escribe cada entidad de protocolo



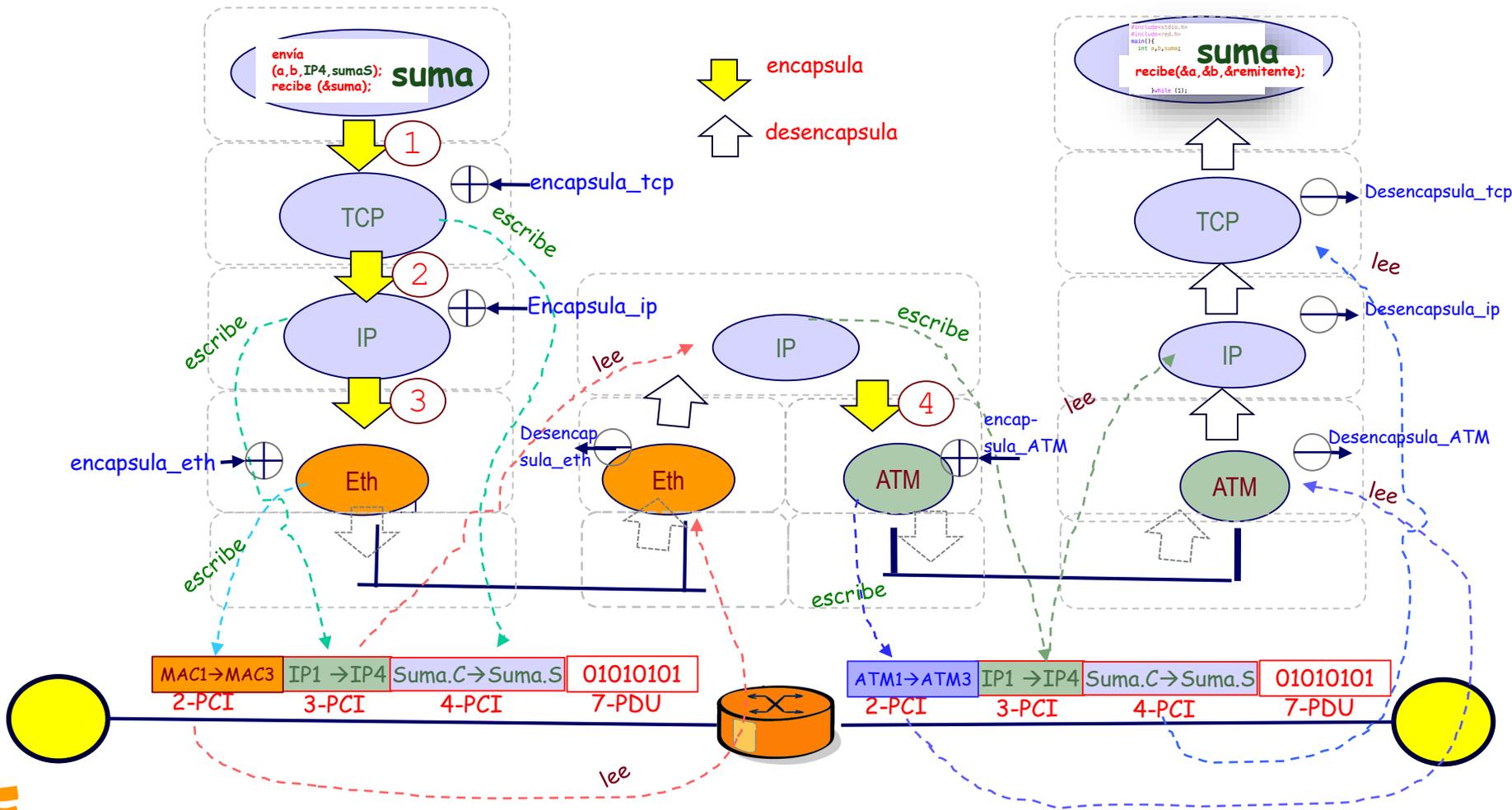
Ejercicio 1

- Dibujar la arquitectura de protocolos (suponiendo "TCP" como protocolo de transporte,, "Eth" como protocolo del 1º enlace y "ATM" como protocolo del segundo enlace) identificando las entidades de cada capa



