

# Redes de datos 1

Carreras de:

Ingeniería Eléctrica

Ingeniería en Sistemas de Comunicación

Facultad de Ingeniería – Universidad de la República

Instituto de Ingeniería Eléctrica

2023

# Docentes

- Gabriel Gómez Sena (ggomez@fing.edu.uy) (Responsable del curso)
- Eduardo Cota (cota@fing.edu.uy)
- Alvaro Valdes (valdes@fing.edu.uy)
- Martín Randall (mrandall@fing.edu.uy)
- Lucas Inglés (lucasi@fing.edu.uy)
- Romina García (rominag@fing.edu.uy)
- Máximo Pirri (mpirri@fing.edu.uy)

# Objetivos

- **Objetivos generales**

- Al finalizar el curso se espera que el estudiante sea capaz de comprender:
  - los conceptos fundamentales de las redes de datos o redes de comunicaciones
  - la utilidad de un modelo organizado en capas
  - los objetivos y funciones principales de cada capa
  - los aspectos fundamentales de las redes de datos basadas en la familia de protocolos TCP/IP que se usan en Internet

- **Objetivos específicos**

- Al finalizar el curso se espera que el estudiante sea capaz de:
  - comprender la complejidad involucrada en la comunicación de aplicaciones de diferentes dispositivos a través de una red de comunicaciones (por ejemplo Internet)
  - comprender el funcionamiento de los principales protocolos y tecnologías utilizados en Internet (TCP, UDP, IP, Ethernet)
  - configurar un equipo en una red TCP/IP
  - adquirir elementos básicos para el diagnóstico de la conectividad de una red

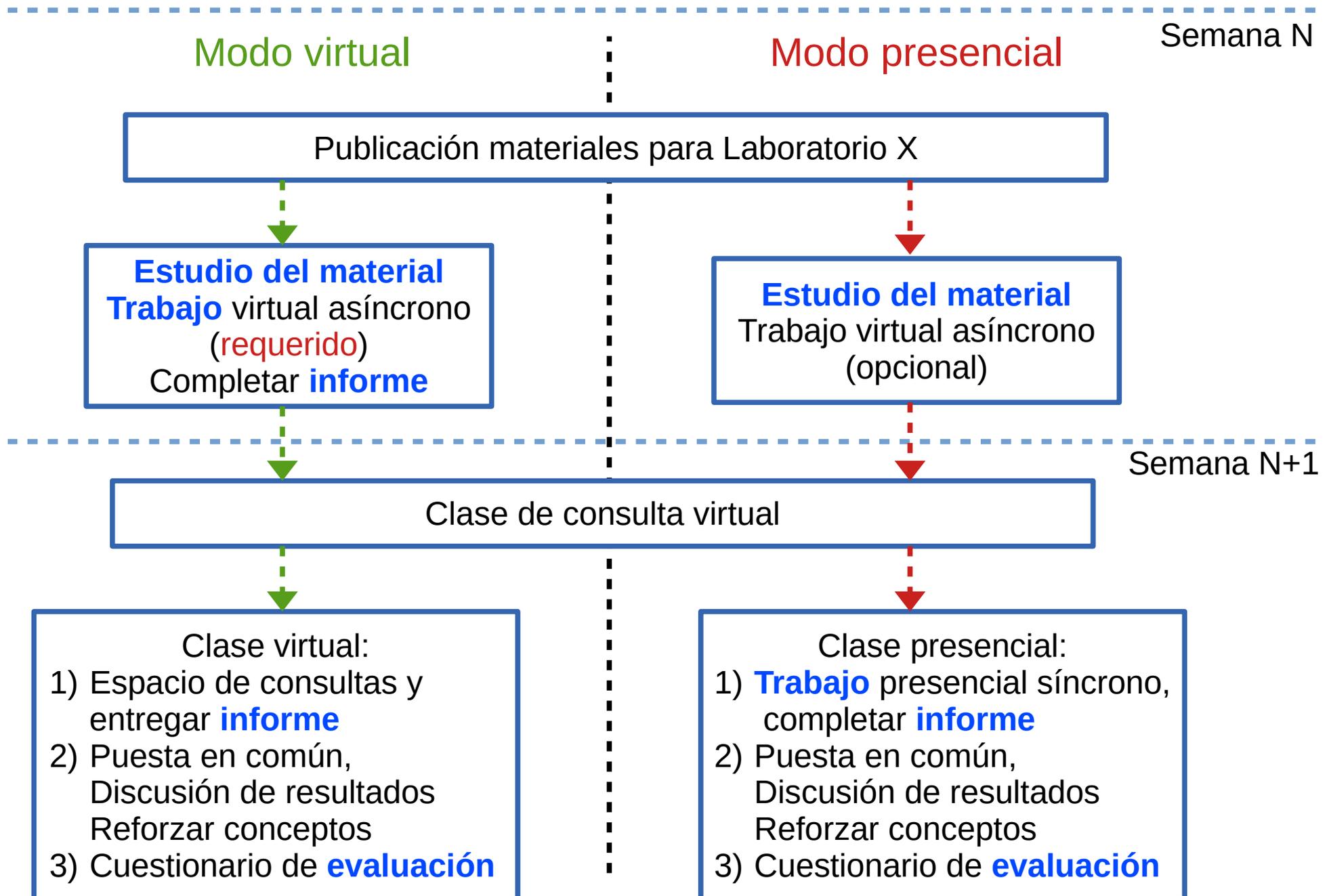
# Detalles de la Unidad Curricular (UC)

- Créditos: 8
- Teórico:
  - Lunes de 18:00 a 20:00 (Aulario B21)
  - Miércoles de 18:00 a 20:00 (3er subsuelo IMFIA 031, piso verde)
- Laboratorios
  - 7 prácticas de laboratorio en el semestre
  - Grupo 1: Martes de 18:00 a 21:00 (LabSoft IIE)
  - Grupo 2: Miércoles de 14:00 a 17:00 (LabSoft IIE)
  - Grupo 3: Jueves de 18:00 a 21:00 (virtual sala zoom)
  - En EVA está publicado un cronograma tentativo de laboratorios
- En [eva.fing.edu.uy](http://eva.fing.edu.uy) se publican las presentaciones del teórico y el material para los laboratorios
- Previa: Curso de Diseño lógico

# Laboratorios

- Los laboratorios podrán realizarse en modalidad **presencial** o **virtual**
- Los laboratorios se realizarán en grupos de a lo sumo **3 integrantes**
- Se pondrá a disposición una **máquina virtual** donde se emularán los componentes de una red adecuada para cada laboratorio
  - La máquina virtual se instala en una computadora real (Linux/Windows/otros)
- Para cada laboratorio se pondrá a disposición:
  - Un archivo conteniendo la **definición del laboratorio** que deberán instalar en la máquina virtual
  - Un **instructivo o guía de laboratorio** que contendrá información específica sobre el tema a abordar
  - Un **procedimiento o informe** a seguir para completar la práctica. En algunos laboratorios se entregarán otros documentos adicionales
- **Evaluación** del laboratorio:
  - Cada laboratorio se evaluará a través de un **cuestionario** individual, en EVA que se realizará en el horario de la clase de laboratorio
  - Los grupos deberán entregar un informe de cada laboratorio el día de la clase
  - Cada cuestionario otorgará puntos

# Laboratorios modalidad virtual o presencial



# Elección de modalidad de laboratorios

- Se disponibilizará una actividad en EVA para conformación de grupos de 3 estudiantes
  - Los grupos tienen nombre de la forma Grupo-Dia-NN
- Se espera que los grupos se mantengan con la misma integración durante el curso
- Los grupos de cada día conformarán agrupamientos
- La modalidad de los horarios es:
  - Martes – presencial
  - Miércoles – presencial
  - Jueves – virtual

# Evaluación del curso

- Se realizarán dos parciales que totalizarán un máximo de 85 puntos
- Se requerirá obtener un mínimo de 10 puntos en el primer parcial y no habrá mínimo para el segundo parcial
- El laboratorio totalizará un máximo 15 puntos
- Se requerirá obtener un mínimo de 8 puntos para la aprobación del laboratorio
- Según los puntos obtenidos en el curso:
  - (a) El estudiante que, habiendo cumplido con los mínimos estipulados en parciales y laboratorio, acumule **60 o más puntos, exonerará la UC**
  - (b) El estudiante que, habiendo cumplido con los mínimos estipulados en parciales y laboratorio, acumule **entre 25 y 59 puntos**, ganará el curso y obtendrá el derecho a rendir un **examen** para aprobar la UC
  - (c) El estudiante que no cumpla con los mínimos estipulados en parciales y laboratorio o acumule **menos de 25 puntos, reprobará** la UC
- Para los estudiantes del caso (b) anterior, el examen consistirá en una prueba que abarque los contenidos de teórico y de laboratorio

# Bibliografía y material para el curso

- Presentaciones del curso teórico
- Material de los laboratorios
- Kurose, James; Ross, Keith (2016), [Computer Networking: A Top-Down Approach](#) (7th Edition). Pearson.
- Tanenbaum, Andrew S.; Wetherall, David J (2010). Computer Networks (5th edition). Prentice Hall.
- Fall, Kevin R.; Stevens, W. Richard (2011). TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition). Addison-Wesley Professional Computing Series
- Openfing:
  - Redes de Datos (IIE)
    - Versión anterior de este curso que tendrá diferencias, pero también similitudes en algunos temas
    - Algunos temas no los vamos a dictar
  - Redes de Computadoras (INCO)
    - Tiene puntos en común

Se disponibilizarán las clases grabadas del teórico del curso de la edición 2021

# ¿Qué son las redes de datos?

- Dispositivos interconectados para intercambiar información (datos)

# ¿Qué son las redes de datos?

- Dispositivos interconectados para intercambiar información (datos)

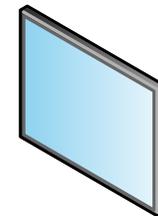
Computadora



Portátil



Teléfono



Televisor

# ¿Qué son las redes de datos?

- Dispositivos interconectados para intercambiar información (datos)

Computadora



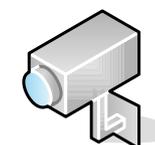
Portátil



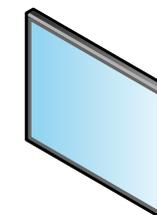
•



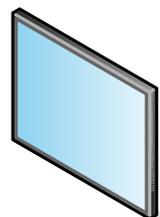
Impresora



Cámara



Televisor

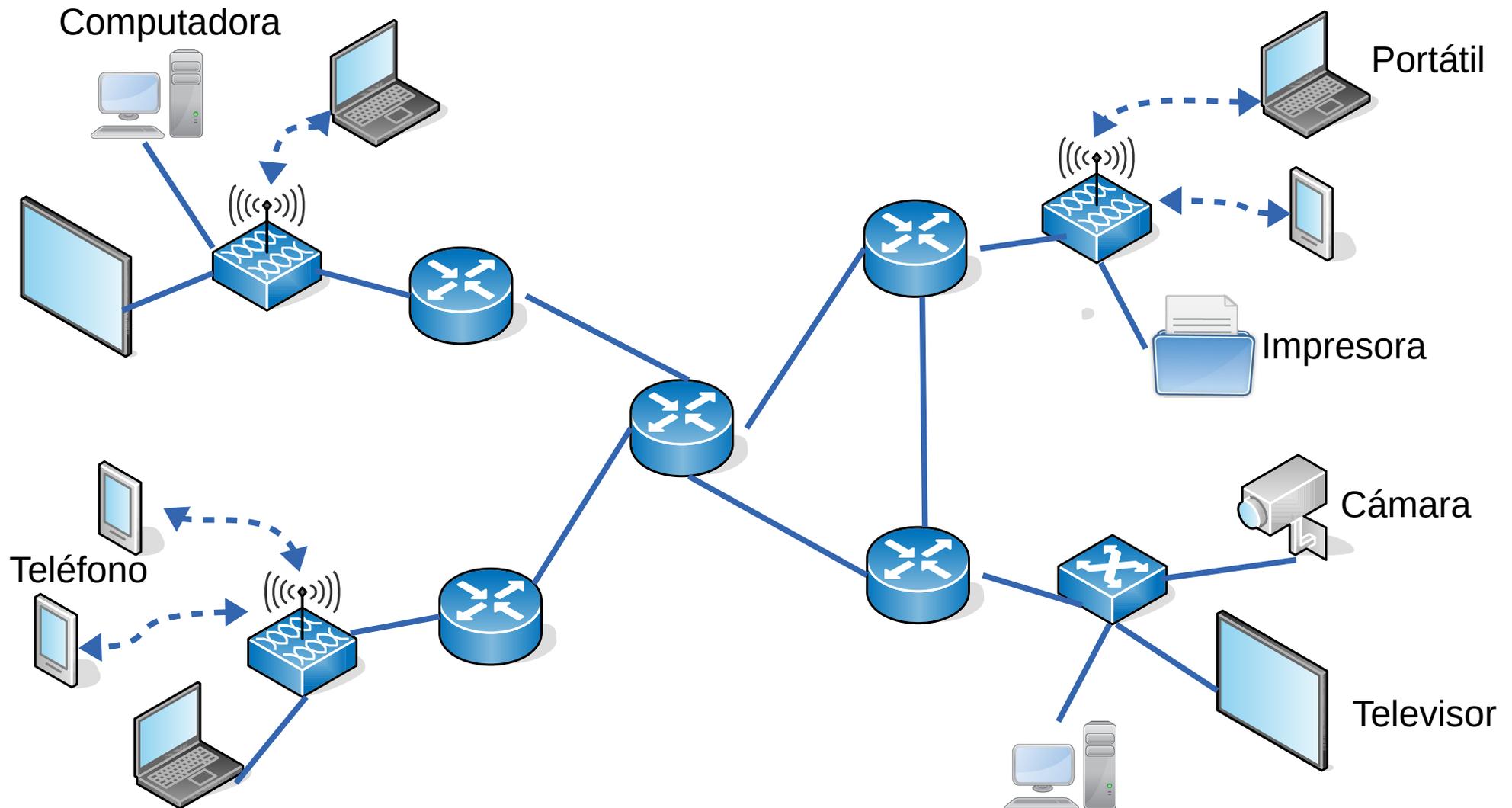


Teléfono



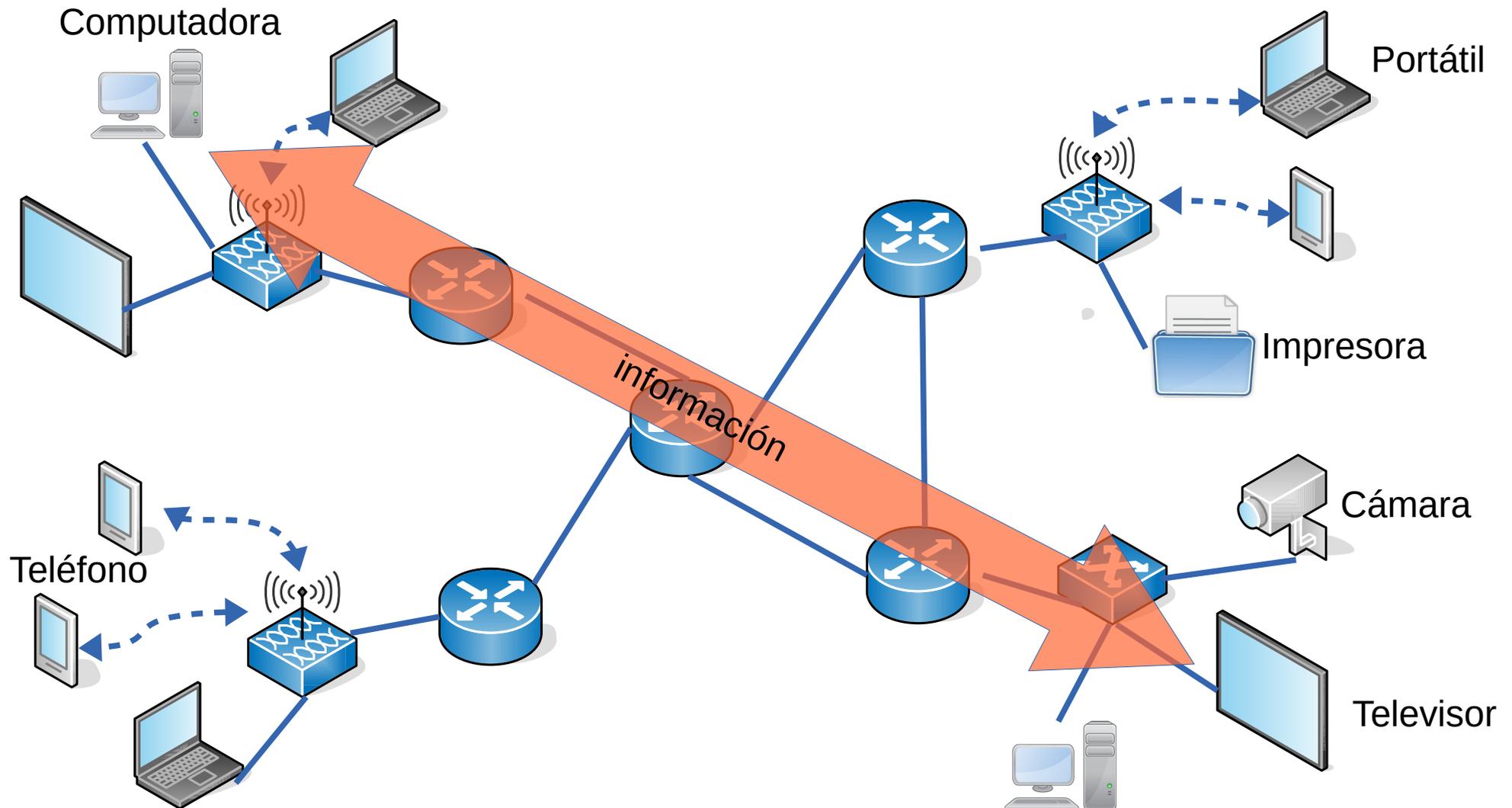
# ¿Qué son las redes de datos?

- Dispositivos interconectados para intercambiar información (datos)



# ¿Qué son las redes de datos?

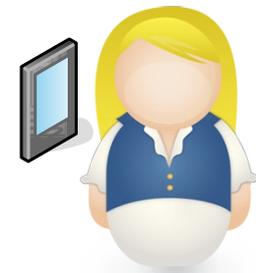
- Dispositivos interconectados para intercambiar información (datos)



# ¿Para qué sirven las redes de datos?

- Medio de comunicación entre personas
- Mensajes, chats, whatsapp
- Correo electrónico
- Reuniones virtuales. Video conferencia
- Búsqueda y obtención de información, navegación, descargas
- Grupos de trabajo dispersos, compartir archivos y documentos
- Trabajo a distancia
- Educación
- Entretenimiento
- Comercio y banca electrónica
- Etc.

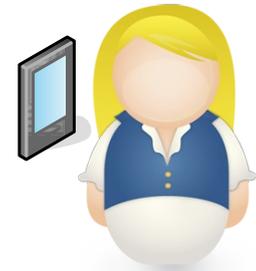
# Ejemplo: comunicación por whatsapp



# Ejemplo: comunicación por whatsapp



Cómo va?

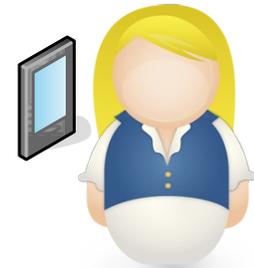


# Ejemplo: comunicación por whatsapp

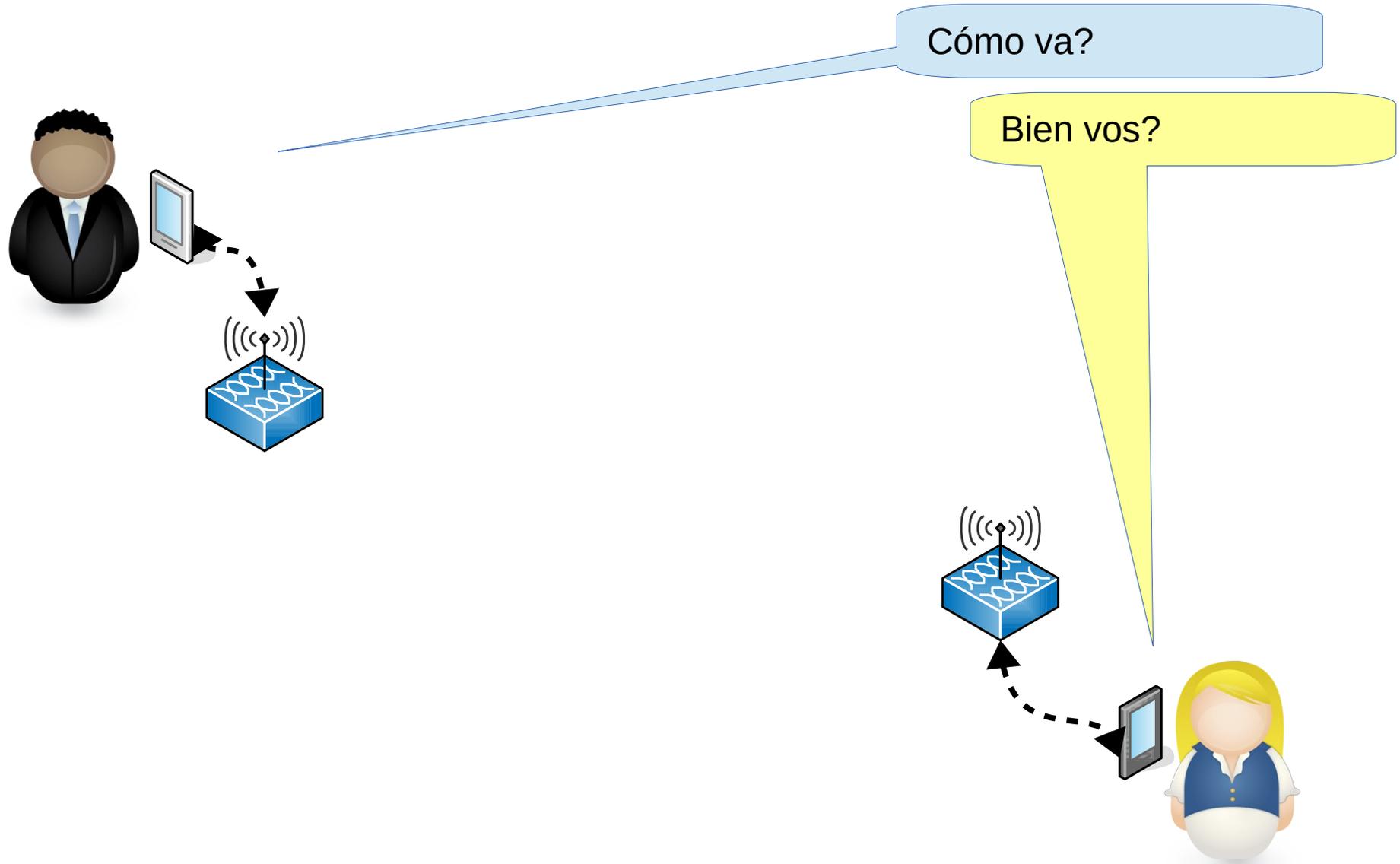


Cómo va?

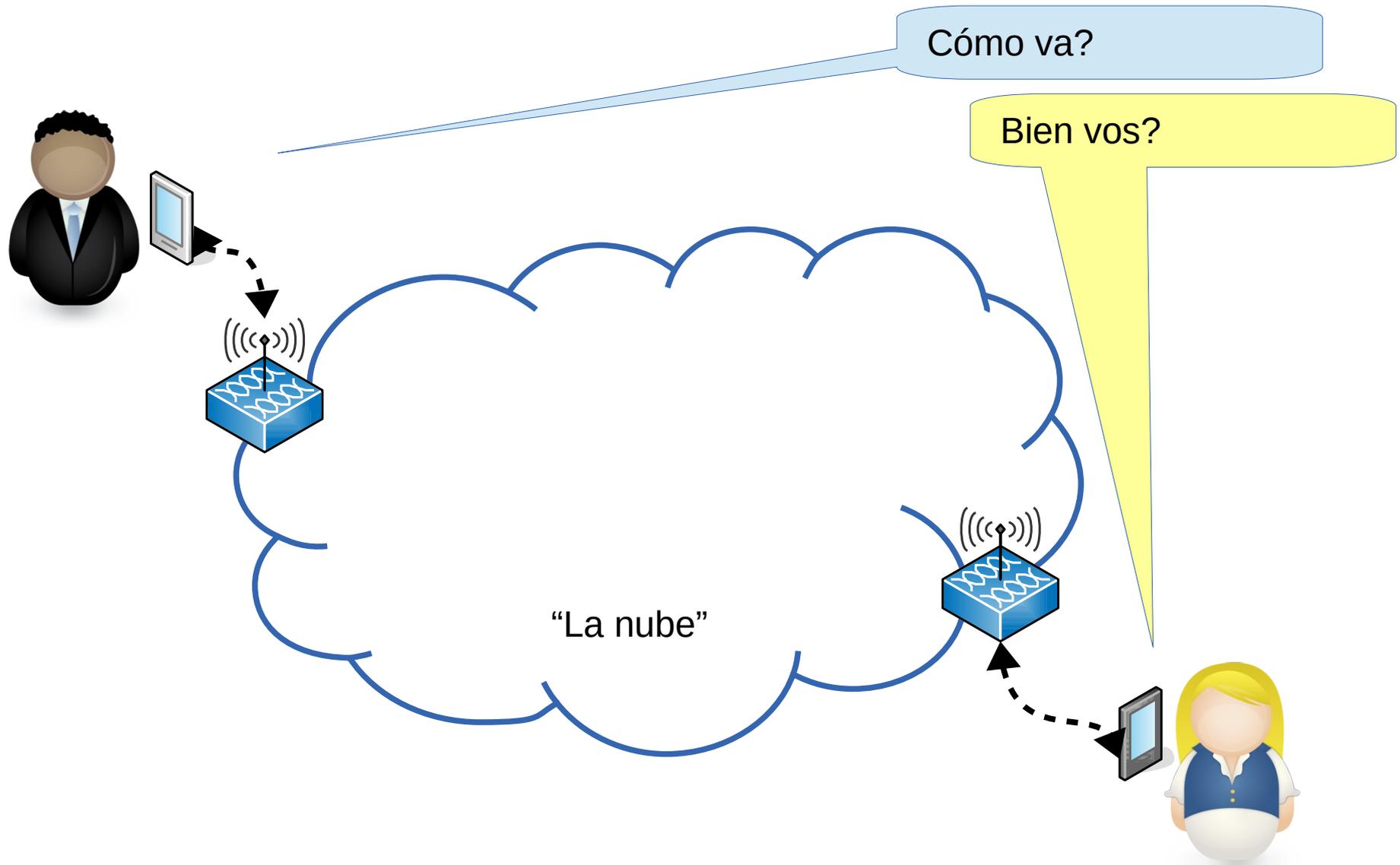
Bien vos?



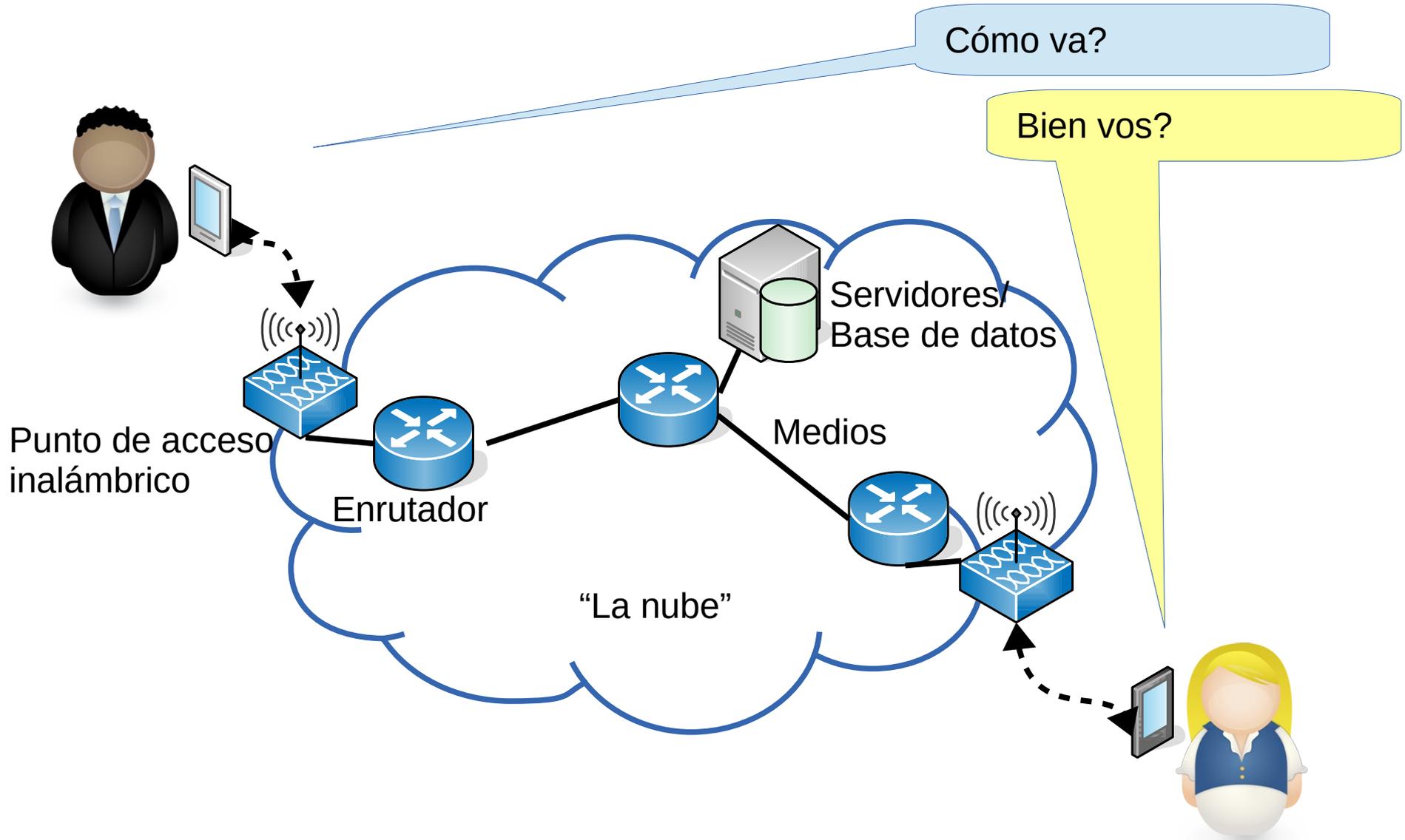
# Ejemplo: comunicación por whatsapp



# Ejemplo: comunicación por whatsapp



# Ejemplo: comunicación por whatsapp

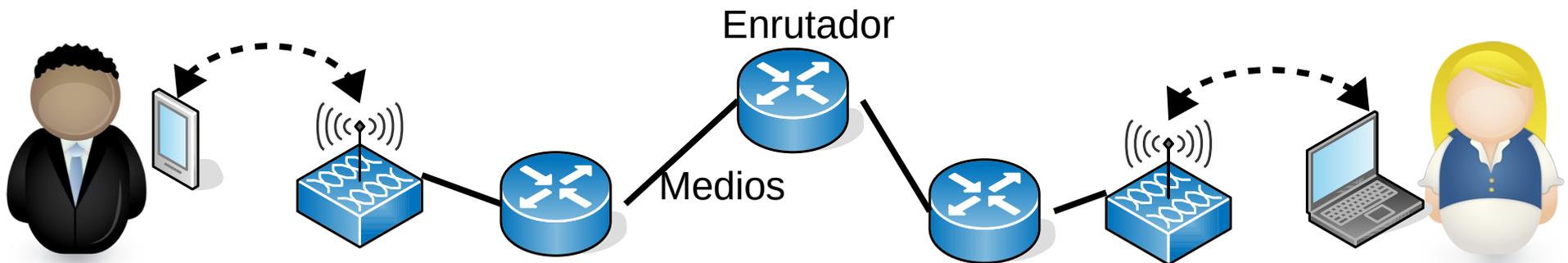


# Componentes de la red

- **Dispositivos** en los extremos de la red (Dispositivos finales, End Points)
  - Computadoras
  - Celulares
  - TVs
  - Impresoras
  - Cámaras
  - Heladeras?, etc., etc.
  - Internet de las Cosas (Internet of Things, IoT)
- Que puedan transformar la información (texto, imágenes, audio, video) a formato digital (representación mediante ceros y unos)
- Que puedan conectarse a otros dispositivos mediante interfaces cableadas o inalámbricas

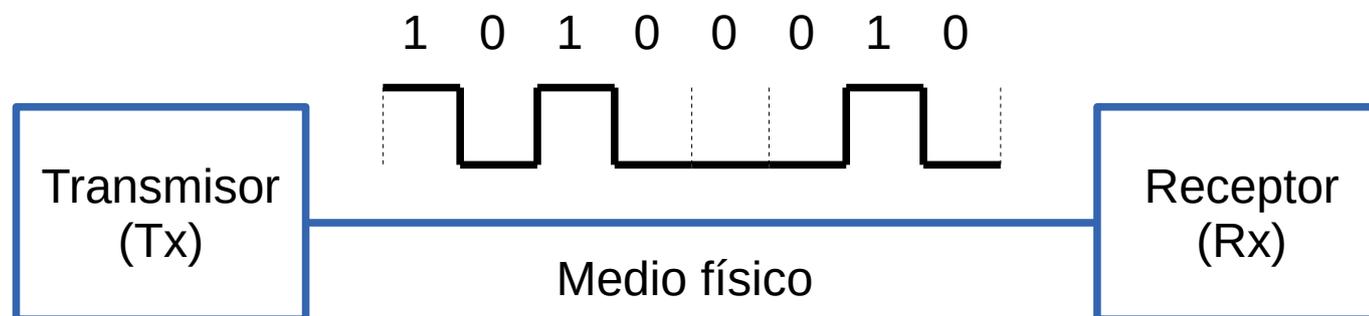
# Componentes de la red

- Medios de comunicación
  - Cableados, inalámbricos
  - Comunicaciones a distancia
  - Permiten transmitir y recibir la información expresada como secuencias de ceros y unos, usando señales eléctricas, ópticas, etc sobre medios cableados o inalámbricos
- Dispositivos intermedios
  - Puntos de acceso, Conmutadores, Enrutadores, Switches
  - Que resuelvan la conectividad global



# Medios físicos

- Para transmitir información (bits) por un medio físico se los traduce a señales electromagnéticas
- Ejemplo: representación de ceros y unos como señales eléctricas en un medio físico



- Normalmente no se usa esta representación simple por problemas de detección, sincronización, ruido, atenuación de los medios físicos, etc (no veremos estos problemas en este curso)
- El camino de un extremo a otro de la red surgirá de la concatenación de varios tramos transmisor-receptor eventualmente sobre diferentes medios
- Estos tramos se llaman [enlaces \(links\)](#)
- La capacidad para transmitir información se mide en [bits por segundo \(bps\)](#)

# Medios físicos

- Medios que se usan típicamente para transmisión de datos
- Medios **guiados** (por un medio sólido)
  - Pares trenzados de cobre
  - Cable coaxial
  - Fibra óptica
- Medios **no guiados** (inalámbricos, por el aire)
  - Radio enlaces terrestres
  - Radio enlaces satelitales
- Los medios tienen diferentes características:
  - Velocidad para la transmisión de datos
    - Se mide en bits por segundo (bps)
      - bps, kbps (kilo,  $10^3$ ), Mbps (mega,  $10^6$ ), Gbps (giga,  $10^9$ )
  - Distancias que pueden cubrir
  - Tasas de error
  - Costos

# Errores de los medios

- Los medios físicos pueden introducir errores en los bits transmitidos
- Las causas pueden ser ruido, interferencias, atenuación, dispersión o imperfecciones de los componentes (conectores, transmisores, receptores)
- Se mide con el “bit error rate” (BER): tasa de bits errados sobre el total de bits transmitidos
- Un BER de  $10^{-2}$  significa que si transmitimos 100 bits, en promedio 1 bit estará errado
- En redes inalámbricas el BER puede ser del orden de  $10^{-2}$  mientras que en fibra óptica el BER puede ser de  $10^{-13}$  o mejor
- Los errores en un canal pueden ser aislados o presentarse en ráfagas
- Se van a necesitar métodos de control de errores para detectar o corregir esos errores

# Par de cobre trenzado

- Muy usado
- Barato
- Se usa para telefonía desde hace muchos años
- Pares trenzados para reducir la interferencia entre pares cercanos
- Recubiertos con plástico para protección
- Usado en cableado de redes en edificios y casas
  - Redes de área local (LAN, Local Area Network)
  - UTP (Unshielded Twisted Pair)
- Velocidades rango de 10 Mbps a 10 Gbps
- Depende del tipo de cable y de las distancias
  - Ejemplo: Cable UTP Categoría 6a
    - Permite 10 Gbps hasta 100 metros de distancia
  - Ejemplo: cable telefónico de cobre en los hogares
    - Permite hasta 56 kbps (con módems) o decenas de Mbps (ADSL)



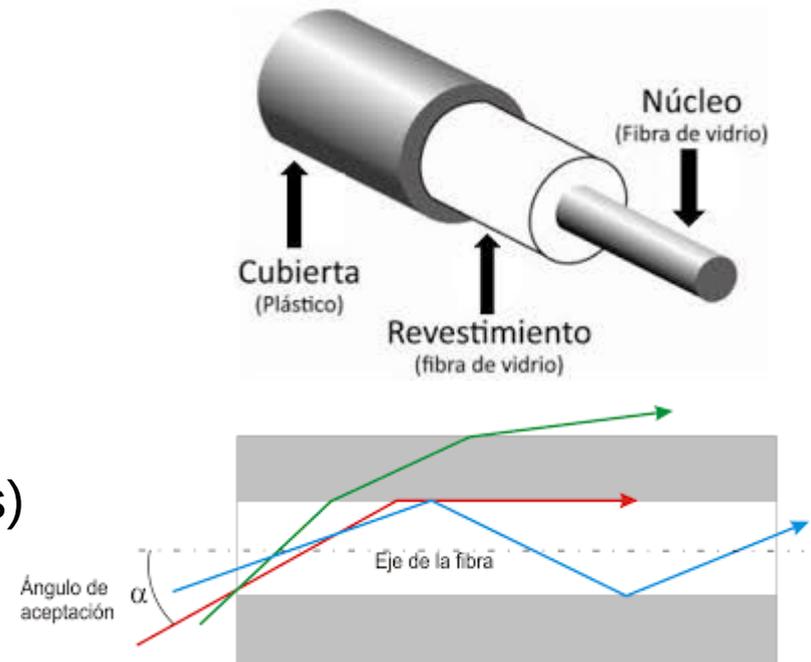
# Cable coaxial

- Un conductor central de cobre con un aislante que lo separa del otro conductor que es la malla
- Protegidos con plástico
- Uso común en TV por cable
- 1 a 2 Gbps hasta 1 km de distancia
- Los sistemas de TV por Cable en otros países se utilizan para dar conectividad a Internet
  - Recientemente cambió la regulación en Uruguay



# Fibra óptica

- Cable que conduce pulsos de luz
- Velocidades de decenas o cientos de Gbps
- Son inmunes a la interferencia
- Baja atenuación
- Ideales para largas distancias  
(por ejemplo entre continentes, ciudades)
- Alto costo por dispositivos ópticos  
(transmisores, receptores, conmutadores)
- Se usa también en centros de datos (datacenters)
- Fibra óptica al hogar (FTTH, fiber to the home)

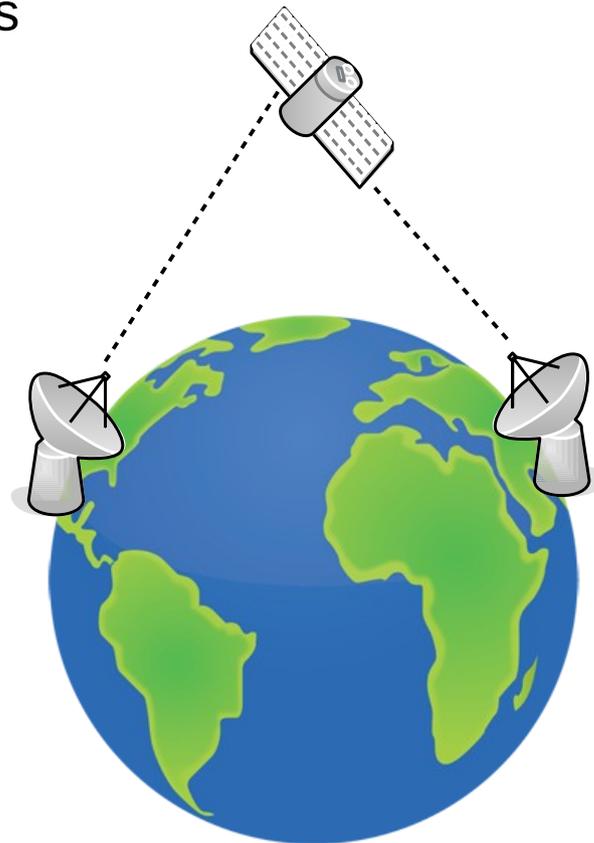


# Radio enlace terrestre

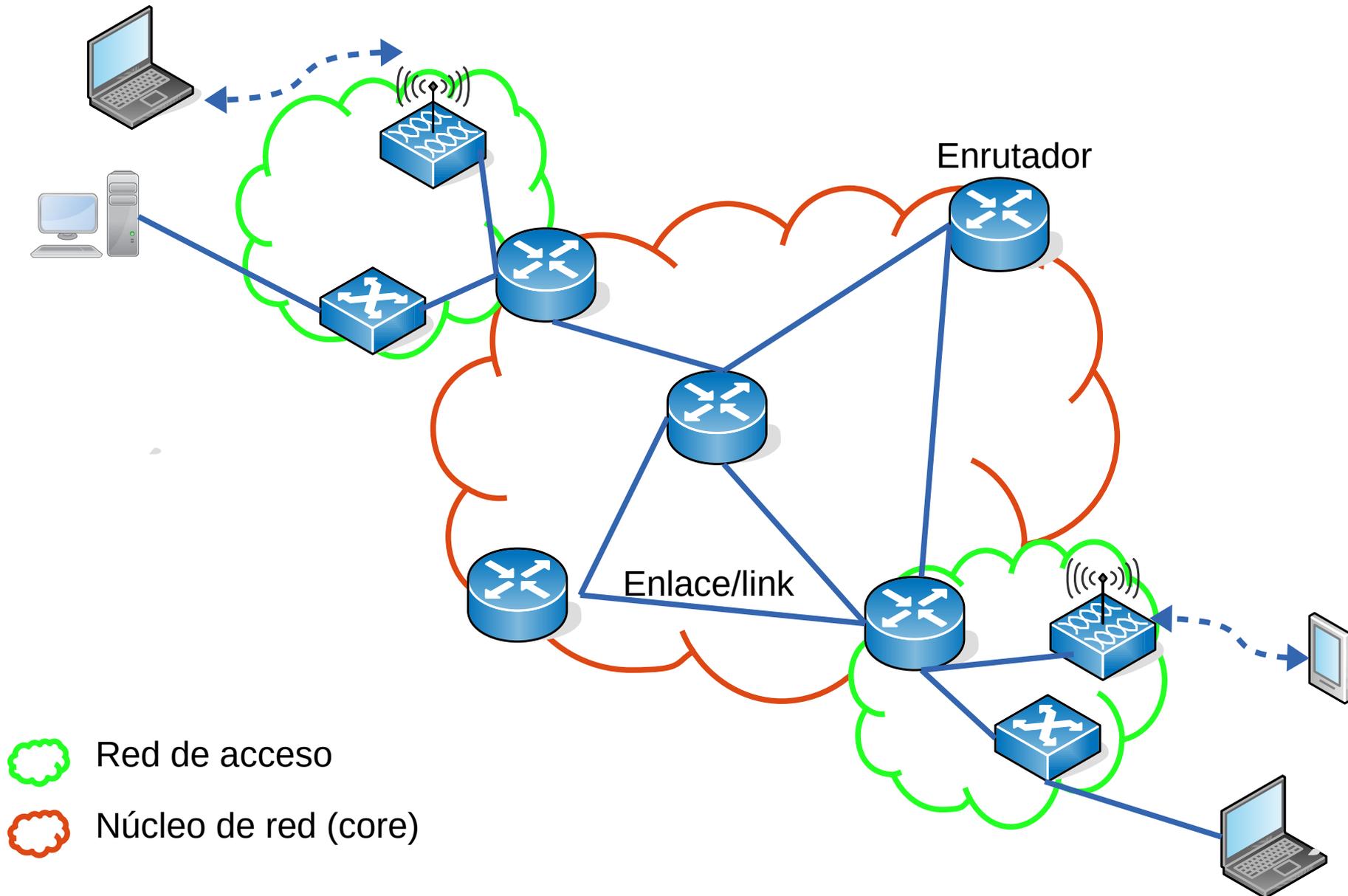
- Se transmiten señales por el aire
- No se necesita cableado
- Pueden atravesar algunos obstáculos (dependiendo de las frecuencias)
- Pueden ser útiles cuando los usuarios son móviles
- Pueden alcanzar grandes distancias
- Dependen del entorno
  - Atenuación por condiciones climáticas (niebla, lluvia)
  - Rebotes en obstáculos (árboles, edificios)
  - Interferencia con otras señales electromagnéticas
- Se usan en:
  - Cortas distancias (pocos metros) por ejemplo Bluetooth
  - Distancias medias (cien metros) por ejemplo WiFi
  - Grandes distancias (decenas de km) por ejemplo Celulares

# Radio enlace satelital

- Estaciones terrenas que envían datos al satélite
- El satélite tiene antenas para recibir y retransmitir las señales de nuevo a la tierra
- Satélites geo-estacionarios (36000 km de la tierra)
- Satélites de órbita baja (750 km de la tierra)
- Centenas de Mbps



# Esquema de una red



-  Red de acceso
-  Núcleo de red (core)

# Acceso y Núcleo de red

- Red de acceso
  - Medios cableados o inalámbricos
  - Acceso fijo o móvil
  - Ejemplos:
    - Wifi (inalámbrica, cierta movilidad)
    - Celular (inalámbrica, móvil)
    - Ethernet (cabelada, fija)
    - TV Cable (cableada, fija) (en Uruguay no reglamentada)
    - Fibra óptica (cableada, fija)
    - Cable telefónico (cableado, fijo) (ADSL)
- Núcleo de la red
  - Conectividad global – múltiples proveedores
  - Mayores velocidades
  - Agregación de datos

# Además de los componentes se necesita...

- Ponerse de acuerdo en los **formatos** en que se va a enviar la información
  - ¿Cómo se codifica un video para que cualquier reproductor lo entienda?
  - ¿Cómo hago para asegurarme que la información que quiero enviar a un usuario X, llegue a X y no a Y?
  - ¿Cómo verifico si la información llega correctamente a destino?
- Ponerse de acuerdo en reglas para el intercambio de la información
  - Reglas de procedimiento: **protocolos de comunicación**
  - Ejemplo: llamada telefónica (llamo, atiendo, hablo, corto)
- Normas, **especificaciones**, reglamentaciones
  - Para que equipos de diferentes fabricantes se entiendan
  - Ejemplo: Para que los conectores sean compatibles
  - Organizaciones que publican estándares, ejemplos:
    - Normas ISO, especificaciones de la IEEE (Institute of Electrical and Electronics Engineers), publicaciones de los grupos de trabajo en Internet (Request for Comments, RFC), etc

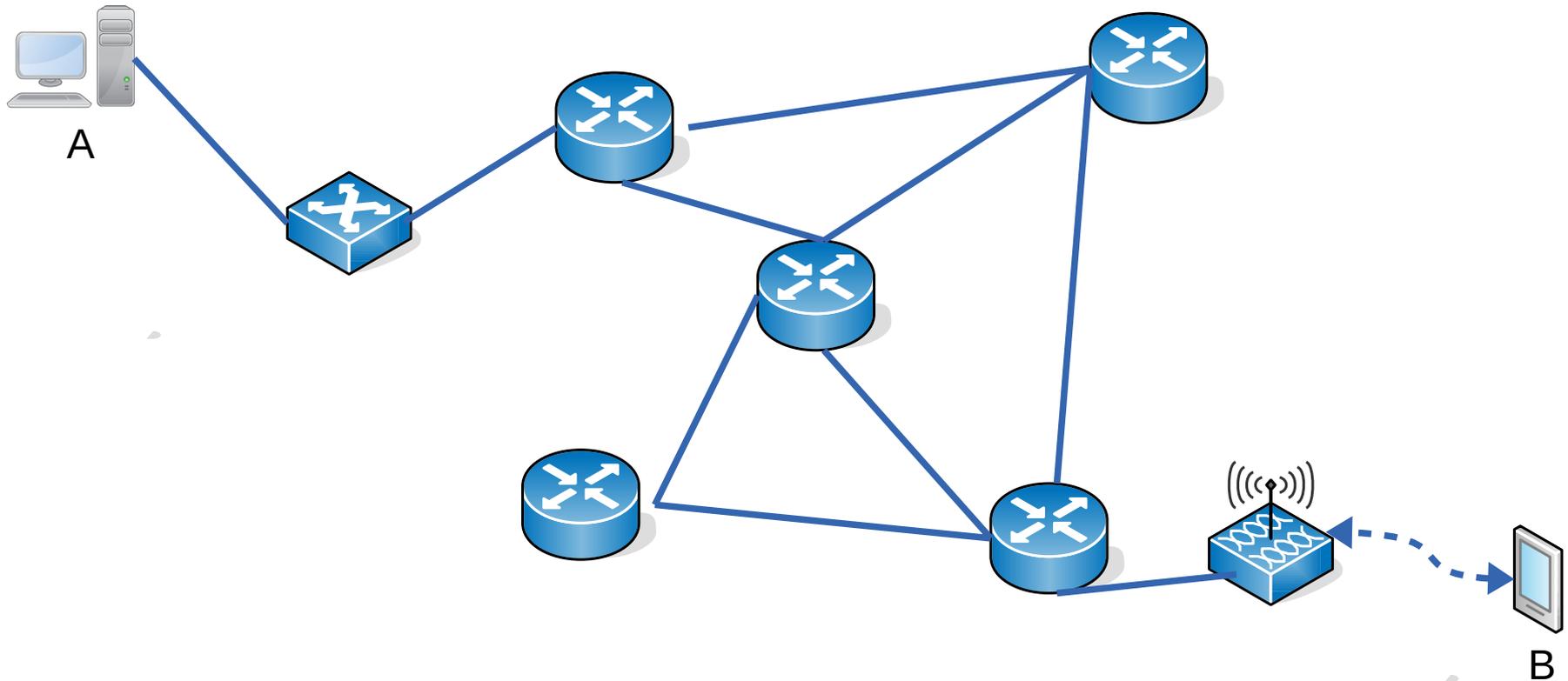
# Protocolos

- Reglas y procedimientos para intercambiar información
- Analogía humana: Le pregunto a otra persona qué hora tiene
  - A: Hola, me podrías decir la hora?
  - B: Si, cómo no, son las 12:00
  - A: Muchas gracias
  - B: De nada
- Para que los dispositivos se entiendan en una red, son necesarios **protocolos** de comunicación que deben definir:
  - El formato del mensaje o paquete a intercambiar
  - Las reglas de intercambio, la secuencia lógica de mensajes
  - Las acciones a tomar frente a un mensaje determinado
  - Los mensajes de respuesta a enviar
- En las redes van a aparecer muchos protocolos
  - Para que un transmisor y receptor se entiendan al enviarse bits
  - Para que un enrutador sepa hacia dónde debe encaminarse un paquete
  - Etc, etc.

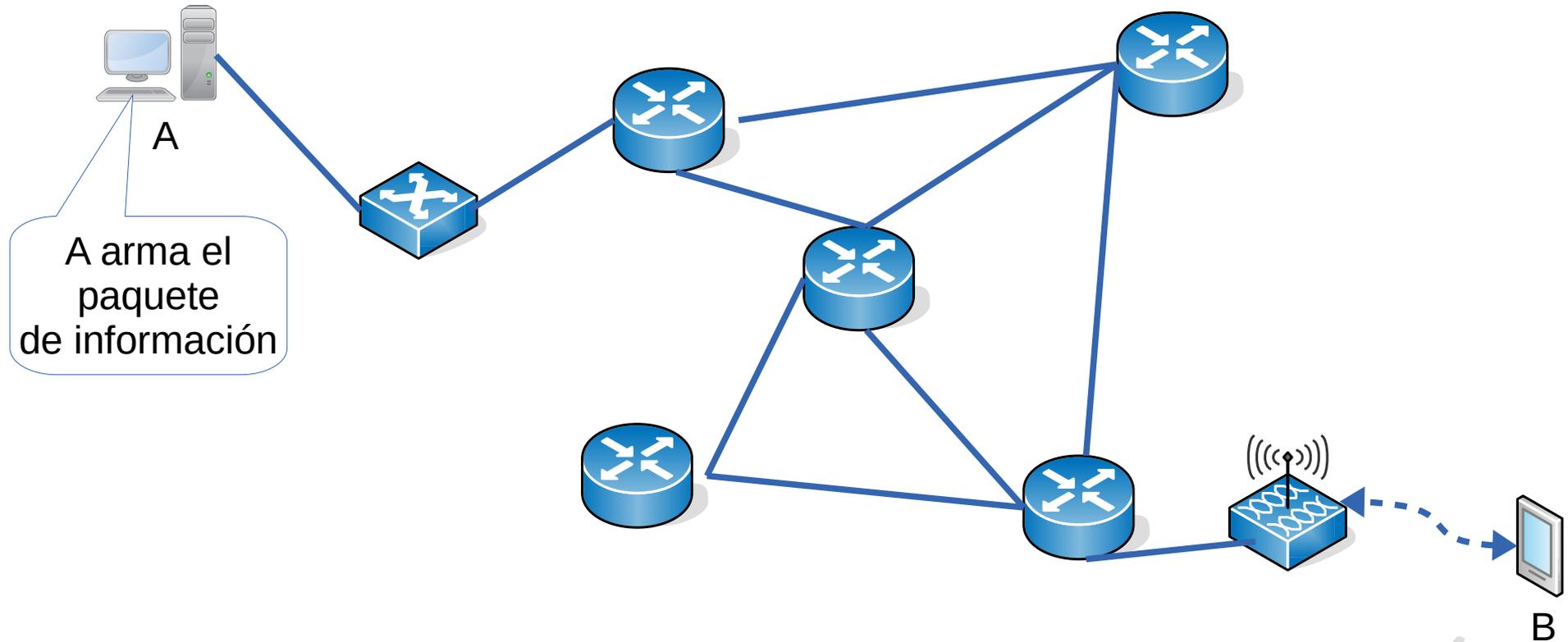
# Paquetes

- La información que queremos enviar de un dispositivo de la red a otro (por ejemplo: un correo electrónico, una imagen) se divide en unidades llamadas genéricamente **paquetes**
- Esos paquetes, además de los datos del usuario, contienen información adicional (**encabezados**)
  - Ejemplos: direcciones, números de secuencia, redundancia para verificación de errores, etc
- Es necesario acordar un formato de paquete
- Los paquetes circulan desde el origen, pasando por los enlaces y dispositivos intermedios (switches, conmutadores, enrutadores) hasta llegar al destino final
- Posibilitan un uso eficiente de los recursos de la red en base a la multiplexación estadística, ya que no todos los usuarios usan todo el tiempo la red

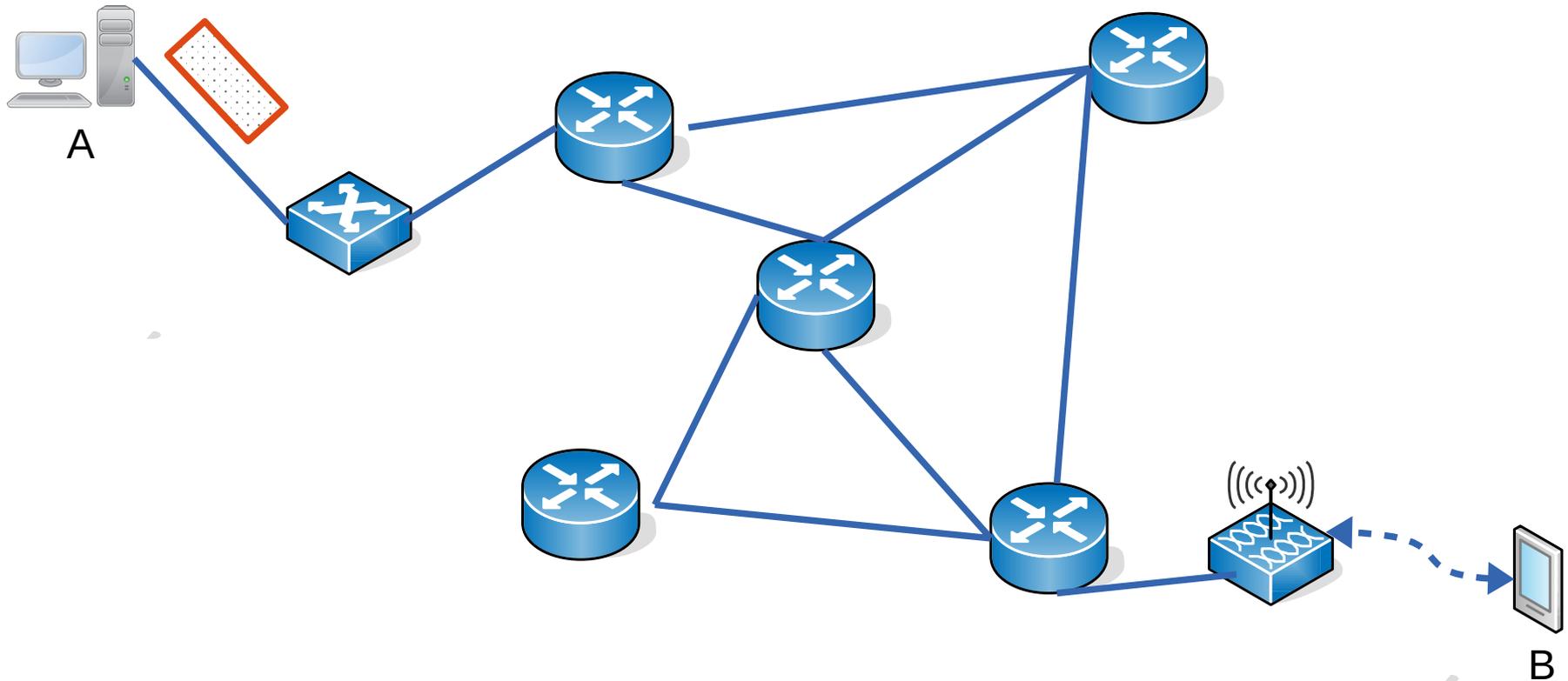
# Conmutación de paquetes



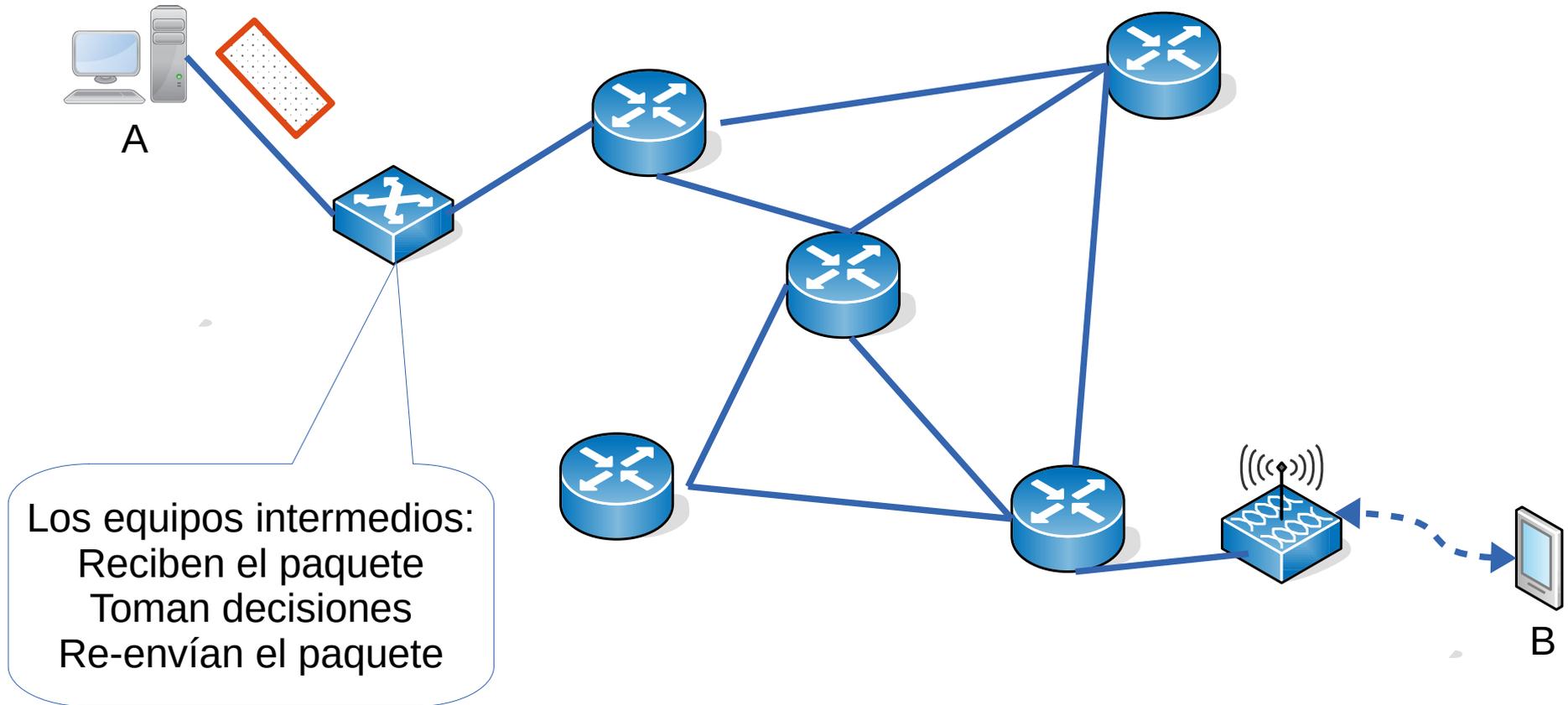
# Conmutación de paquetes



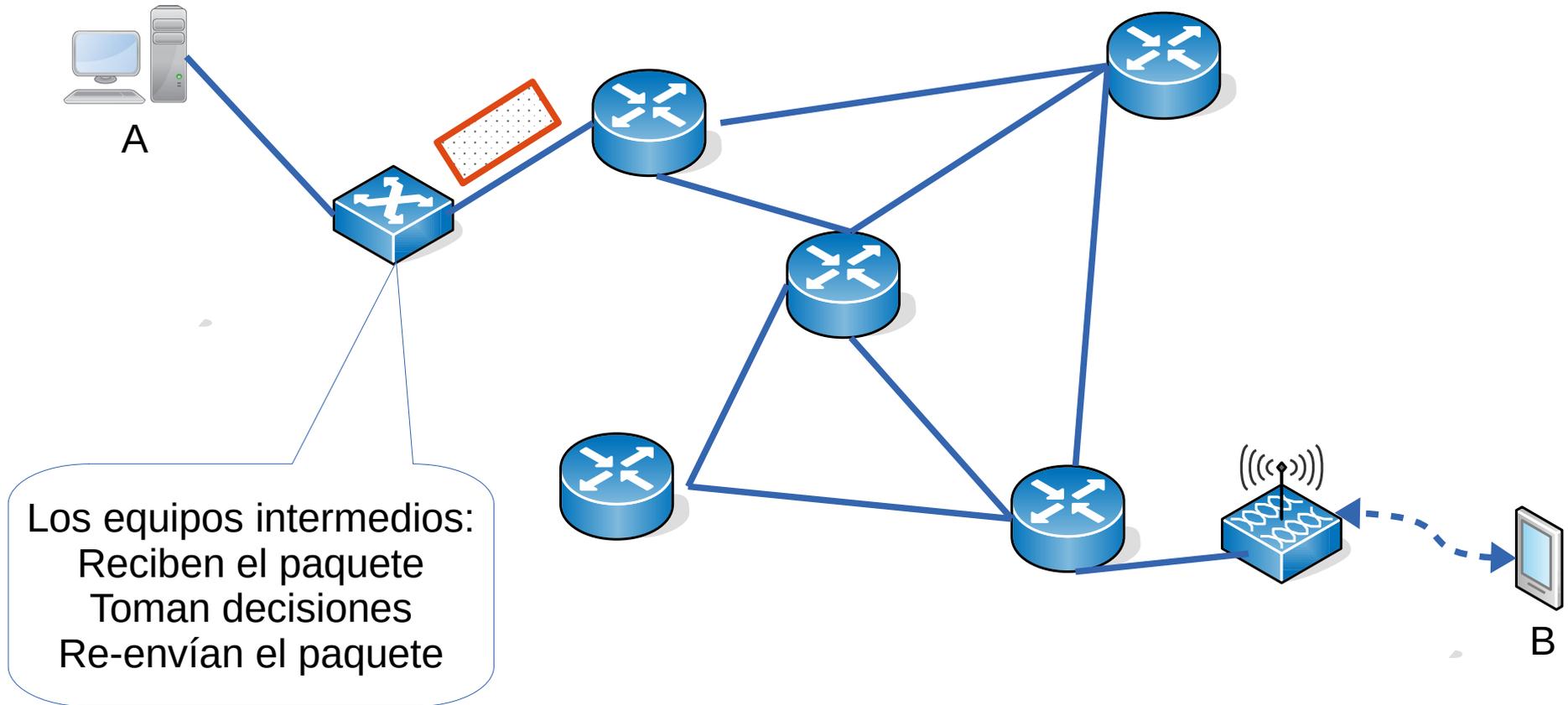
# Conmutación de paquetes



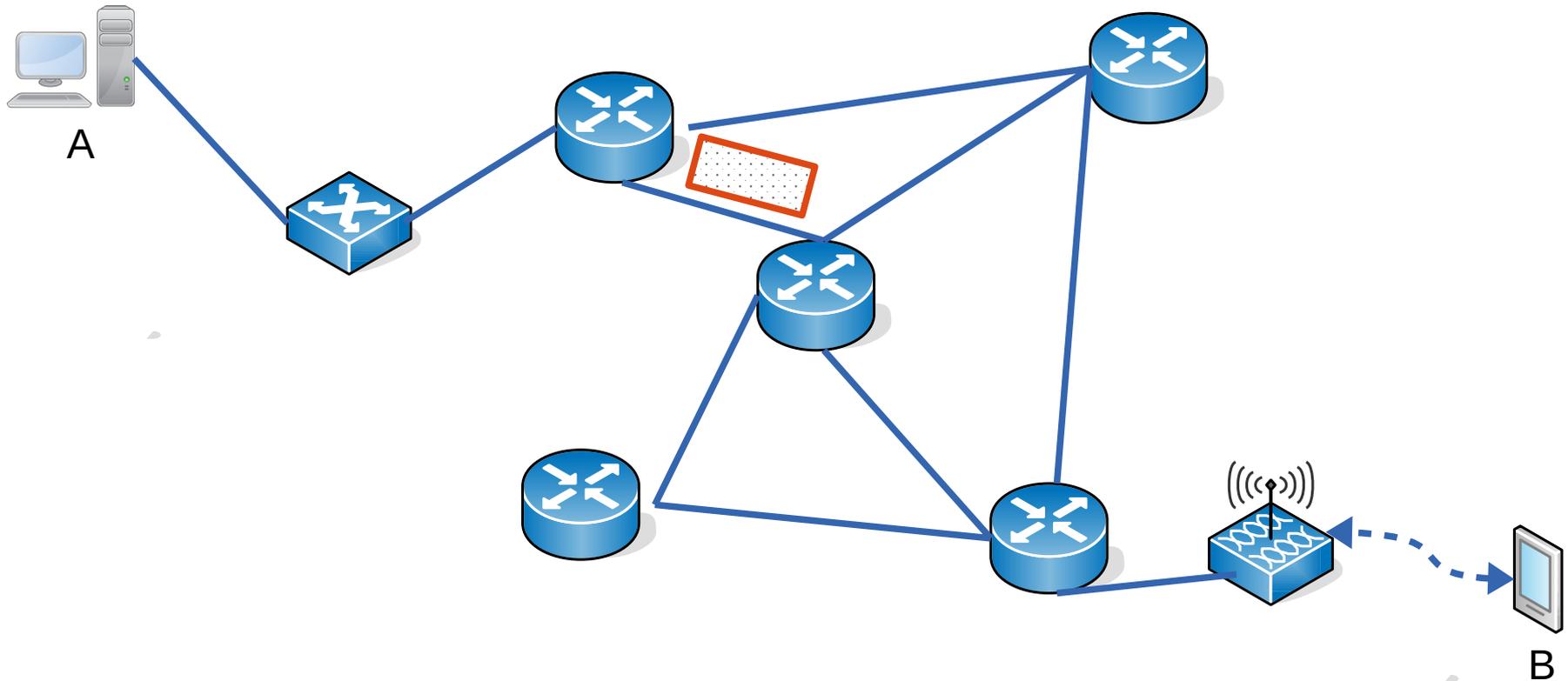
# Conmutación de paquetes



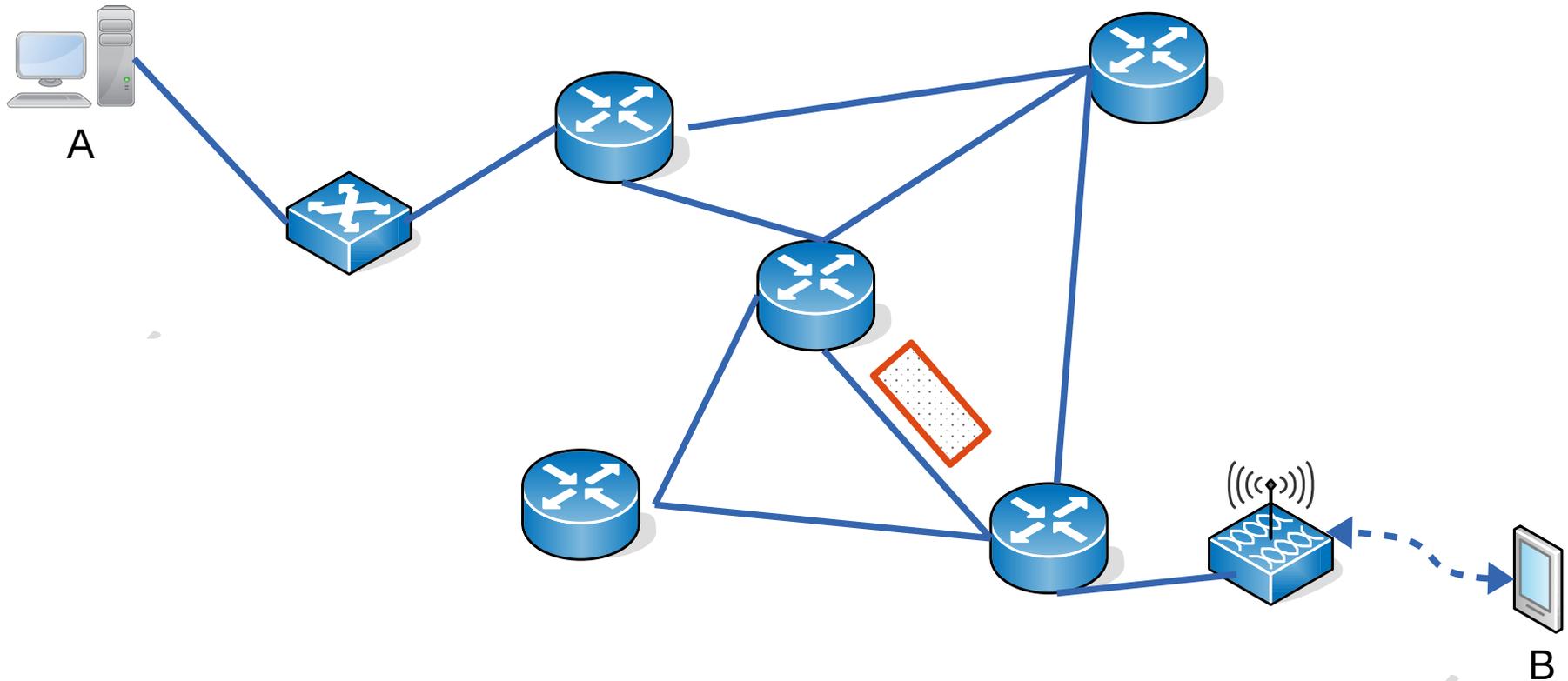
# Conmutación de paquetes



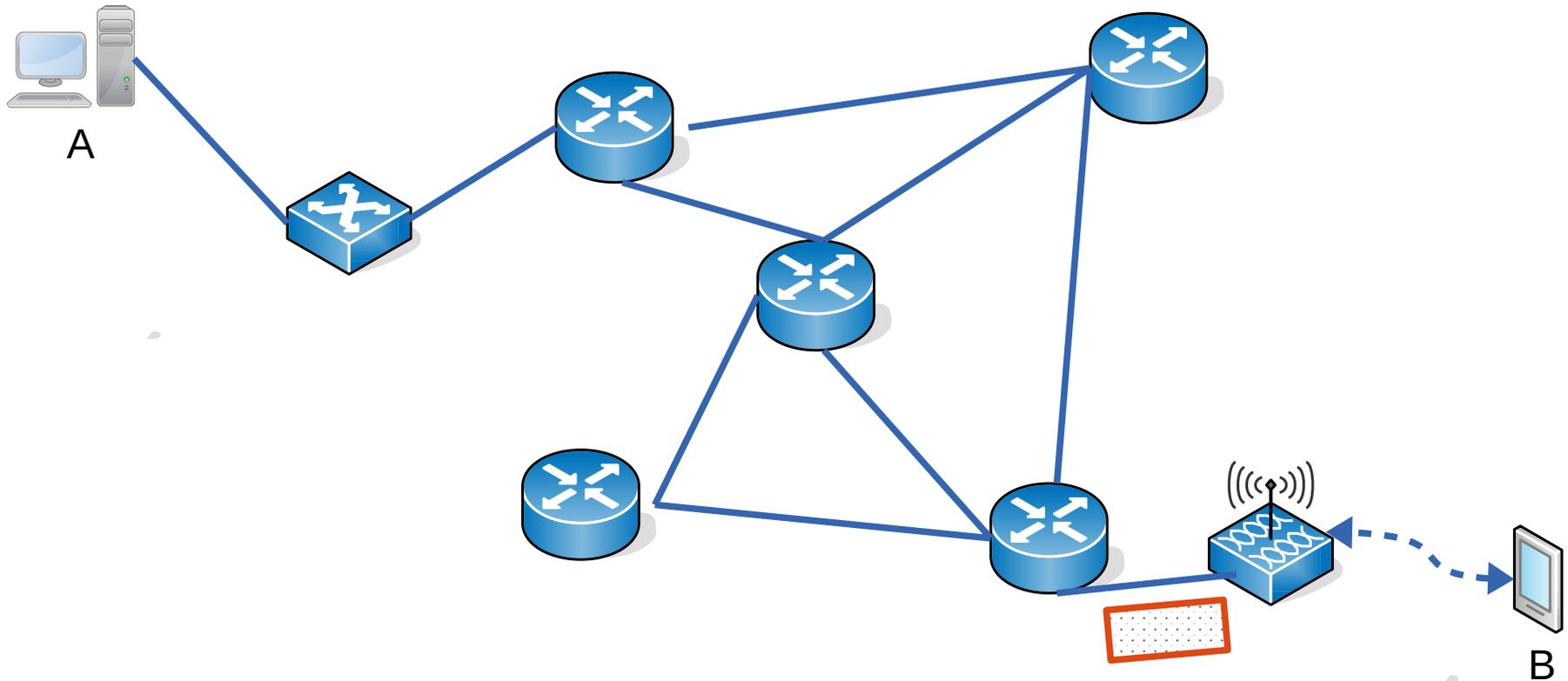
# Conmutación de paquetes



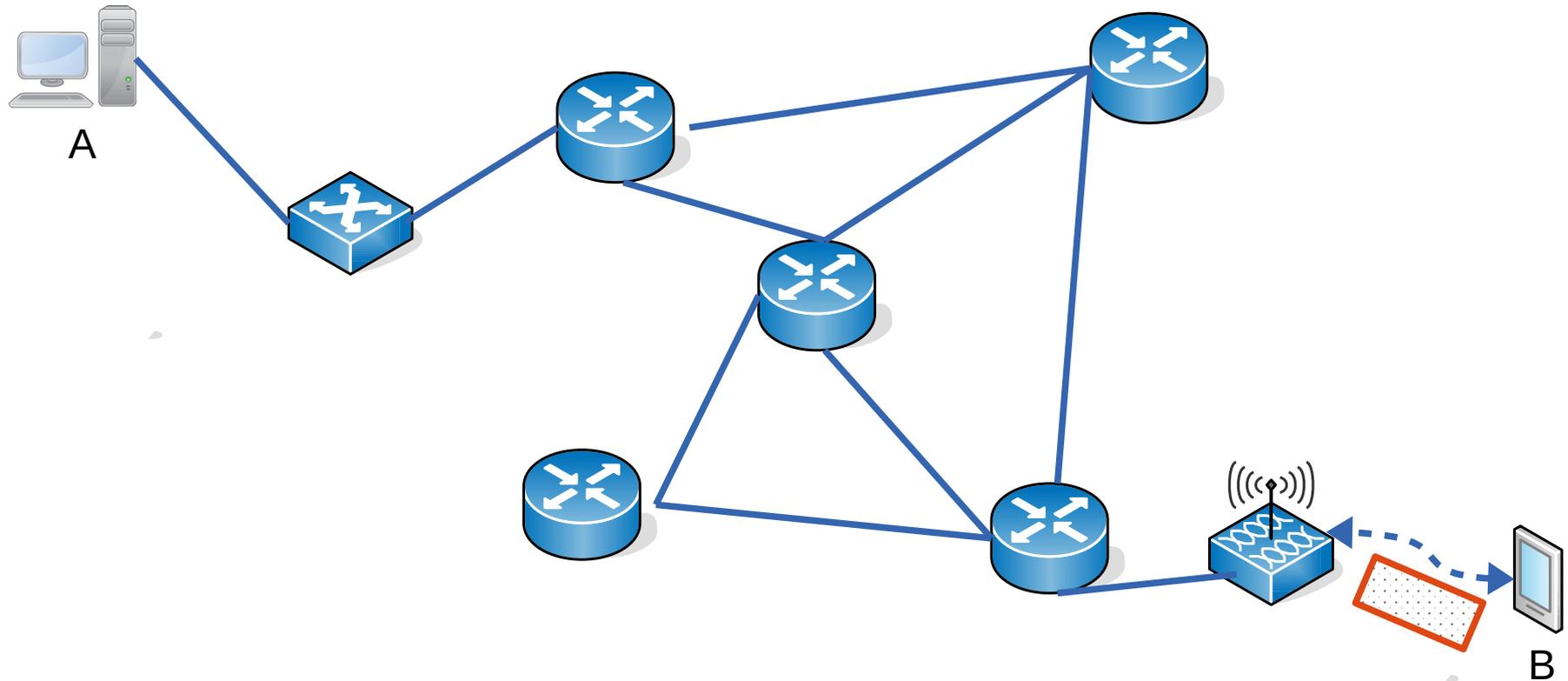
# Conmutación de paquetes



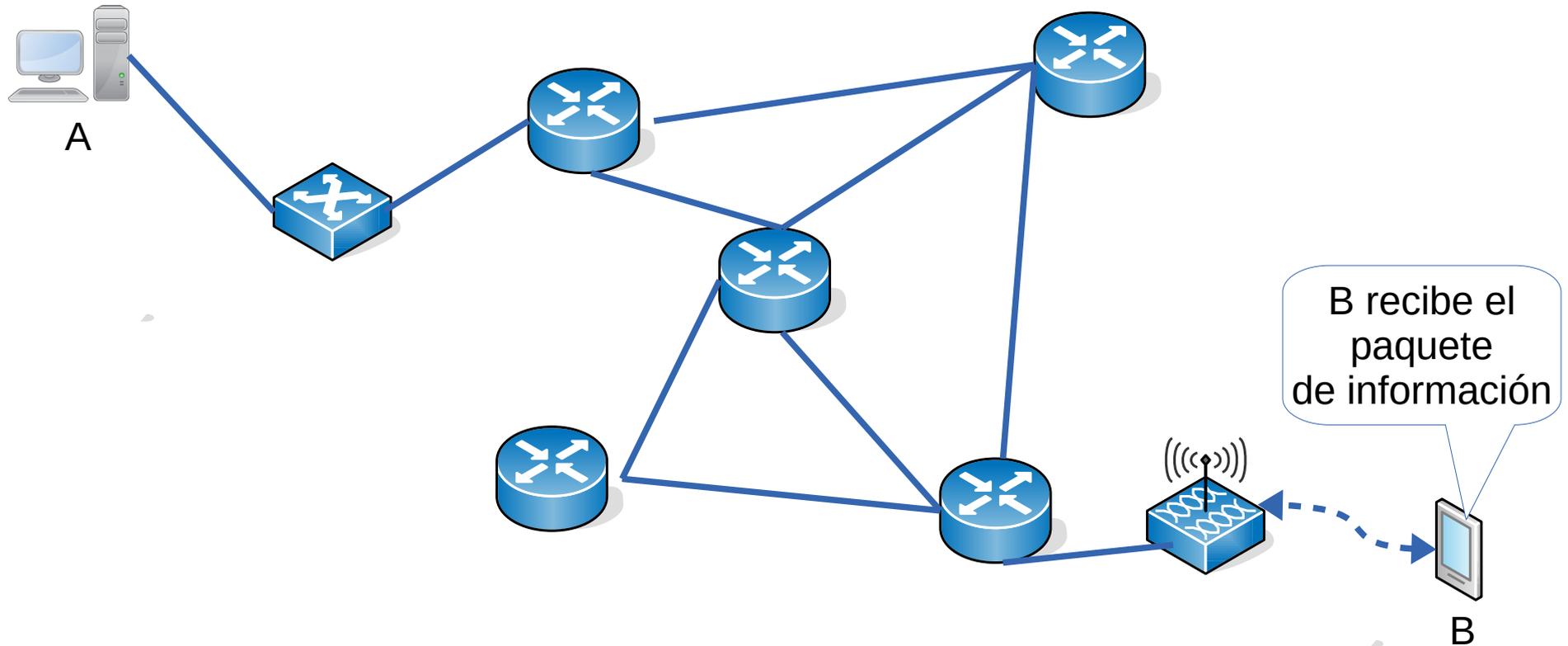
# Conmutación de paquetes



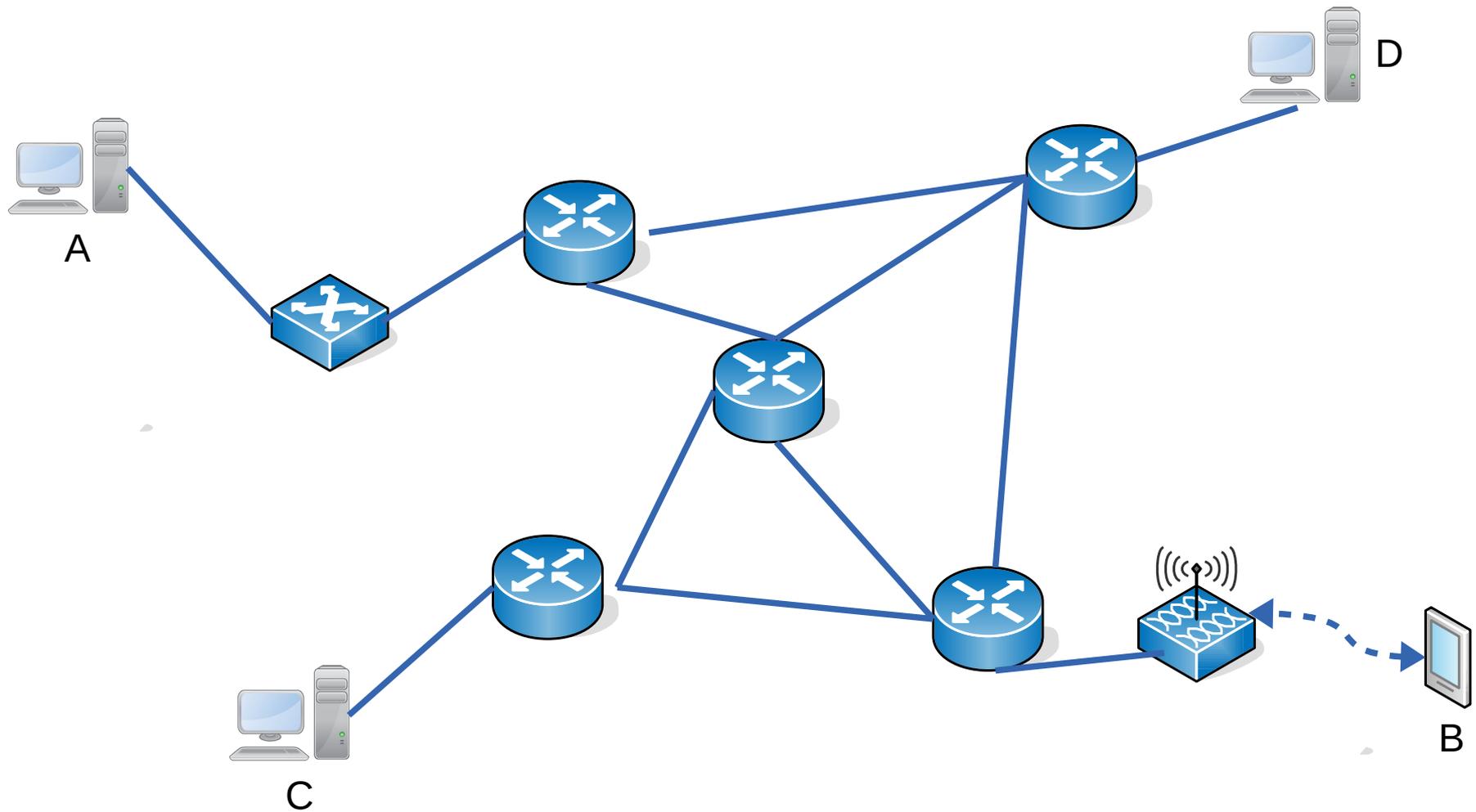
# Conmutación de paquetes



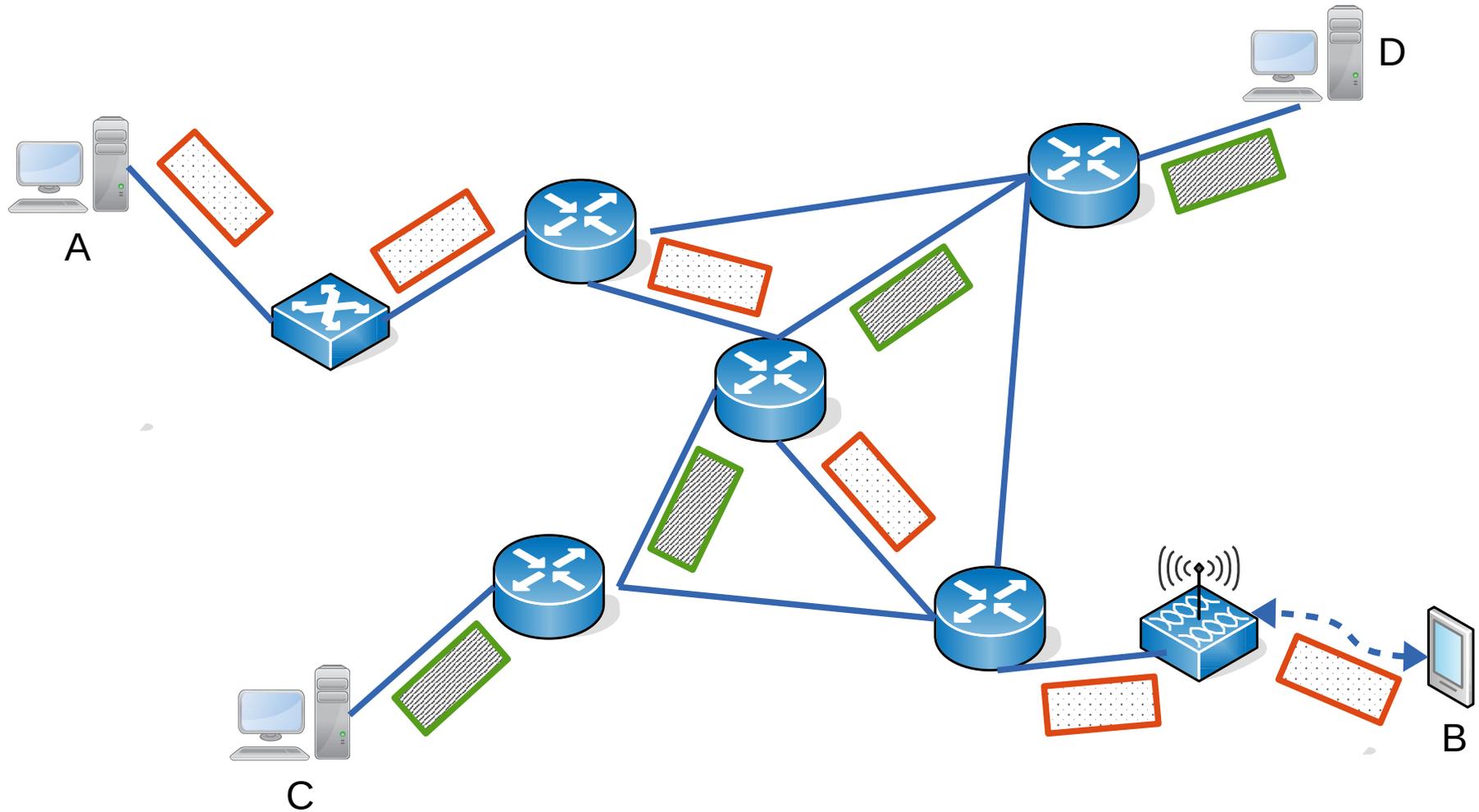
# Conmutación de paquetes



# Conmutación de paquetes



# Conmutación de paquetes



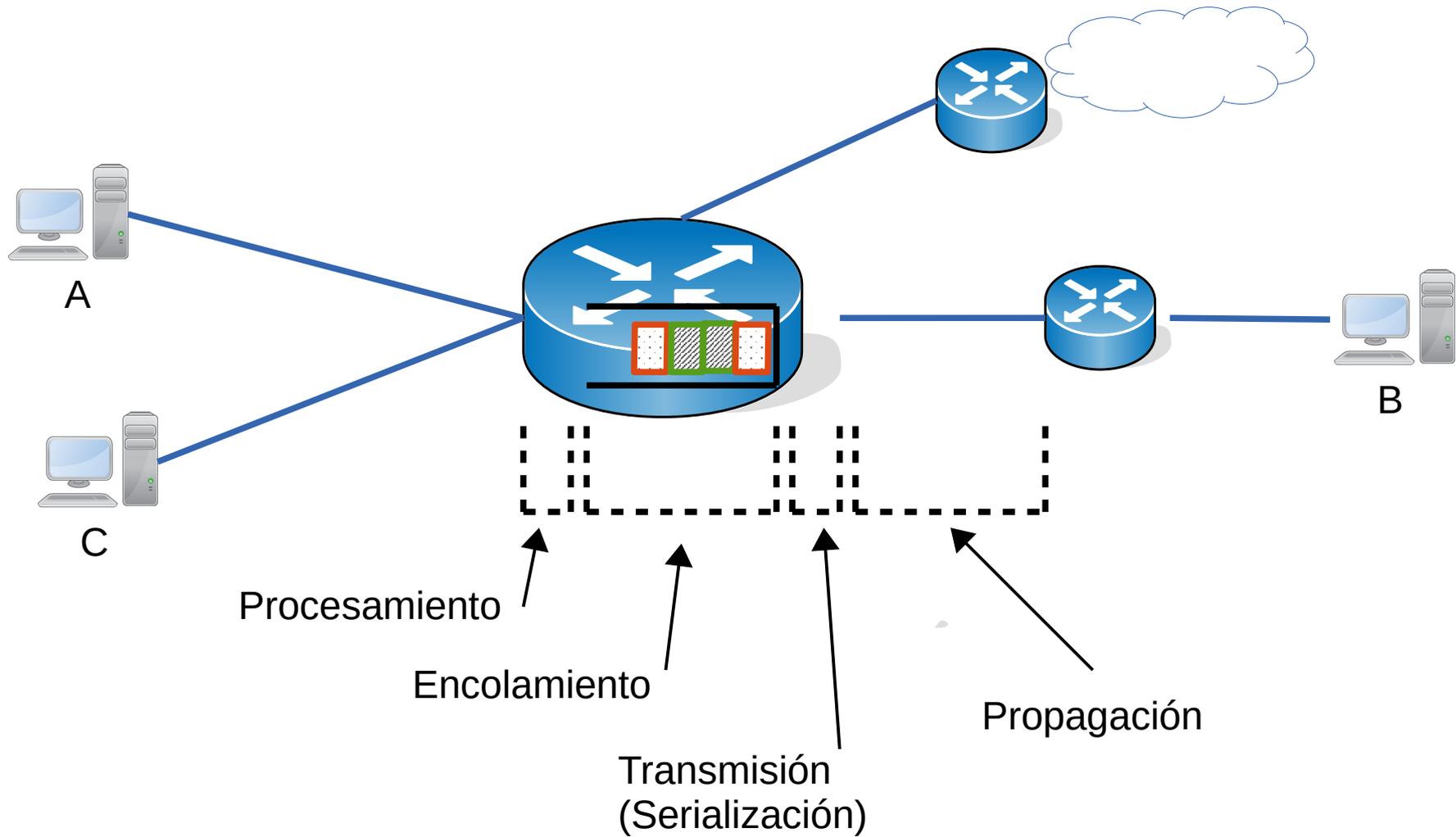
# Retardos, pérdidas y throughput

- Idealmente querríamos transmitir la información desde un extremo al otro de la red de forma instantánea y sin pérdidas
- En la realidad, la red:
  - Introduce **retardos** en el camino entre el origen y el destino
  - Introduce o puede introducir **pérdida** de datos
  - Limita el **throughput** (cantidad de bytes por segundo que puedo transmitir entre los dispositivos de los extremos de la red)
- Por algunas de las siguientes causas:
  - Capacidad de los equipos y enlaces
  - Leyes de la física
  - Tiempos de procesamiento en los dispositivos
  - Eventual sobre carga o congestión
  - Errores

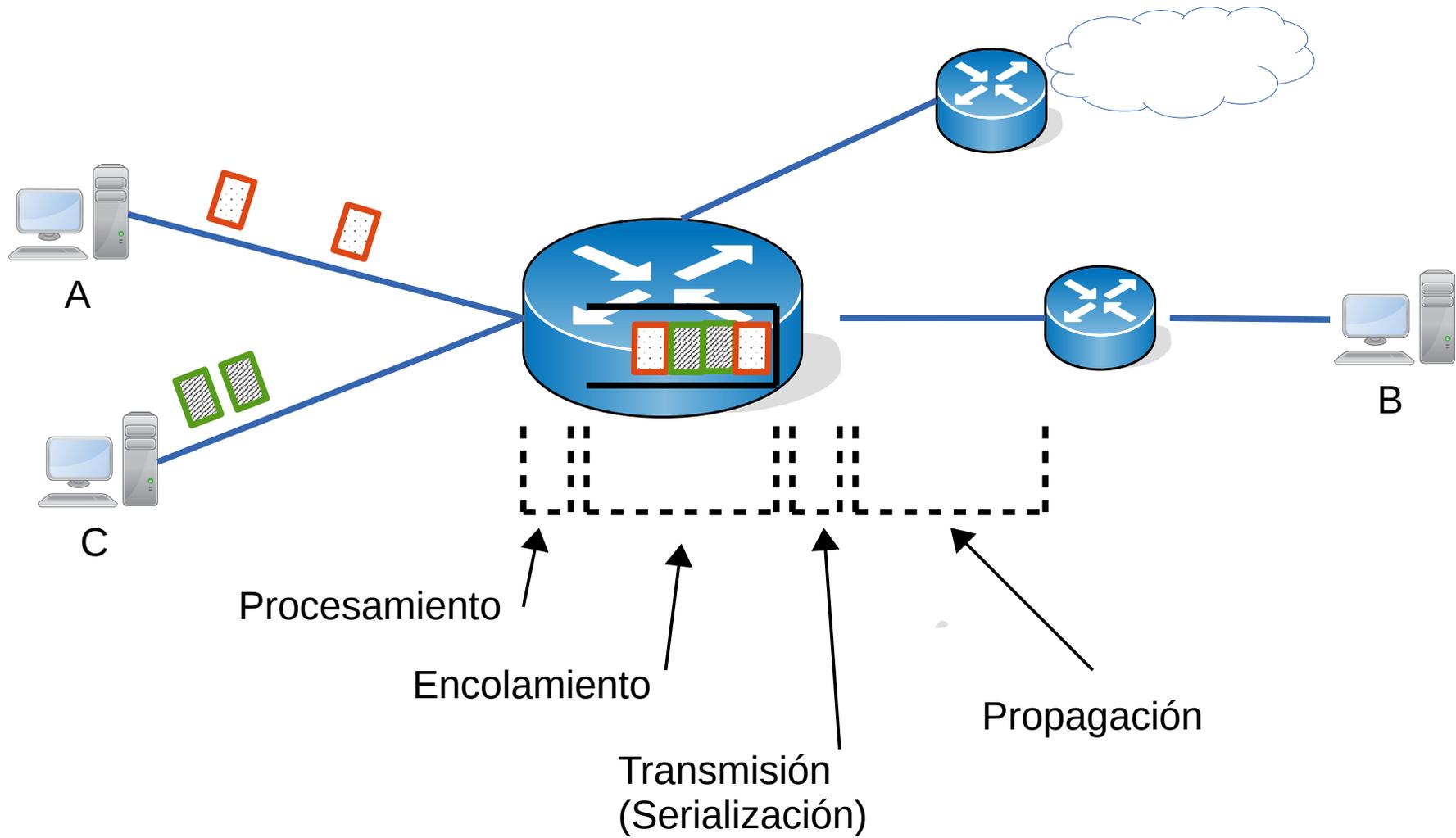
# Retardo en redes de paquetes

- Un paquete que sale del nodo origen, atraviesa varios nodos intermedios y enlaces, y finalmente llega al nodo destino, se ve afectado por varios retardos
- Tipos de retardo:
  - **Procesamiento** en cada nodo
  - Atención en cola de espera en cada nodo (**encolamiento**)
  - Transmisión o **Serialización** en cada enlace
  - **Propagación** en cada enlace
- Entre origen y destino habrá un retardo total que acumulará todos estos tipos de retardo para cada nodo o enlace que recorra
- Este retardo total afecta sensiblemente los servicios que se brindan sobre la red (a algunos los afecta más que a otros)
  - Por ejemplo para descargas de archivos no es muy relevante, pero para navegación o para servicios de telefonía, puede ser crítico