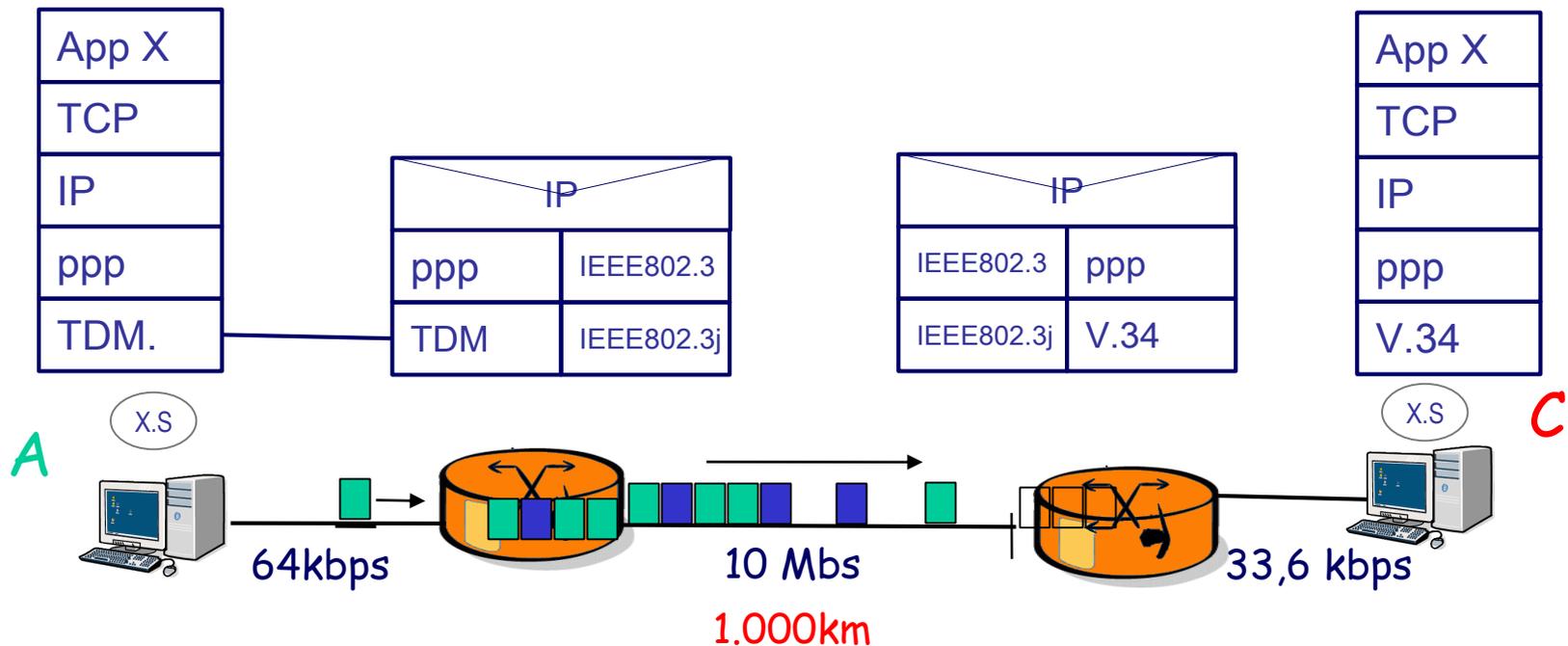
Ejercicio 1

- Identifica Solicitudes de Servicio entre capas y los procesos de encapsulamiento y desencapsul.
- Dibuja las tramas que se envían por cada enlace
- Identifica qué cabecera lee/escribe cada entidad de protocolo