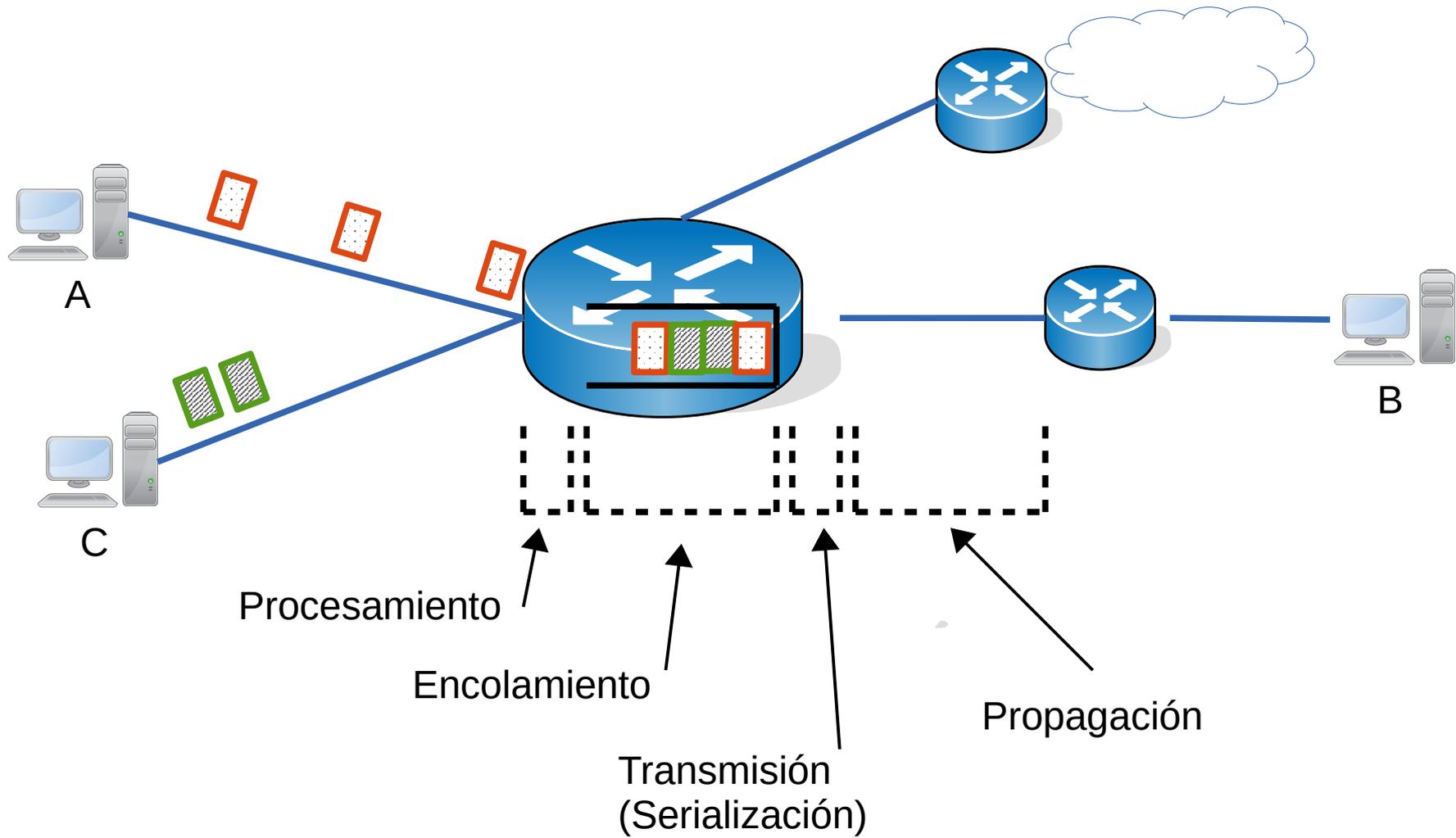
# Tipos de retardo



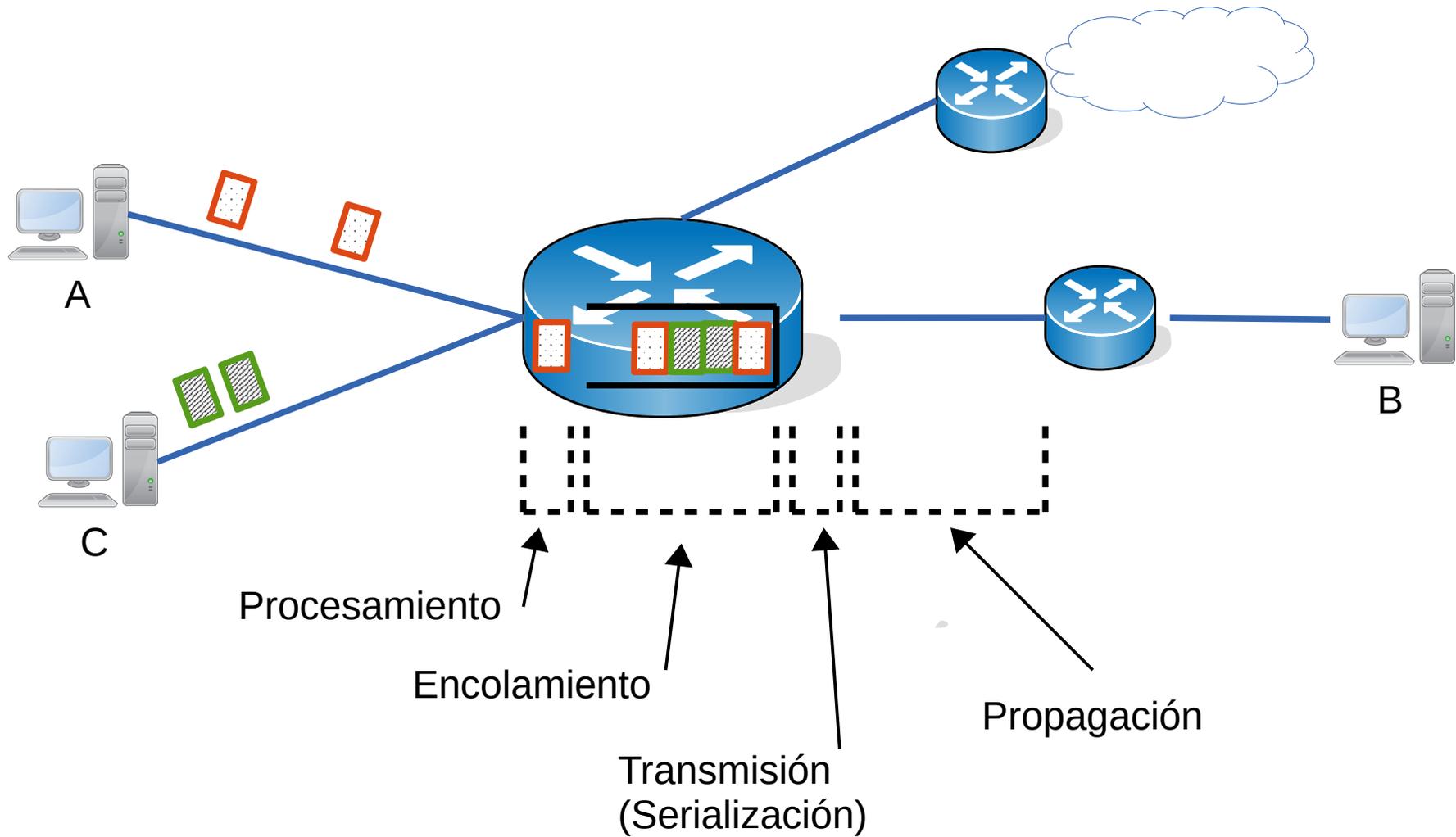
# Tipos de retardo



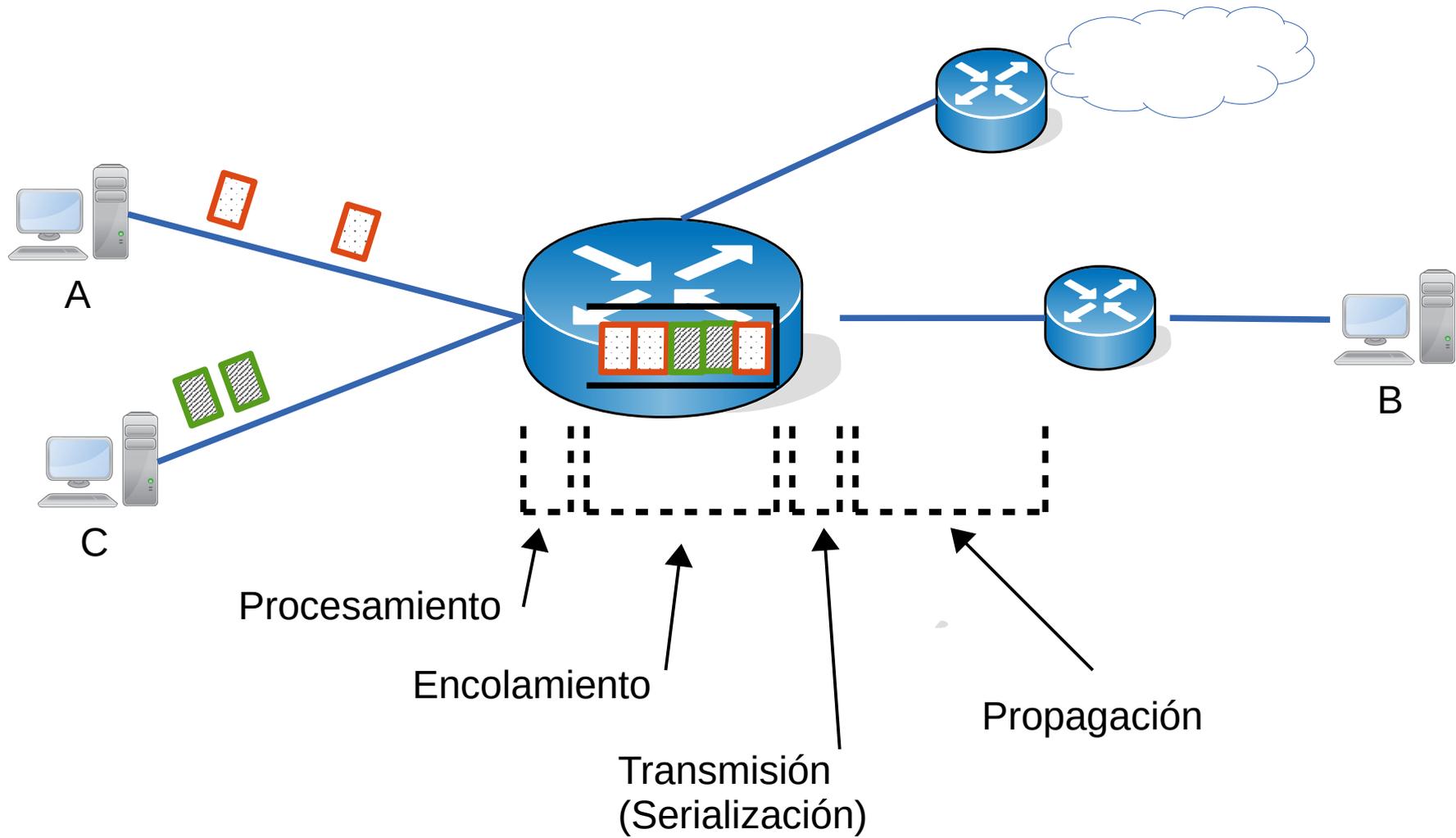
# Tipos de retardo



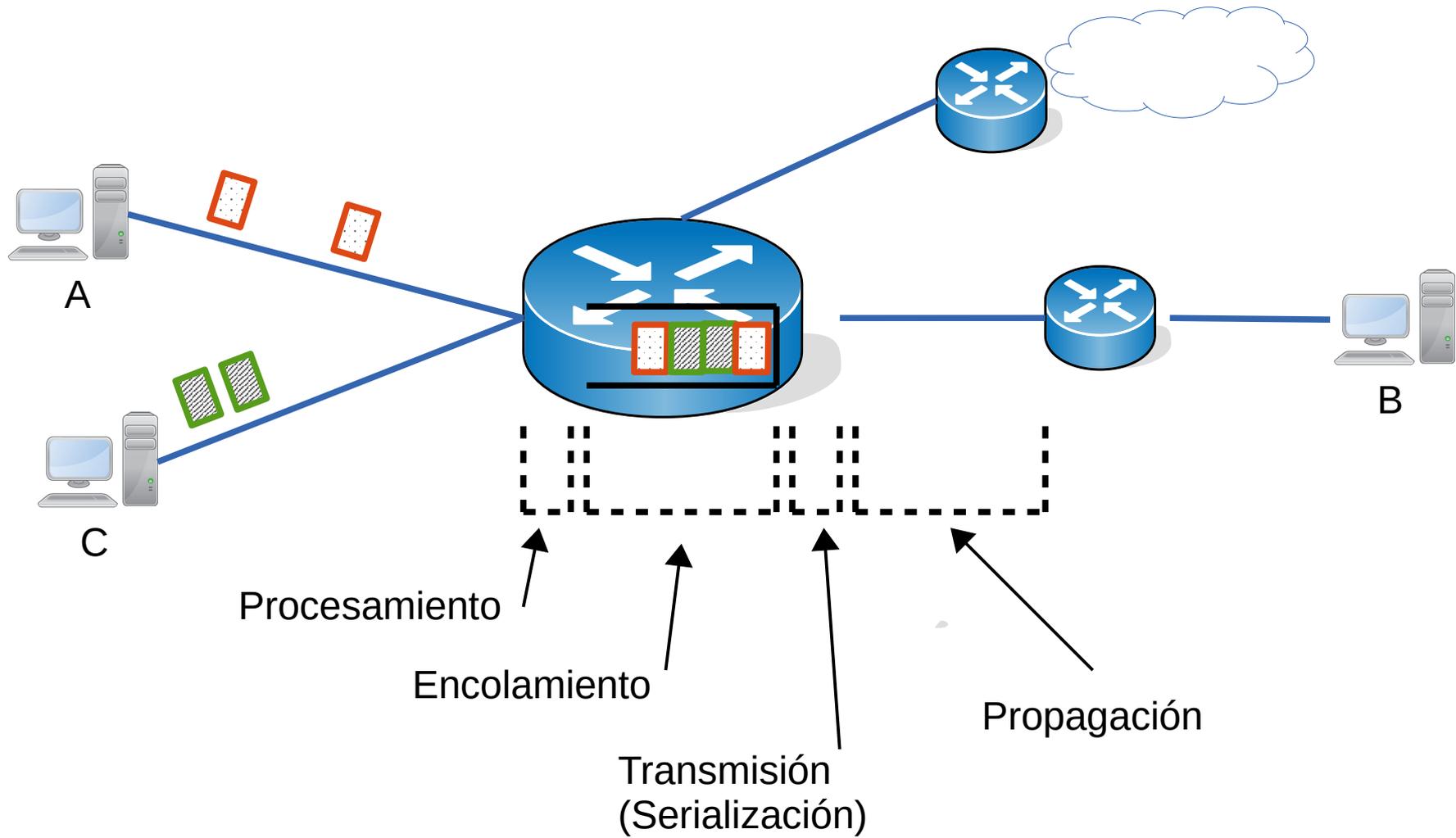
# Tipos de retardo



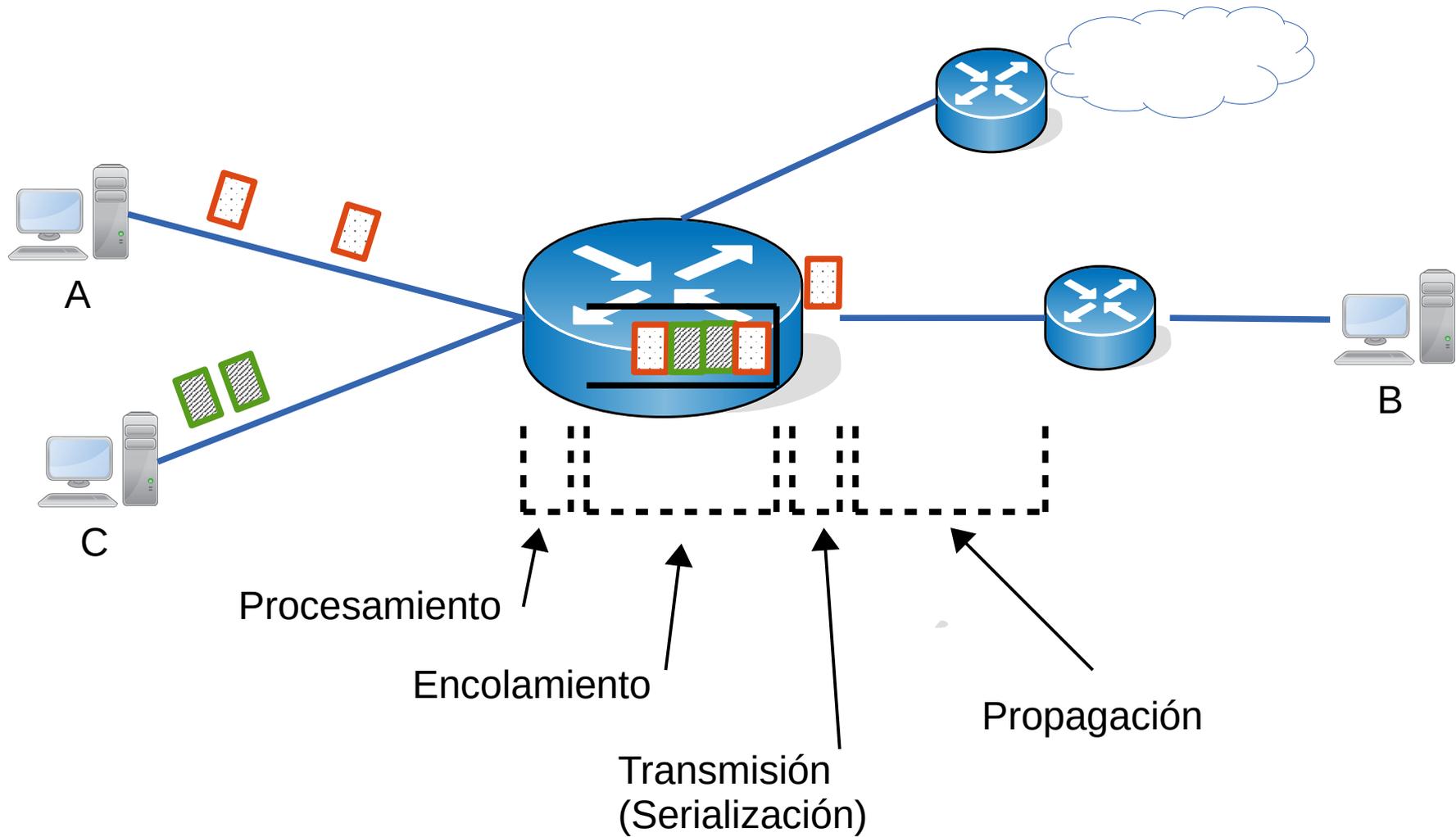
# Tipos de retardo



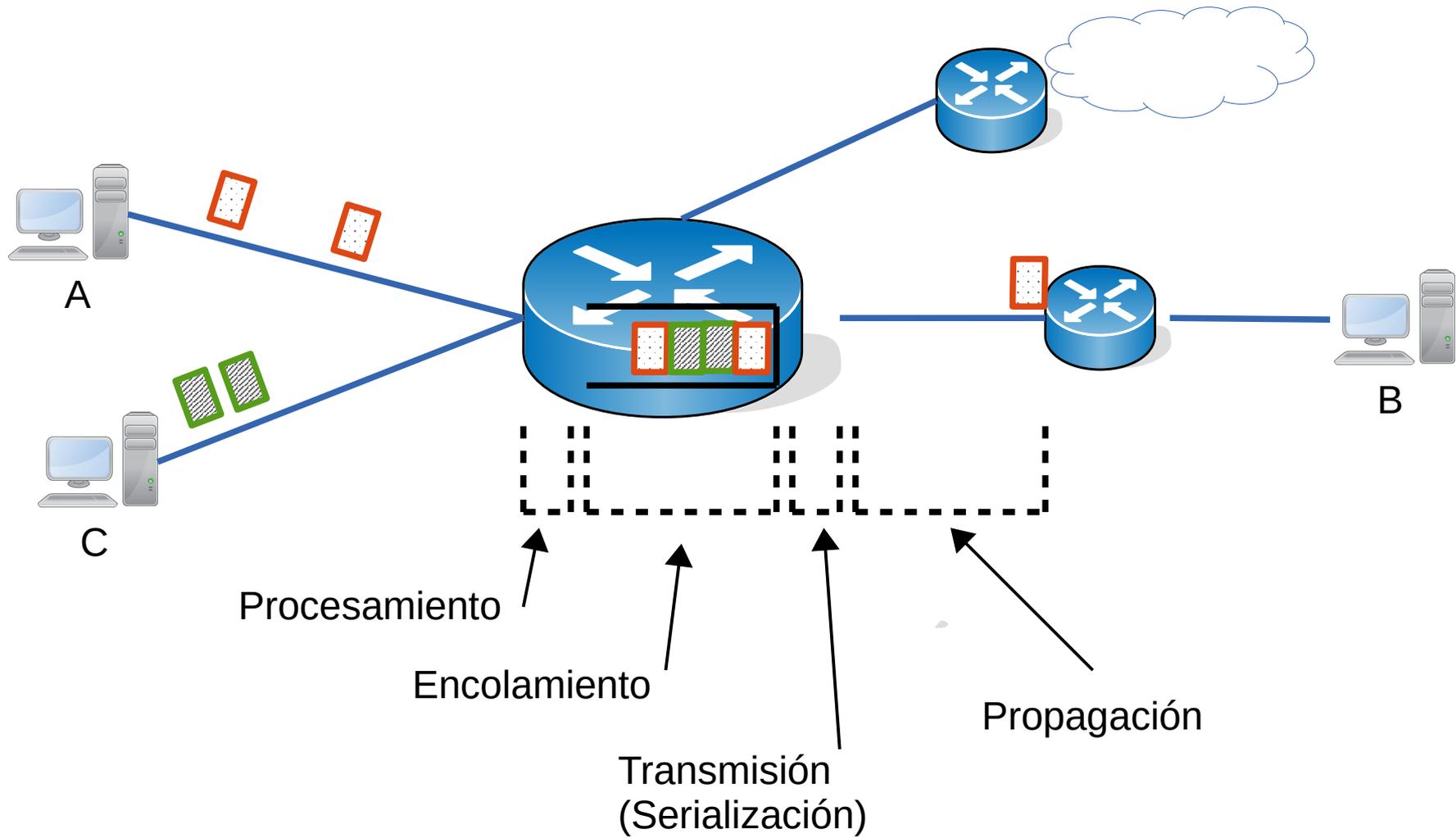
# Tipos de retardo



# Tipos de retardo



# Tipos de retardo



# Tipos de retardo

- Retardo de **procesamiento**
  - Examinar el paquete, verificar posibles errores
  - Decidir hacia dónde debe encaminarse el paquete (próximo salto en el camino al destino)
  - Orden de microsegundos
- Retardo de **encolamiento**
  - Espera para ser transmitidos por un enlace
  - Depende de otros paquetes que hayan llegado antes y deban salir por el mismo enlace
  - Varía por tanto con la carga de la red (nivel de congestión)
  - Depende del tamaño de los paquetes
  - Orden de micro a milisegundos

# Tipos de retardo

- Retardo de transmisión o **serialización**
  - Tiempo que lleva transmitir todos los bits de un paquete por el enlace
  - Si el enlace permite transmitir R bits/segundo, transmitir un paquete de L bits tardará L/R segundos
  - Orden de micro a milisegundos
- Retardo de **propagación**
  - Tiempo que le lleva a un bit atravesar el enlace hasta llegar al destino
  - Depende de la velocidad de propagación de la señal en el medio
    - en general un poco menos que la velocidad de propagación de la luz en el vacío: en el entorno de  $2 \times 10^8$  a  $3 \times 10^8$  metros por segundo
  - El retardo depende de la velocidad de propagación (V) y del largo del medio físico (D) :  $D/V$
  - Puede ser muy variable, desde despreciable (en un datacenter) a cientos de milisegundos (en enlaces satelitales)

# Retardo de encolamiento y pérdida de paquetes

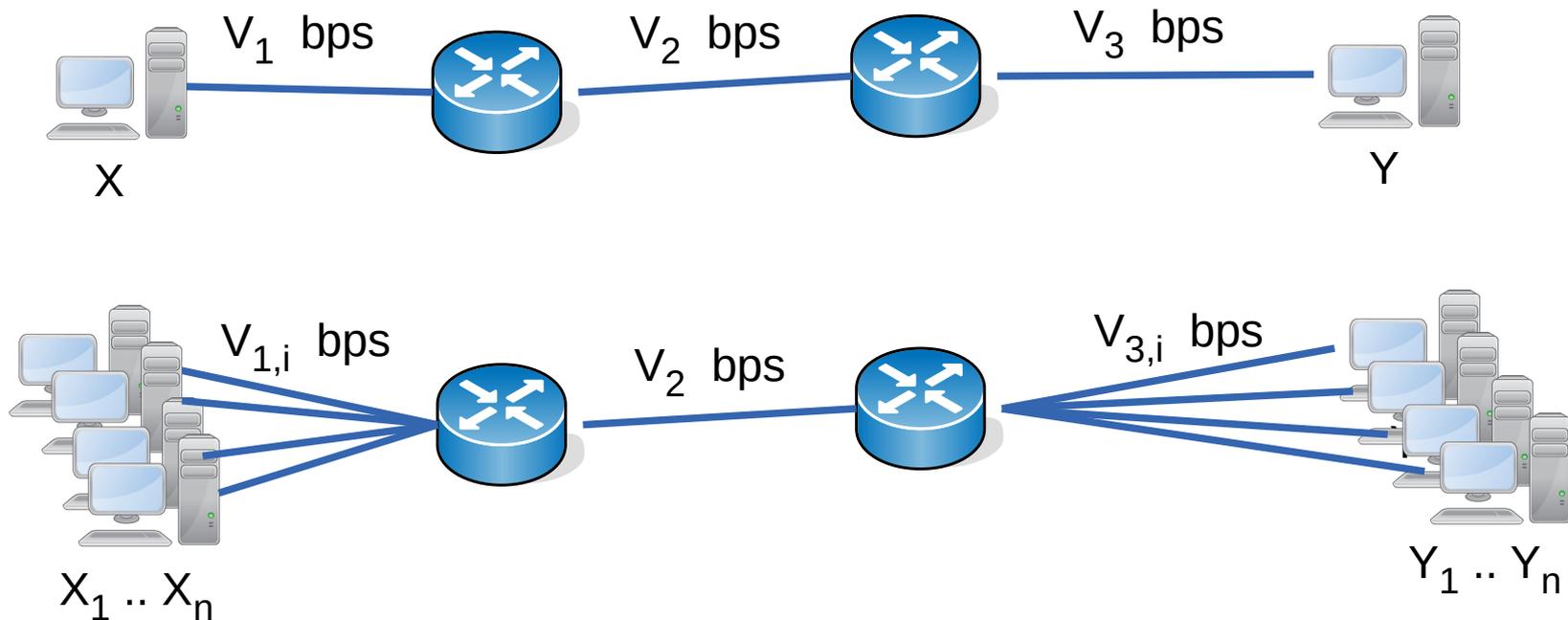
- Si los paquetes llegan a una cola a mayor tasa de la que se transmiten, entonces el tamaño de la cola tendería a crecer
- El retardo por encolamiento sería creciente
- Como en la realidad la capacidad de almacenamiento de paquetes en una cola está acotada, podría haber pérdida de los paquetes que llegan cuando la cola está llena
- Estas **pérdidas** provocarían errores en la transmisión de origen a destino y eventualmente requerirán re-transmisión de los datos perdidos

# Throughput

- Es otro parámetro relevante para caracterizar la performance de la red además de los retardos y las pérdidas
- Si la transferencia de  $B$  bits (con  $B$  grande) desde el nodo  $X$  al  $Y$  de la red, requiere  $T$  segundos, decimos que el **throughput** promedio entre  $X$  e  $Y$  es:

$B/T$  bits/seg

- El throughput depende de la velocidad de cada enlace, pero también de cómo el tráfico circula por la red y los “cuellos de botella”
- Ejemplos:



# Control de flujo

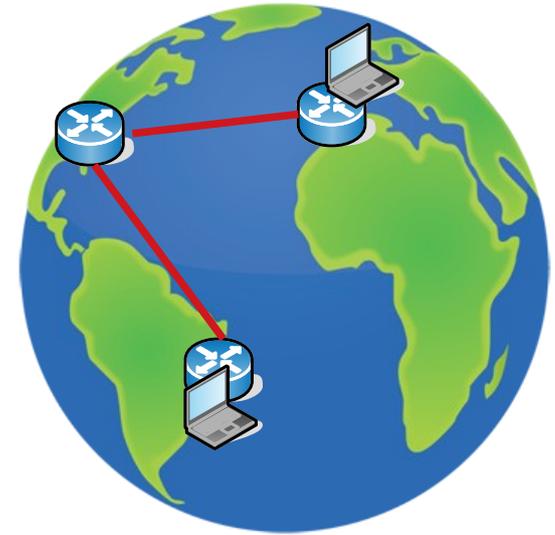
- El **control de flujo** trata de evitar que un transmisor “rápido” sature a un receptor “lento” o sobrecargado
  - El receptor puede tener menos capacidad (de procesamiento, de memoria) que el transmisor
  - El receptor puede estar atendiendo otras comunicaciones
- Generalmente se basa en limitar la cantidad de datos que puede enviar el transmisor, **ajustándola a la capacidad del receptor**
- Si no se controla el flujo, lo que puede pasar es que el receptor no pueda procesar todos los datos que el transmisor envía y descarte algunos de ellos
- En caso que la comunicación sea confiable, estos datos seguramente serán retransmitidos y por tanto se ocupará innecesariamente el canal de comunicación repitiendo la misma información
- Desde el transmisor puede verse como pérdidas en la información

# Control de congestión

- La **congestión** se presenta cuando en una zona de la red se requiere procesar más paquetes de los que los nodos pueden procesar
- Puede deberse a:
  - Que circunstancialmente hay muchos transmisores enviando datos a altas tasas
  - Que la capacidad de algunos equipos está sub-dimensionada
- Puede manifestarse como:
  - Aumento del retardo en el camino al destino
    - Los paquetes se encolan en algún nodo intermedio mientras el nodo no puede procesarlos
  - Pérdida de paquetes
    - En caso que no haya lugar en las colas
- Desde el transmisor puede verse también como pérdidas en la información, pero por una causa diferente al problema de control de flujo
- Para poder controlar la congestión se requerirá alguna **realimentación sobre el estado de carga de la red**

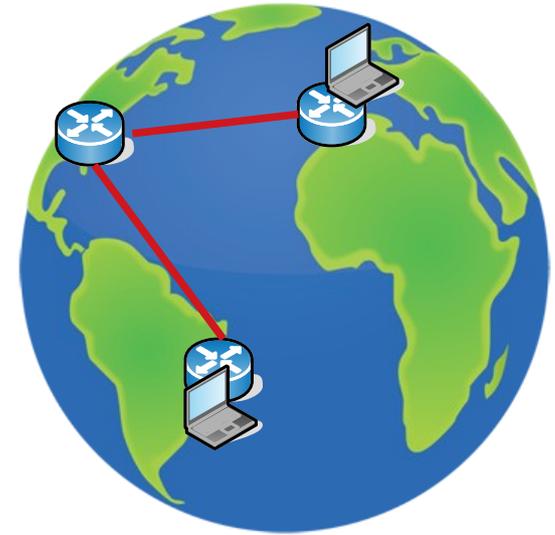
# ¿Cómo atacar la complejidad del problema?

- Se necesita **conectividad global**
- Grandes **distancias**
- Diversidad de **medios** de comunicación
- Diversidad de **tecnologías** de acceso a la red
- Diversidad de **fabricantes** de dispositivos  
(en dispositivos end-point e intermedios)
- Múltiples **proveedores** de servicio
- Diversidad de **aplicaciones** (ejemplo: varios desarrolladores de navegadores)



# ¿Cómo atacar la complejidad del problema?

- Se necesita **conectividad global**
- Grandes **distancias**
- Diversidad de **medios** de comunicación
- Diversidad de **tecnologías** de acceso a la red
- Diversidad de **fabricantes** de dispositivos  
(en dispositivos end-point e intermedios)
- Múltiples **proveedores** de servicio
- Diversidad de **aplicaciones** (ejemplo: varios desarrolladores de navegadores)

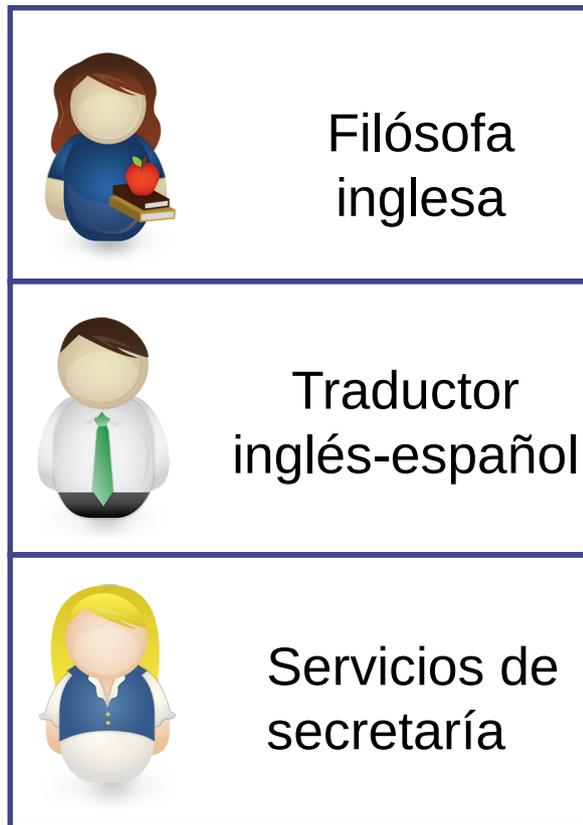


Es necesario dividir el problema para poder analizarlo

# Modelo de capas

- Idea:
  - Dividir un problema complejo en problemas más pequeños y por tanto más fáciles de abordar
- Ejemplo:
  - Intercambio de ideas entre filósofos/as
  - La filósofa A
    - sólo escribe y lee inglés
    - vive en Inglaterra
  - El filósofo B
    - sólo escribe y lee alemán
    - vive en Alemania
  - Complejidad:
    - Diferentes idiomas
    - Comunicación a distancia

# Modelo de capas: flujo de información

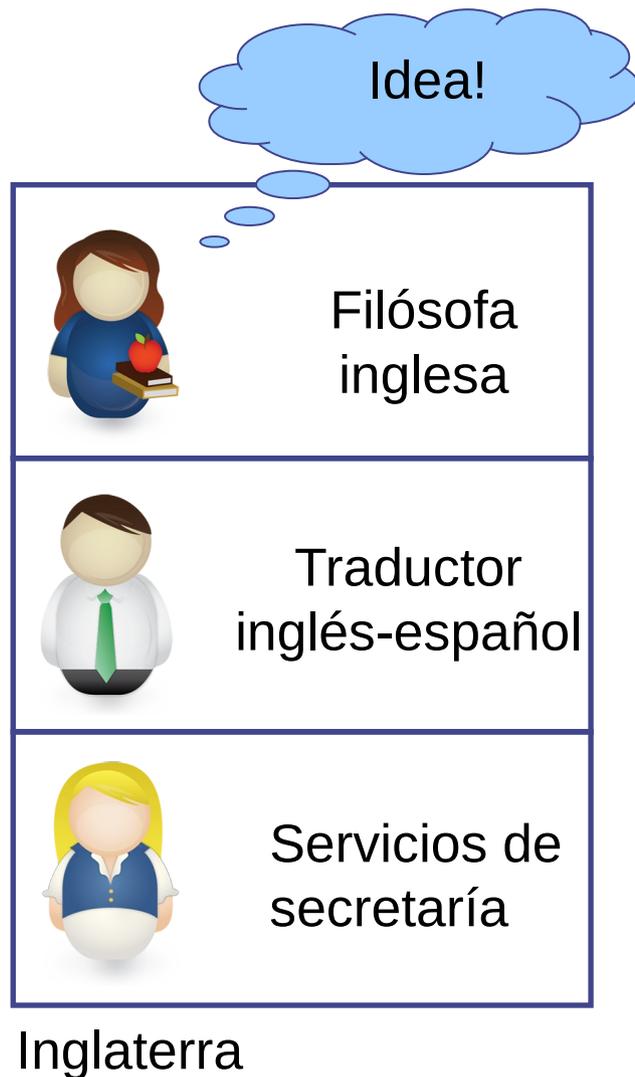


Inglaterra

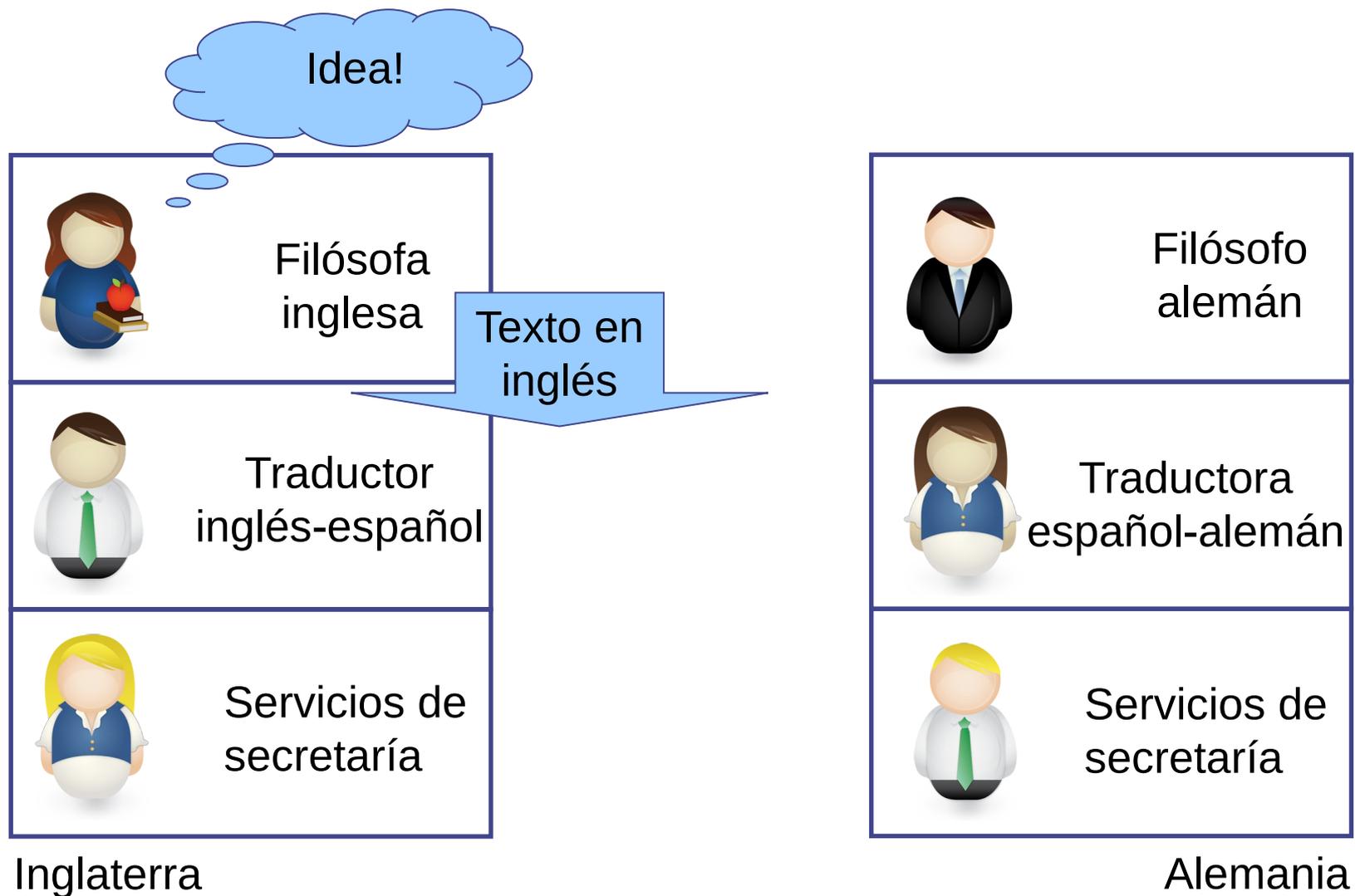


Alemania

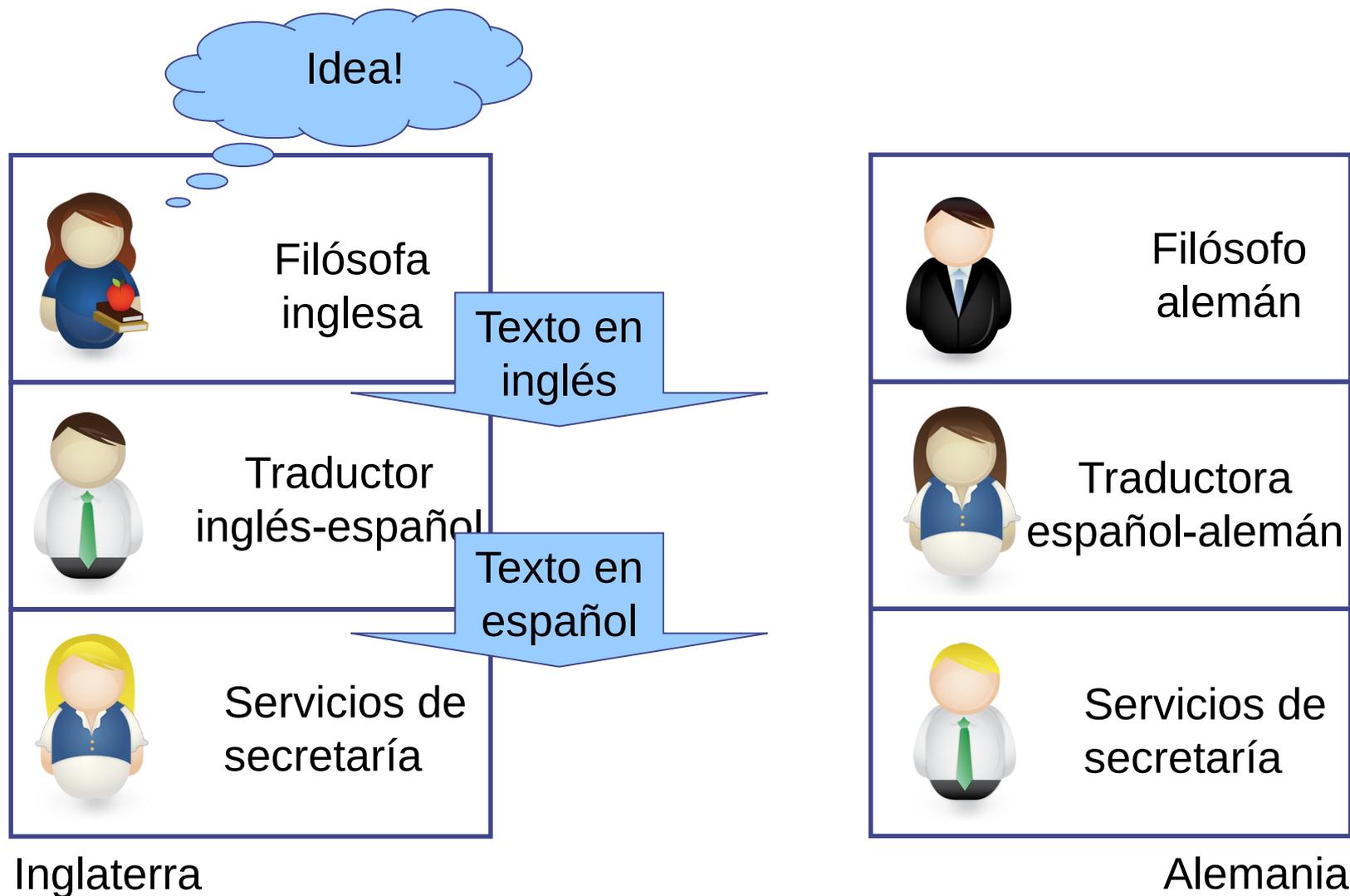
# Modelo de capas: flujo de información



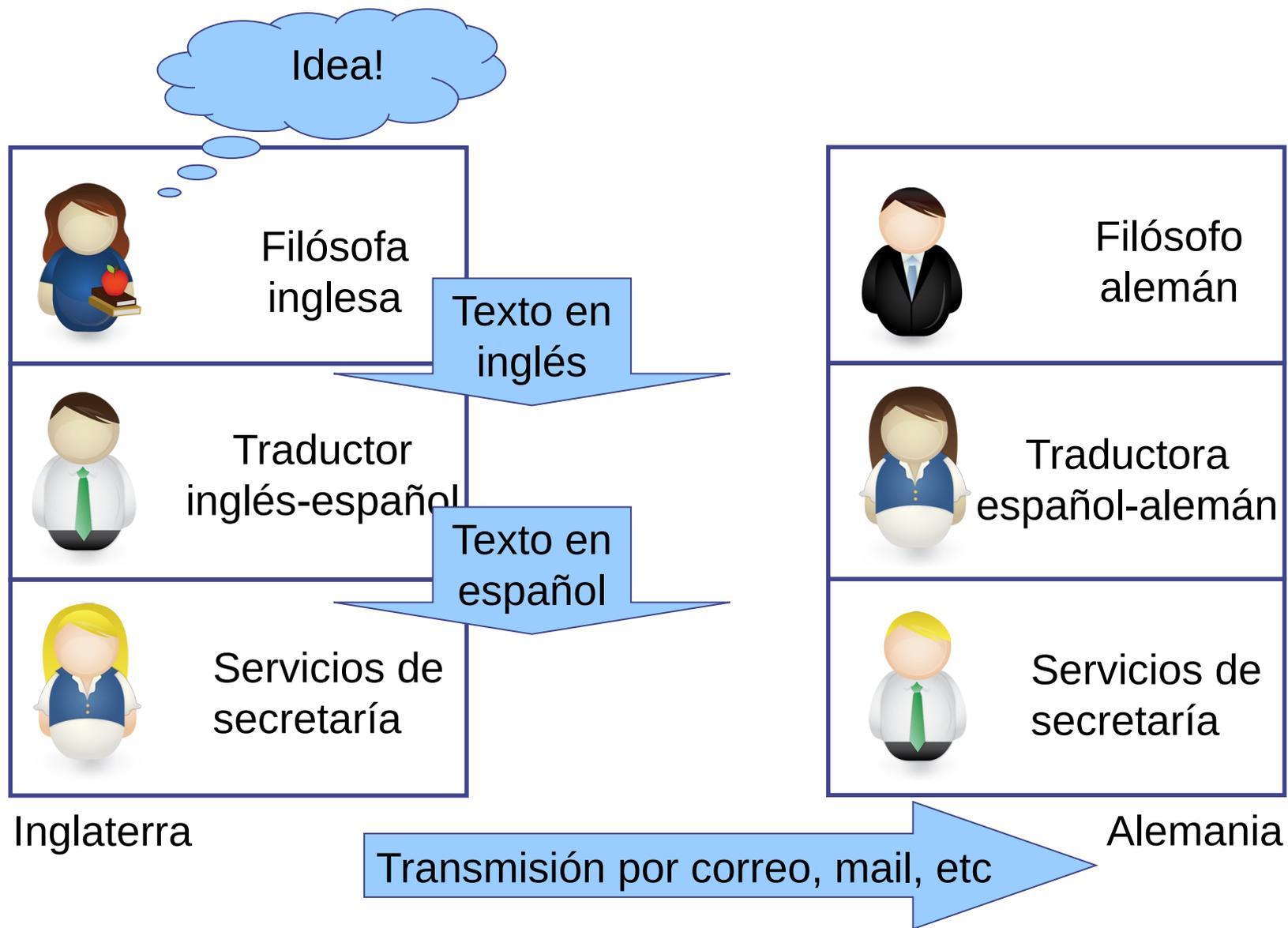
# Modelo de capas: flujo de información



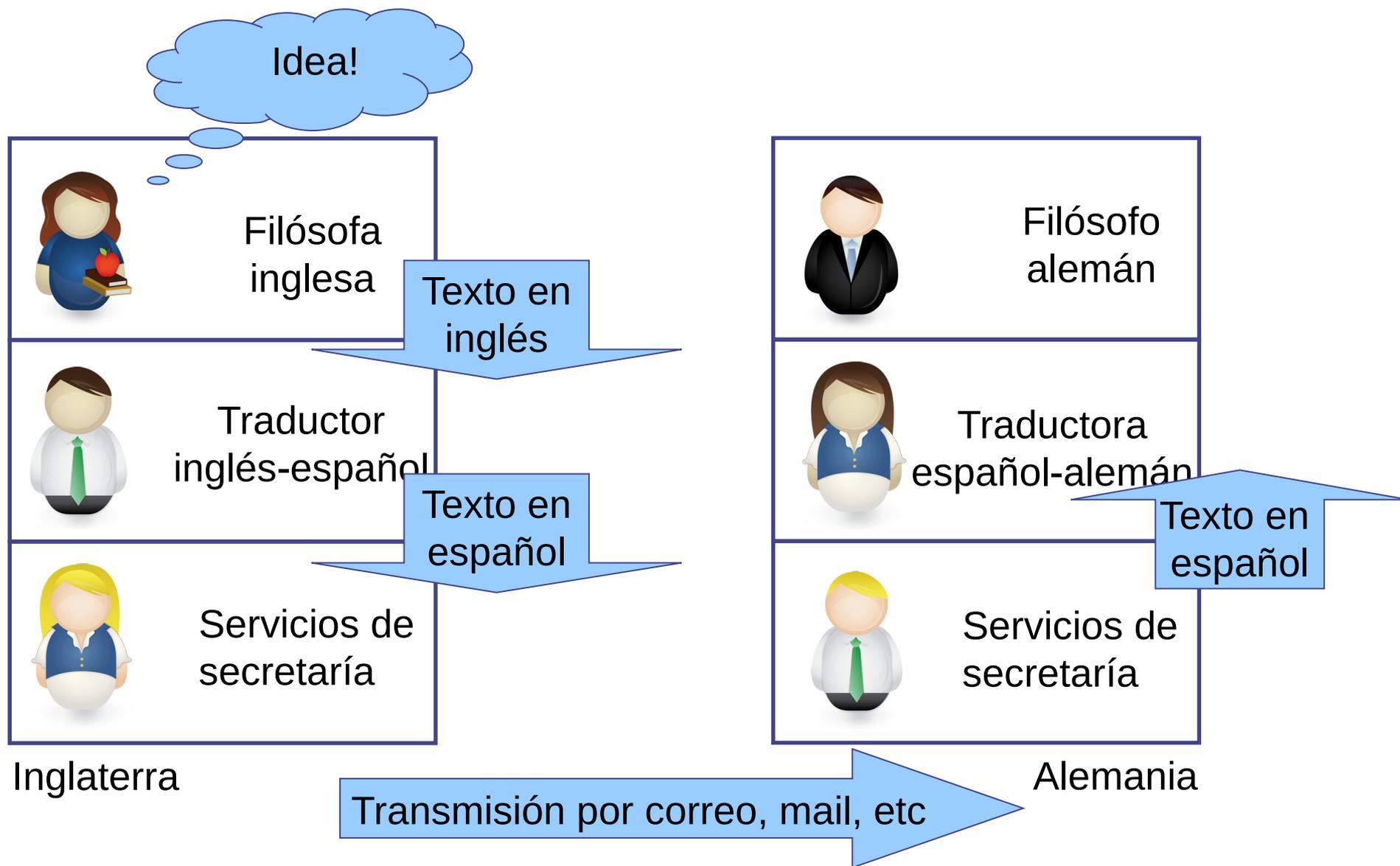
# Modelo de capas: flujo de información



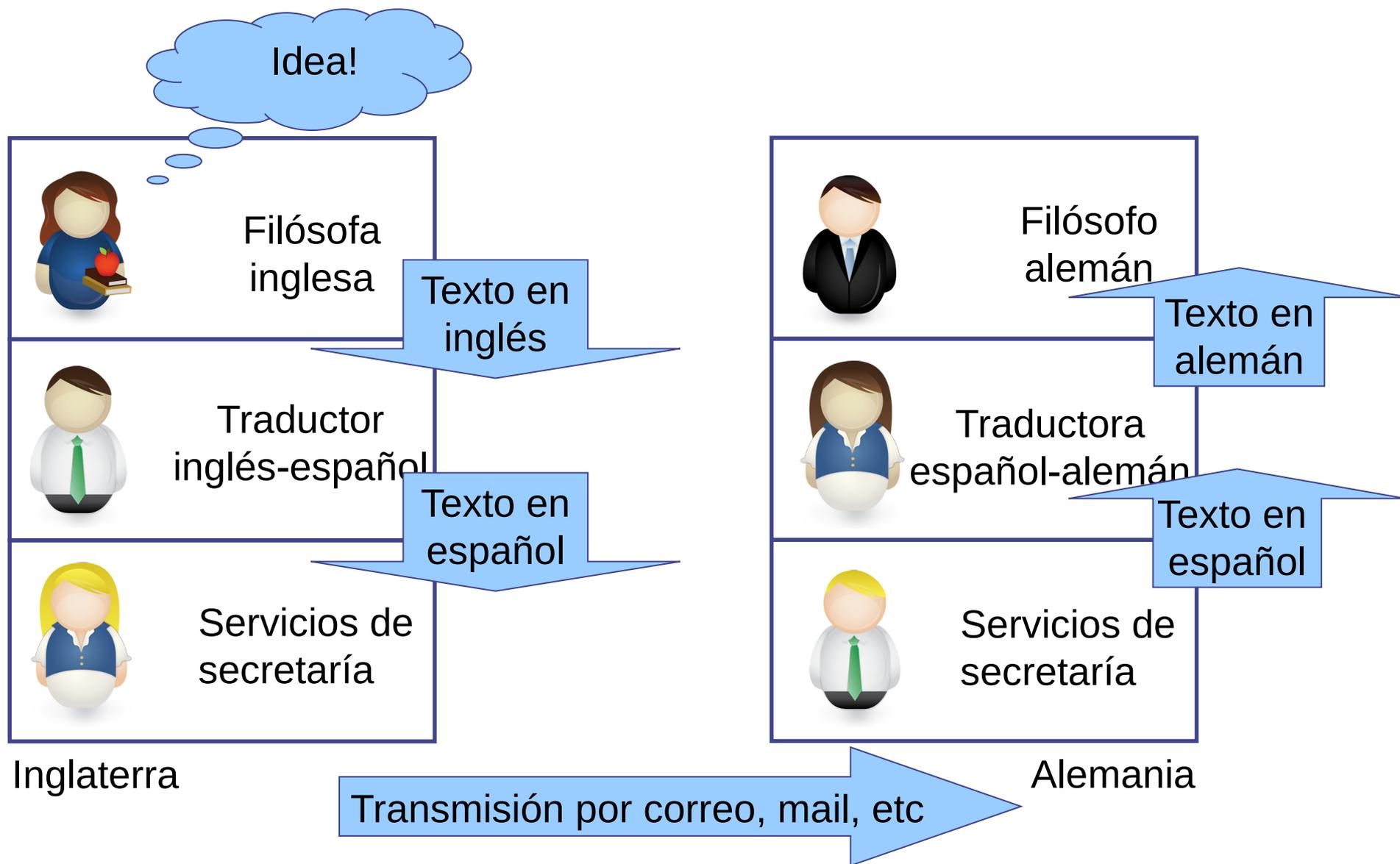
# Modelo de capas: flujo de información



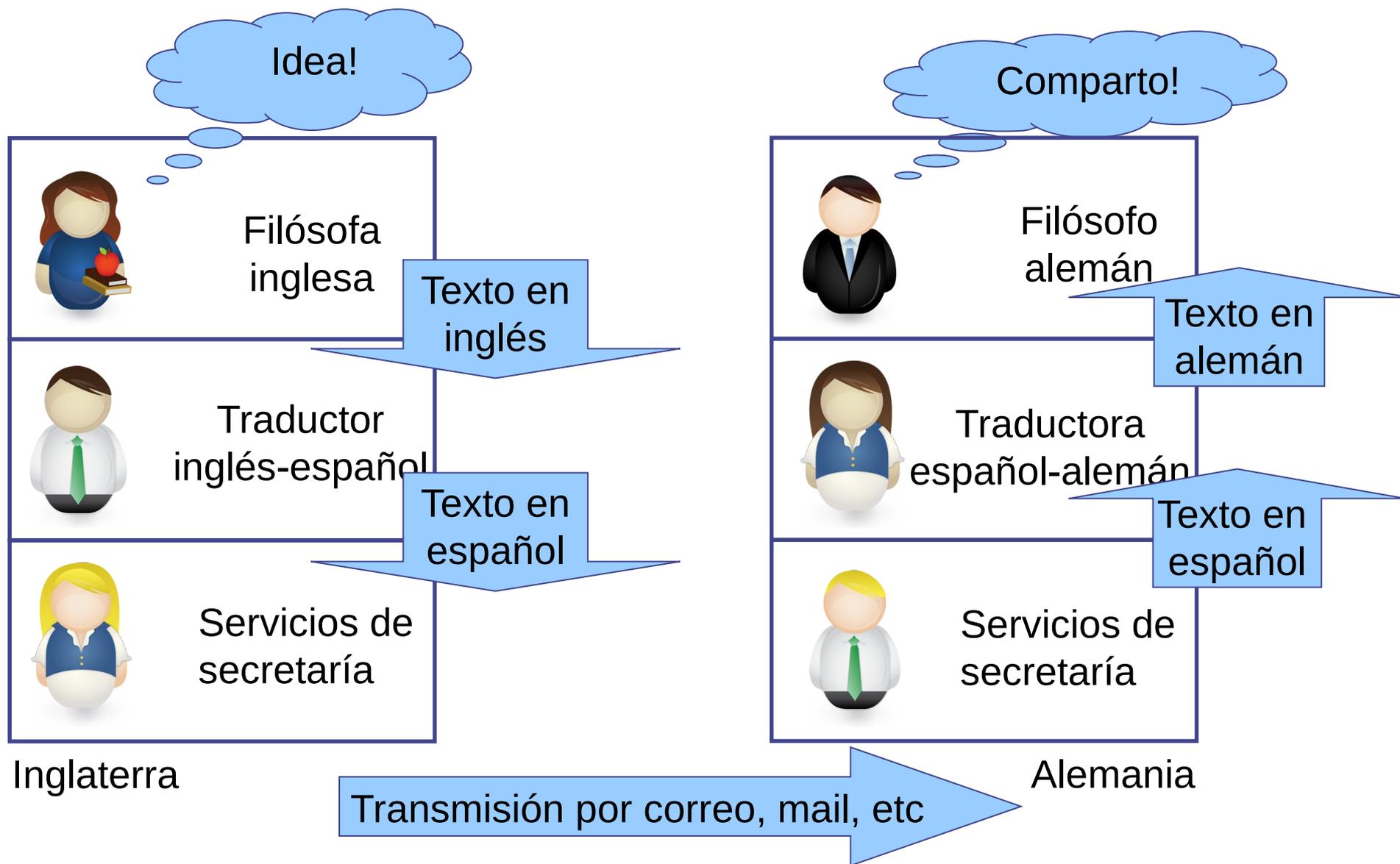
# Modelo de capas: flujo de información



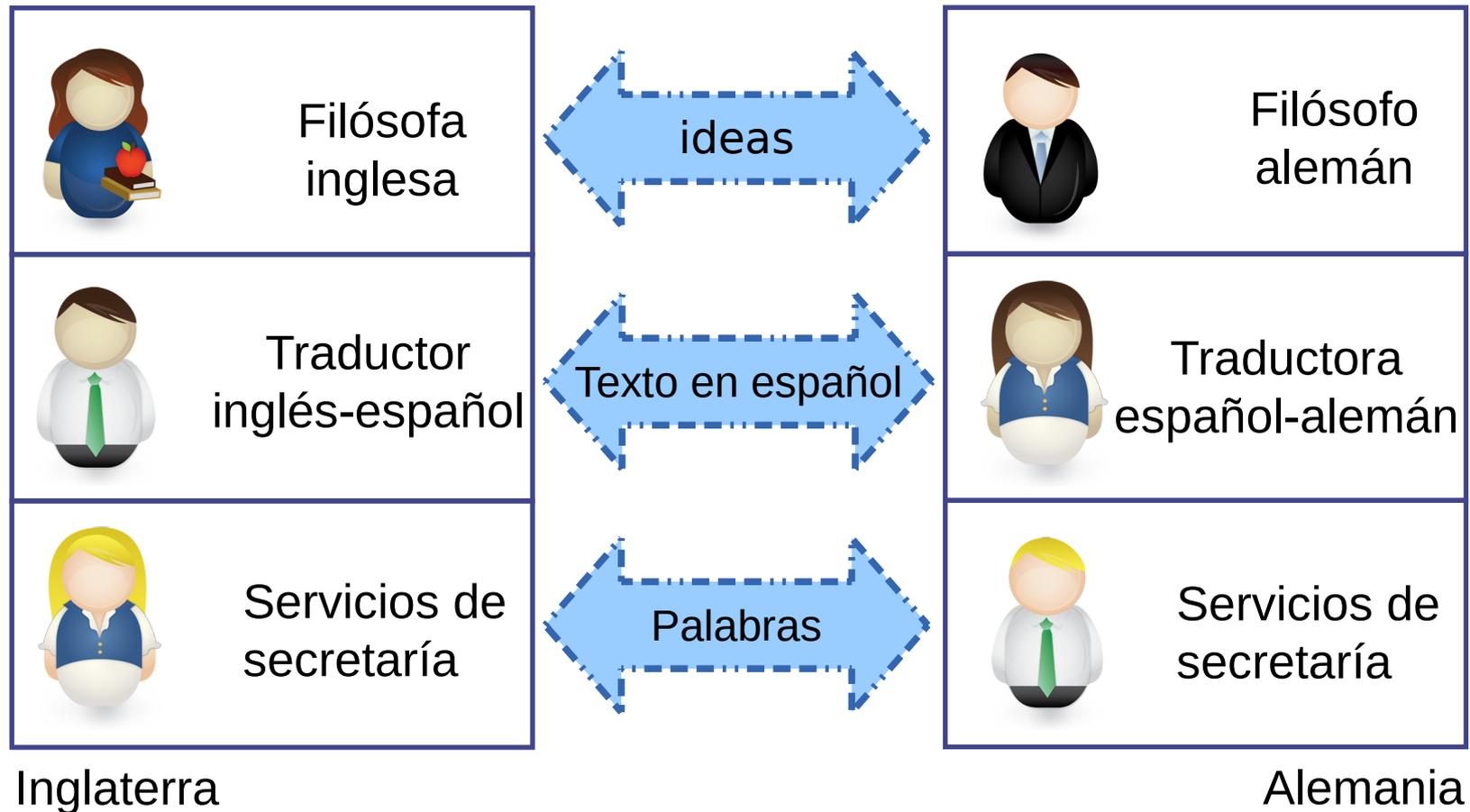
# Modelo de capas: flujo de información



# Modelo de capas: flujo de información



# Modelo de capas: diálogos virtuales



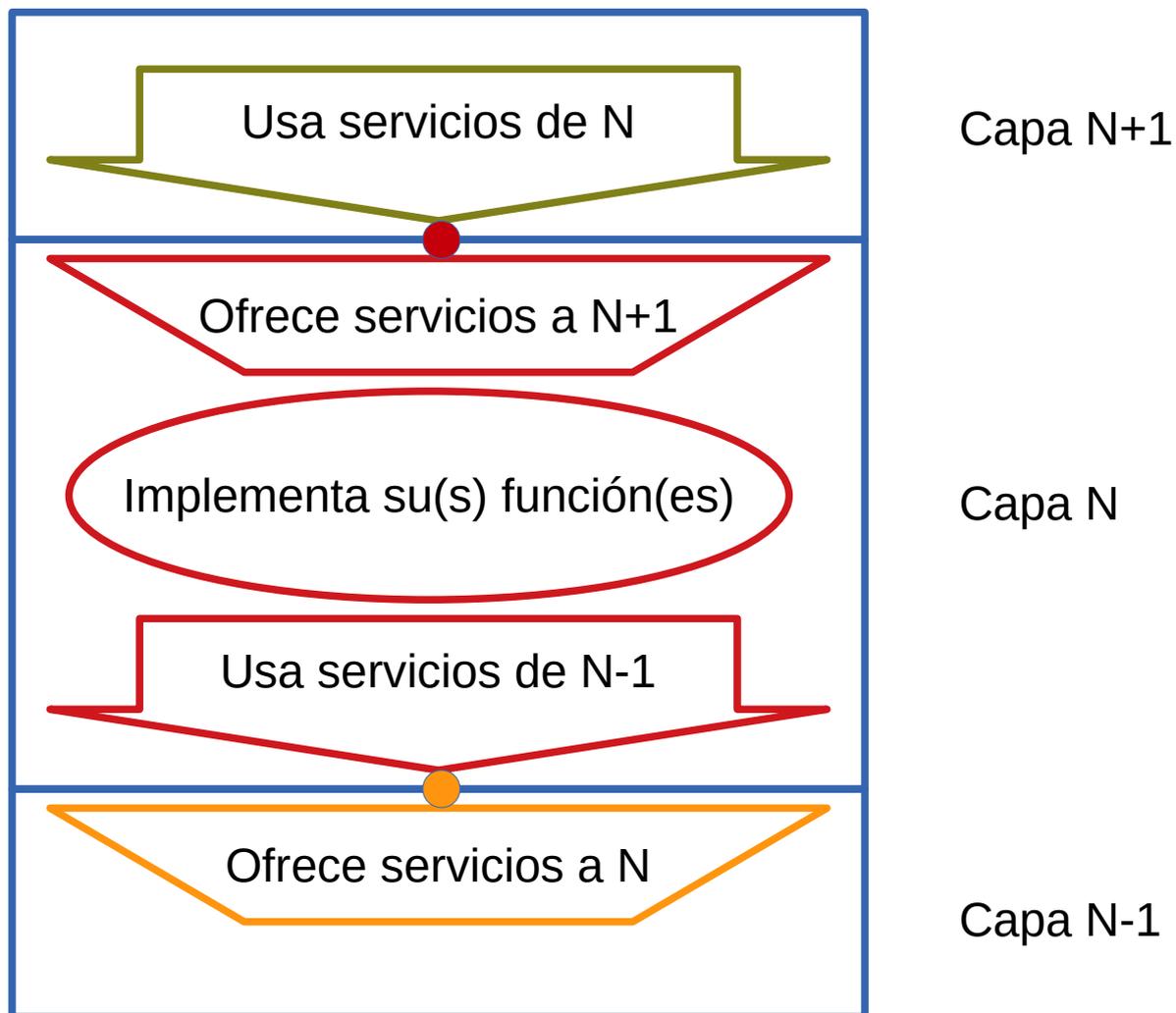
# Modelo de capas

- Ventajas:
  - Problemas más pequeños para abordar
  - Vista del problema en **diferentes niveles de abstracción**
  - Se ofrecen **servicios independientes de la implementación**
  - Permite avance (cambio) de las tecnologías
- Desventajas:
  - Puede ser ineficiente o sub óptimo
  - Pueden repetirse funciones

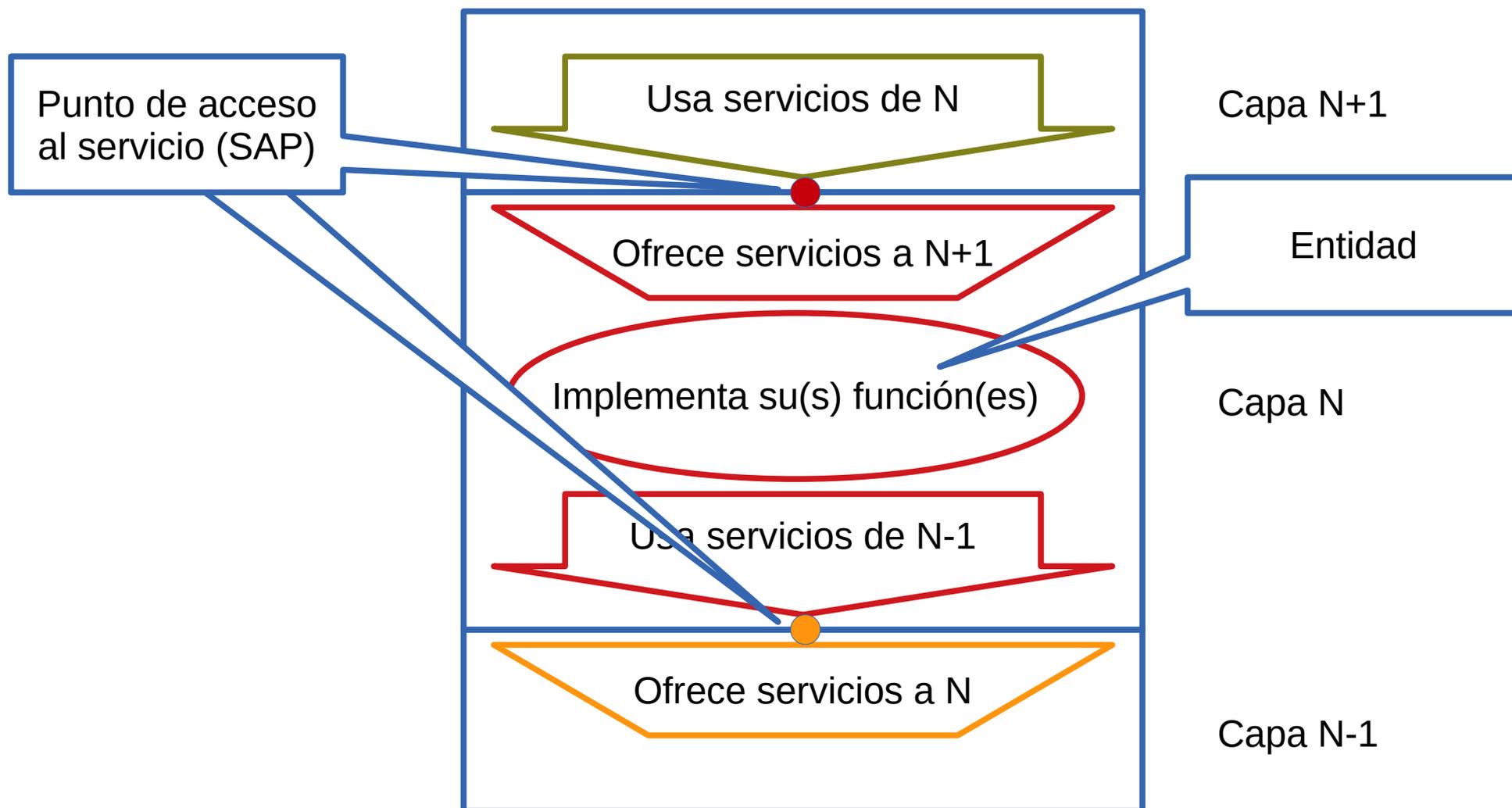
# Modelo de capas para comunicación de datos

- Cada capa debe cumplir una o más **funciones** bien definidas
- Las unidades de hardware, software o combinadas que implementan las funciones de cada capa se llaman “**entidades**”
  - Las entidades más cercanas a los medios físicos se implementan total o parcialmente en hardware, mientras que las de las capas superiores se implementan en software
- Las funciones de una capa se ofrecen como “**servicios**” a las capas superiores
- Los servicios se ponen a disposición mediante funciones o métodos, también llamados “**primitivas de servicio**”
- Los servicios se acceden en los llamados “**puntos de acceso al servicio**” o SAP (service access point)
- Para implementar de las funciones de una capa se utilizan los servicios brindados por la capa inferior

# Comunicación entre capas



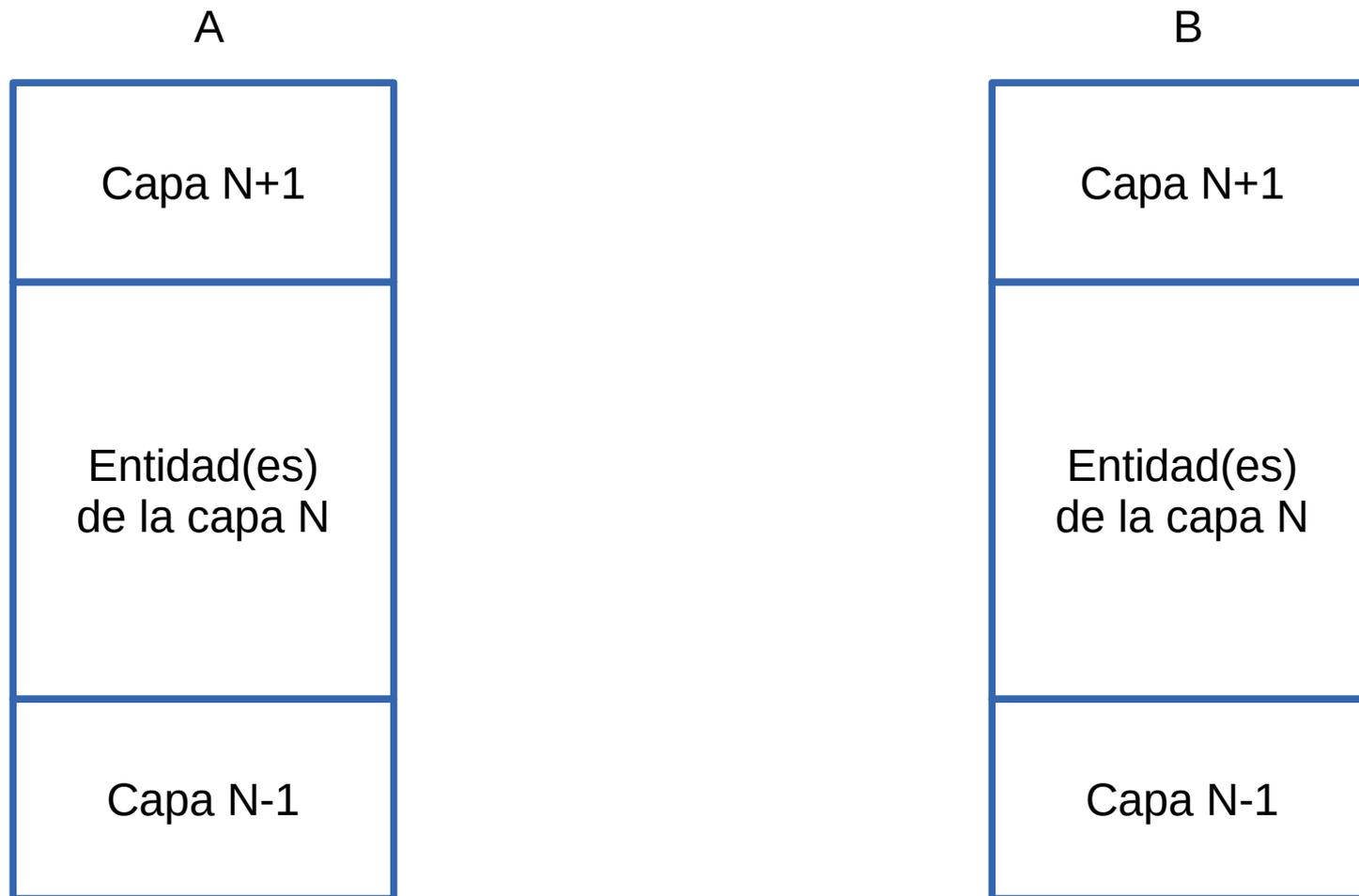
# Comunicación entre capas



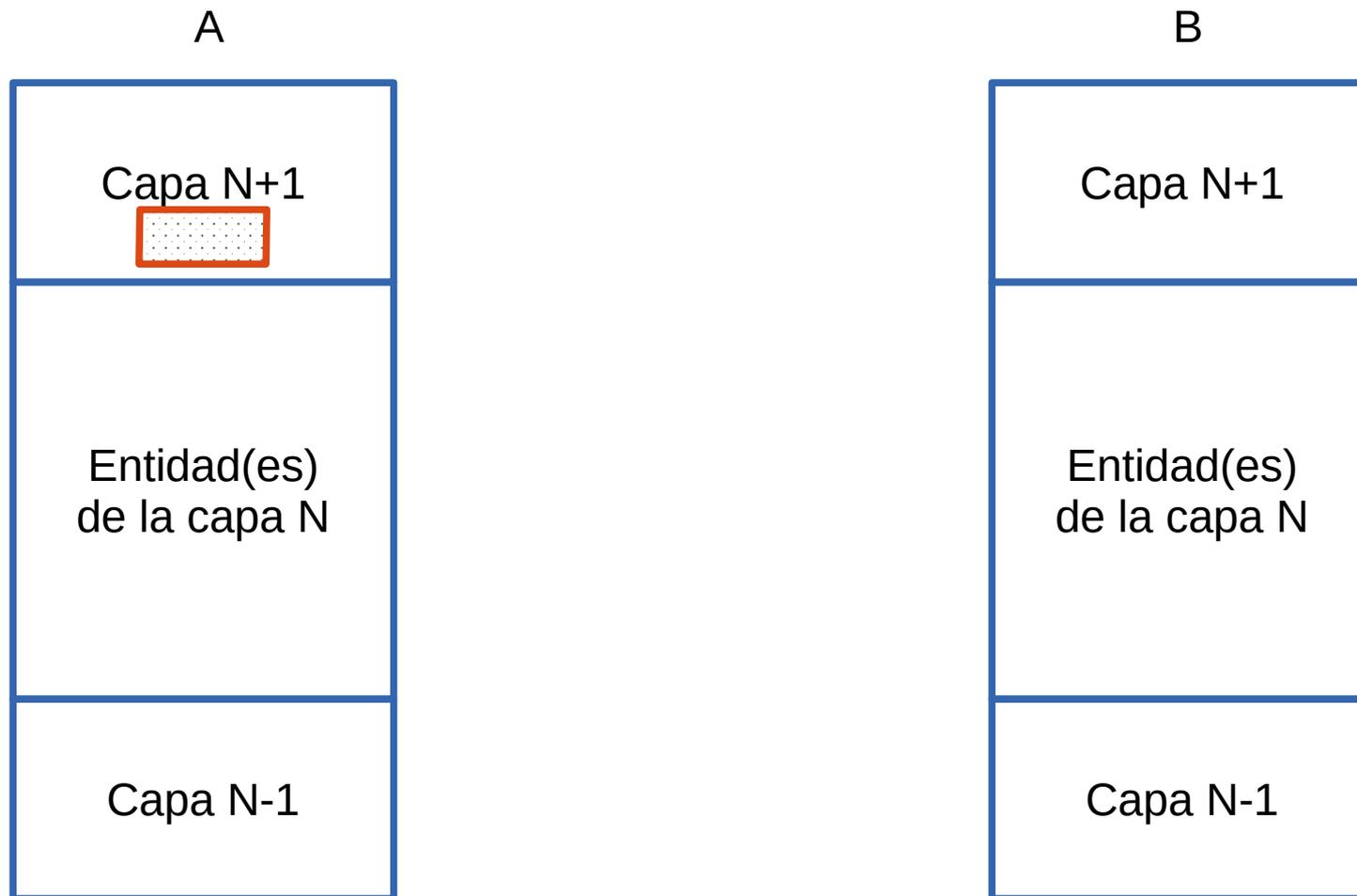
# Unidades de datos, protocolos

- Cada capa definirá las **unidades de datos** específicas para hacer su tarea
  - La unidad de datos de cada capa contiene:
    - **Encabezado**: campos necesarios para cumplir sus funciones
      - Ejemplos: para el control de errores, para la identificación o direccionamiento de equipos o servicios
    - **Carga útil (payload)**: la información de la capa superior
- En el modelo de capas, las entidades del mismo nivel (“**entidades pares**”) intercambian las unidades de datos de acuerdo a “**protocolos**”

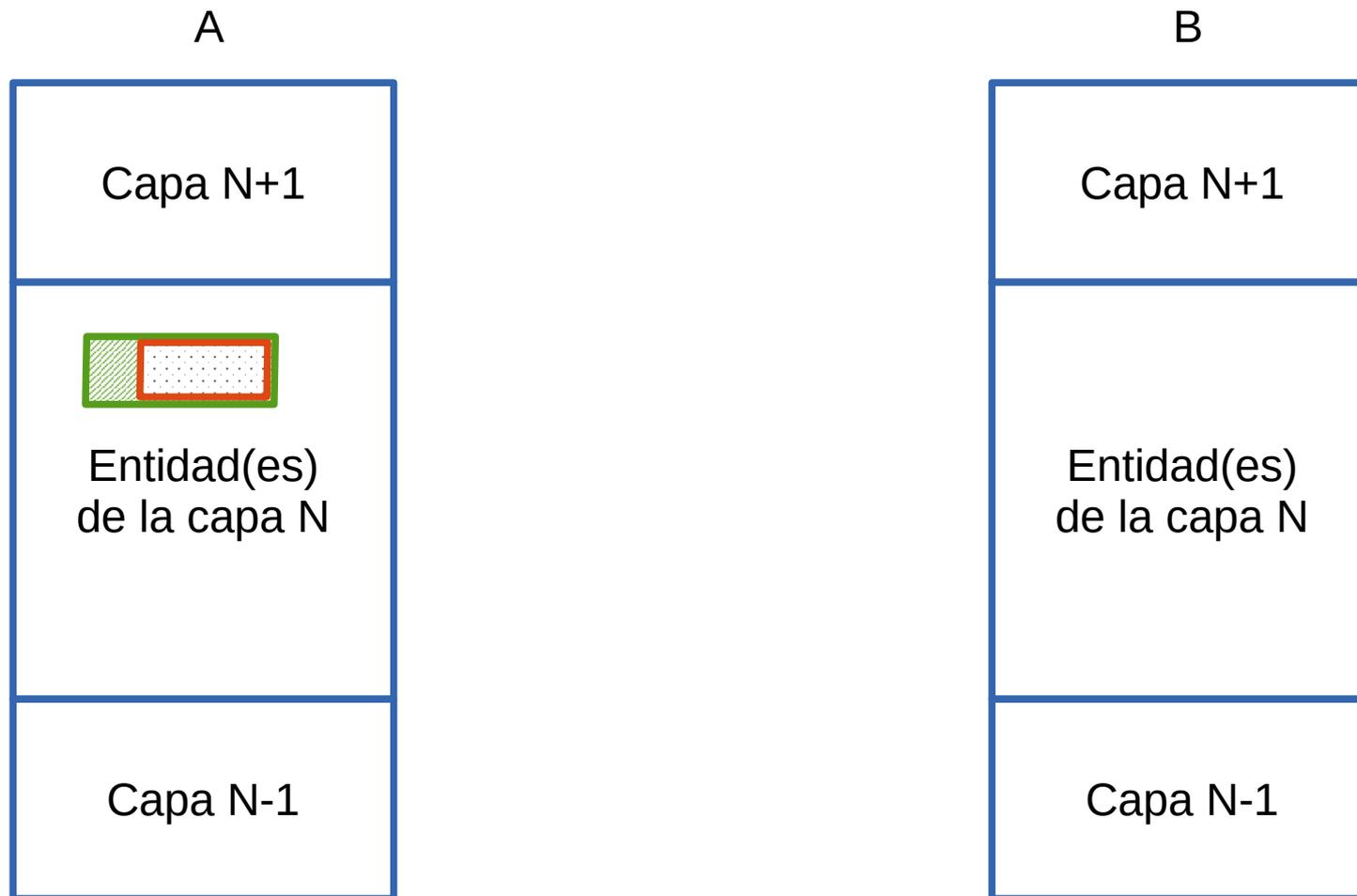
# Concepto de encapsulación o anidado



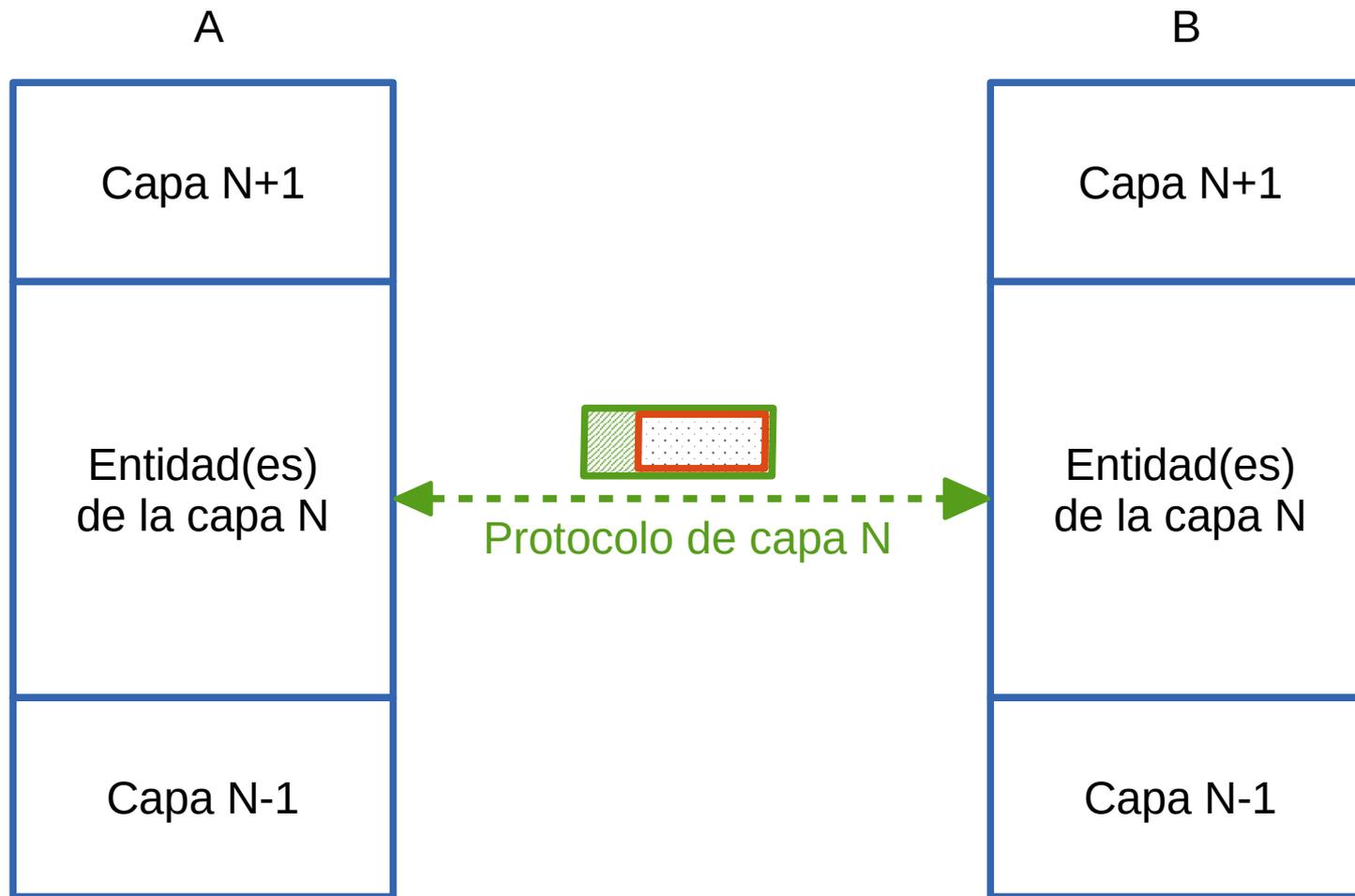
# Concepto de encapsulación o anidado



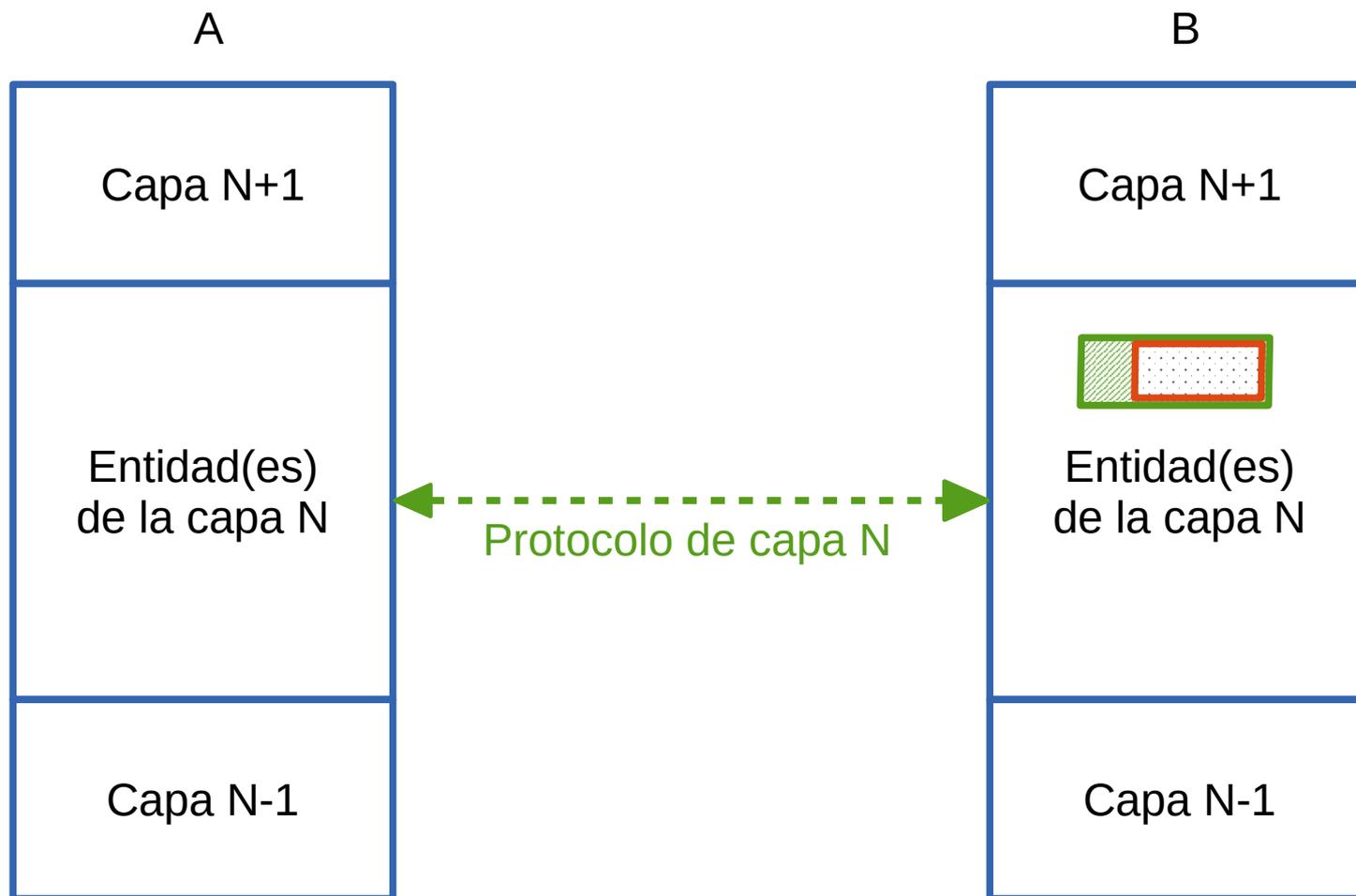
# Concepto de encapsulación o anidado



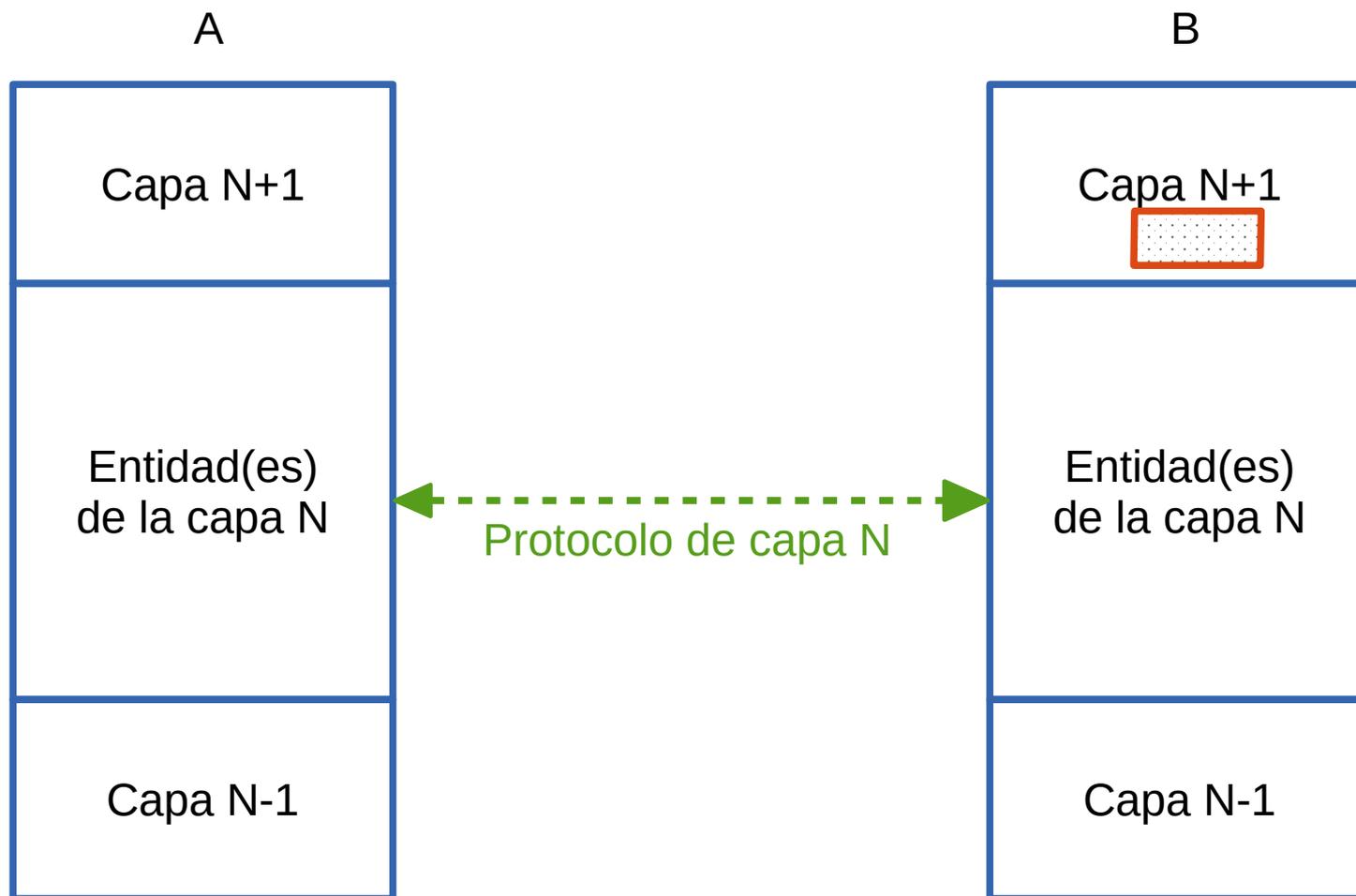
# Concepto de encapsulación o anidado



# Concepto de encapsulación o anidado



# Concepto de encapsulación o anidado

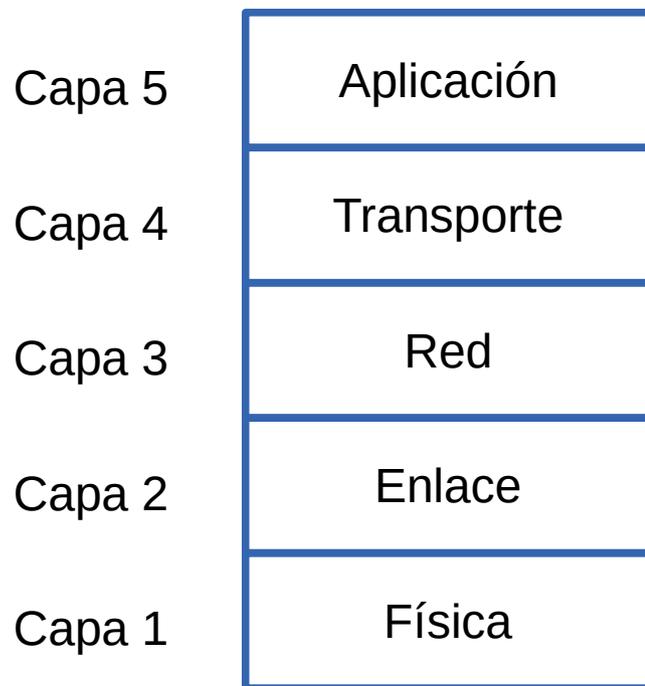


# Modelo de capas

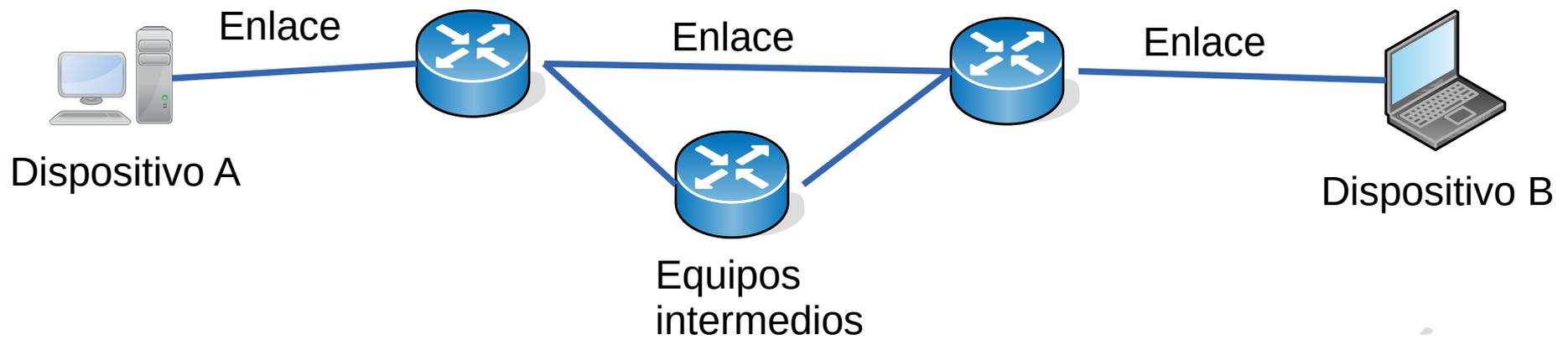
- ¿Qué **funciones** necesitamos para lograr el objetivo de comunicar dispositivos de forma remota con las complejidades que ya se mencionaron?
- ¿Cómo **organizamos** o agrupamos esas funciones en capas?
  - Es necesario identificar claramente los diferentes niveles de abstracción en que podemos analizar el problema
- ¿Qué funciones asociamos a cada capa?
- ¿**Cuántas capas** necesitamos?
  - Es conveniente:
    - No agrupar funciones muy diferentes en una misma capa
    - No tener un número inmanejable de capas

# Modelo de capas en Internet

- En Internet se usa un modelo de 5 capas
- Al conjunto de los protocolos que se implementan se los llama:  
pila de protocolos (protocol stack)

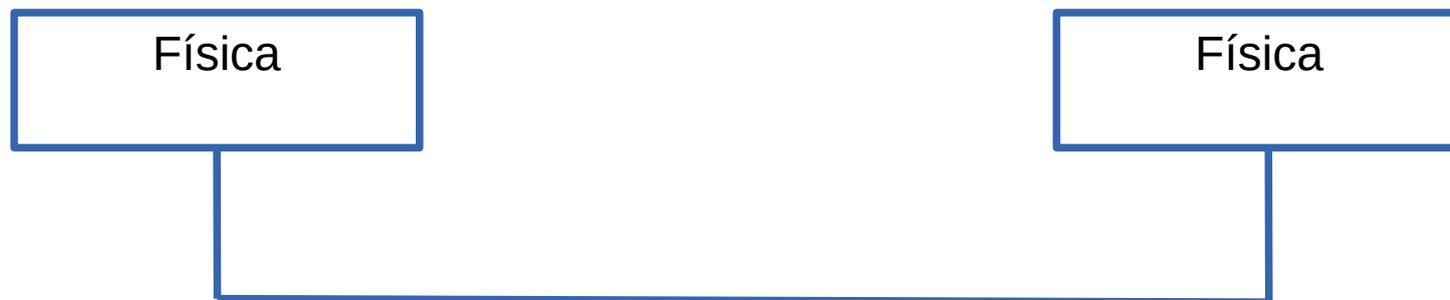


# Esquema simplificado de la red



# Capa 1: Capa Física

- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**



# Capa 1: Capa Física

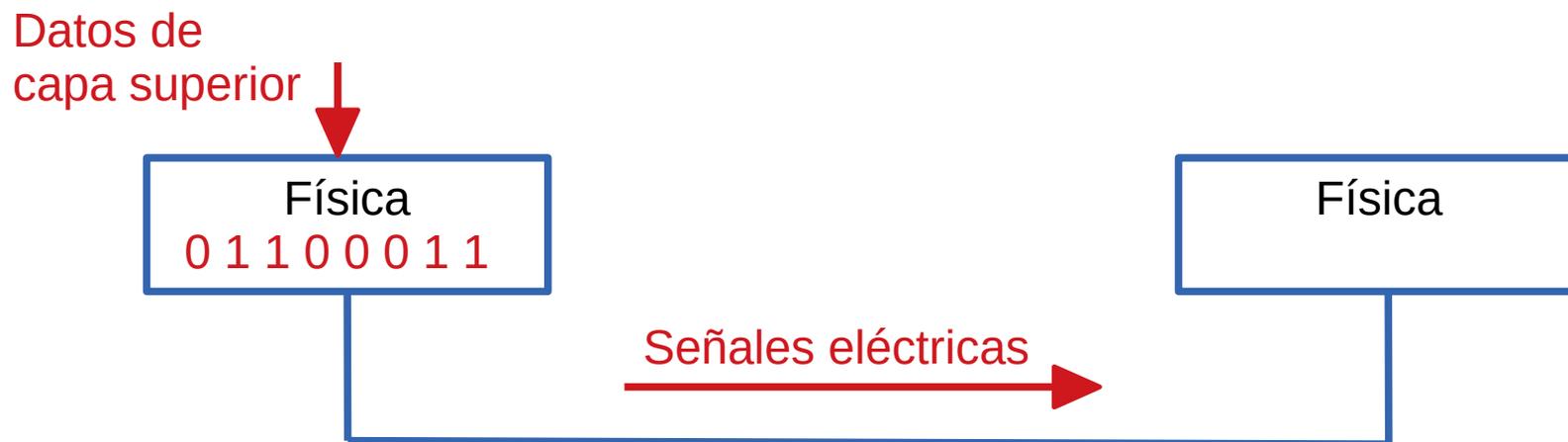
- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**

Datos de  
capa superior



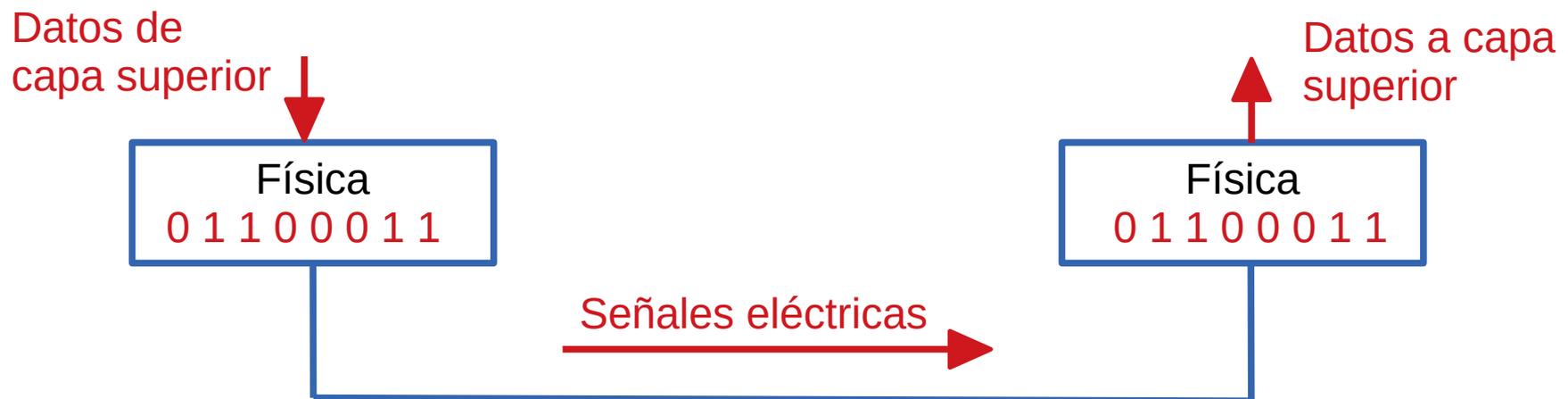
# Capa 1: Capa Física

- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**



# Capa 1: Capa Física

- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**



# Capa 1: Capa Física

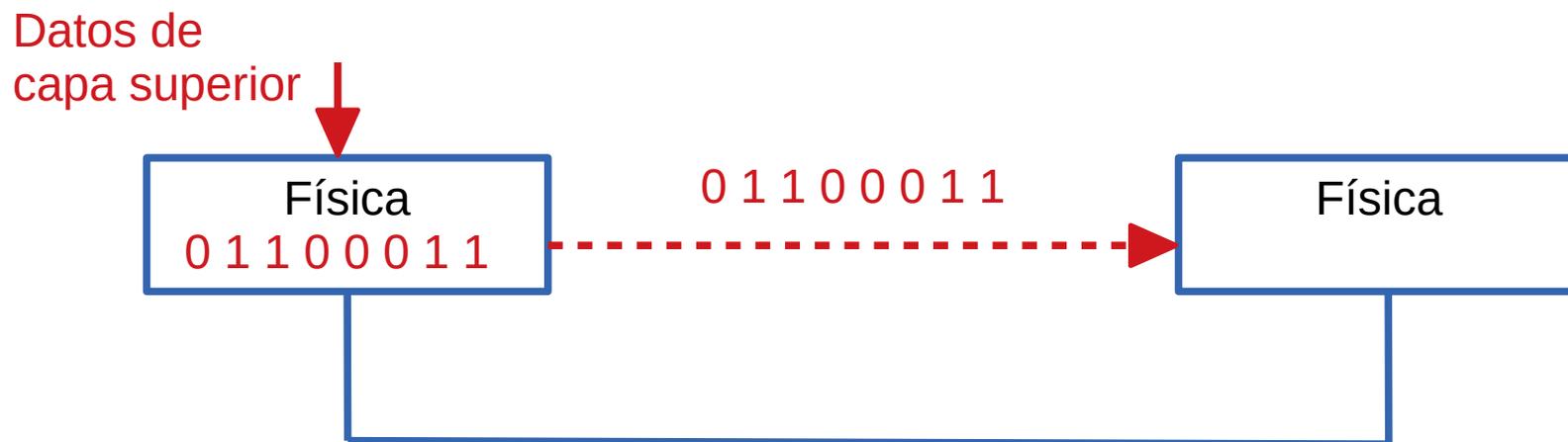
- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**

Datos de  
capa superior



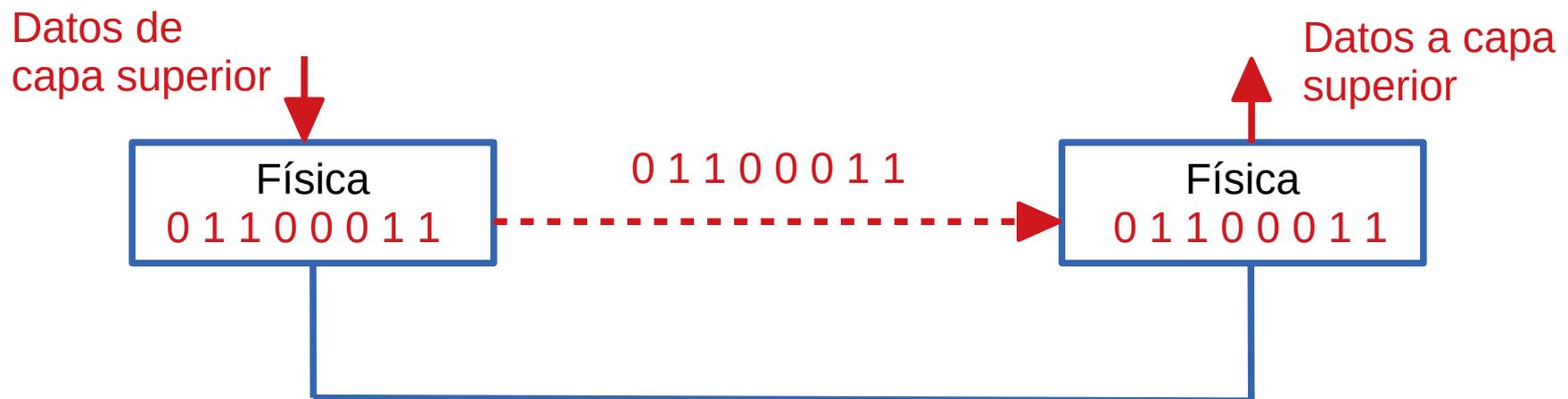
# Capa 1: Capa Física

- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**



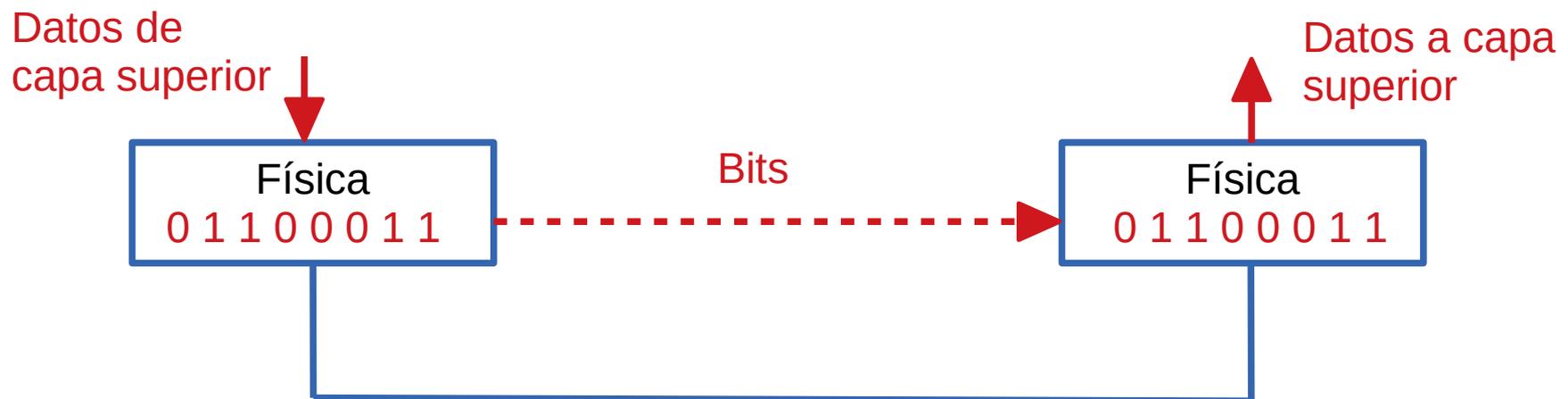
# Capa 1: Capa Física

- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**



# Capa 1: Capa Física

- **Objetivo** principal:
  - Transmitir bits por un medio físico
- Por ejemplo hay que definir:
  - Qué medio físico se usa? Qué tipo de cable y conectores?
  - Cómo se representa un bit en el canal de comunicación?
- Una vez resueltas estos aspectos se puede considerar que la capa física brinda el servicio de transmitir bits sobre un enlace
  - La unidad de datos de esta capa es el **bit**



# Capa 2: Capa de Enlace

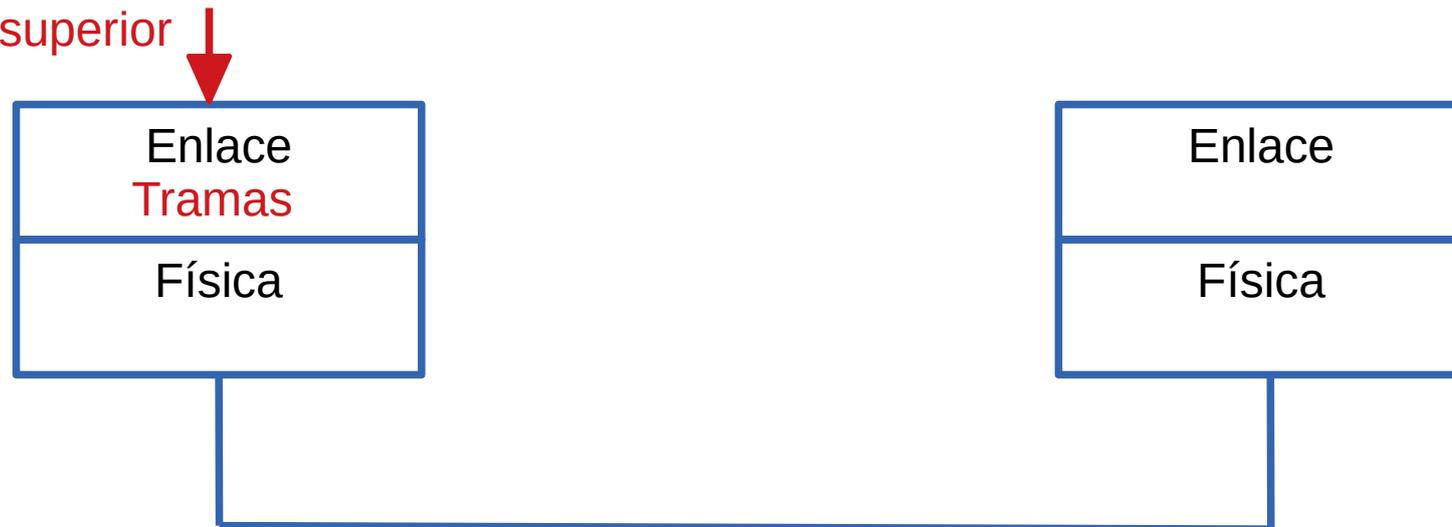
- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

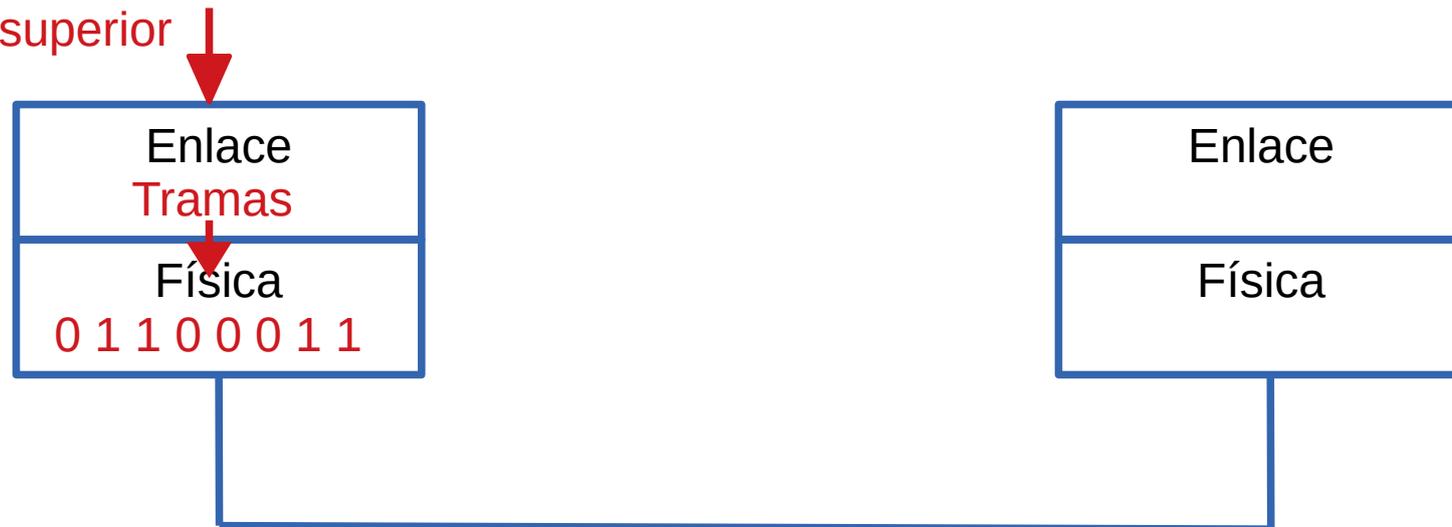
Datos de  
capa superior



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

Datos de  
capa superior



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

Datos de  
capa superior



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

Datos de  
capa superior



# Capa 2: Capa de Enlace

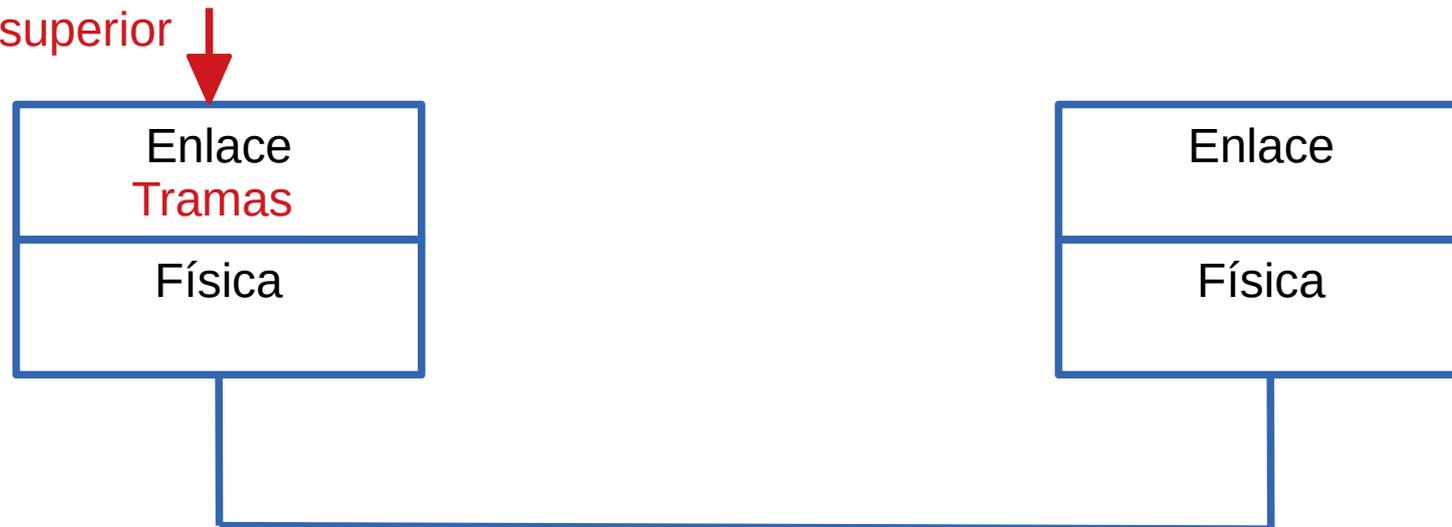
- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