Ejercicio 2: retardo y pérdidas en Internet

- ❑ El proceso X del equipo A envía un mensaje de aplicación de 2KB a C. ¿cuánto tarda el equipo C en recibir todo el mensaje?
 - ❖ Considerar MTU = 1.000B en todos los enlaces
 - ❖ Cabeceras: de transporte (20B), inter-red (8 bytes), PPP(10B), IEEE802.3 (24B).
 - ❖ Considerar: vel. prop. $2 \cdot 10^5 \text{ Km/s}$; $t_{\text{proc}} = 0$; $t_{\text{cola_router}} = 0.1 \text{ ms}$.
 - ❖ Considerar que el origen fragmenta en TCP (en lugar de IP).



REPITE EL EJERCICIO (a) SI NO EXISTIESE MTU; (b) Si Enlace R-R 10kbps

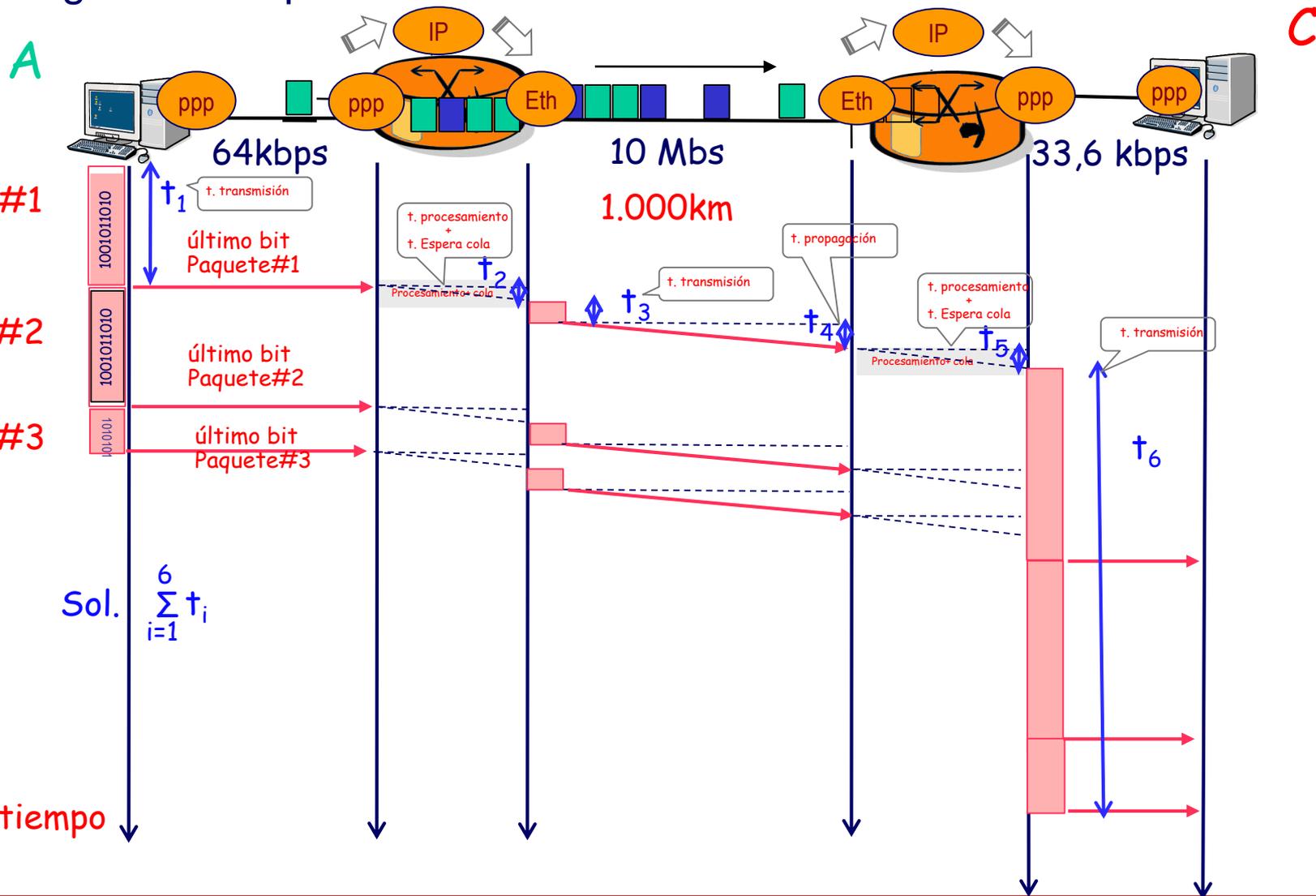


Ejercicio 2

MTU = 1.000Bytes
(NO INCLUYE LA CAB. ENLACE)

$\frac{2.000 \text{ Bytes}}{\text{MTU} - 8 - 20} = 3 \text{ paquetes}$

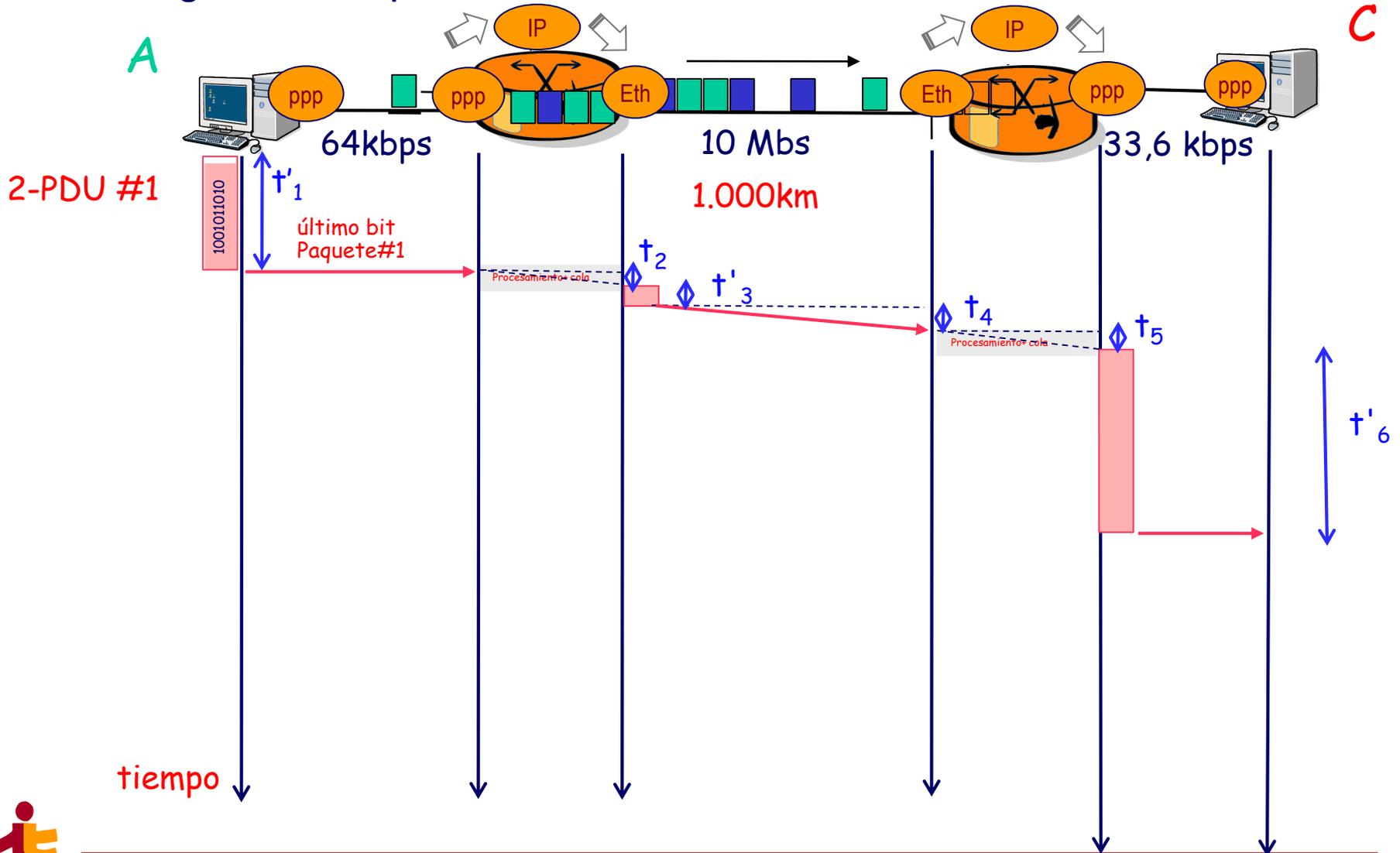
Diagrama Temporal



Ejercicio 2 (a)