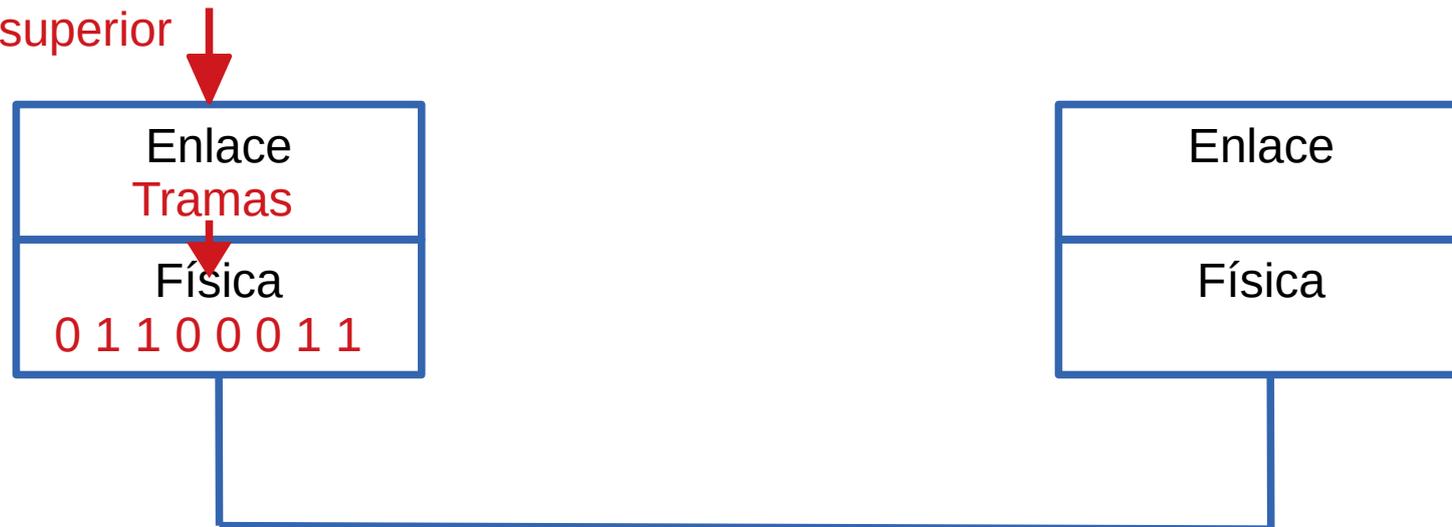
Datos de  
capa superior



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

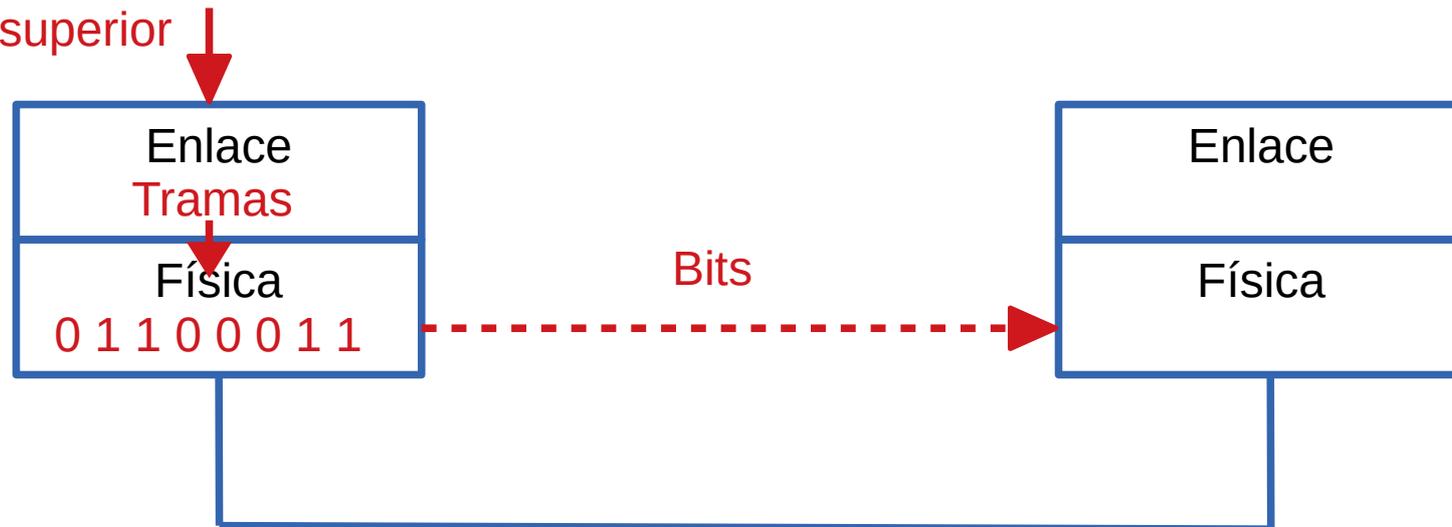
Datos de  
capa superior



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

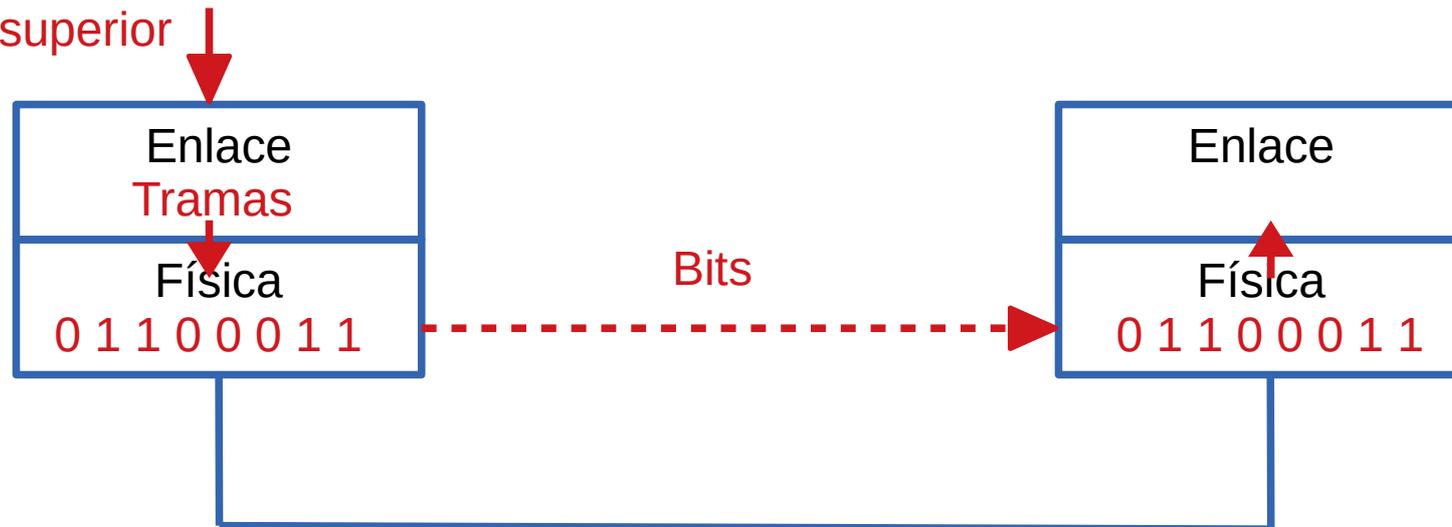
Datos de  
capa superior



# Capa 2: Capa de Enlace

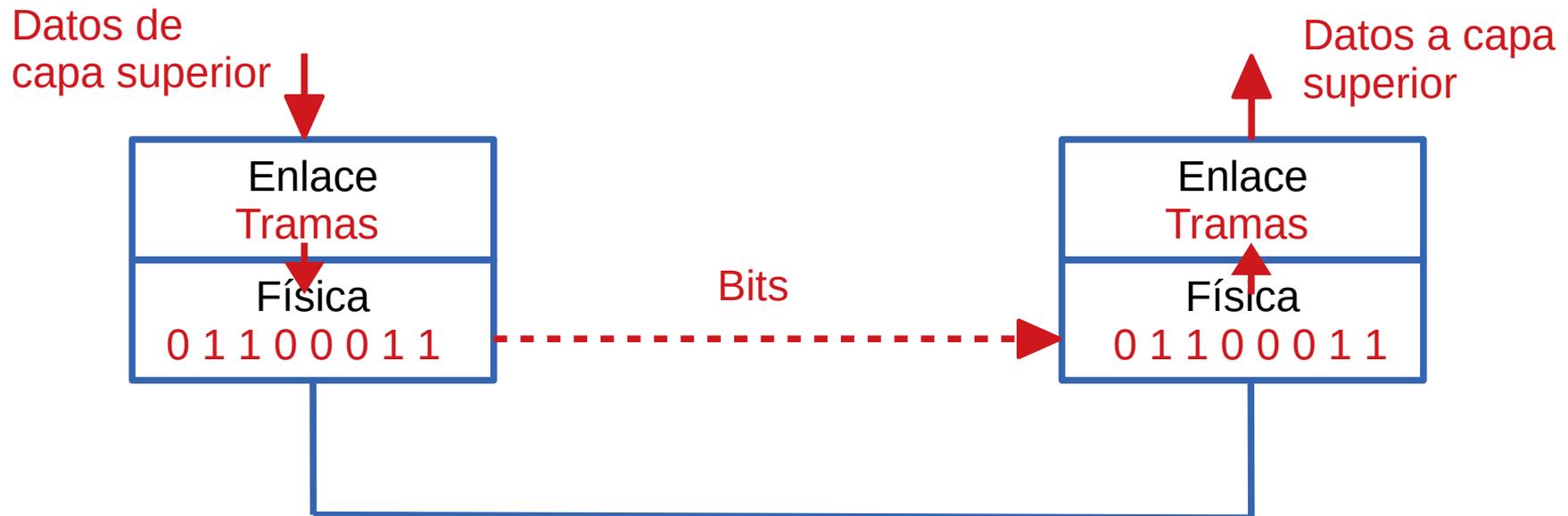
- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

Datos de  
capa superior



# Capa 2: Capa de Enlace

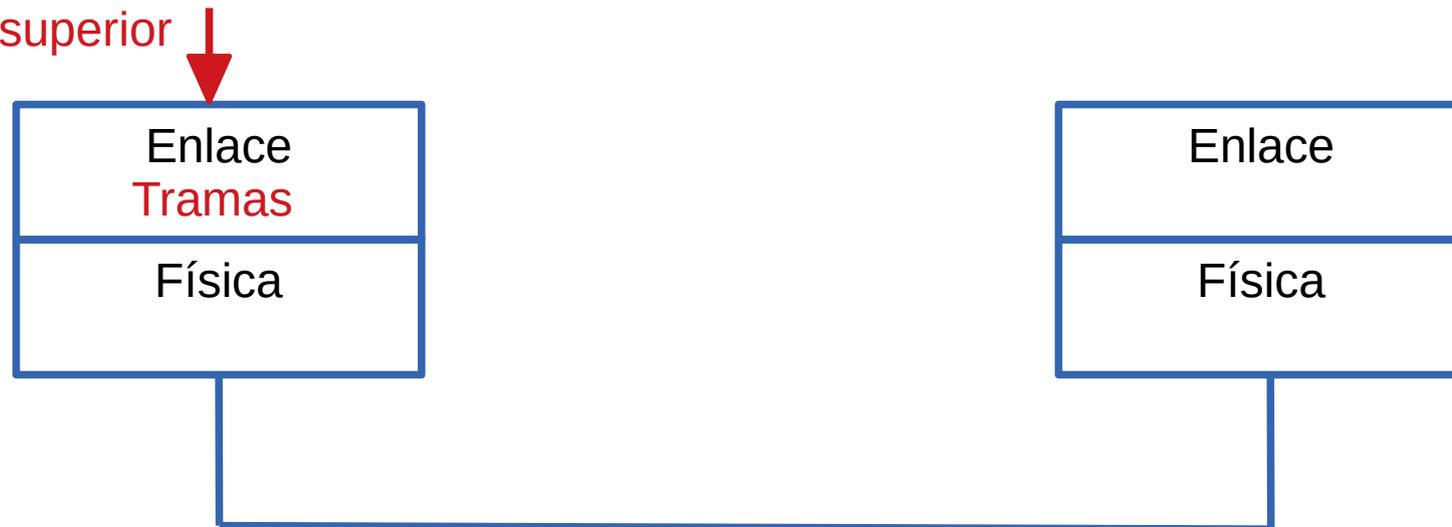
- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones



# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

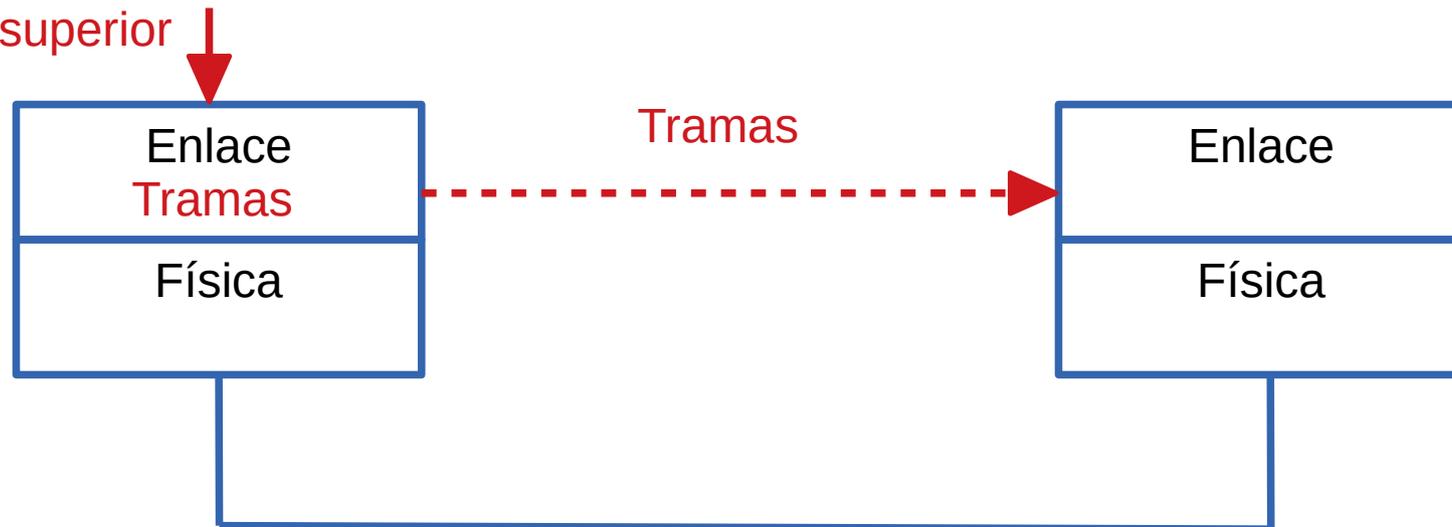
Datos de  
capa superior



# Capa 2: Capa de Enlace

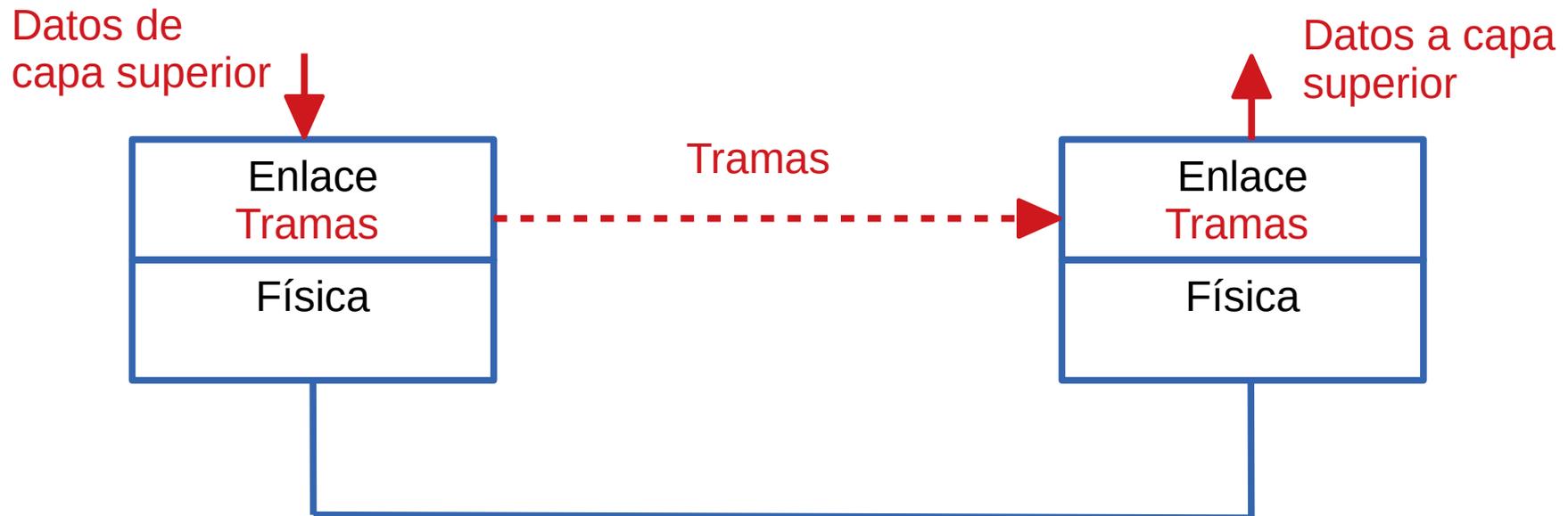
- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones

Datos de  
capa superior



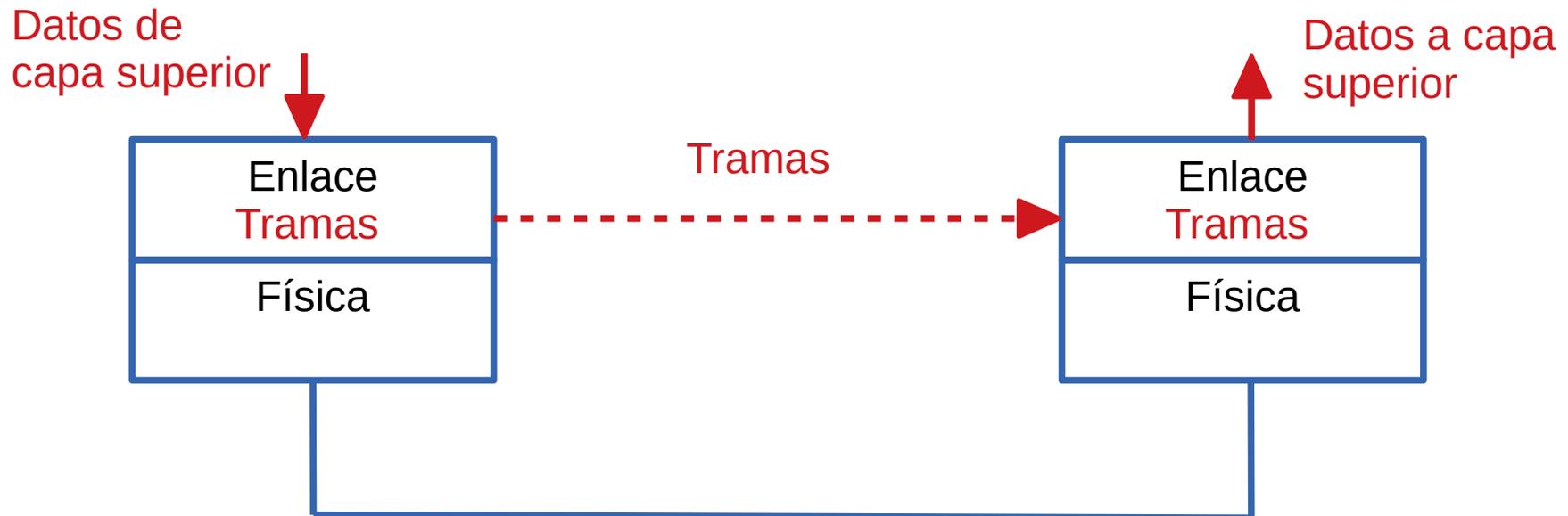
# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones



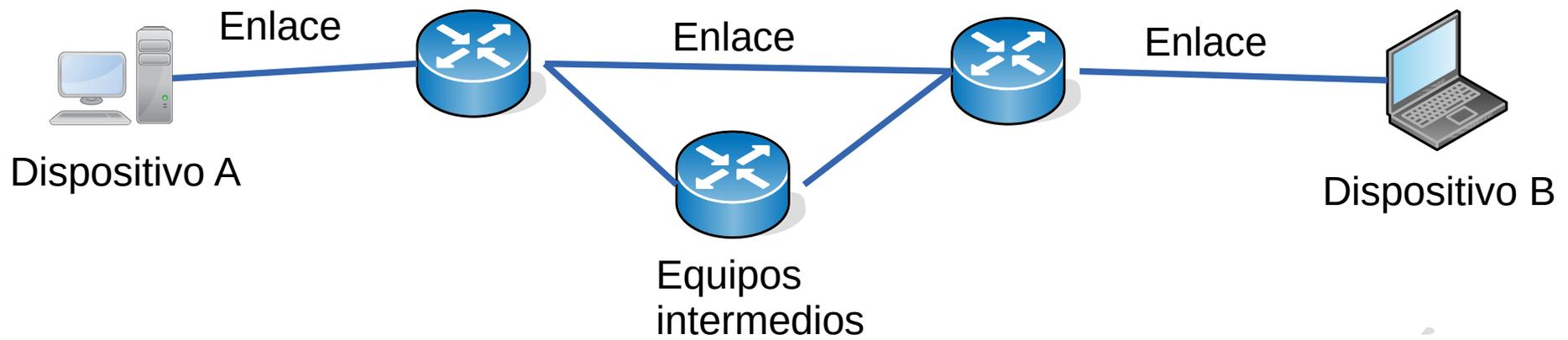
# Capa 2: Capa de Enlace

- **Objetivo** principal:
  - Controlar los errores de bits del medio físico
- Todos los canales tienen errores (interferencias, atenuación)
- Es necesario saber si la información llegó correctamente al destino
- Se agrupan los datos en unidades de información llamadas **tramas** y se agrega redundancia para controlar los errores de cada trama
- Otras funciones: control de flujo, diferentes tipos de servicios, direcciones



# Capa 2: Capa de Enlace

- La capa de enlace resuelve el control de errores, control de flujo y eventualmente garantiza que todas las tramas llegan al otro extremo en cada enlace
- Tendríamos resuelto el control del intercambio de información en cada tramo del camino de A hasta B
- Pero:
  - Cómo se decide el camino de A a B?
  - Cuál camino es el más conveniente si hay múltiples alternativas?

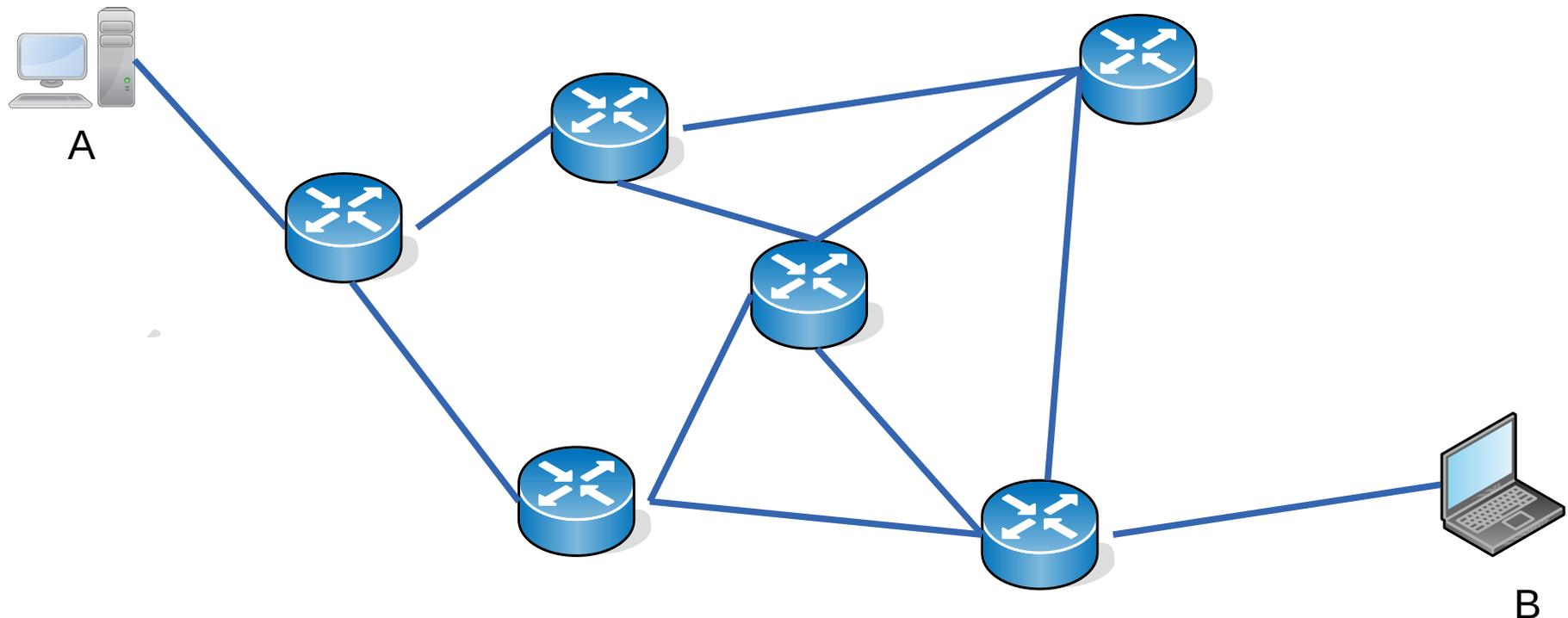


# Capa 3: Capa de Red

- **Objetivo** principal:
  - Hacer llegar los paquetes desde A hasta B
- La información se organiza en unidades de datos llamadas **paquetes**
- Se necesita establecer cómo se identifican y localizan los equipos: **direcciones**
- Función de **ruteo** (routing) : **Plano de Control**
  - Encontrar el (mejor) camino desde A hasta B
  - Se usan algoritmos para determinar los mejores caminos
  - Tradicionalmente los caminos se reflejan en tablas en cada nodo para que sepa cómo llegar a cada destino
- Función de **encaminamiento** (forwarding) : **Plano de Datos**
  - Cada nodo recibe paquetes por una línea de entrada o interfaz, decide cuál es la línea de salida más adecuada para ese destino y encamina el paquete hacia allí
  - Los paquetes se encaminan usando tablas
- Otras funciones: tarificación, control de congestión (en algunos casos)

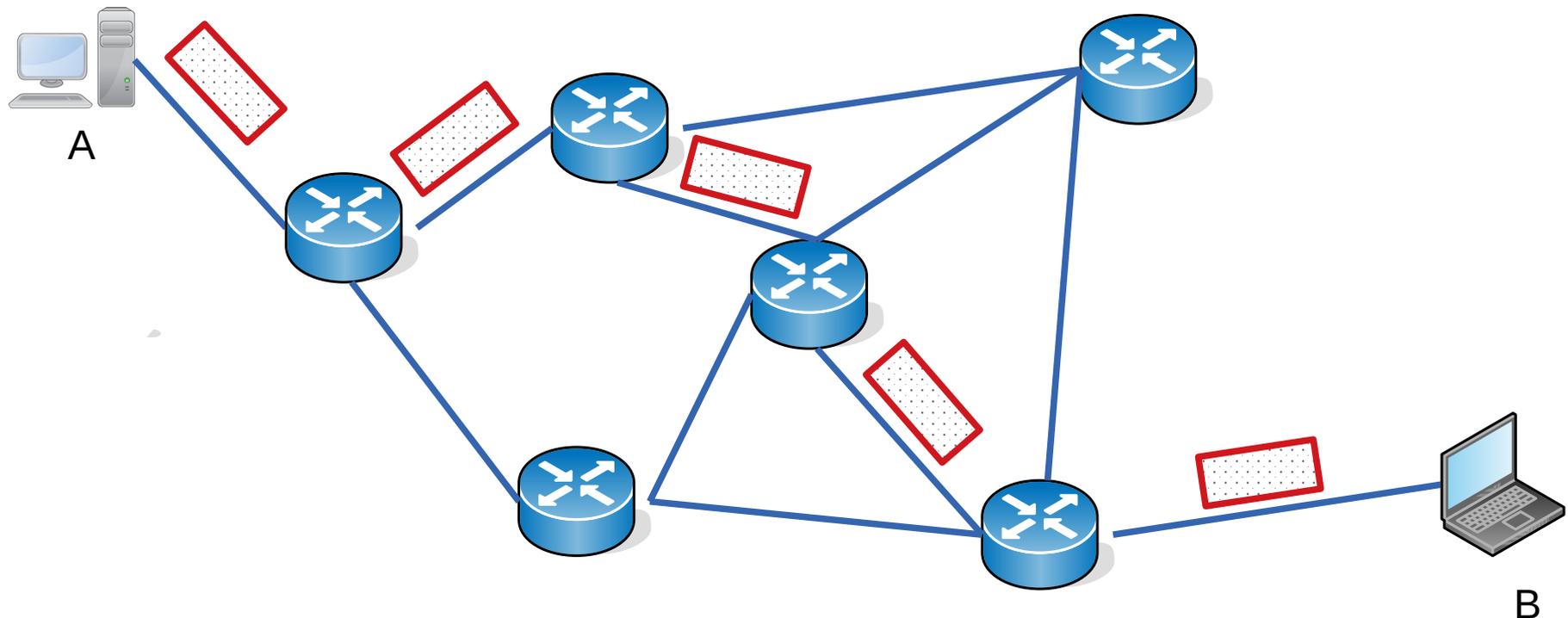
# Capa 3: Función de Ruteo

- Cuál es el **mejor camino** para ir de A a B?
- La decisión se plasma en **tablas** configuradas en cada nodo



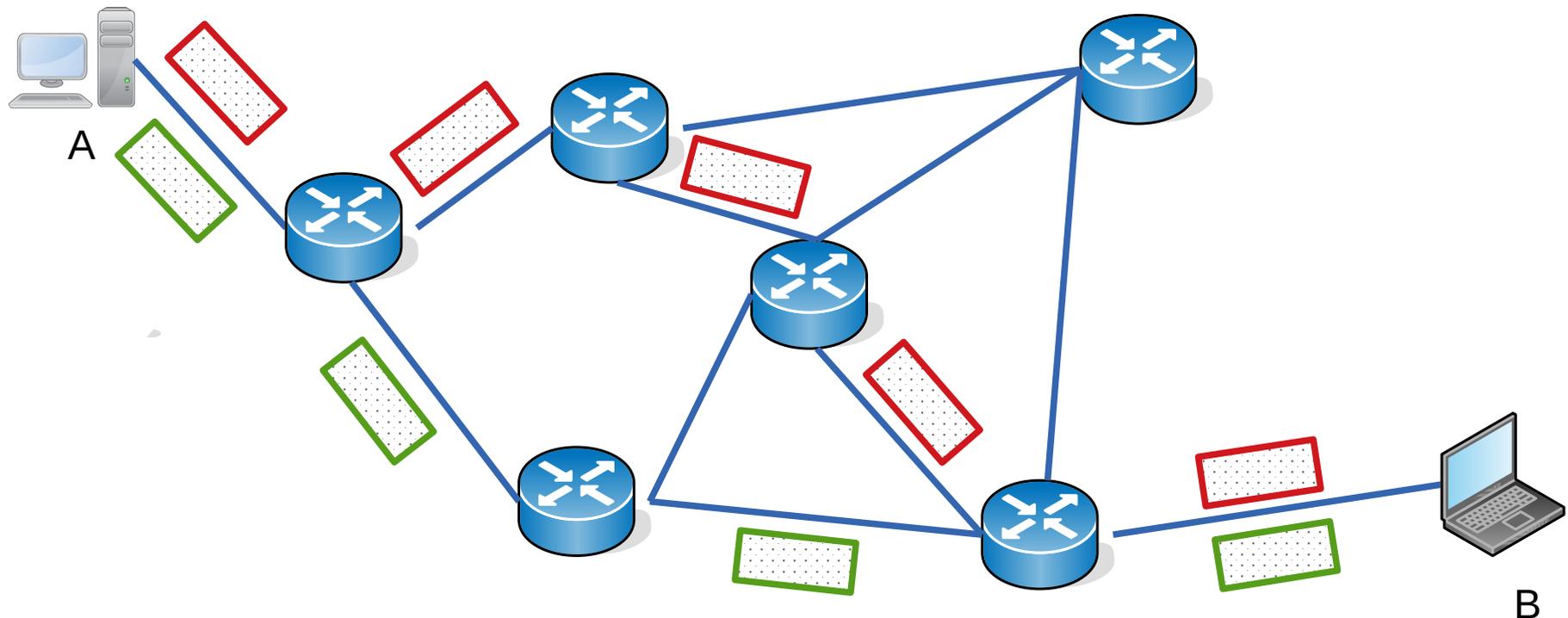
# Capa 3: Función de Ruteo

- Cuál es el **mejor camino** para ir de A a B?
- La decisión se plasma en **tablas** configuradas en cada nodo

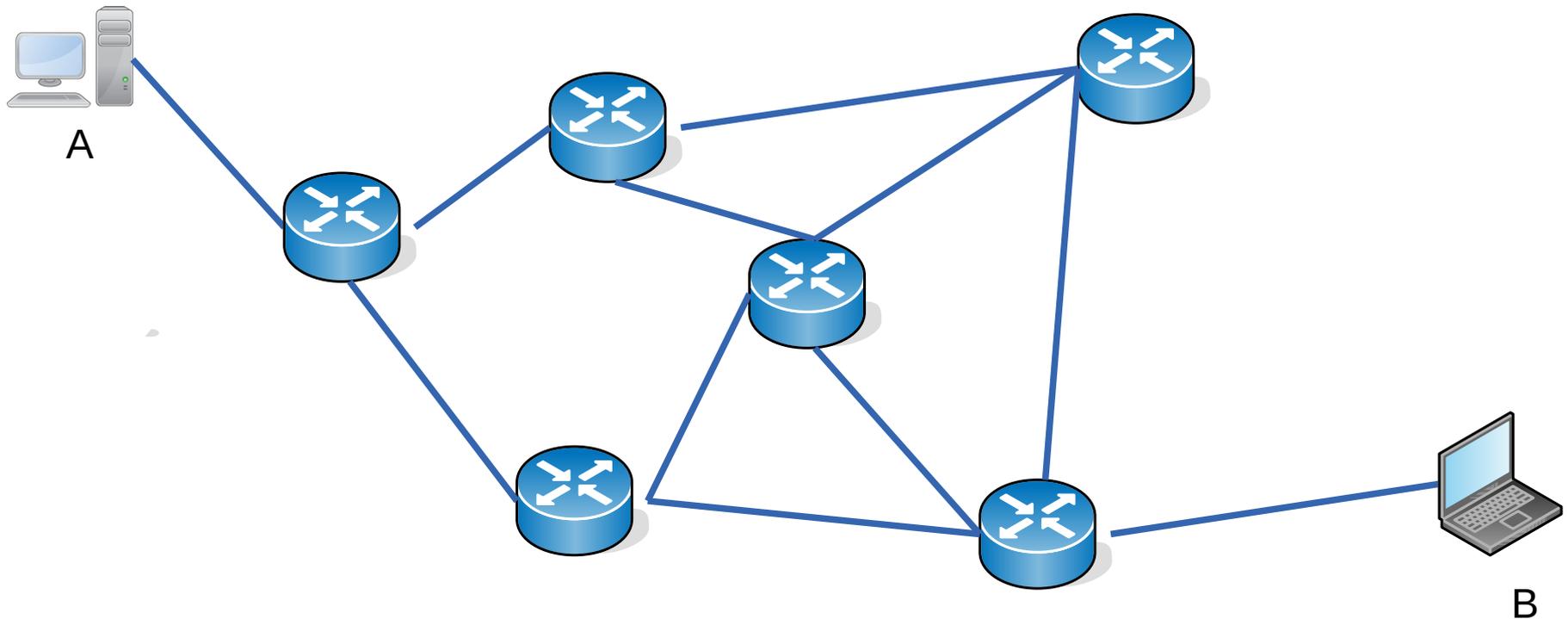


# Capa 3: Función de Ruteo

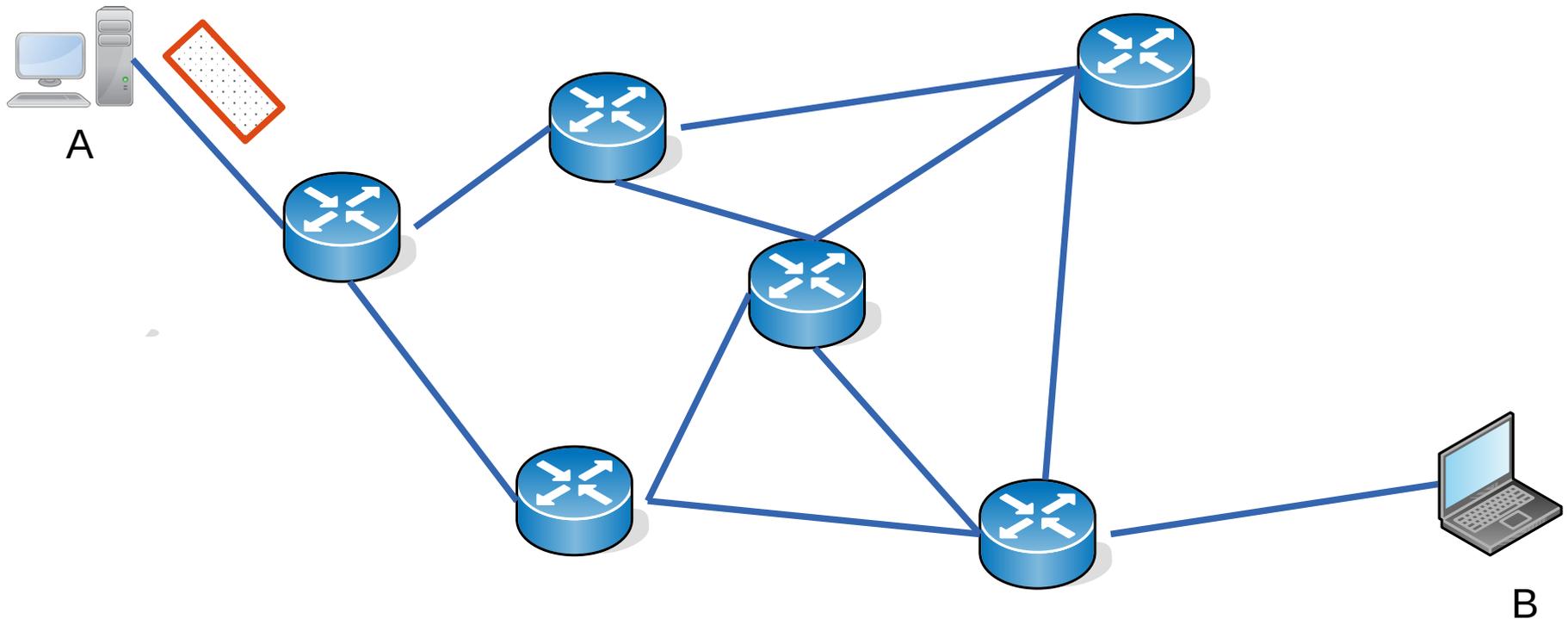
- Cuál es el **mejor camino** para ir de A a B?
- La decisión se plasma en **tablas** configuradas en cada nodo