MTU infinita = 1 paquete

Diagrama Temporal

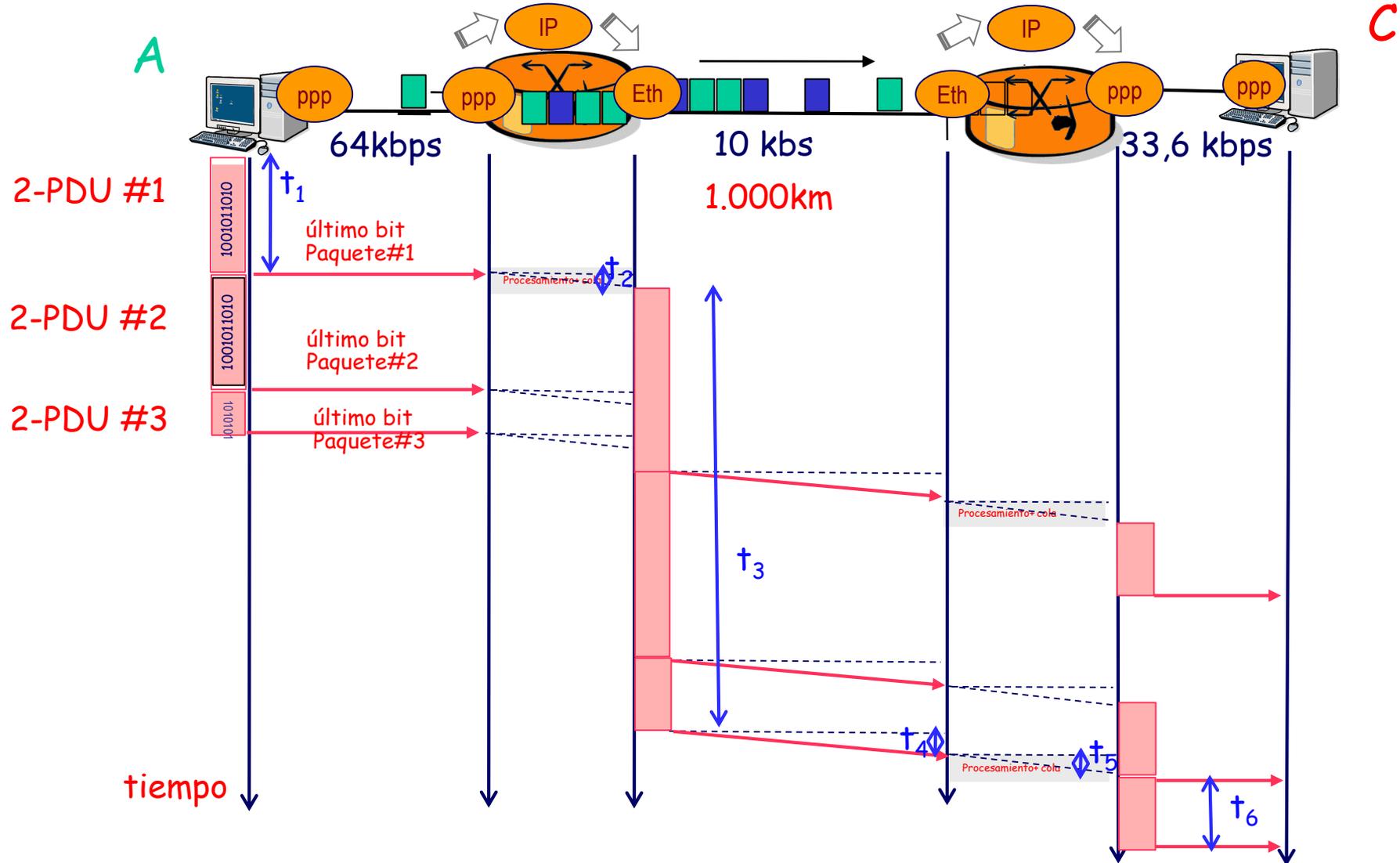


Ejercicio 2 (b)

$$\frac{2.000 \text{ Bytes}}{\text{MTU} - 8 - 20} = 3 \text{ paquetes}$$

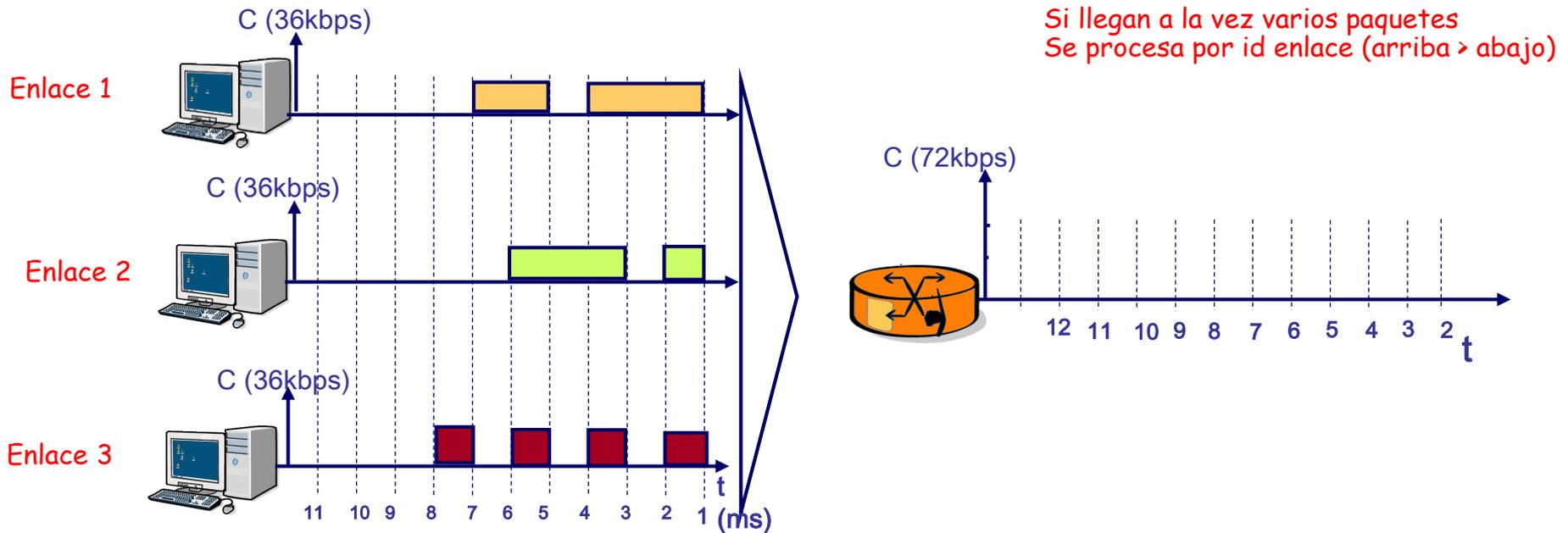
MTU - 8 - 20

(MTU = 1.000Bytes)



Ejercicio 3: multiplexión estadística

□ Dibuje la ocupación del enlace de salida (trafico periodico 10.i.t)