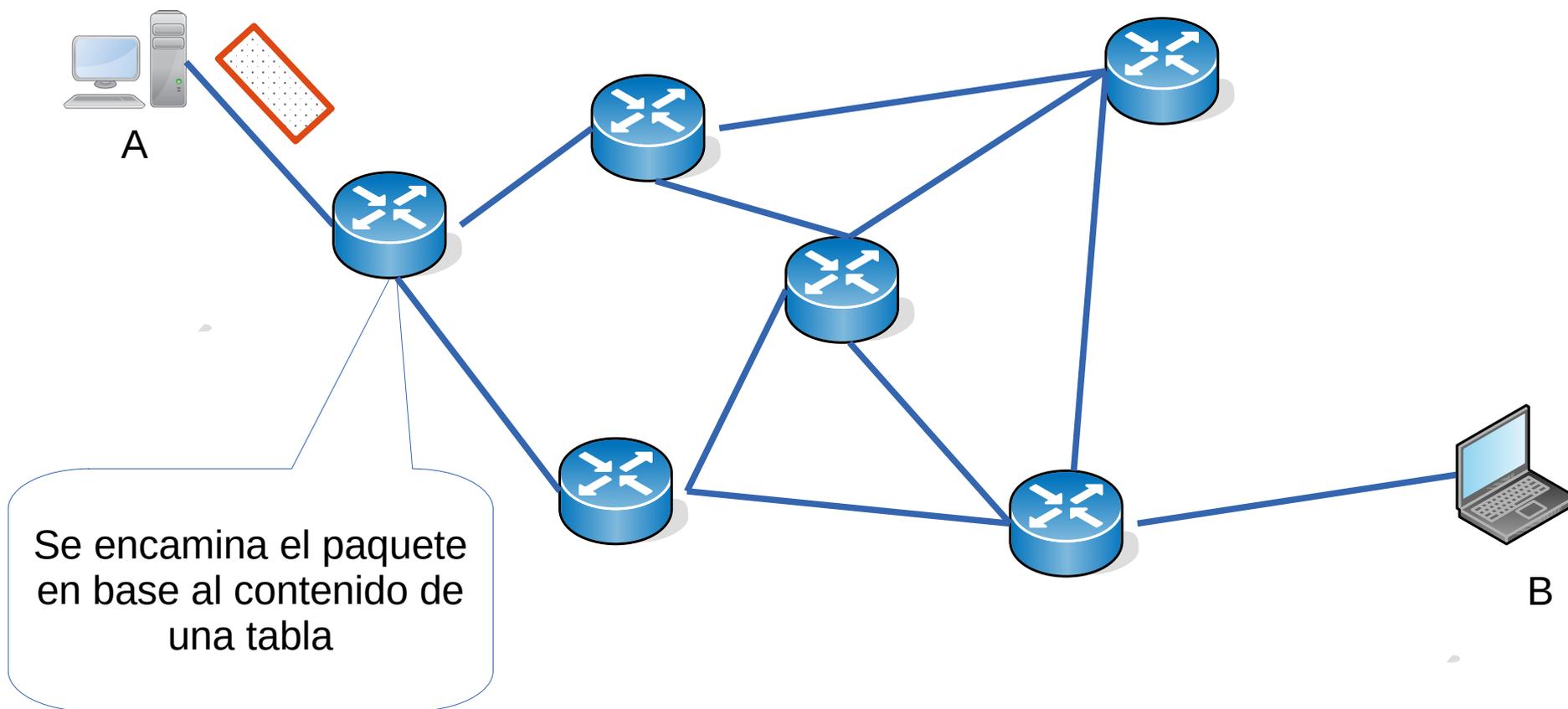
# Capa 3: Función de Encaminamiento



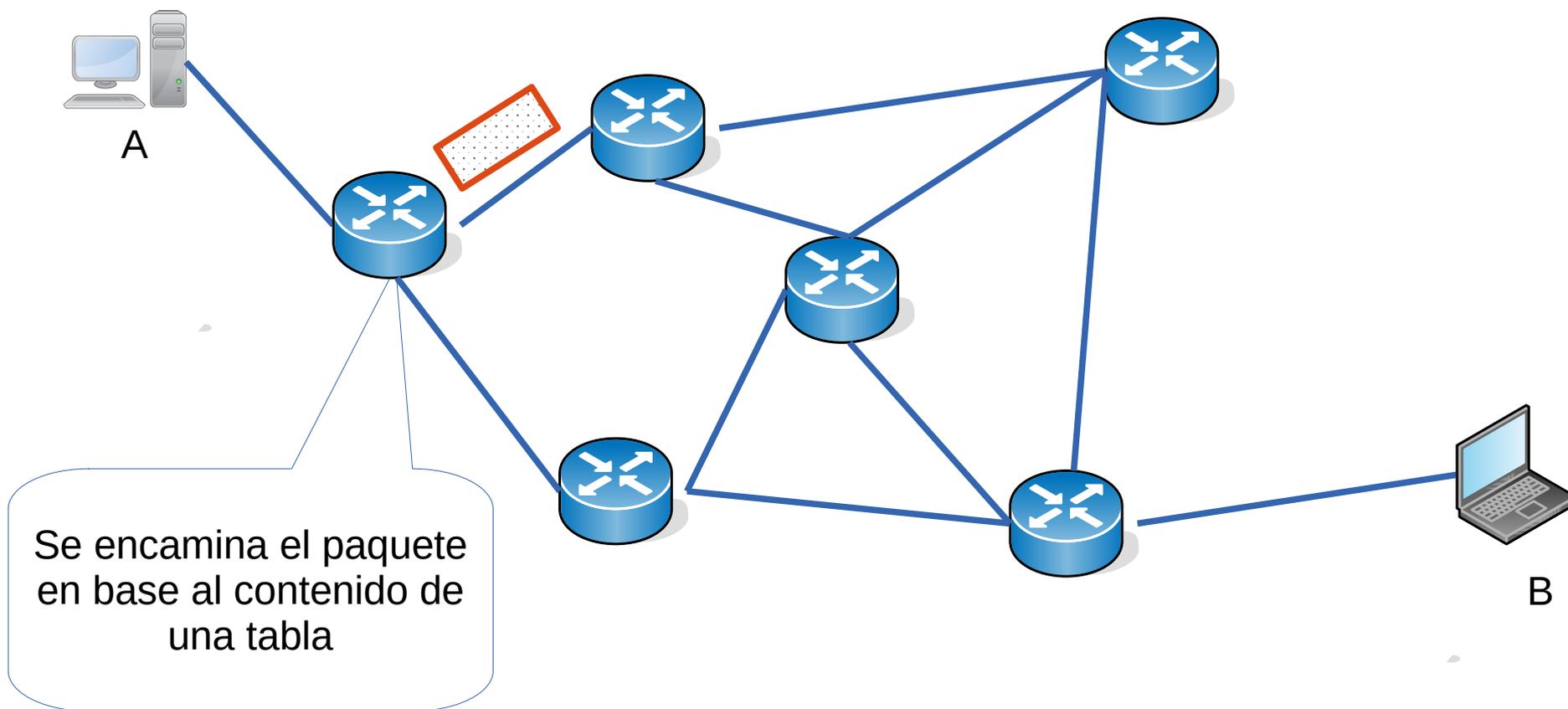
# Capa 3: Función de Encaminamiento



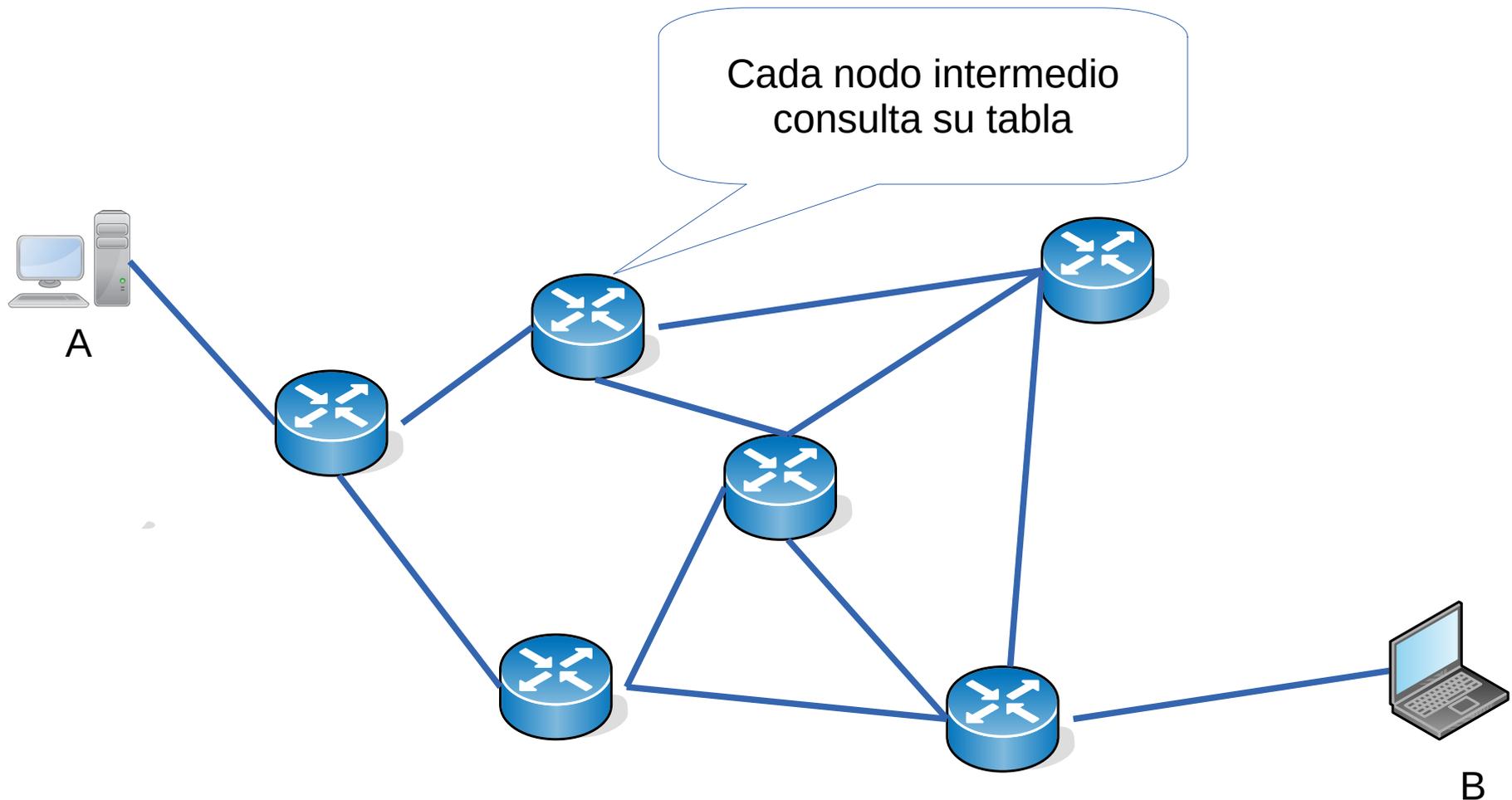
# Capa 3: Función de Encaminamiento



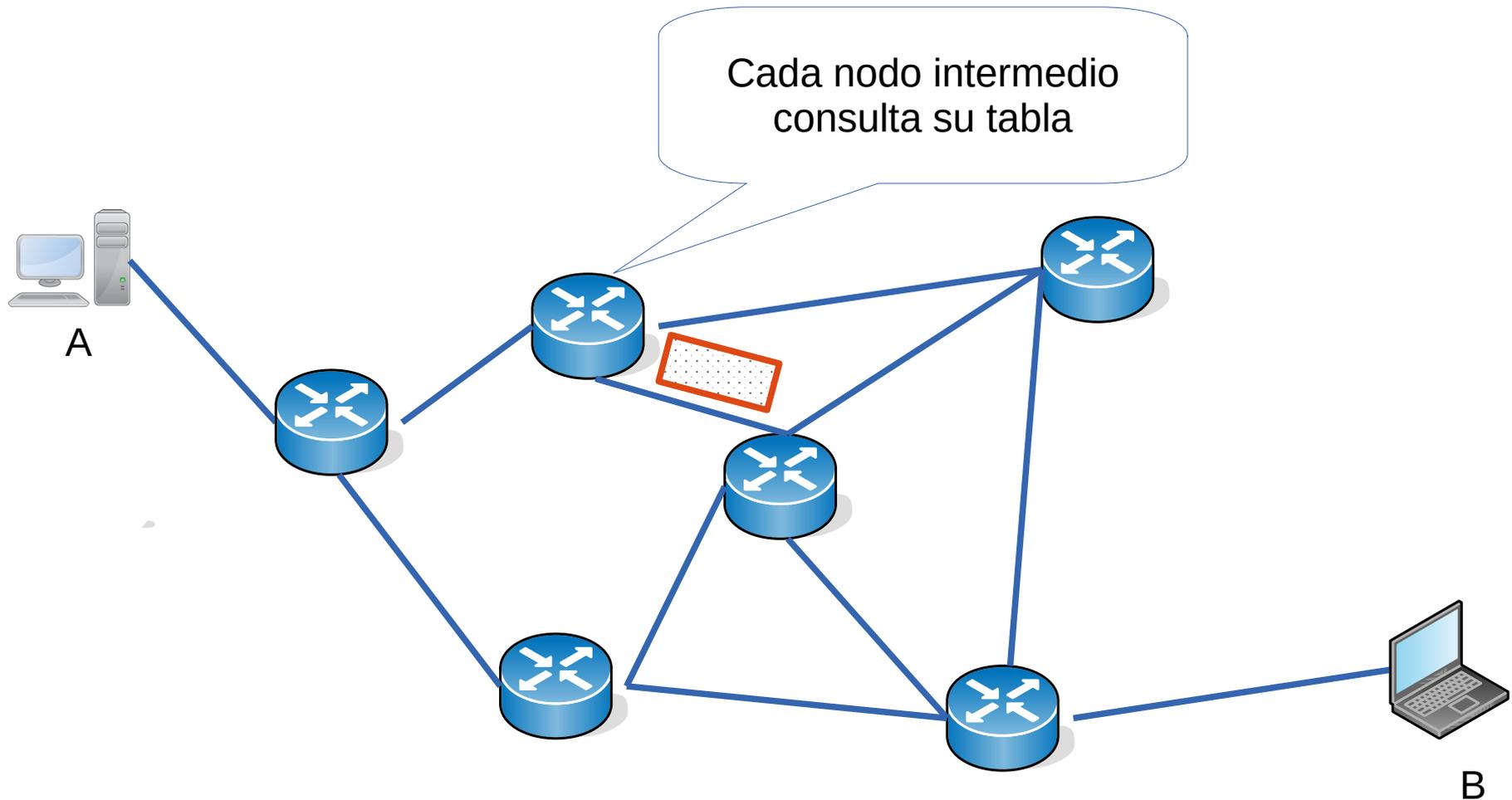
# Capa 3: Función de Encaminamiento



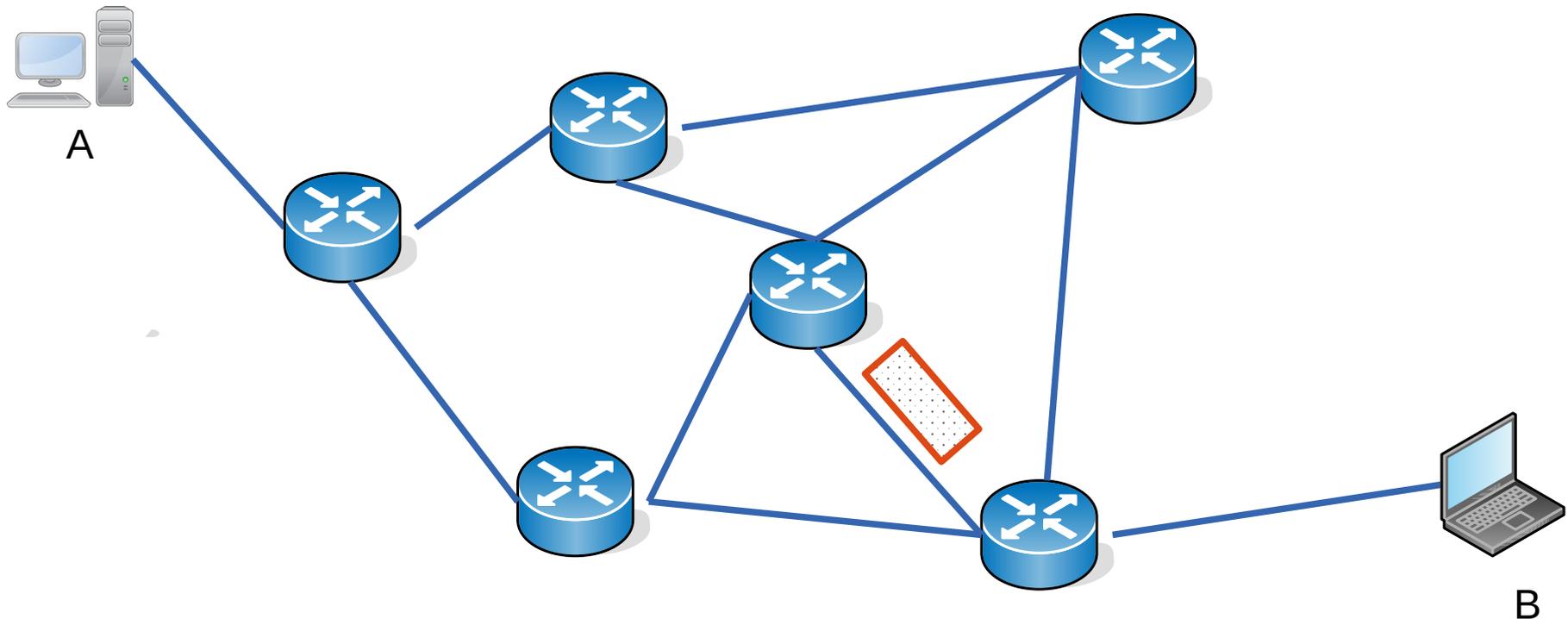
# Capa 3: Función de Encaminamiento



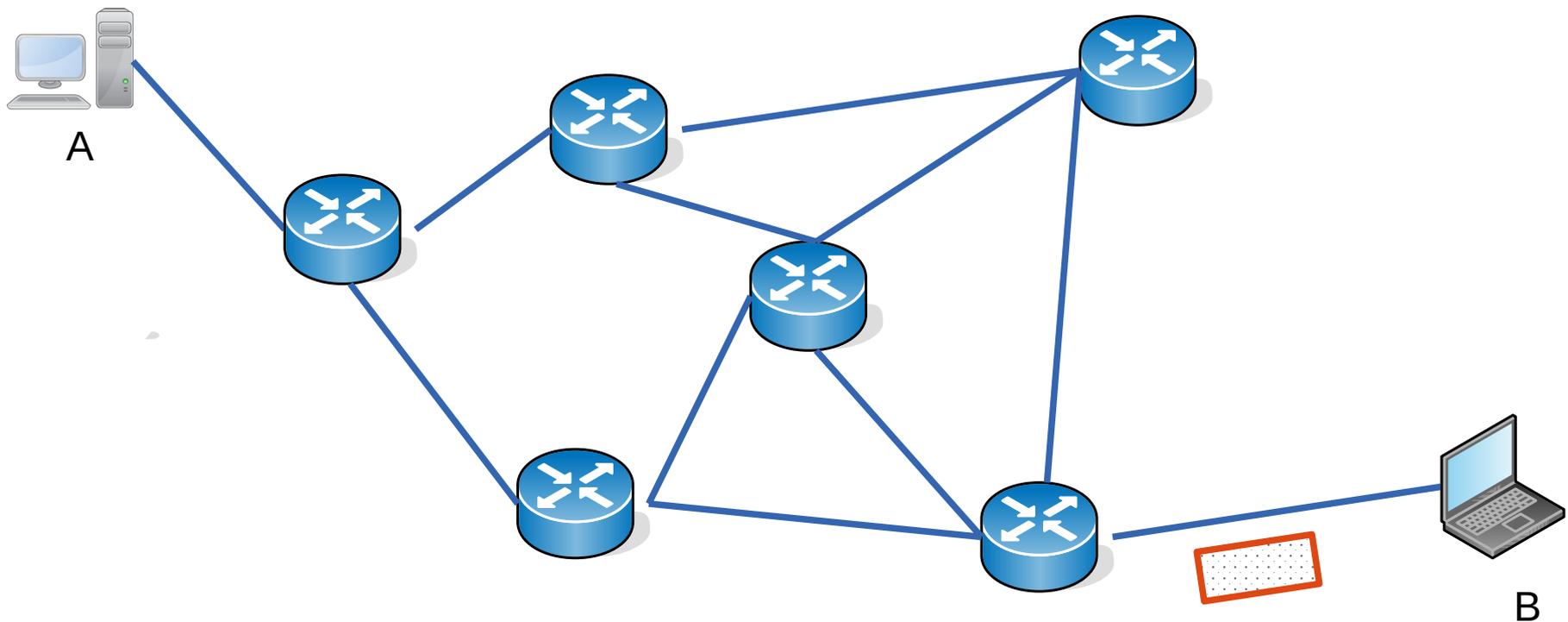
# Capa 3: Función de Encaminamiento



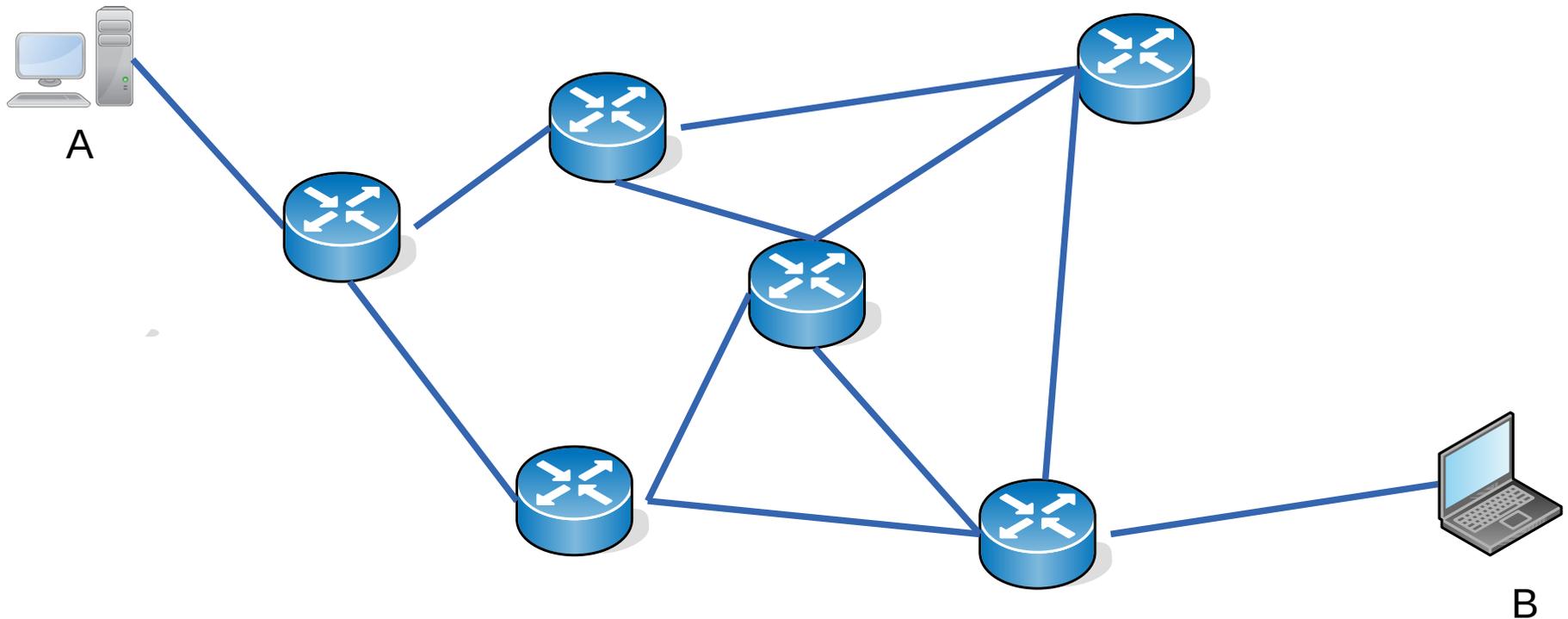
# Capa 3: Función de Encaminamiento



# Capa 3: Función de Encaminamiento

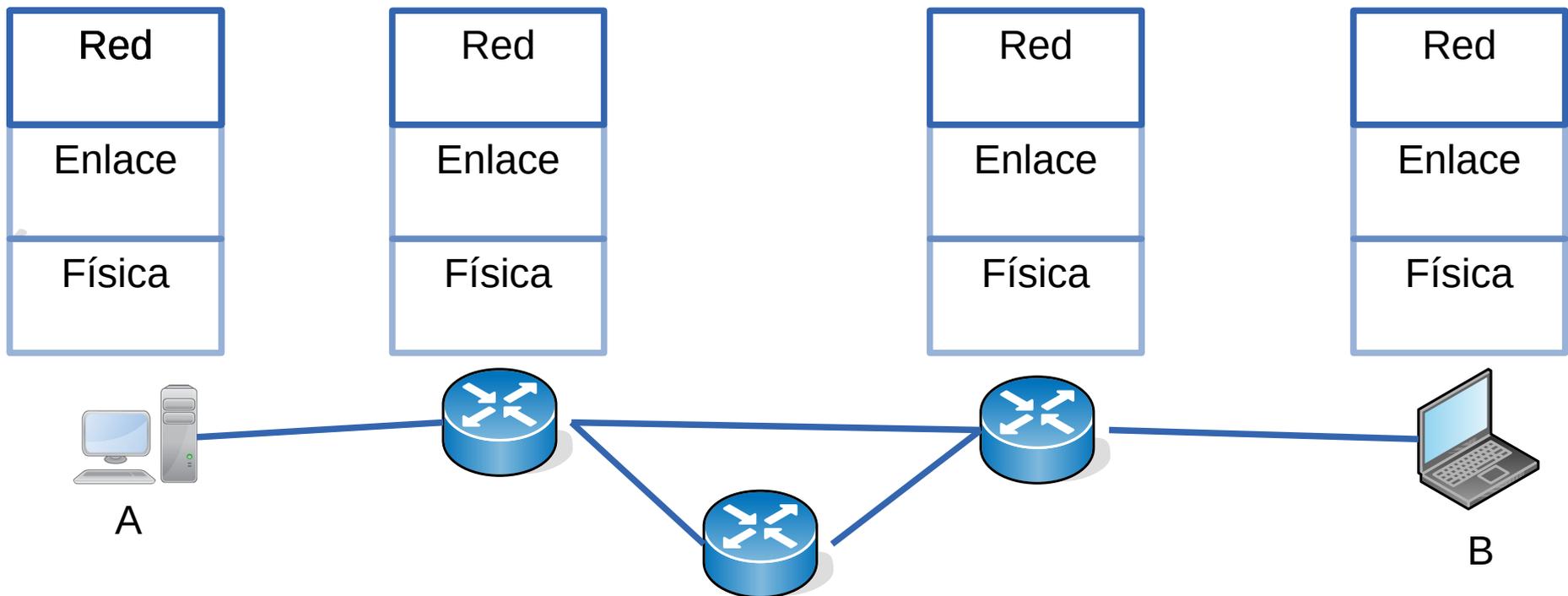


# Capa 3: Función de Encaminamiento



# Capa 3: Capa de red

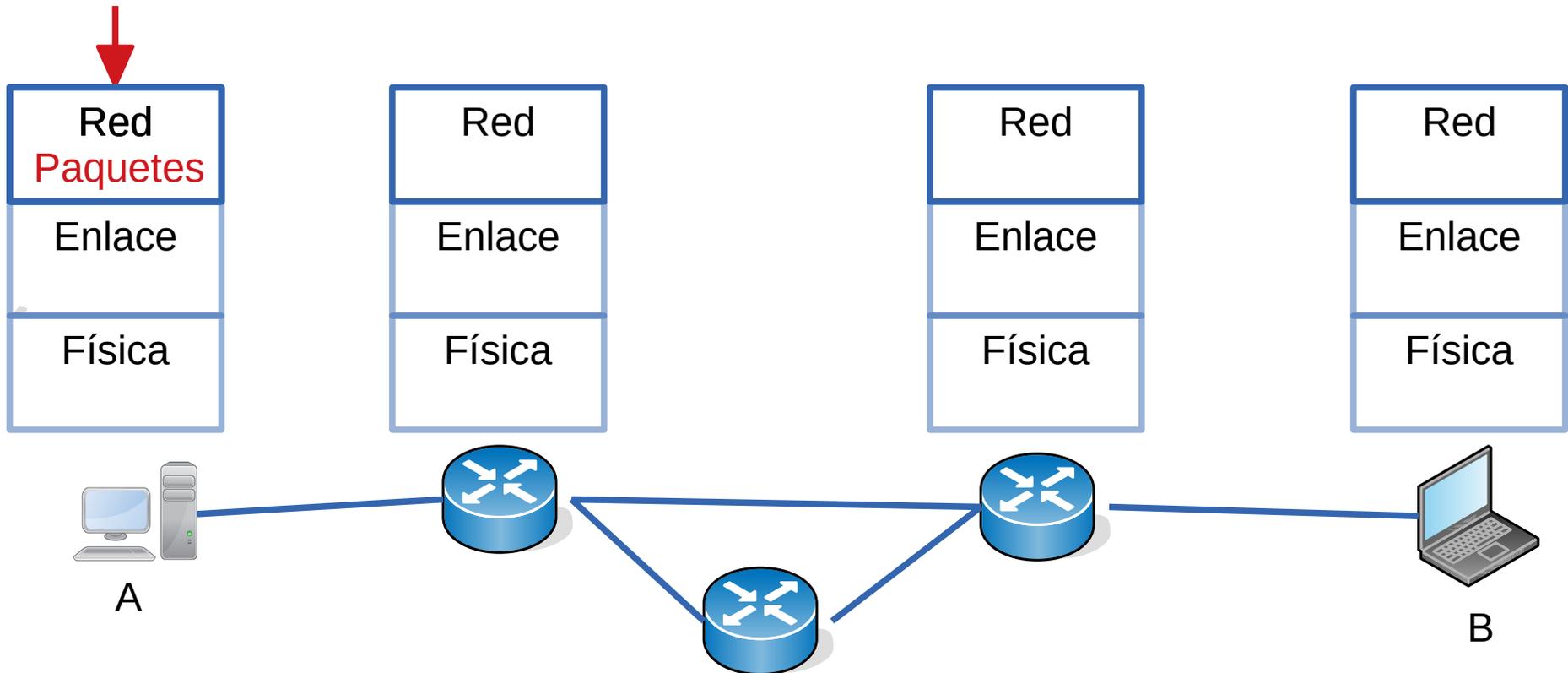
- A este nivel de abstracción podemos ver un intercambio de paquetes entre nodos
- Nos **independizamos** de lo que pasa en las capas inferiores



# Capa 3: Capa de red

- A este nivel de abstracción podemos ver un intercambio de paquetes entre nodos
- Nos **independizamos** de lo que pasa en las capas inferiores

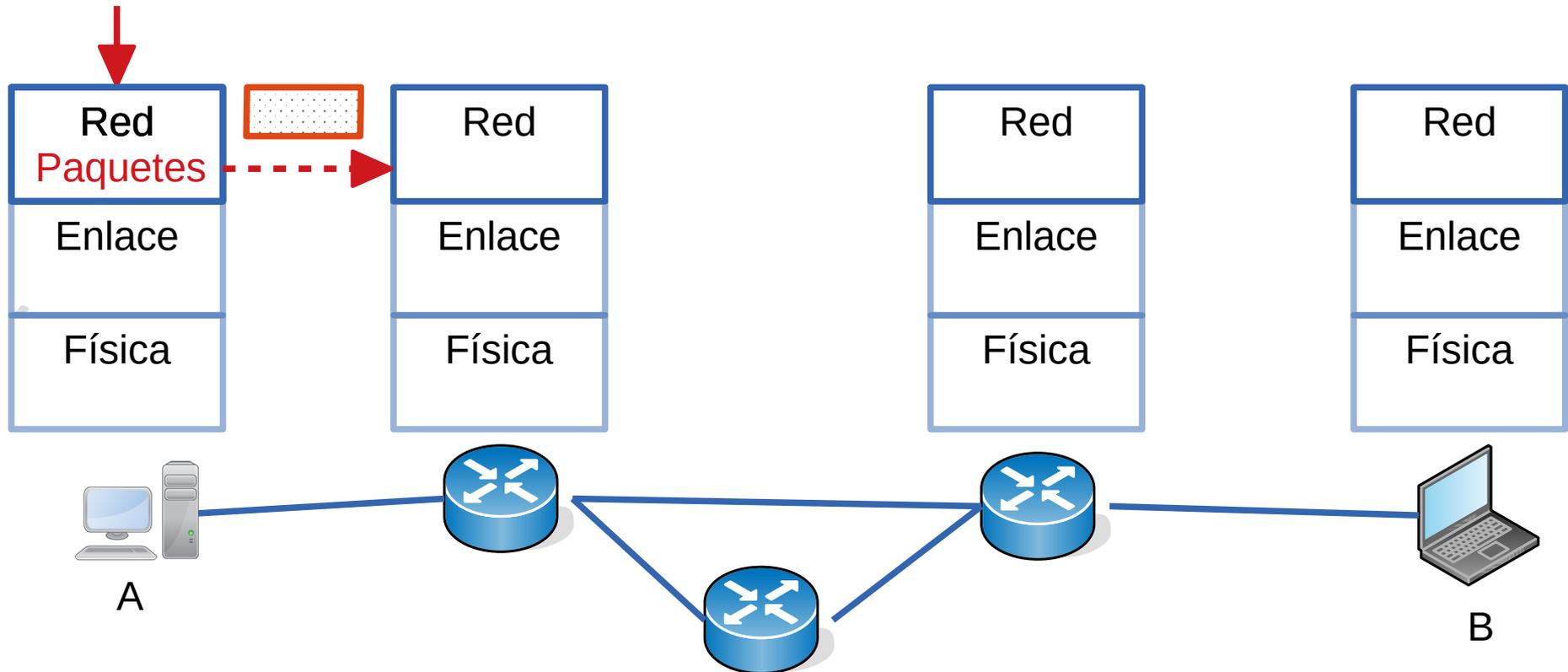
Datos de  
capa superior



# Capa 3: Capa de red

- A este nivel de abstracción podemos ver un intercambio de paquetes entre nodos
- Nos **independizamos** de lo que pasa en las capas inferiores

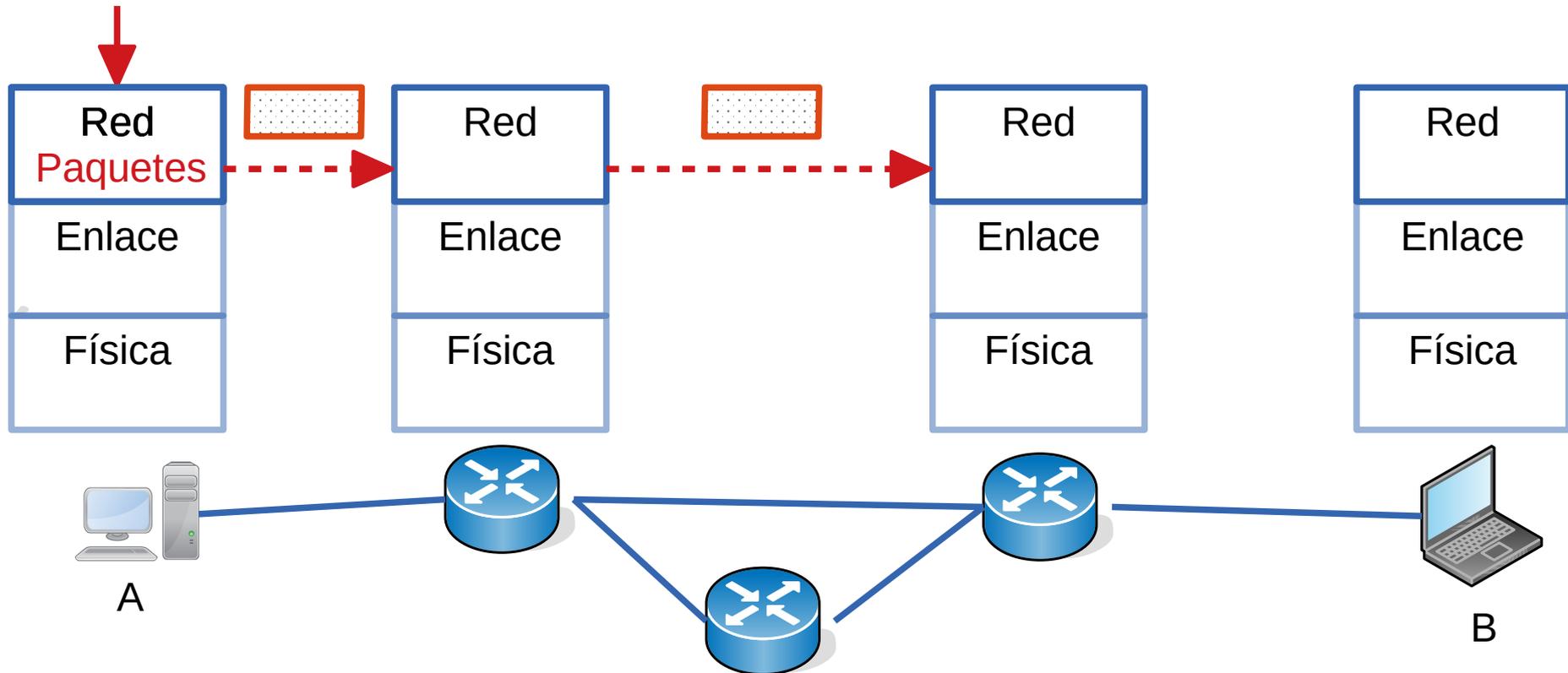
Datos de  
capa superior



# Capa 3: Capa de red

- A este nivel de abstracción podemos ver un intercambio de paquetes entre nodos
- Nos **independizamos** de lo que pasa en las capas inferiores

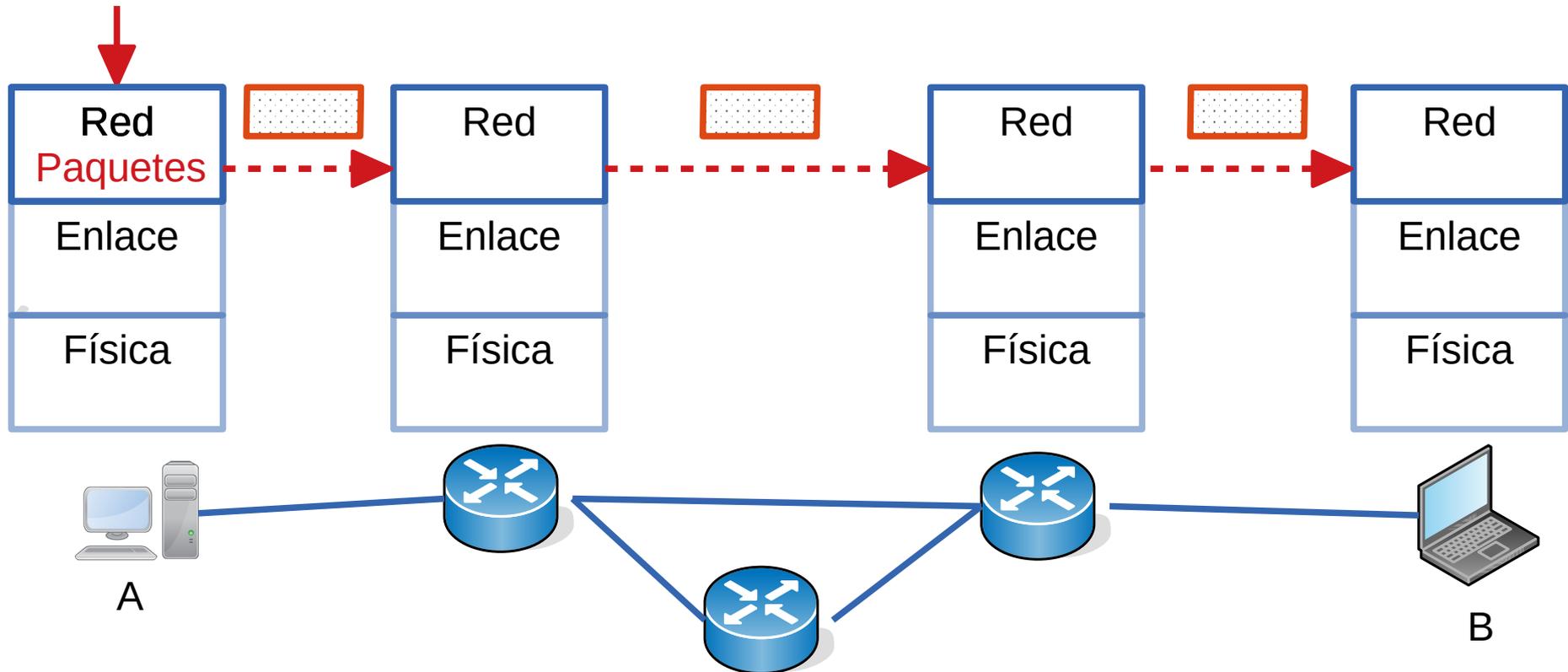
Datos de capa superior



# Capa 3: Capa de red

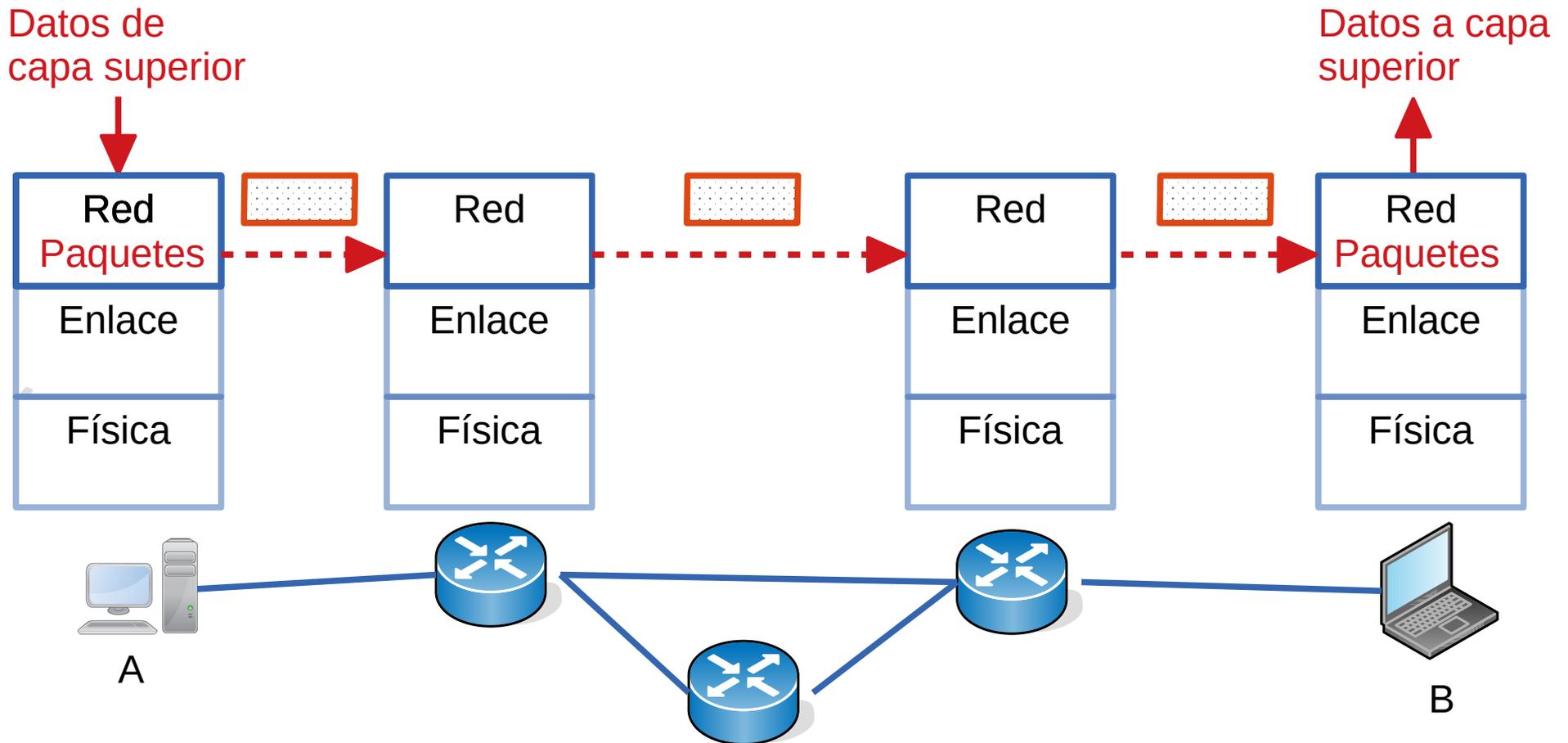
- A este nivel de abstracción podemos ver un intercambio de paquetes entre nodos
- Nos **independizamos** de lo que pasa en las capas inferiores

Datos de  
capa superior



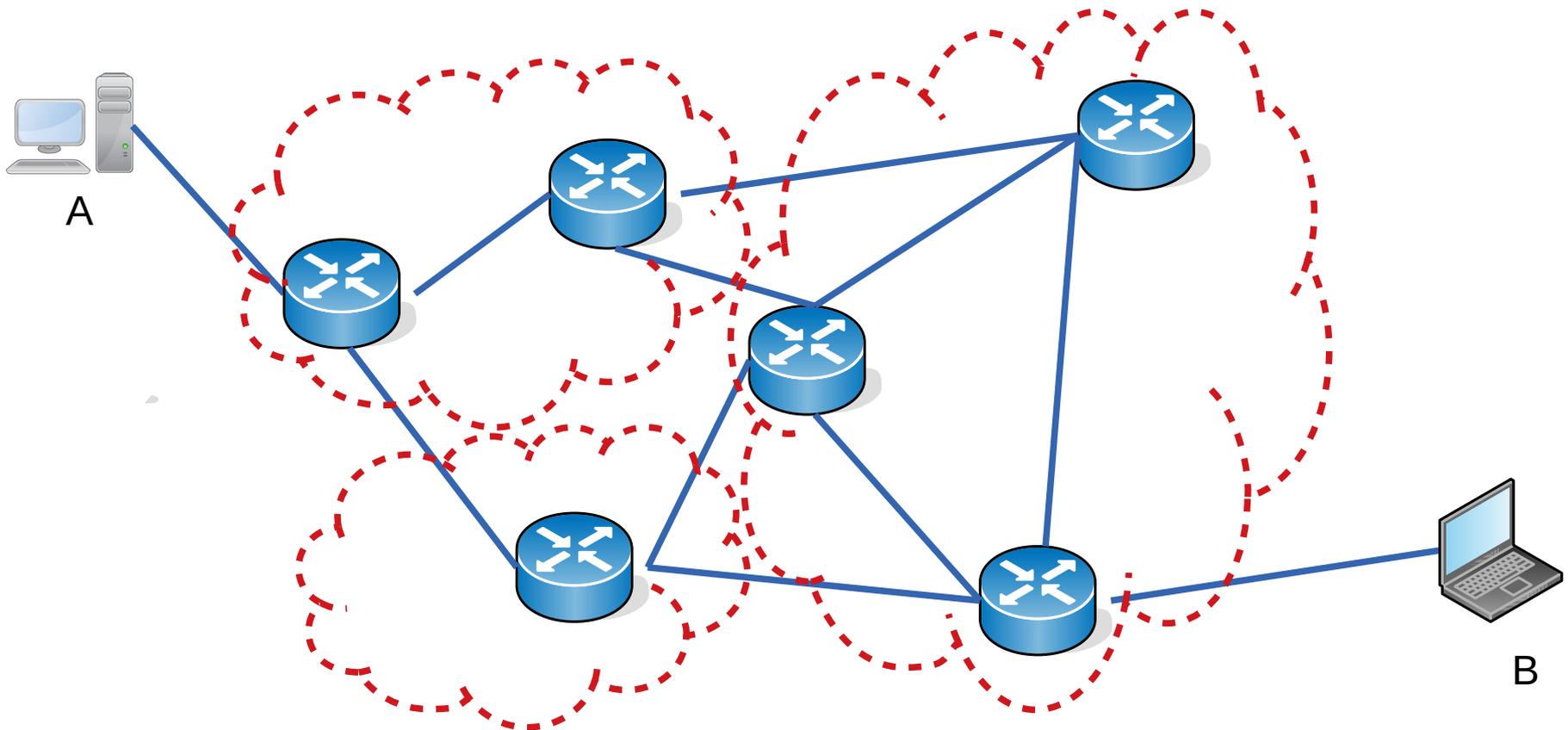
# Capa 3: Capa de red

- A este nivel de abstracción podemos ver un intercambio de paquetes entre nodos
- Nos **independizamos** de lo que pasa en las capas inferiores



# Capa 3: Capa de red

- La capa de red permite que los paquetes lleguen de A a B
- Pero ... en general, el camino pasa por **varios proveedores** que gestionan partes de la red global
- A y B no tienen **control** de lo que pasa en el camino (por ejemplo caídas en nodos o enlaces)

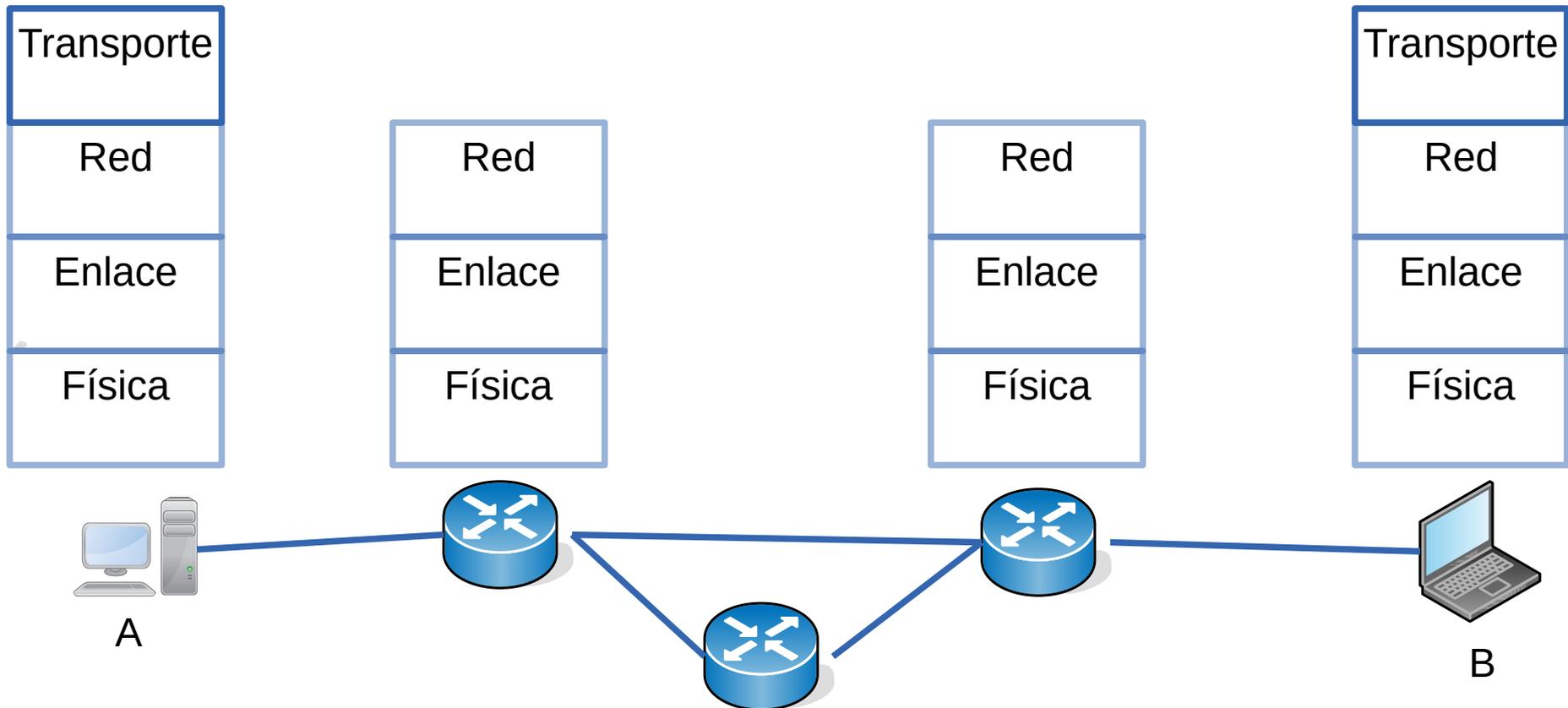


# Capa 4: Capa de transporte

- **Objetivo principal:**
  - Brindar una comunicación bajo control de los extremos A y B
- Es la primer capa del modelo donde el **diálogo virtual es entre A y B**
  - Esto permite **independizarse de lo que pasa en la red** y eventualmente tomar acciones si en la red subyacente hay problemas
- Organiza la información de la capa superior en unidades llamadas **segmentos**
- Ofrece diferentes tipos de servicio a la capa superior
  - Servicios orientados a conexión y no orientados a conexión
    - Se requiere o no un establecimiento antes de enviar datos
  - Servicios confiables y no confiables
    - Se garantiza que los datos llegan correctamente al receptor
- Se necesita identificar el servicio o aplicación en el equipo local y en el remoto
- Control de flujo
- Control de congestión (cuando no se puede hacer en capa de red)

# Capa 4: Capa de transporte

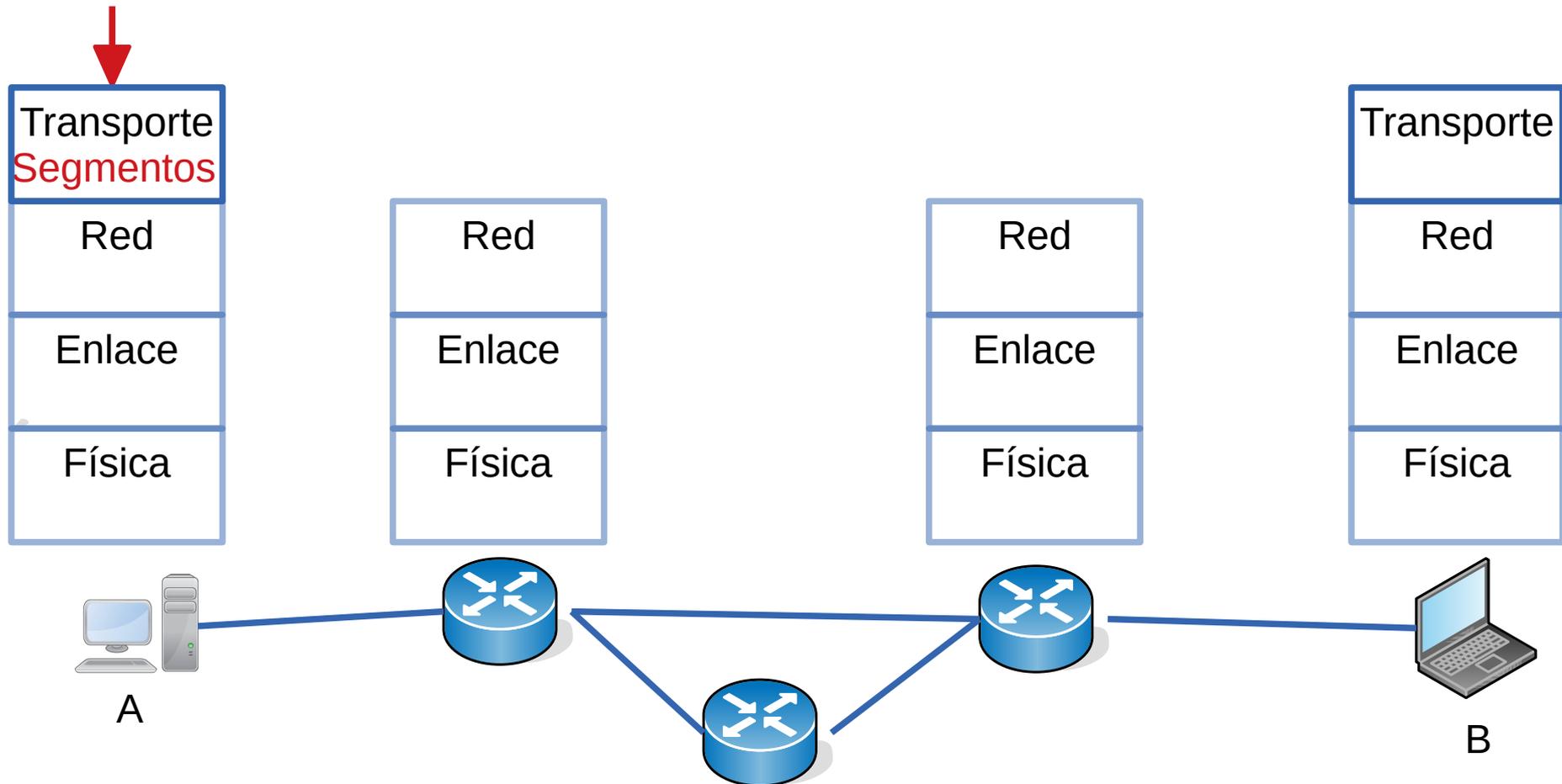
- A este nivel de abstracción lo vemos como un intercambio de segmentos entre A y B



# Capa 4: Capa de transporte

- A este nivel de abstracción lo vemos como un intercambio de segmentos entre A y B

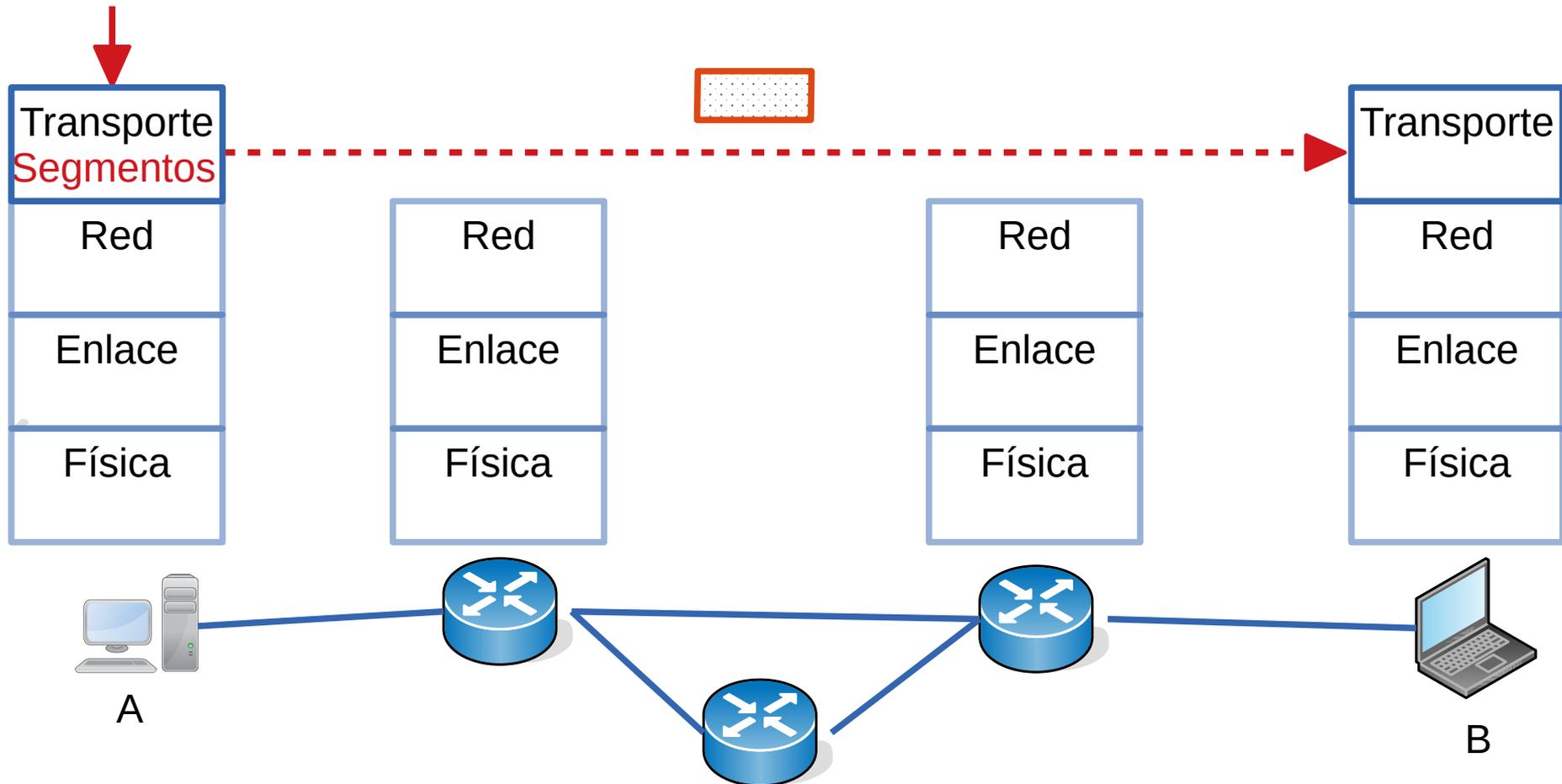
Datos de capa superior



# Capa 4: Capa de transporte

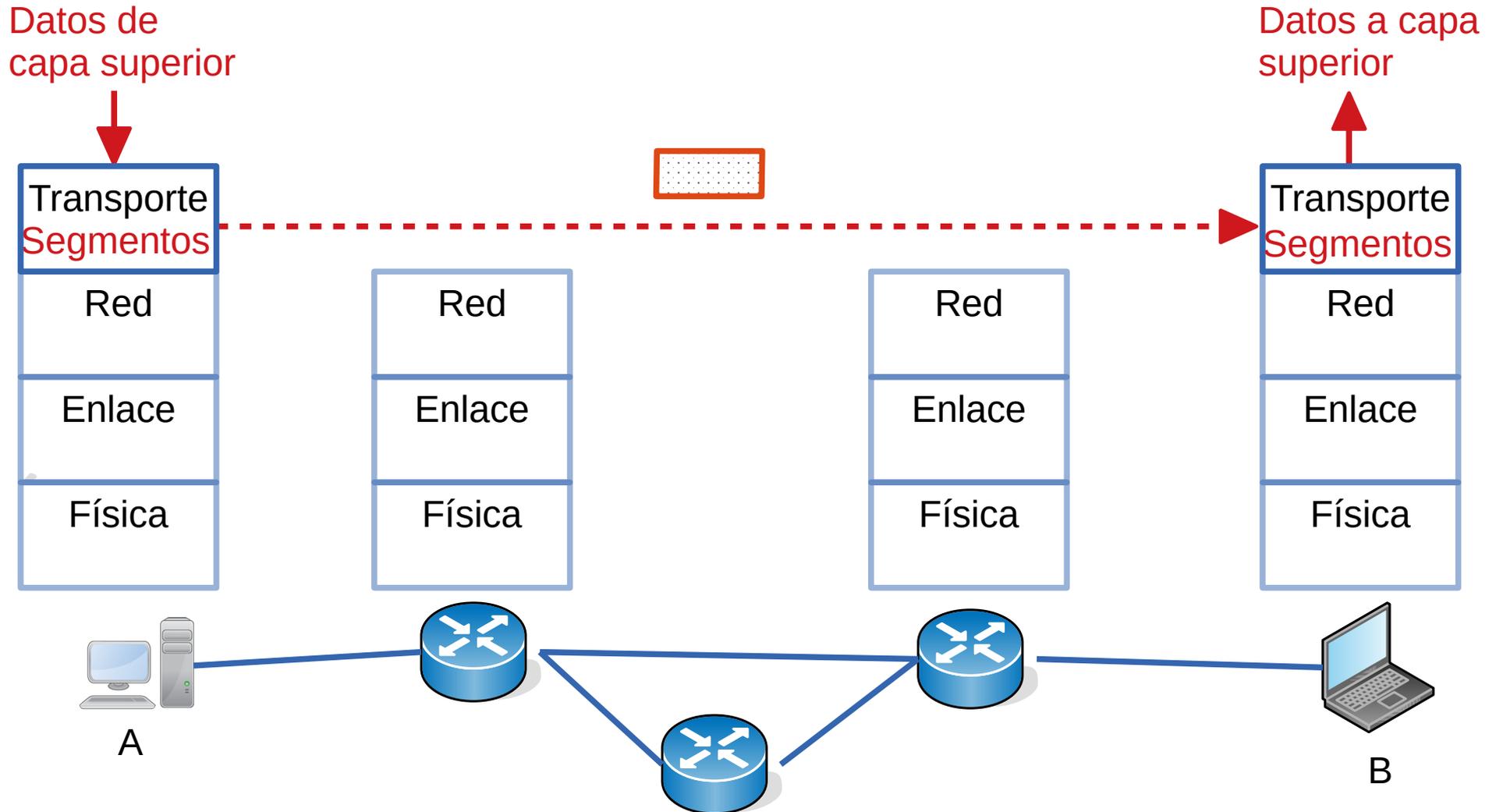
- A este nivel de abstracción lo vemos como un intercambio de segmentos entre A y B

Datos de capa superior



# Capa 4: Capa de transporte

- A este nivel de abstracción lo vemos como un intercambio de segmentos entre A y B

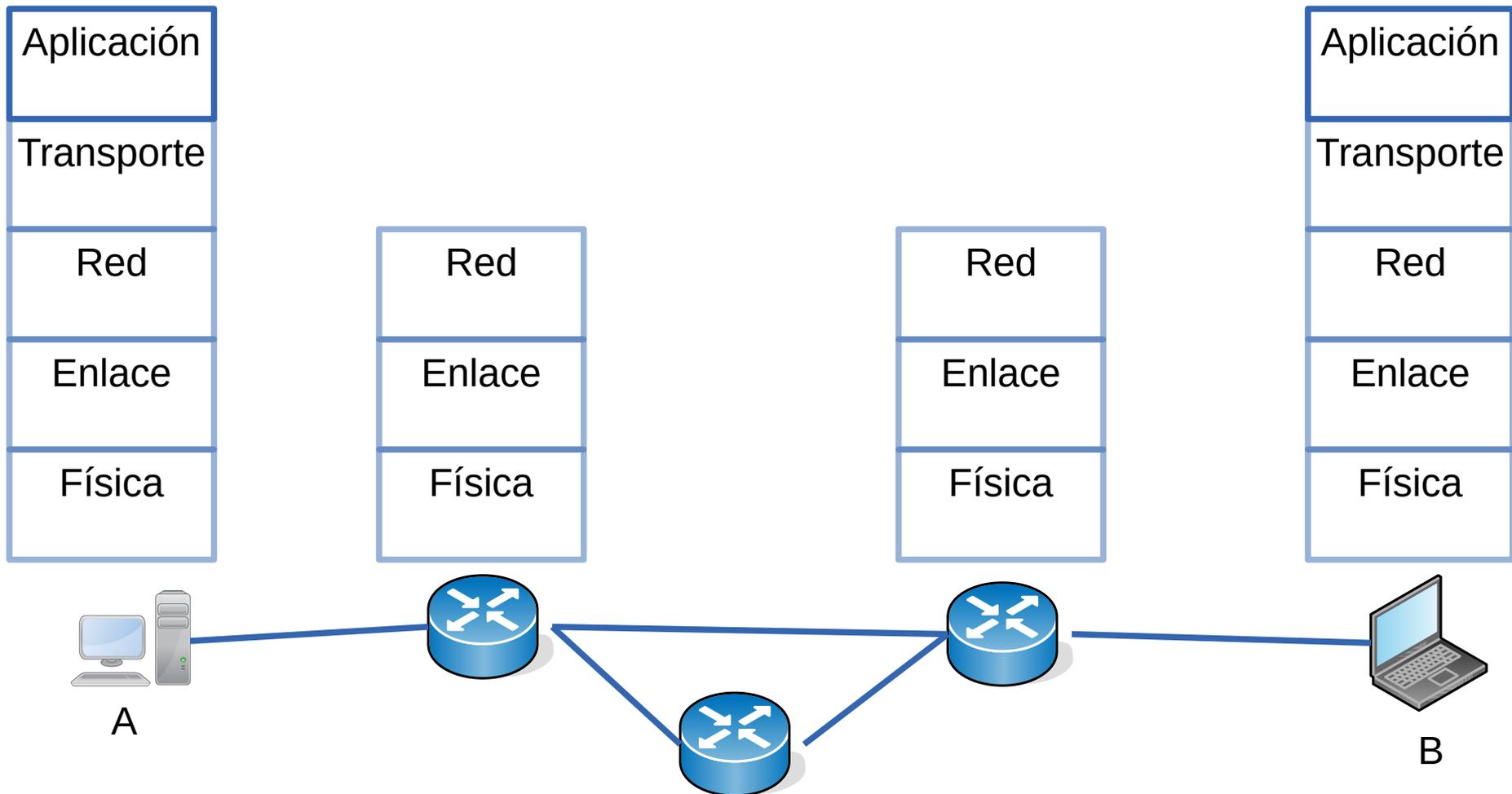


# Capa 5: Capa de aplicación

- **Objetivo** principal:
  - Ofrecer aplicaciones al **usuario**
- Las unidades de datos las llamaremos **mensajes**
- Ejemplos de aplicaciones:
  - Correo electrónico (Thunderbird, Outlook, etc)
  - Chat y voz (Whatsapp, Hangout, Skype)
  - Video conferencias
  - Navegadores (Firefox, Chrome, Safari, Opera, Edge, etc)
  - Compartir información (Google Drive, One, Owncloud)
  - Trabajo a distancia (ssh, Escritorio Remoto, VNC, etc)
  - Educación
  - Entretenimiento (Youtube, Netflix, Juegos en línea)
  - Redes sociales (Facebook, Instagram, etc)

# Capa 5: Capa de aplicación

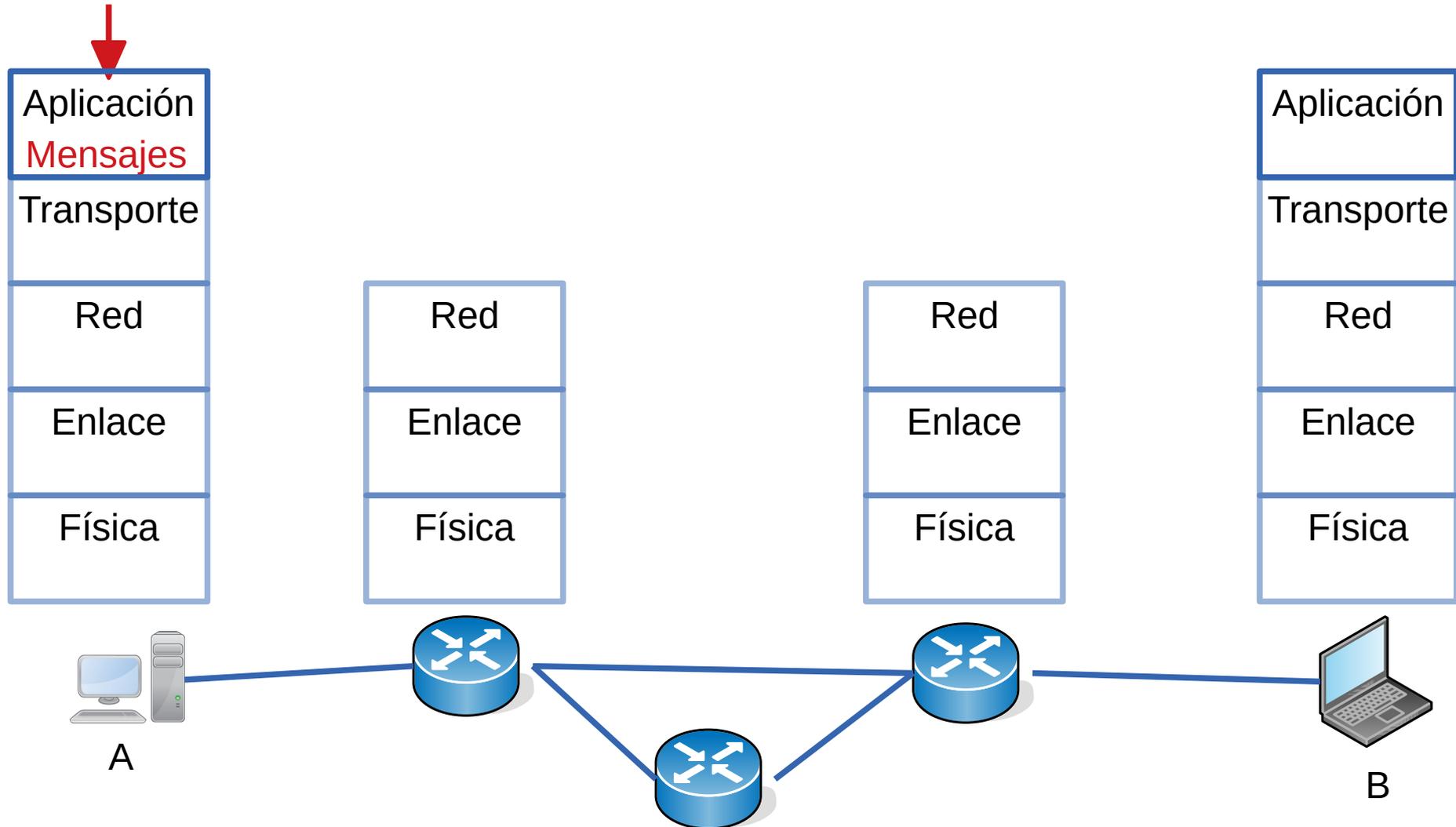
- Las aplicaciones en A y B lo ven como un intercambio directo de mensajes