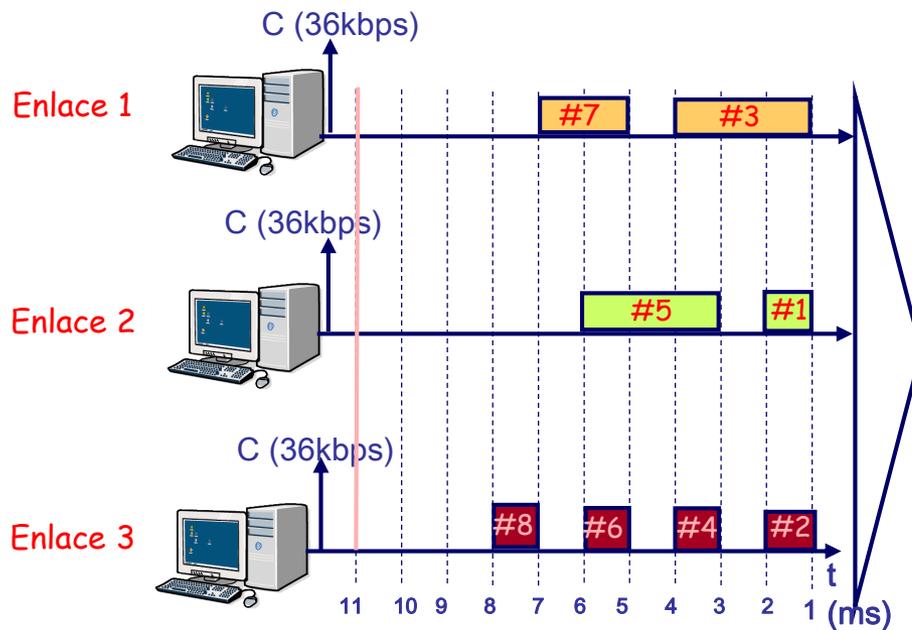


- ❖ Throughput que atraviesa el nodo
- ❖ ¿cuál es el tiempo medio de espera en cola que sufren los paquetes?
- ❖ Si el patrón de tráfico se repite periódicamente cada 10 i.t. ¿qué porcentaje de tiempo esta ocupado el enlace de salida?
- ❖ ¿qué tamaño de cola necesitamos para que no haya pérdidas?

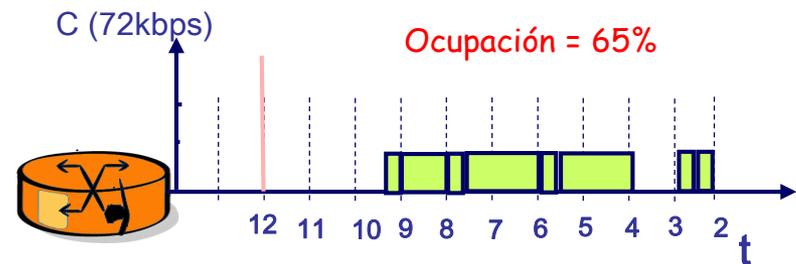


Ejercicio 3:

□ Dibuje la ocupación del enlace de salida (trafico periodico 10.i.t)



Si llegan a la vez varios paquetes
Se procesa por id enlace (arriba > abajo)



#PAQ	t_arriv	t_inicioS	t_finS	W
1	2	2	2.5	0
2	2	2.5	3	0.5
3	4	4	5.5	0
4	4	5.5	6	1.5
5	6	6	7.5	0
6	6	7.5	8	1.5
7	7	8	9	1
8	8	9	9.5	1

Throughput = 65% de 72kbps = 46,8 kbps

(50% 36kbps + 40% 36kbps + 40% 36kbps)

En el instante 7, esperan los paquetes #6 y #7.

#6 → ttx = 1ms → 36 bits

#7 → ttx = 2ms → $0.002 \times 36000 = 72$ bits

Total = 36 + 72 = 108 bits.

AVE(W) = 5.5ms/8