# Capa 5: Capa de aplicación

- Las aplicaciones en A y B lo ven como un intercambio directo de mensajes

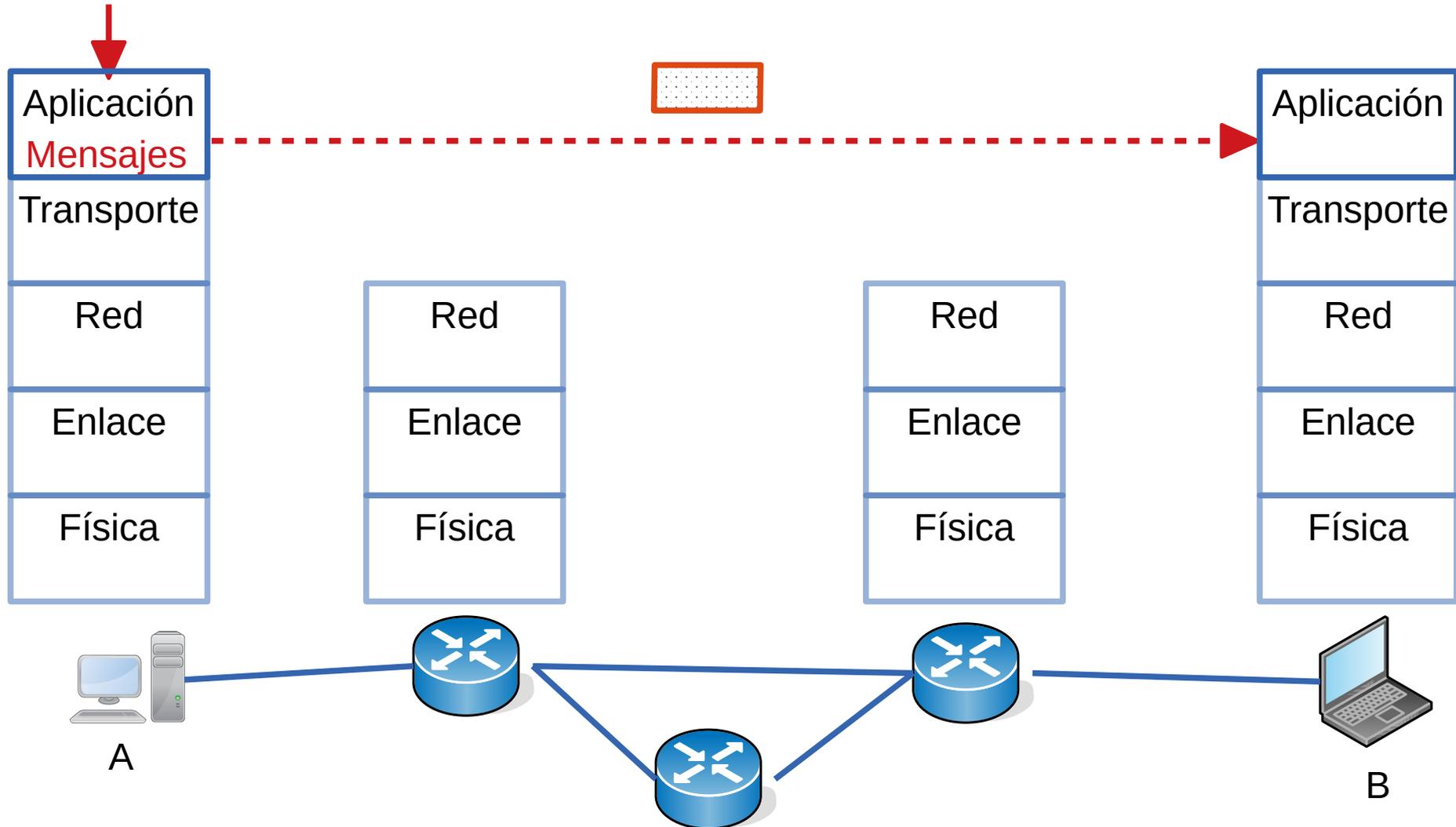
Datos del usuario en A



# Capa 5: Capa de aplicación

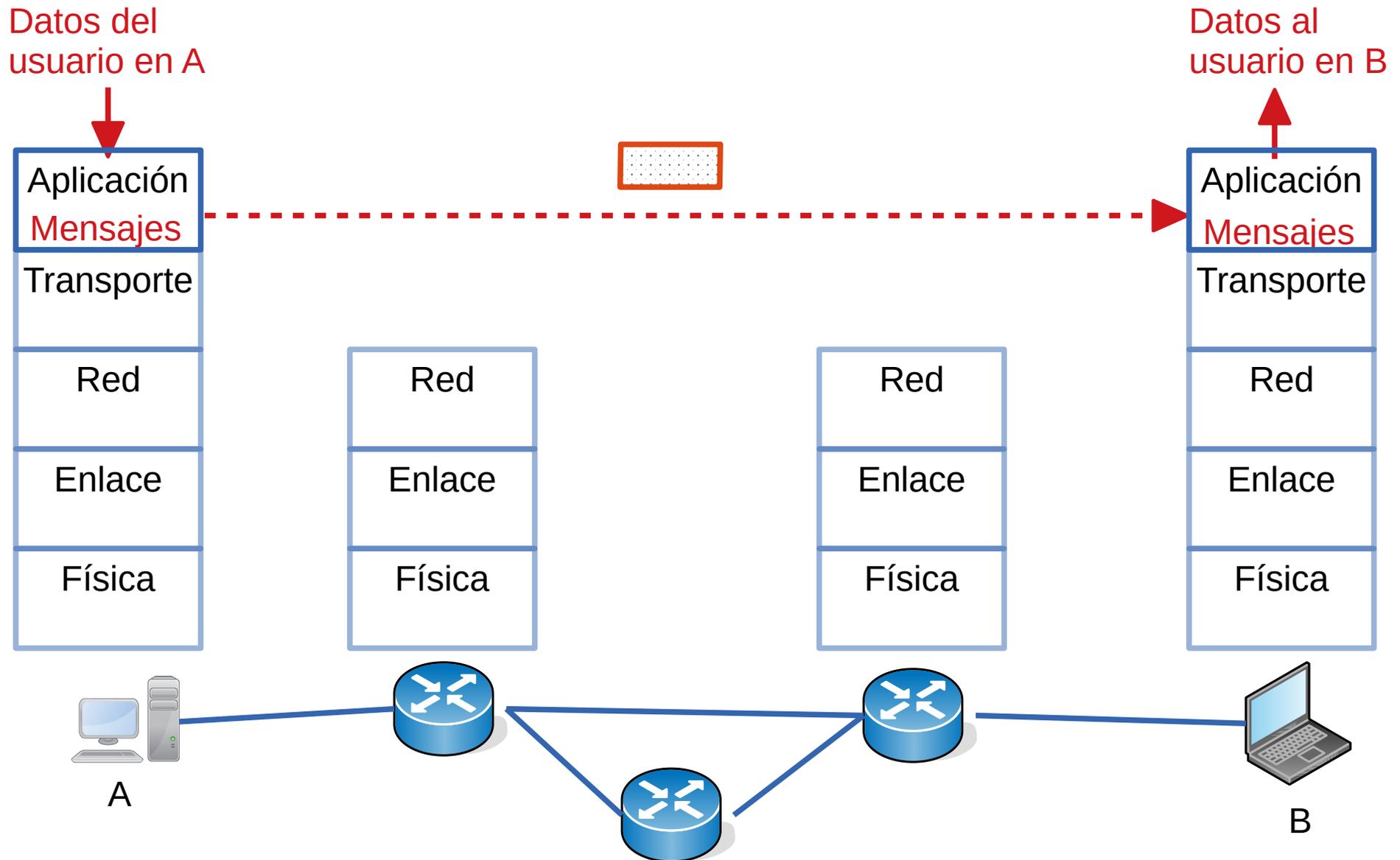
- Las aplicaciones en A y B lo ven como un intercambio directo de mensajes

Datos del usuario en A



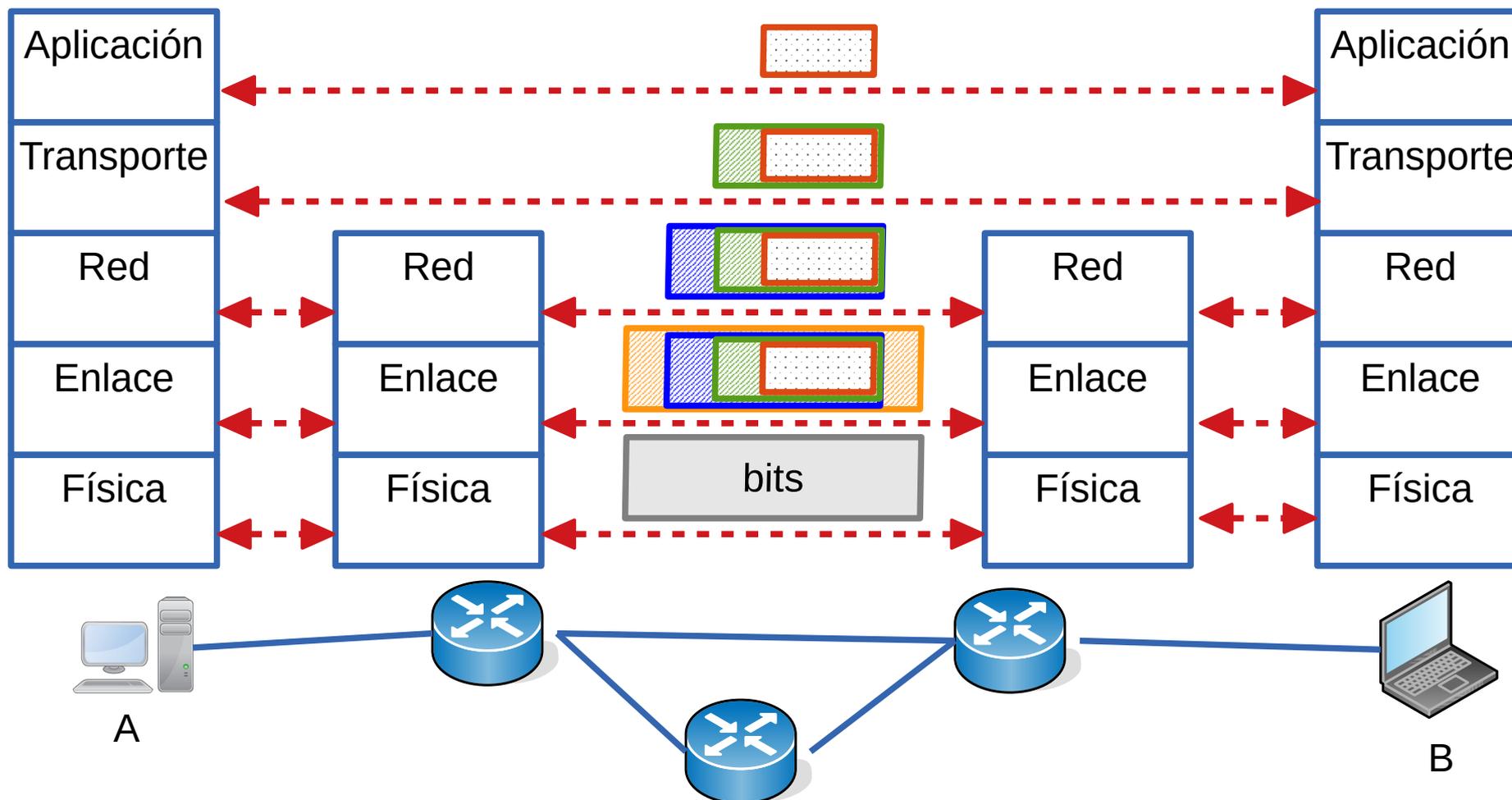
# Capa 5: Capa de aplicación

- Las aplicaciones en A y B lo ven como un intercambio directo de mensajes

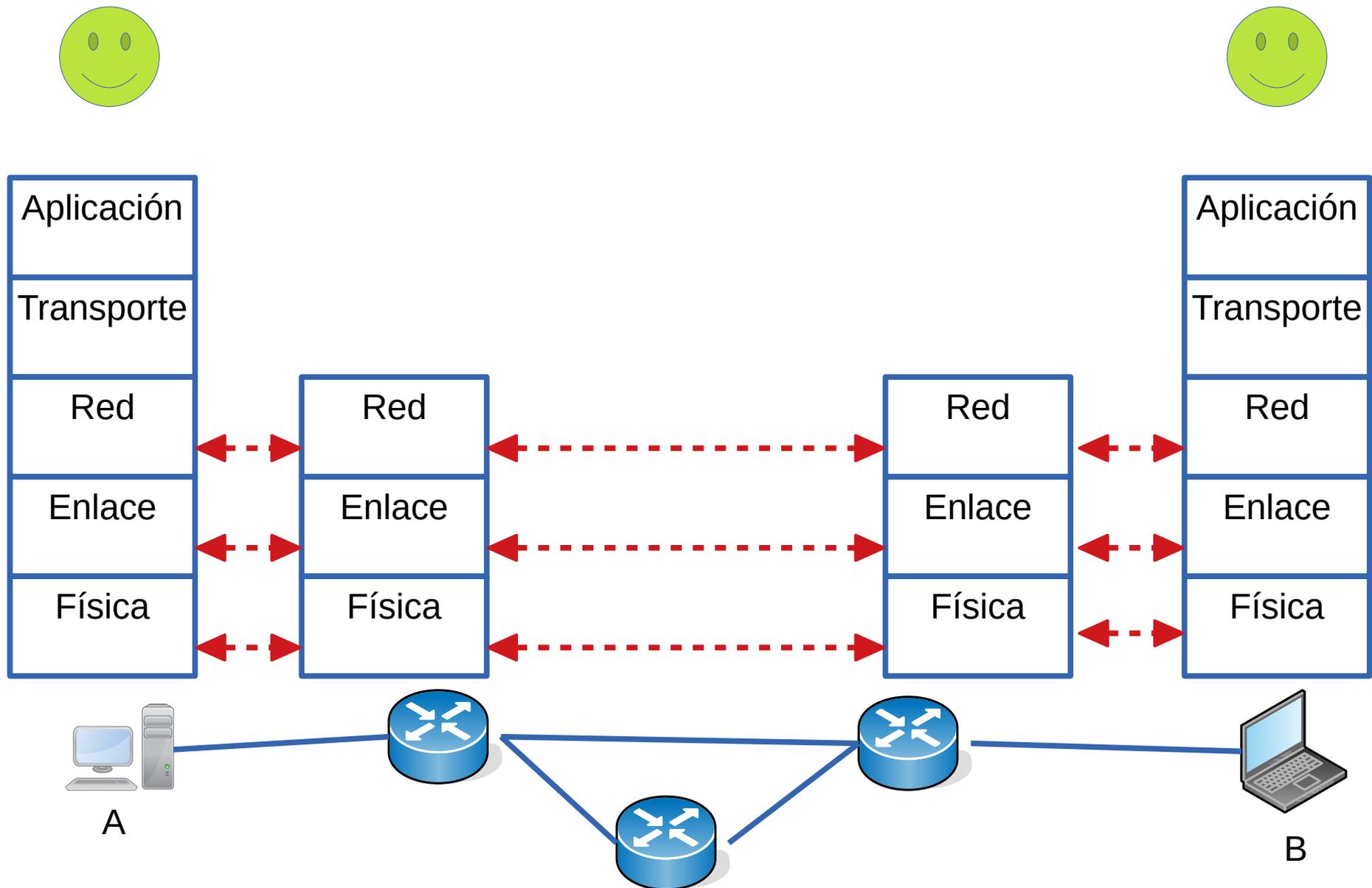


# Encapsulamiento o anidamiento

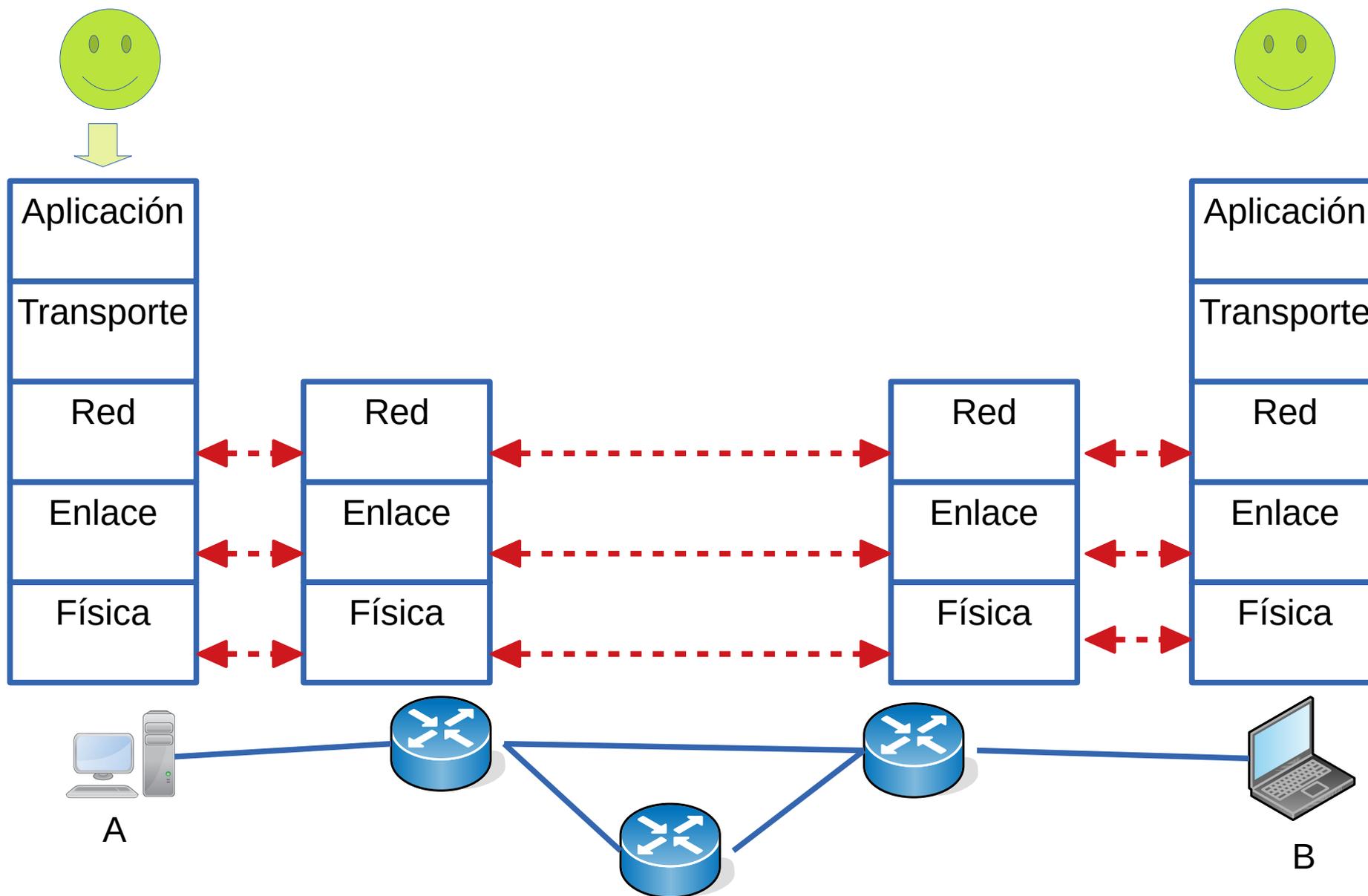
- Las unidades de una capa son la carga útil de la capa inferior
- Los encabezados pueden estar antes y/o después de la carga útil
- Sobrecarga de los protocolos: **cada capa agrega información propia**



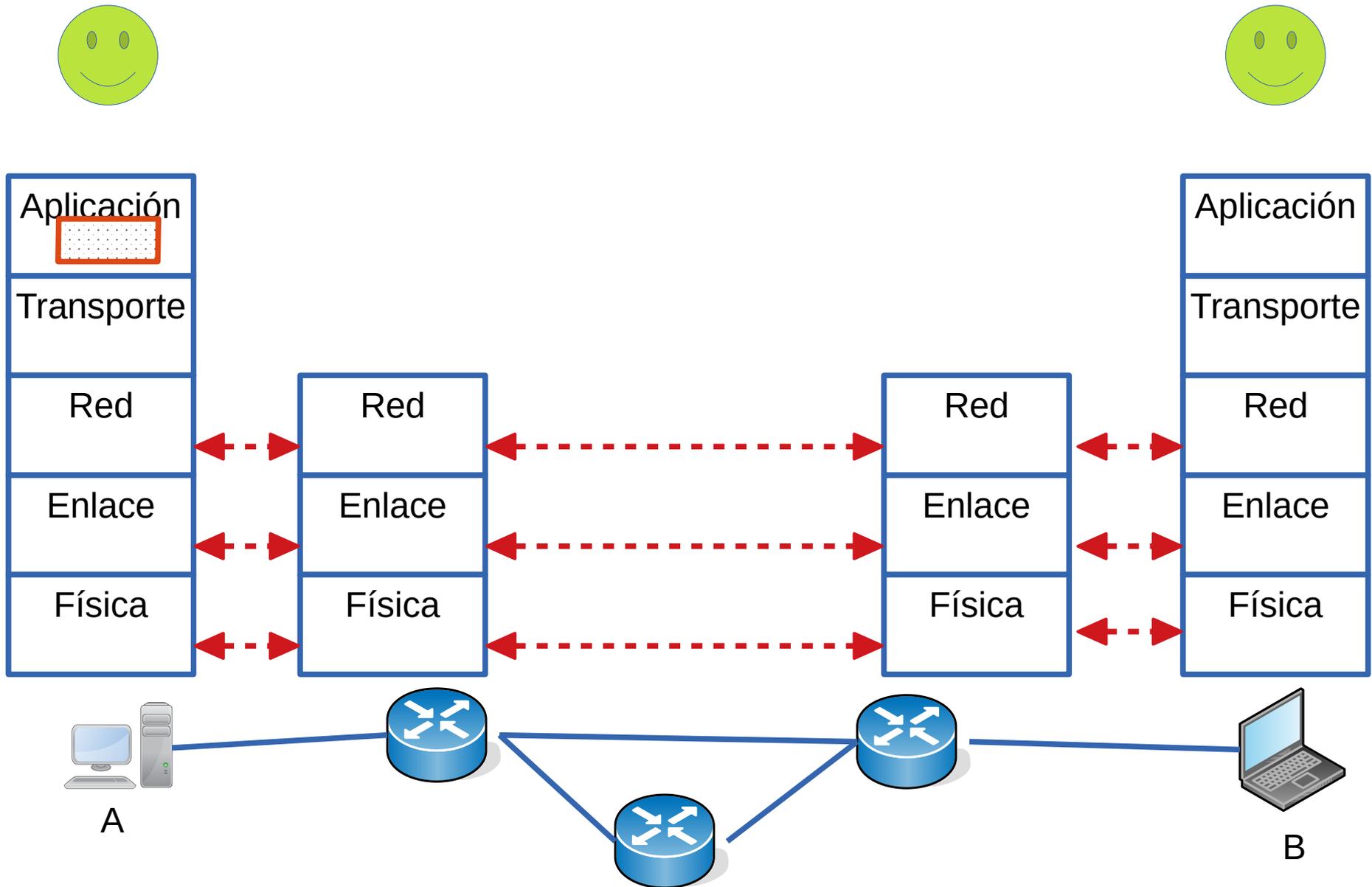
# Encapsulamiento y niveles de abstracción



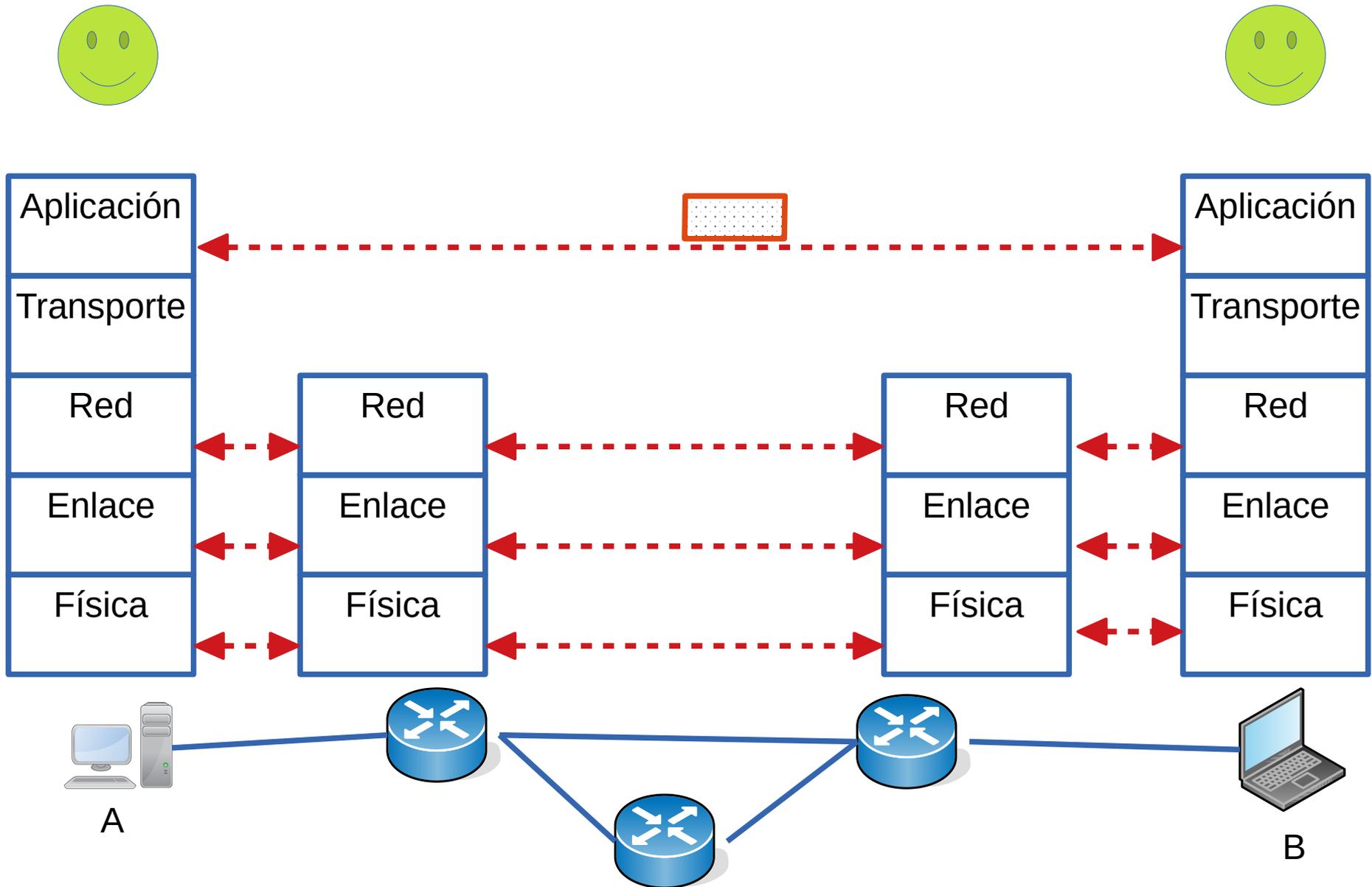
# Encapsulamiento y niveles de abstracción



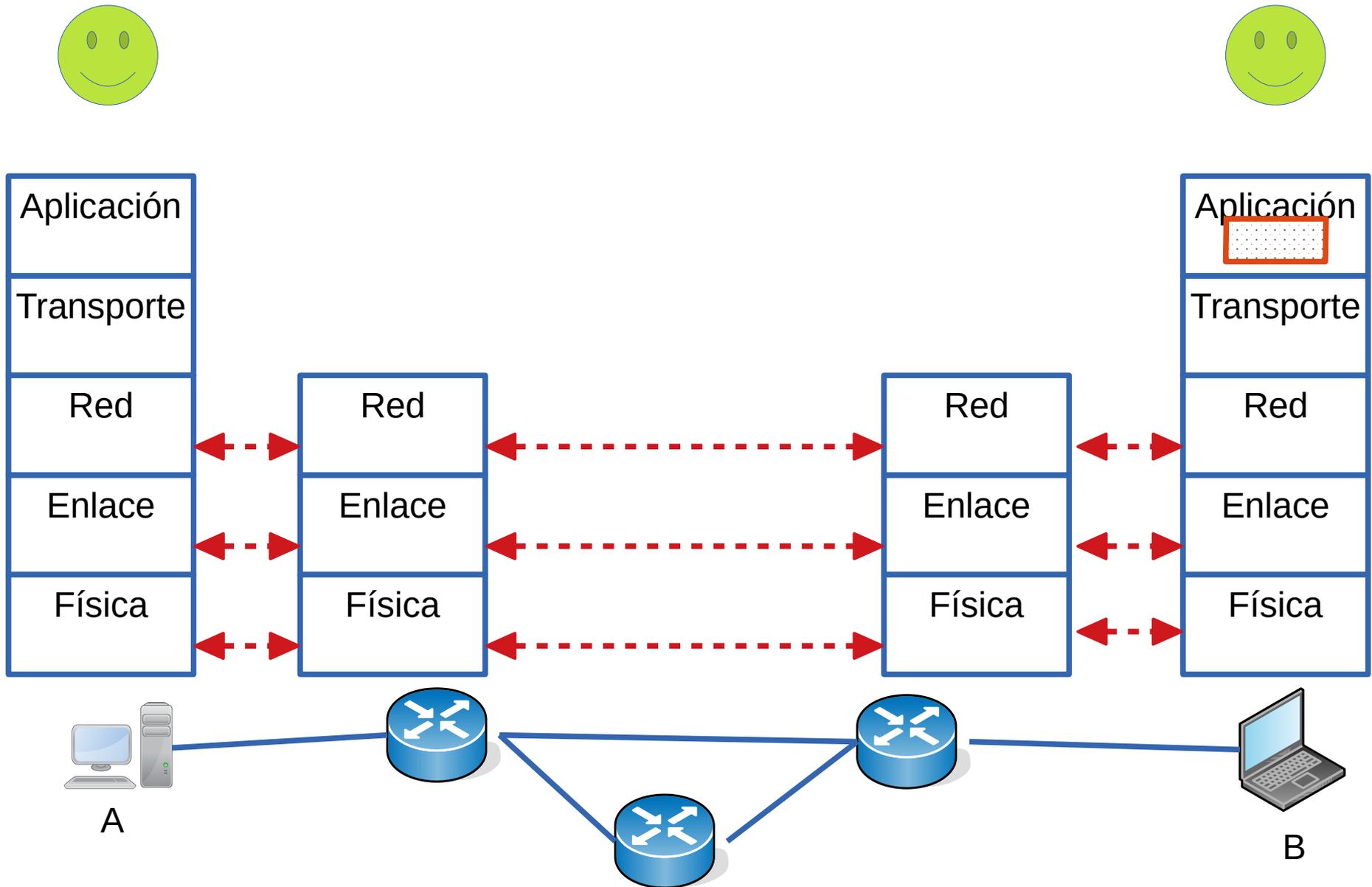
# Encapsulamiento y niveles de abstracción



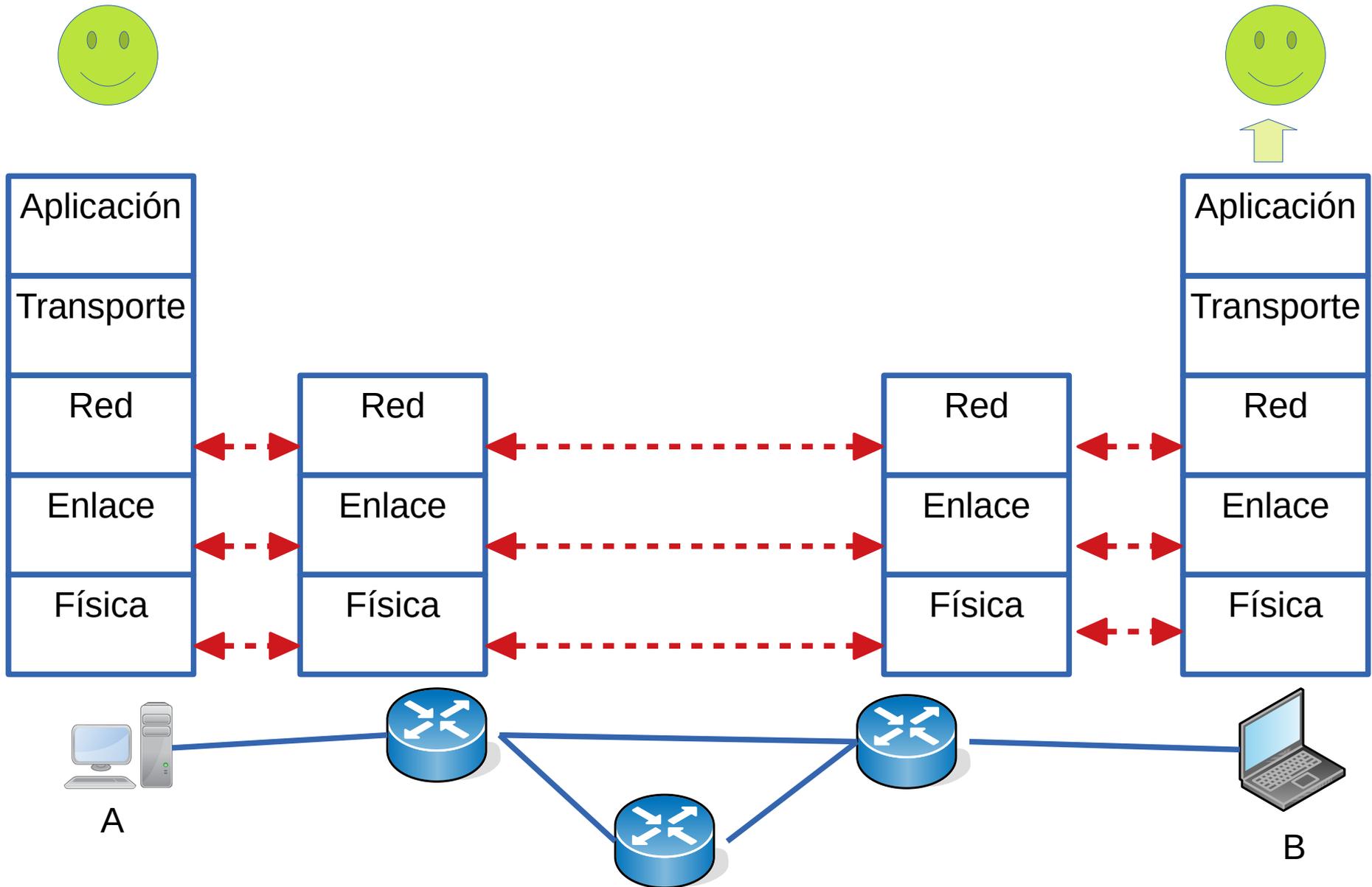
# Encapsulamiento y niveles de abstracción



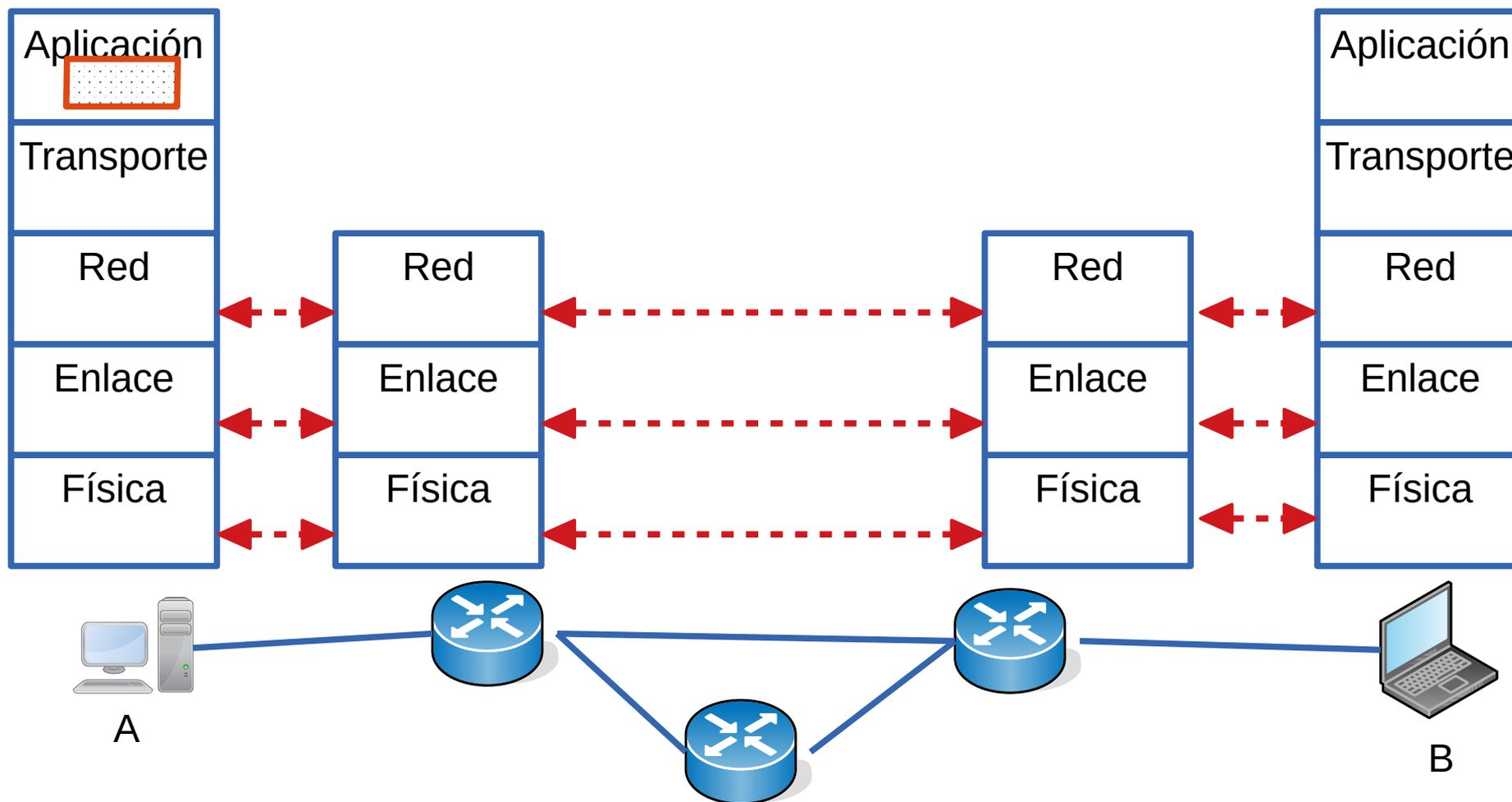
# Encapsulamiento y niveles de abstracción



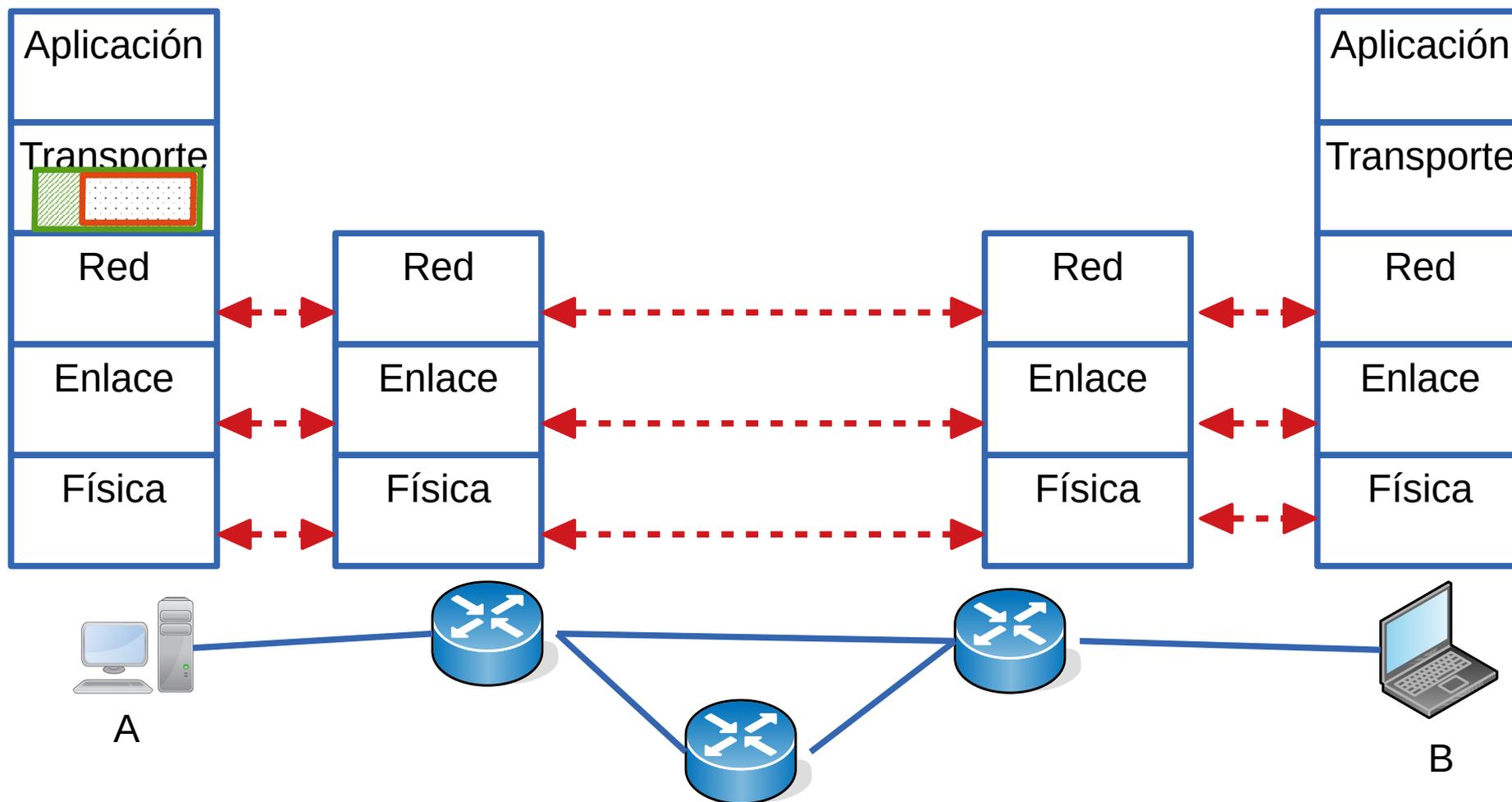
# Encapsulamiento y niveles de abstracción



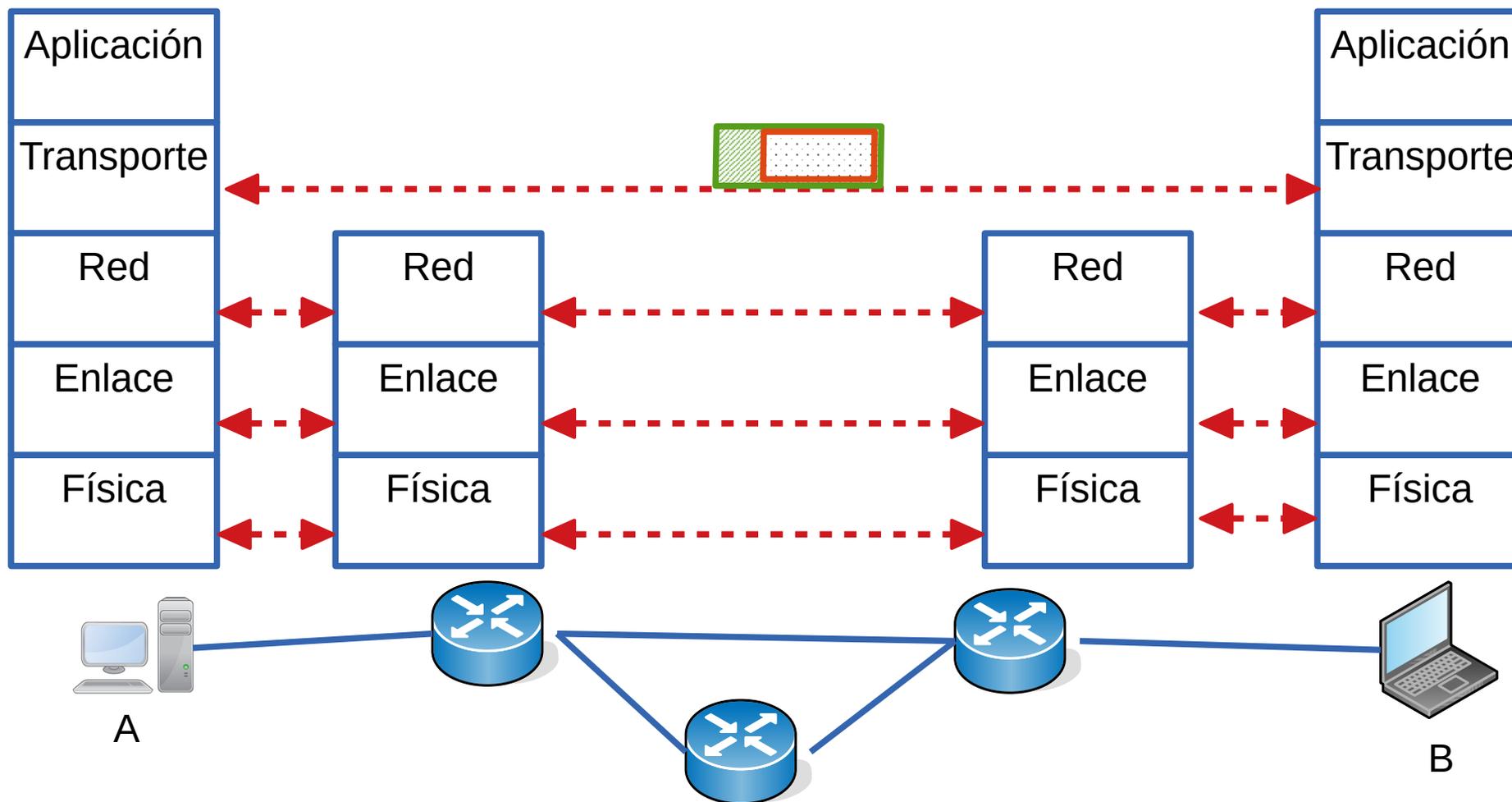
# Encapsulamiento y niveles de abstracción



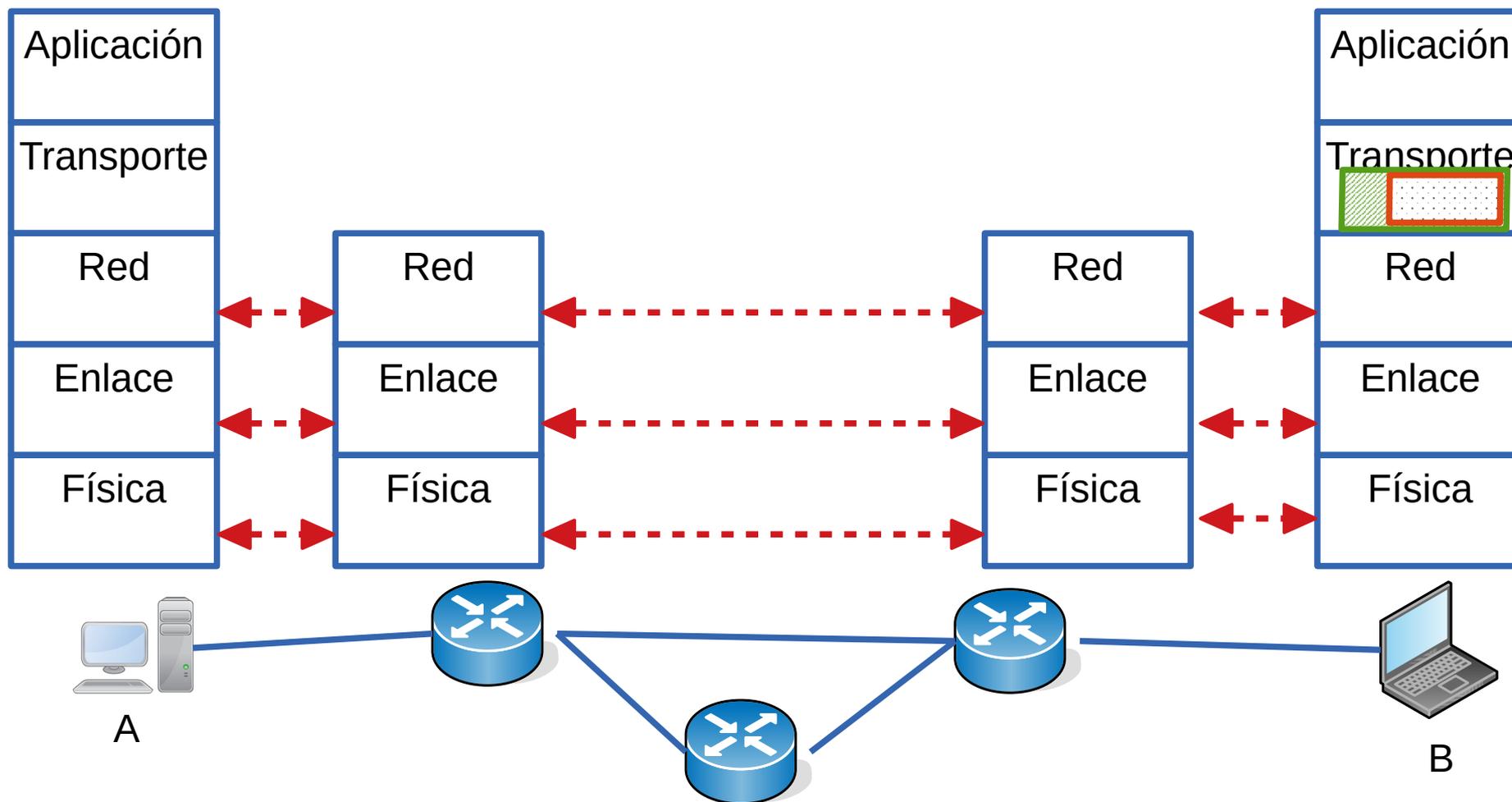
# Encapsulamiento y niveles de abstracción



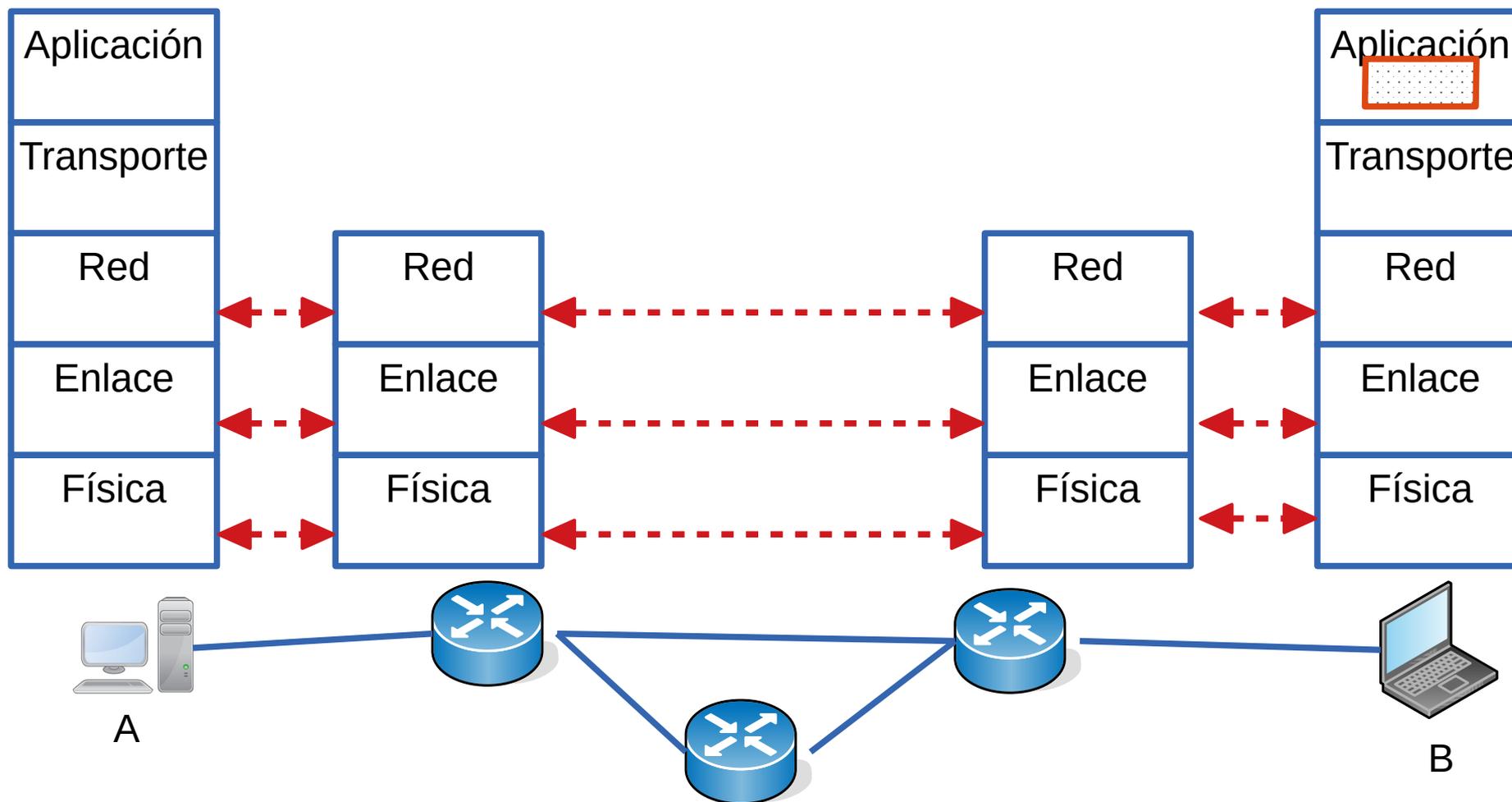
# Encapsulamiento y niveles de abstracción



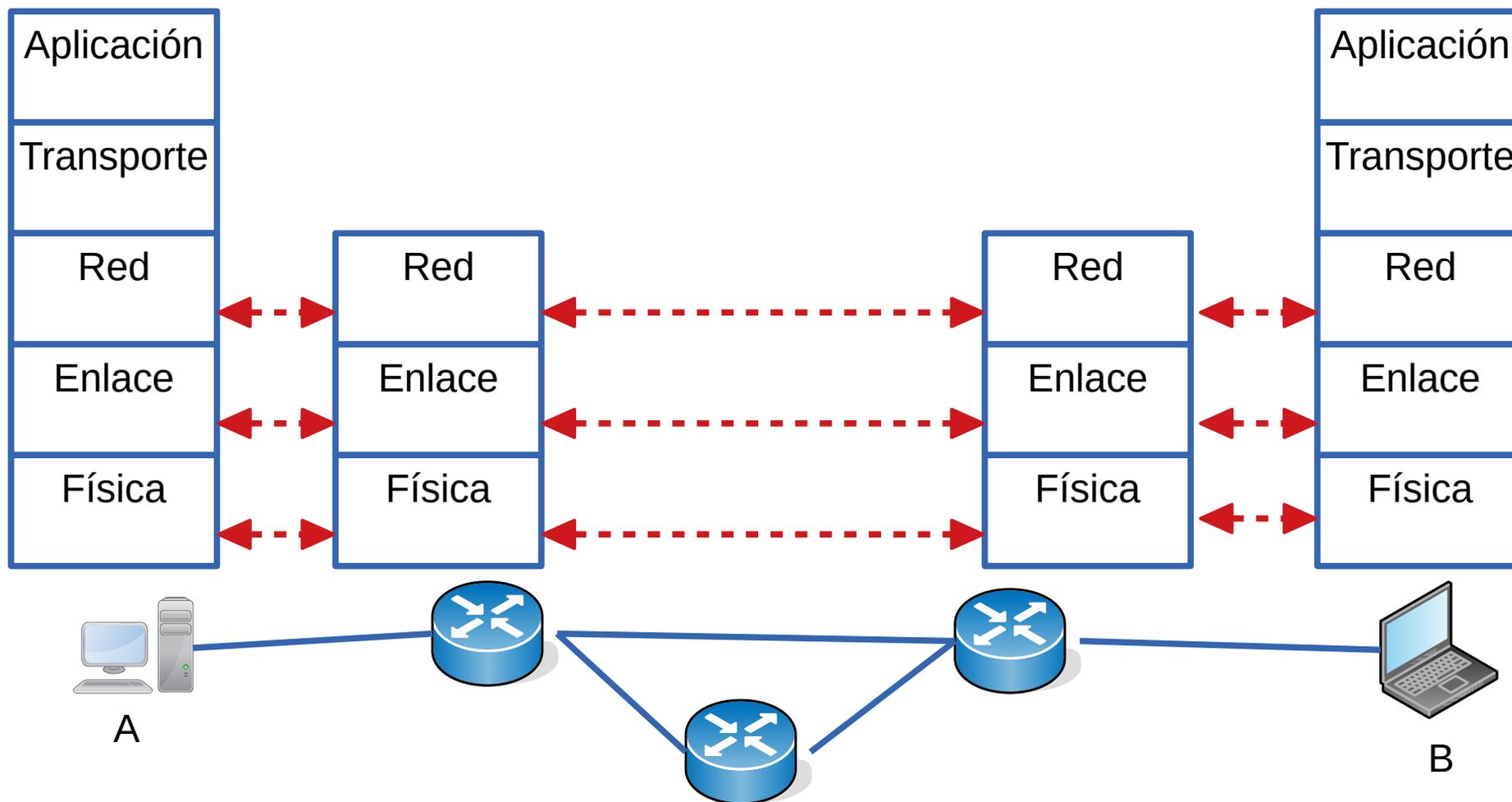
# Encapsulamiento y niveles de abstracción



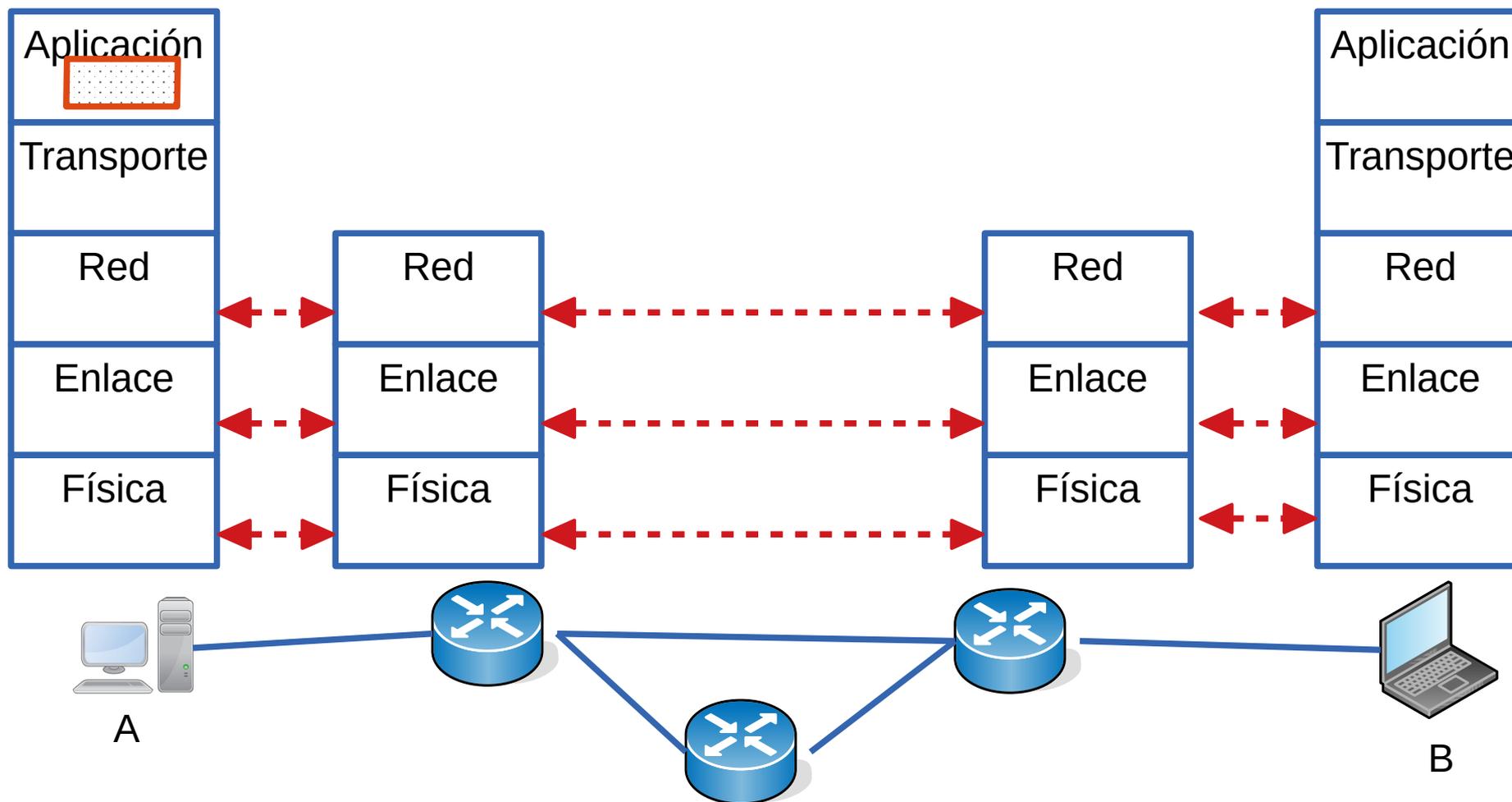
# Encapsulamiento y niveles de abstracción



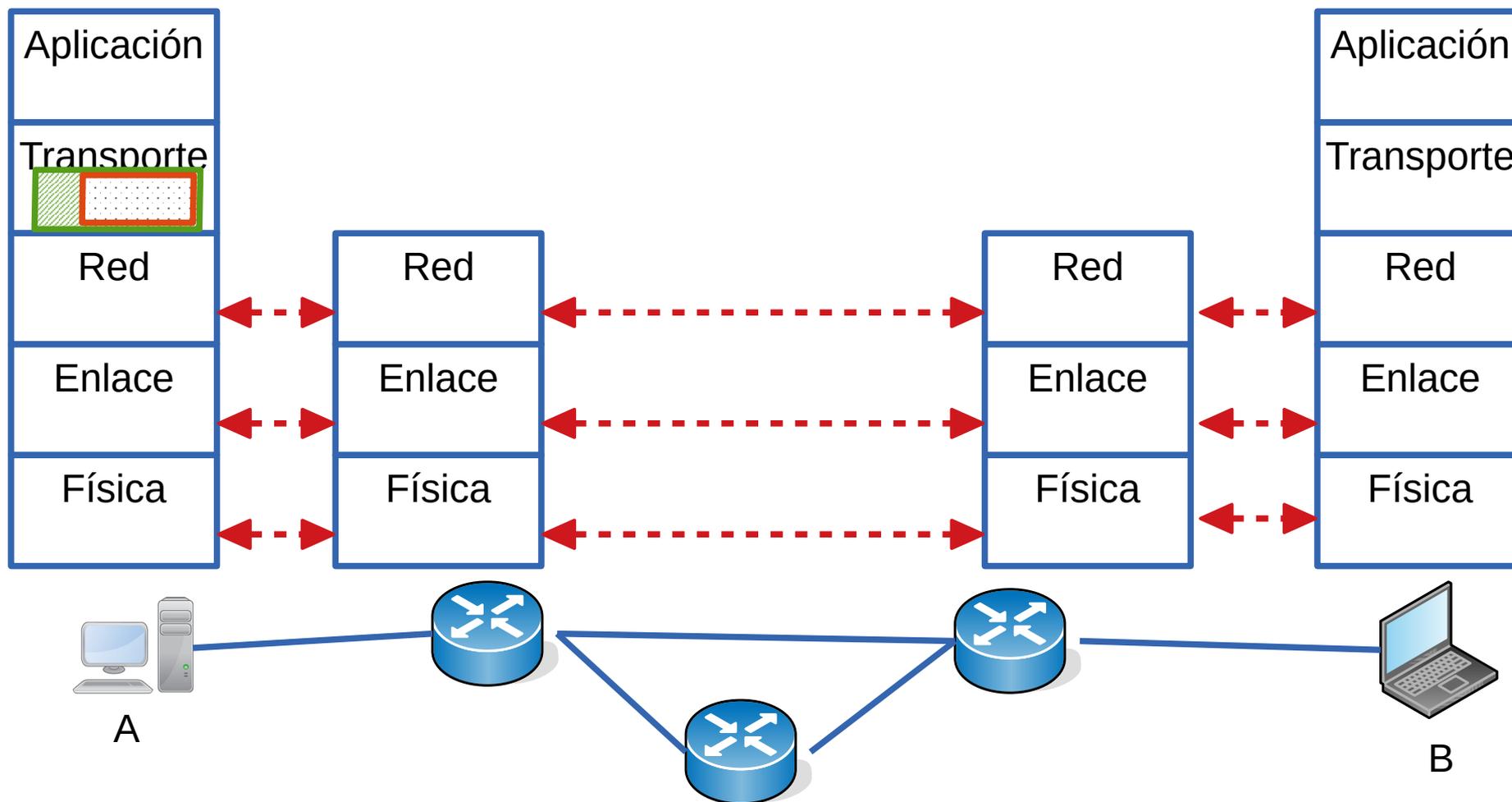
# Encapsulamiento y niveles de abstracción



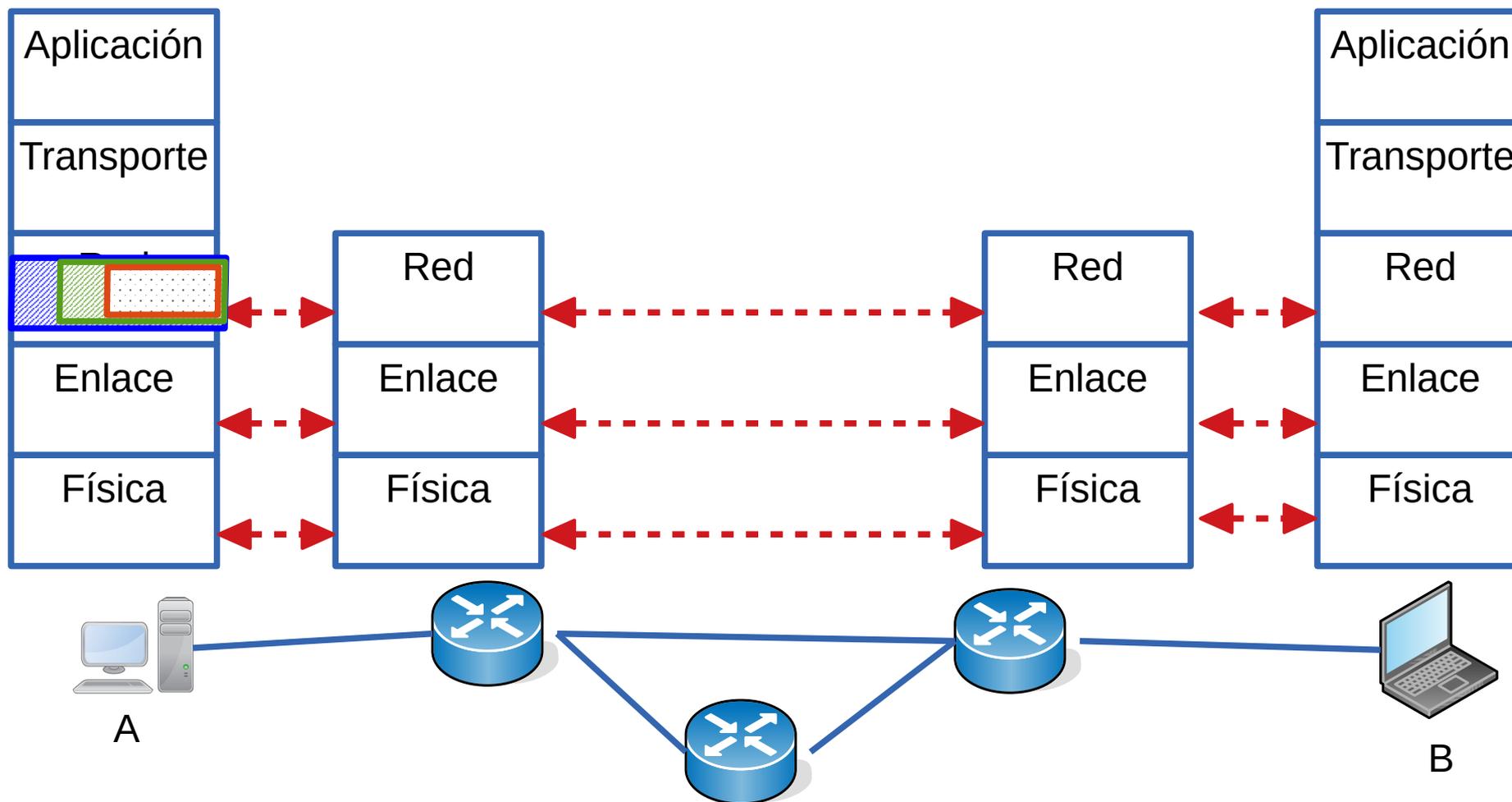
# Encapsulamiento y niveles de abstracción



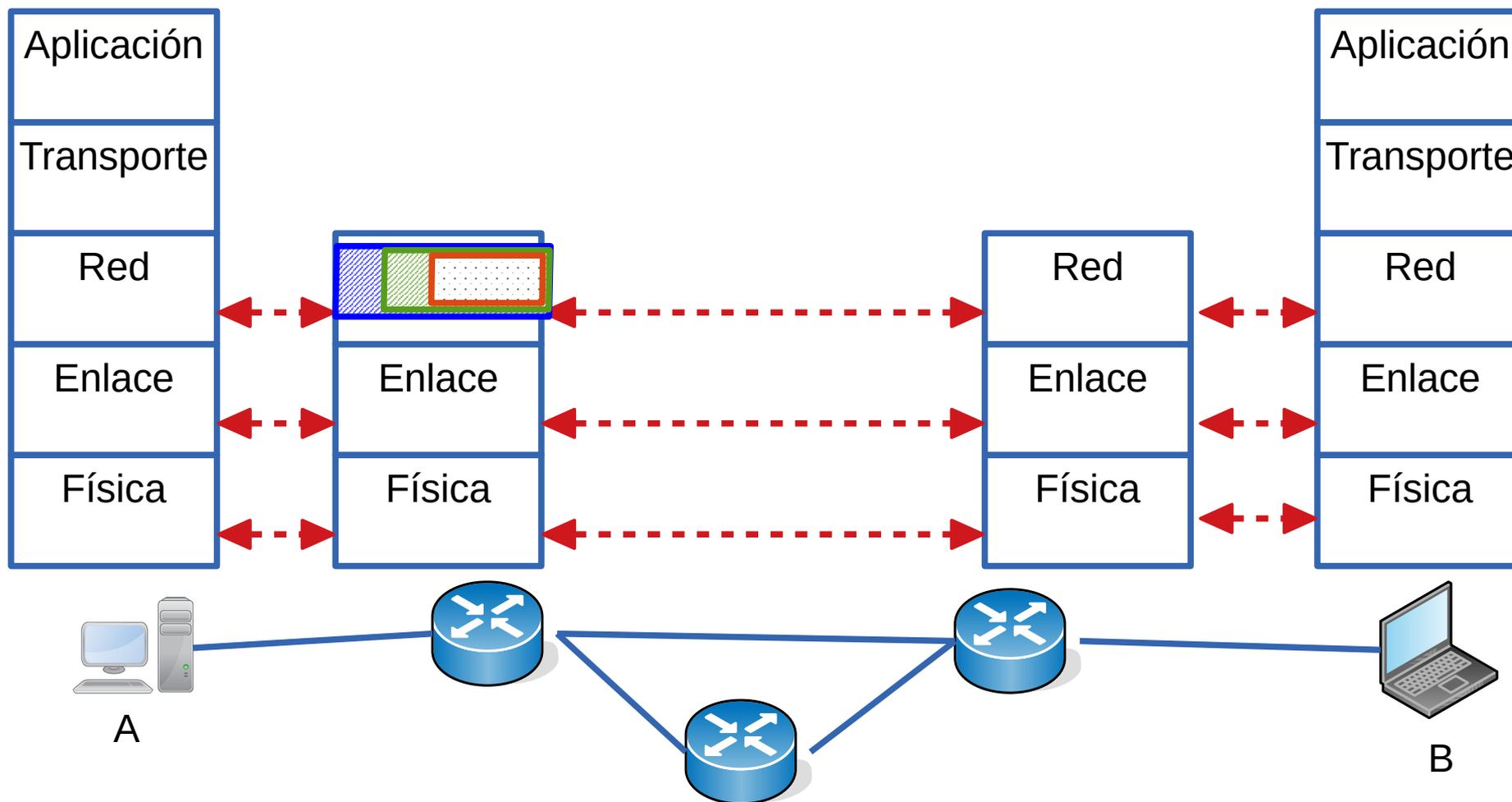
# Encapsulamiento y niveles de abstracción



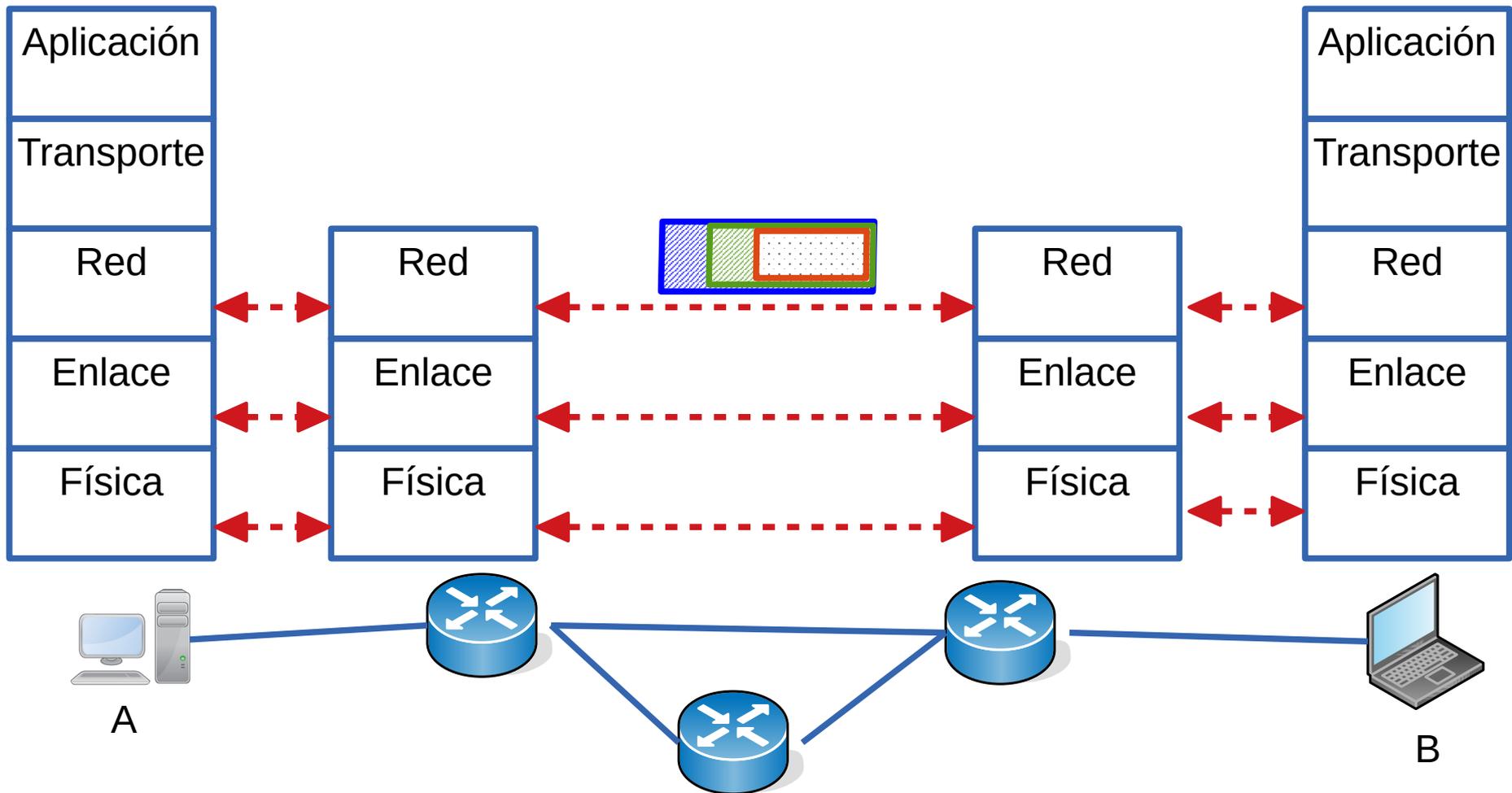
# Encapsulamiento y niveles de abstracción



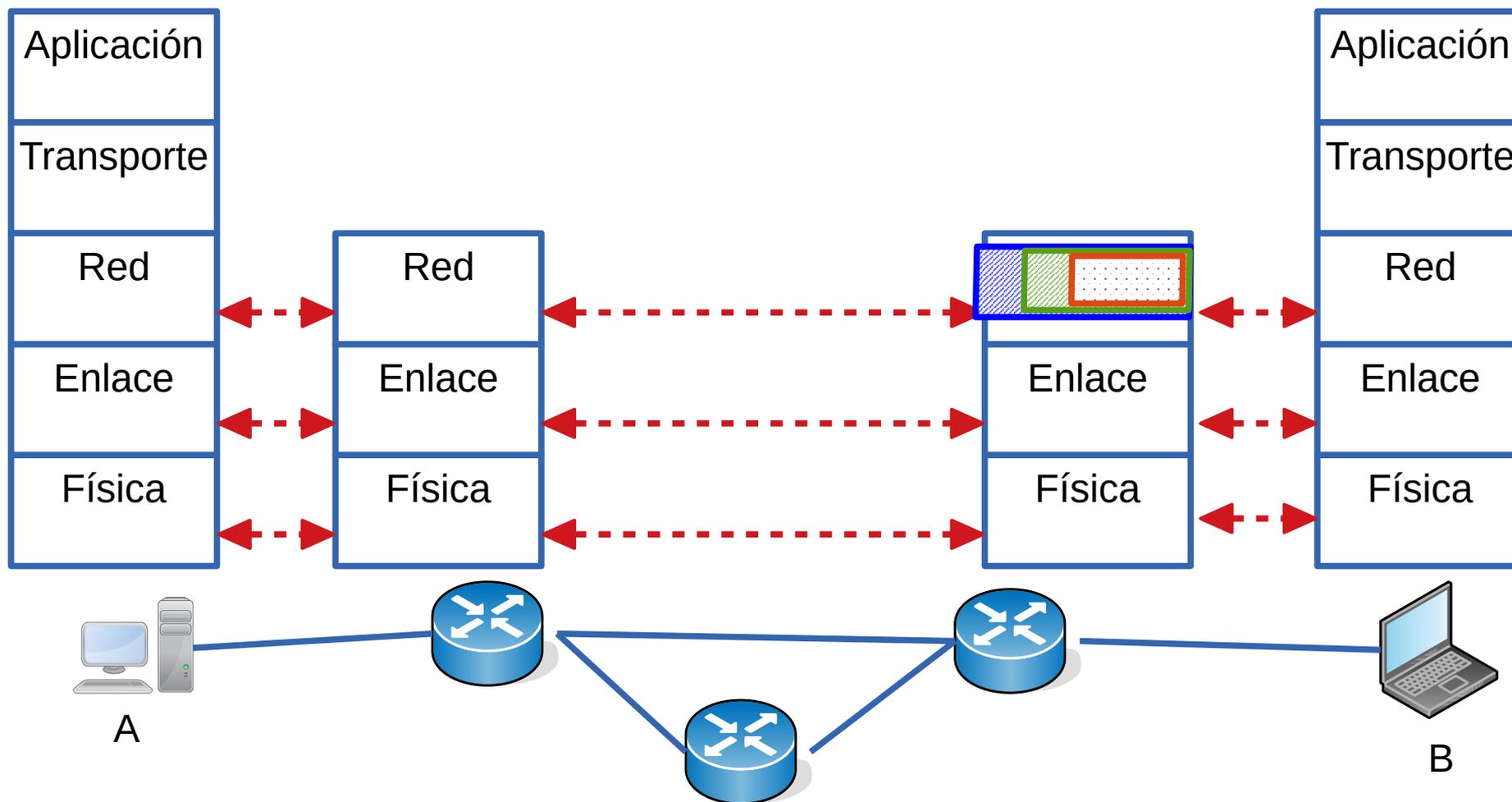
# Encapsulamiento y niveles de abstracción



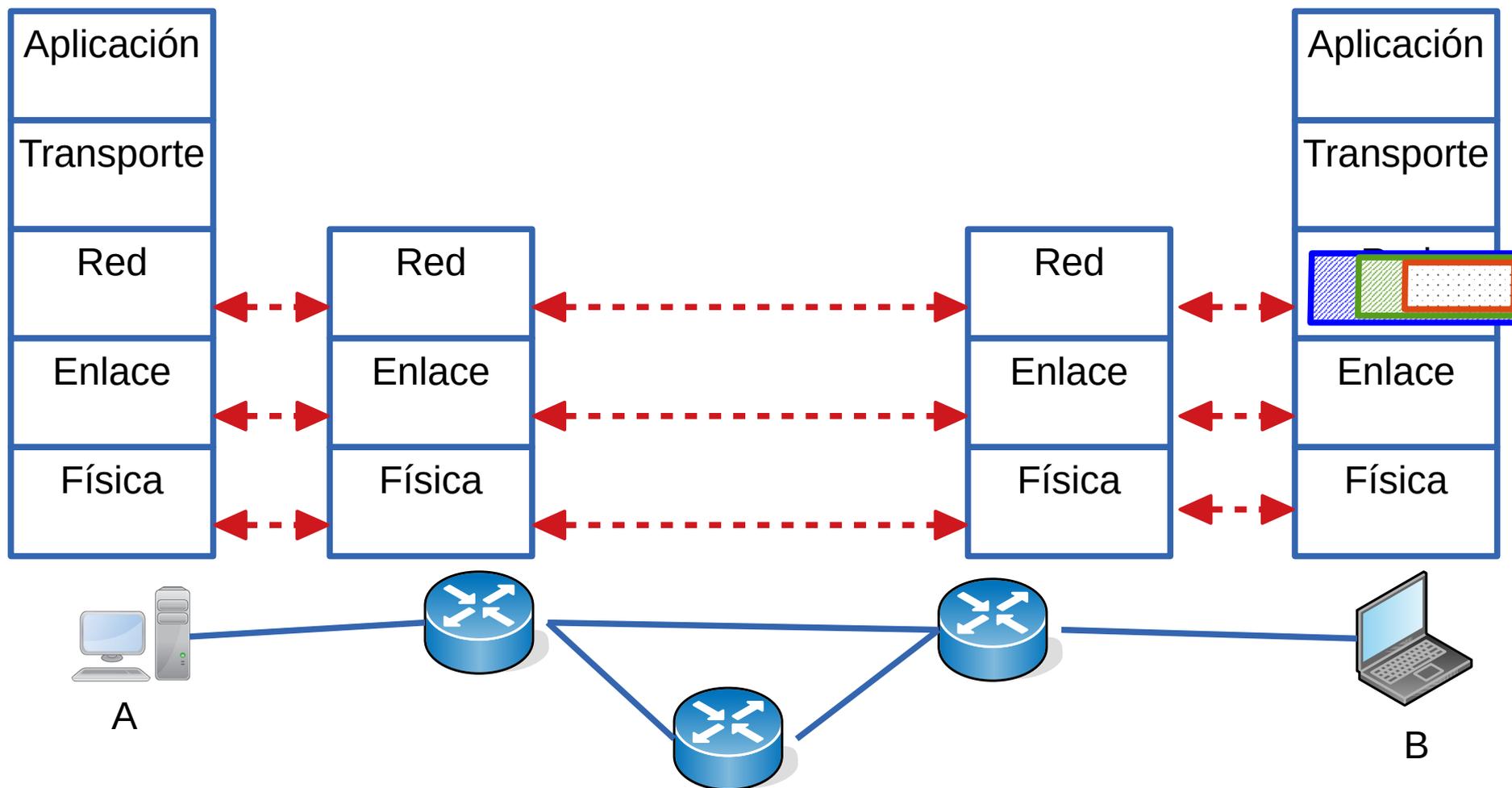
# Encapsulamiento y niveles de abstracción



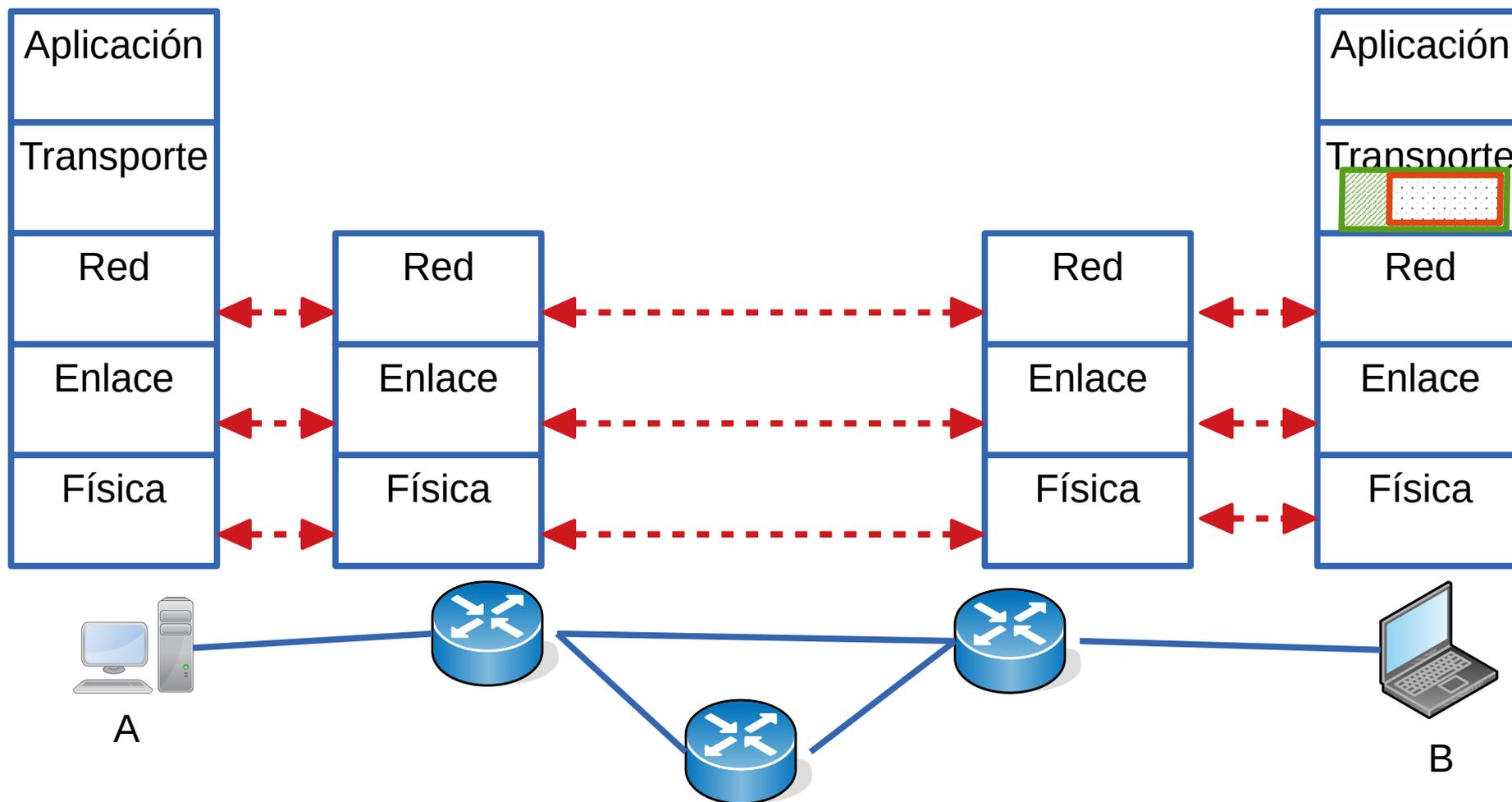
# Encapsulamiento y niveles de abstracción



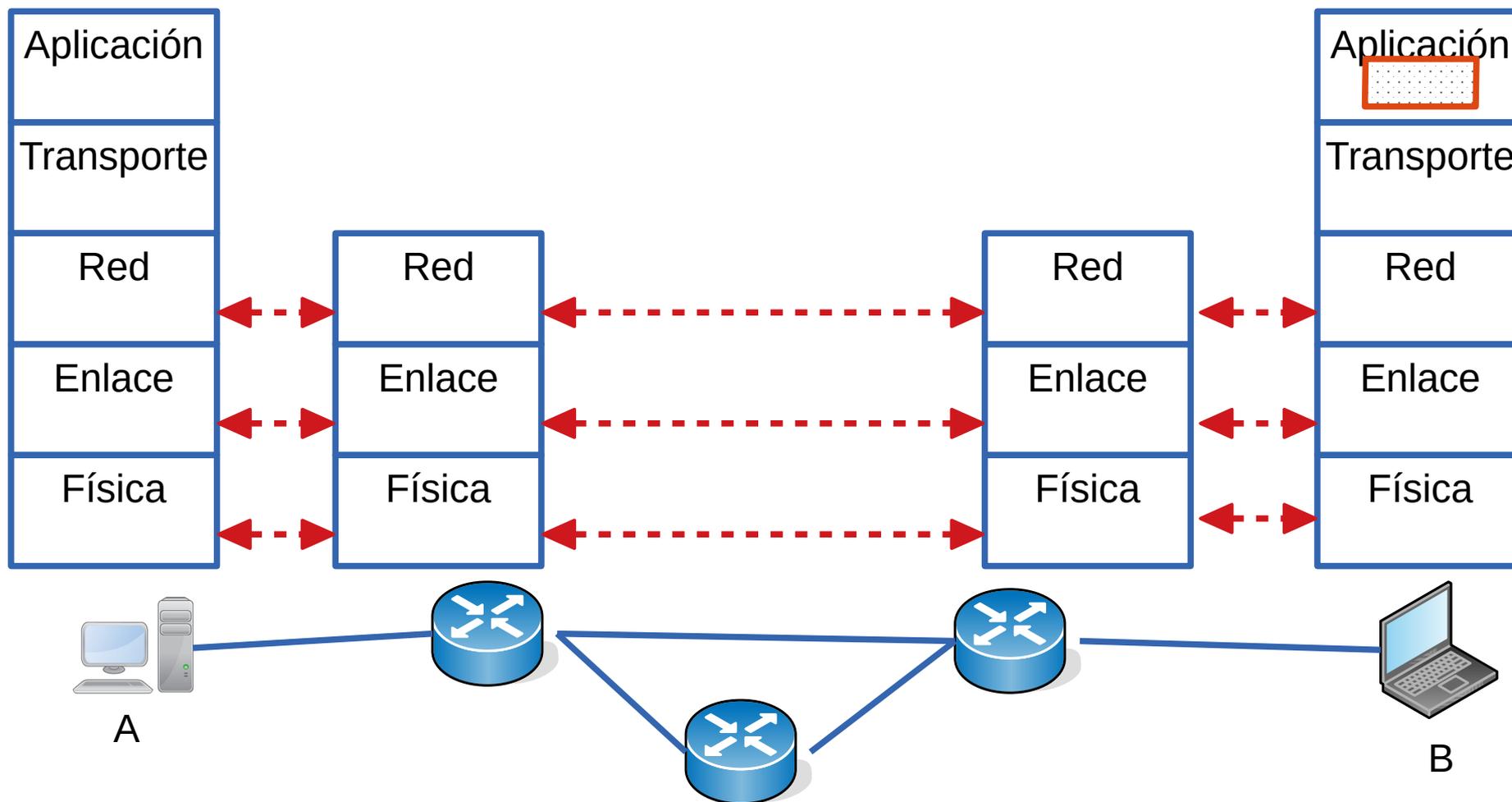
# Encapsulamiento y niveles de abstracción



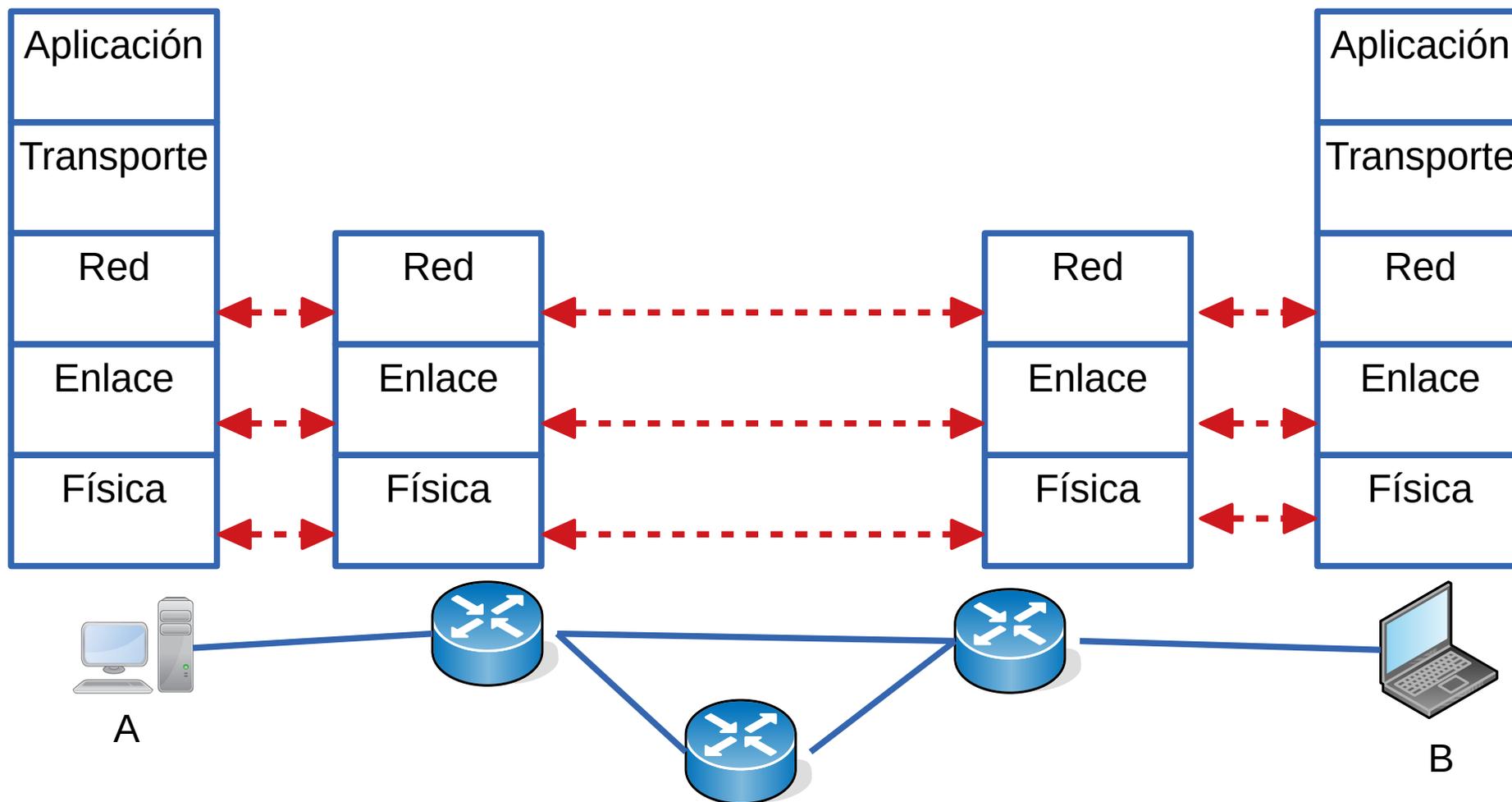
# Encapsulamiento y niveles de abstracción



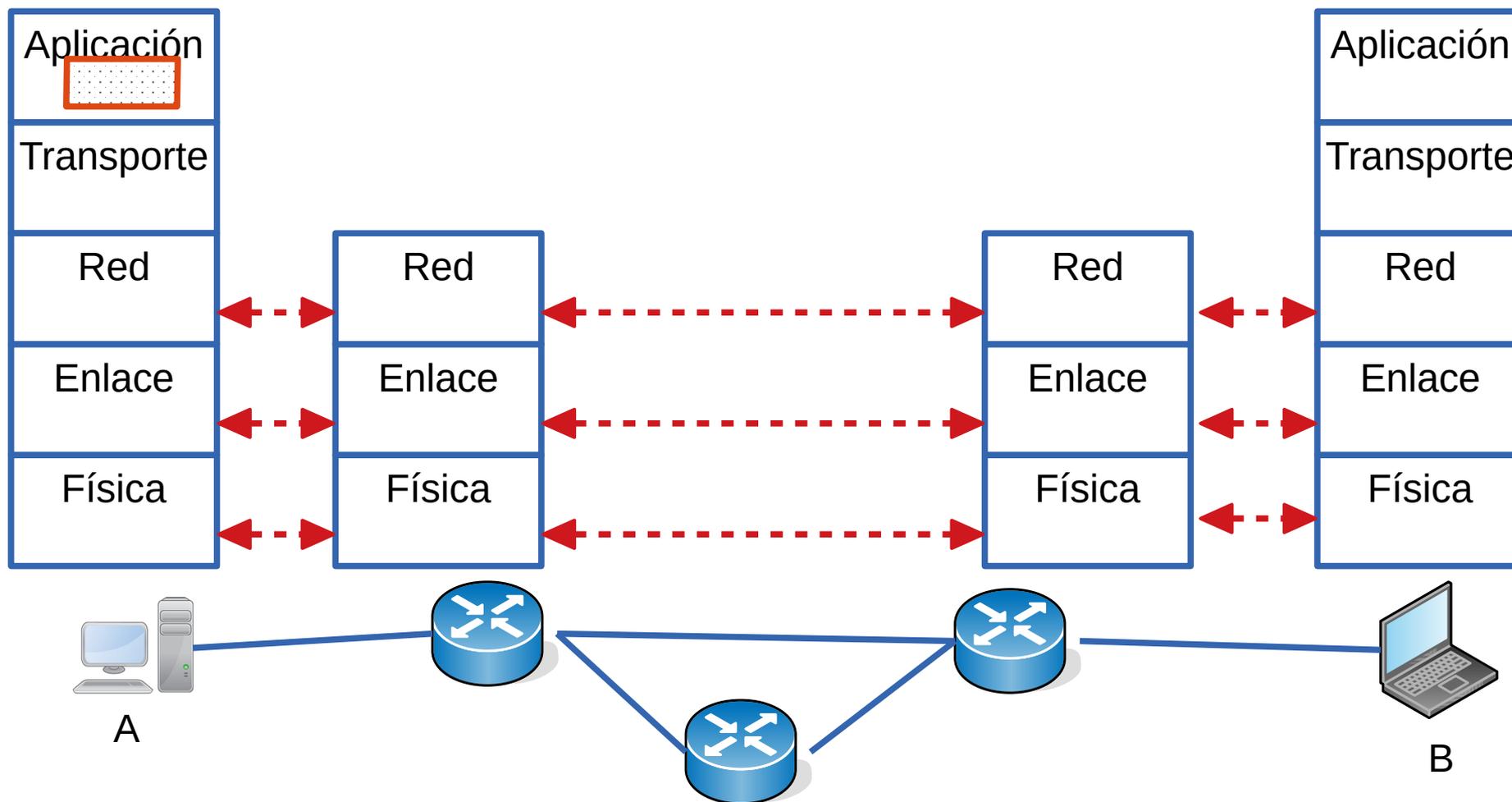
# Encapsulamiento y niveles de abstracción



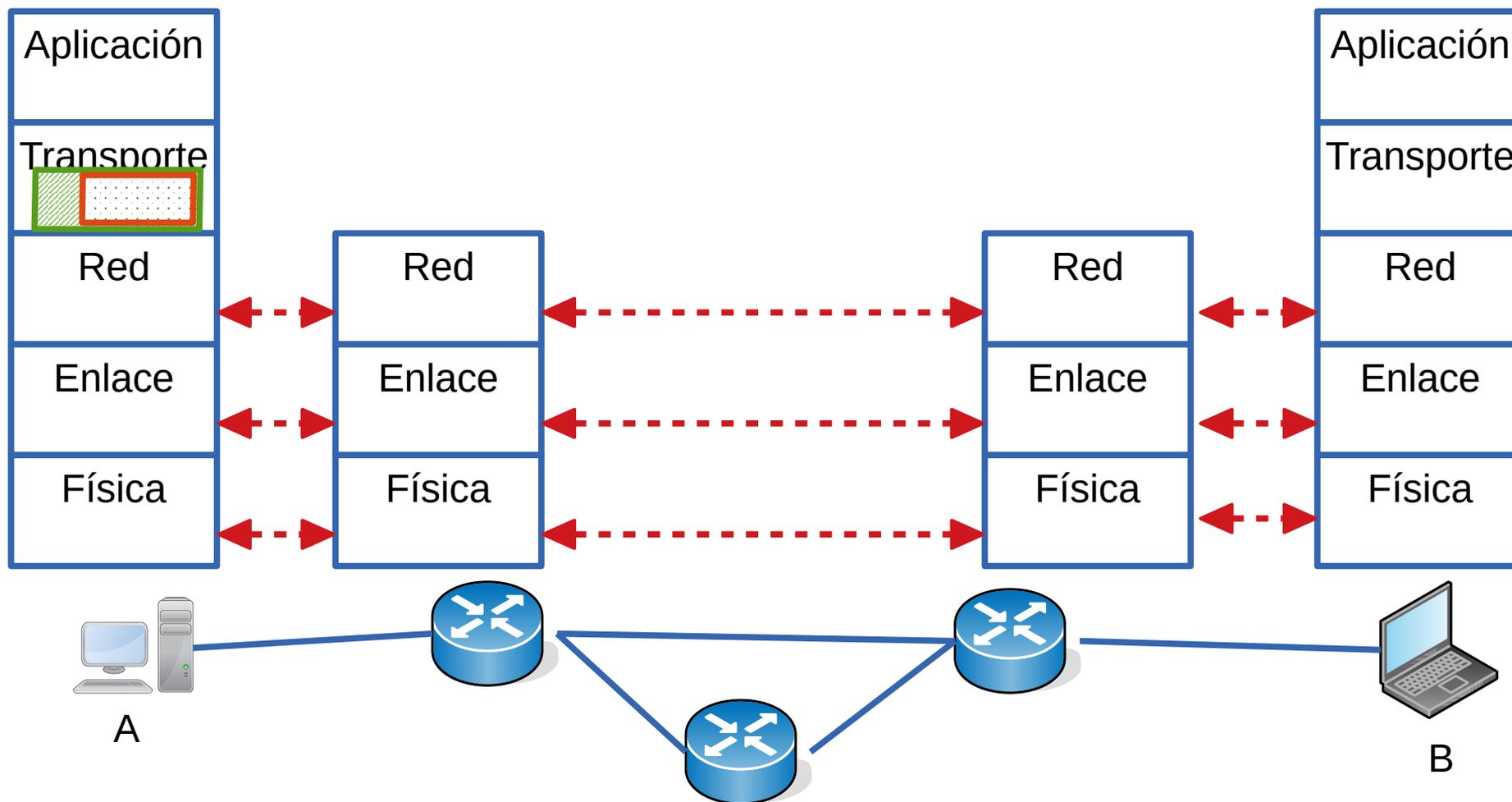
# Encapsulamiento y niveles de abstracción



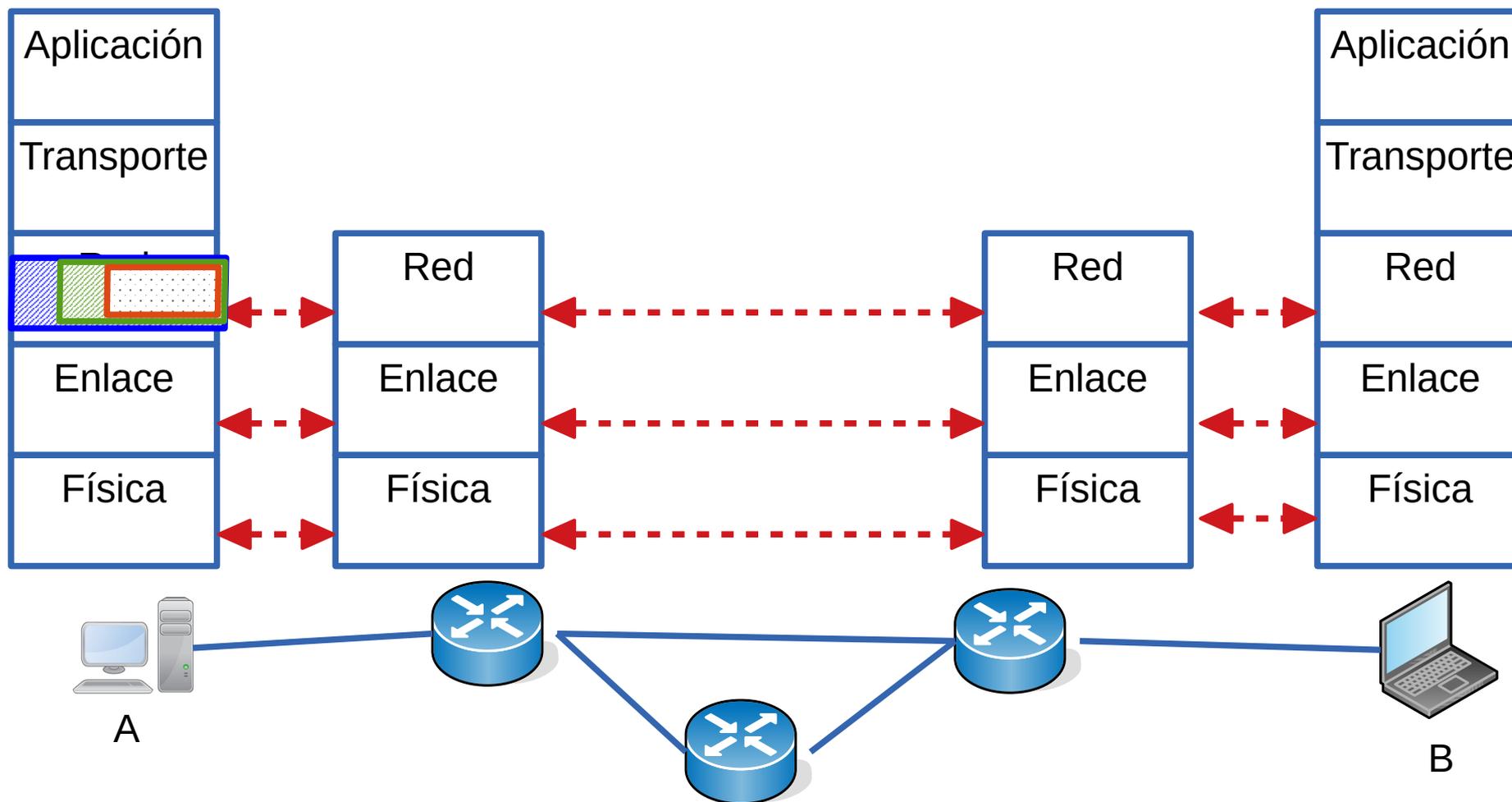
# Encapsulamiento y niveles de abstracción



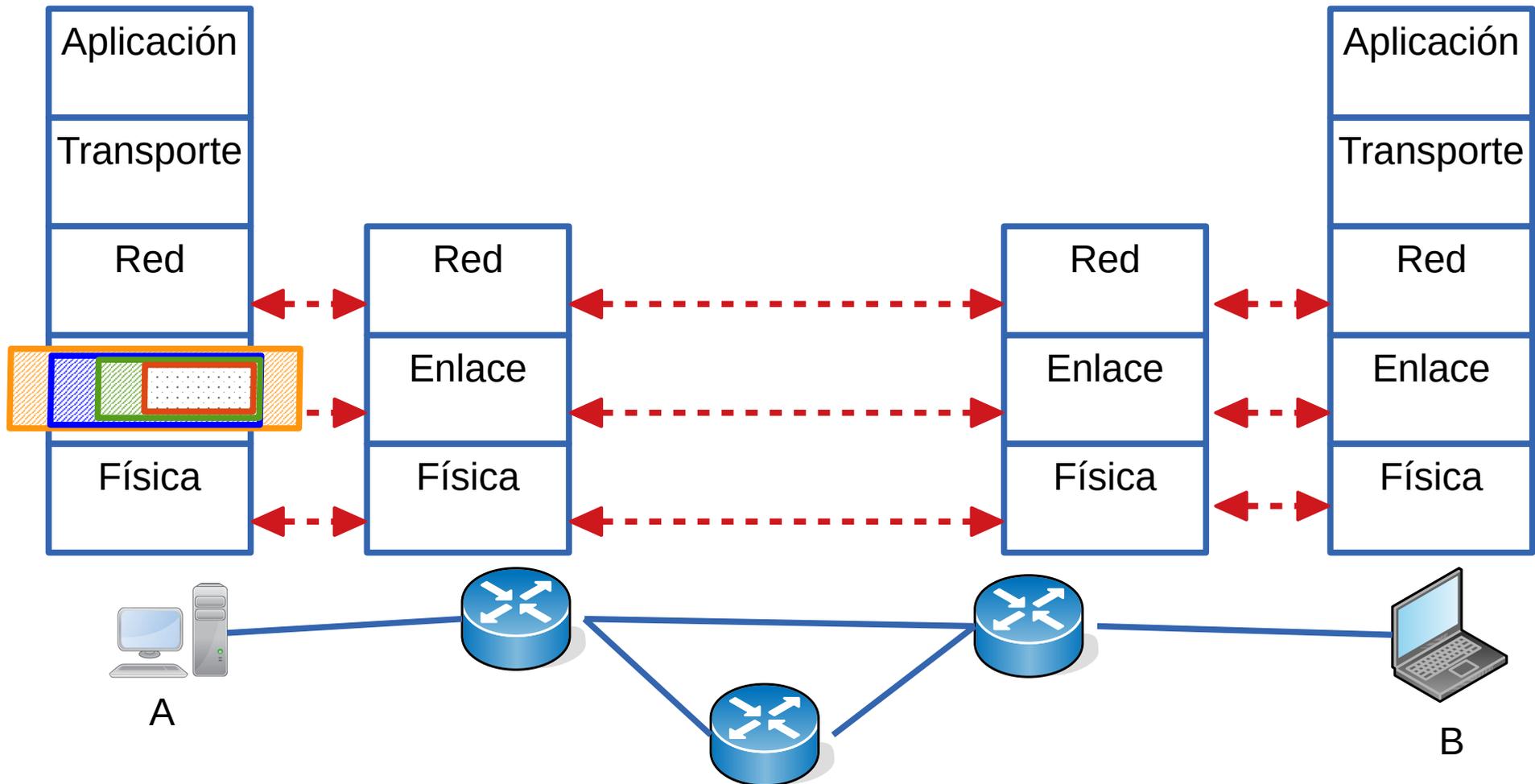
# Encapsulamiento y niveles de abstracción



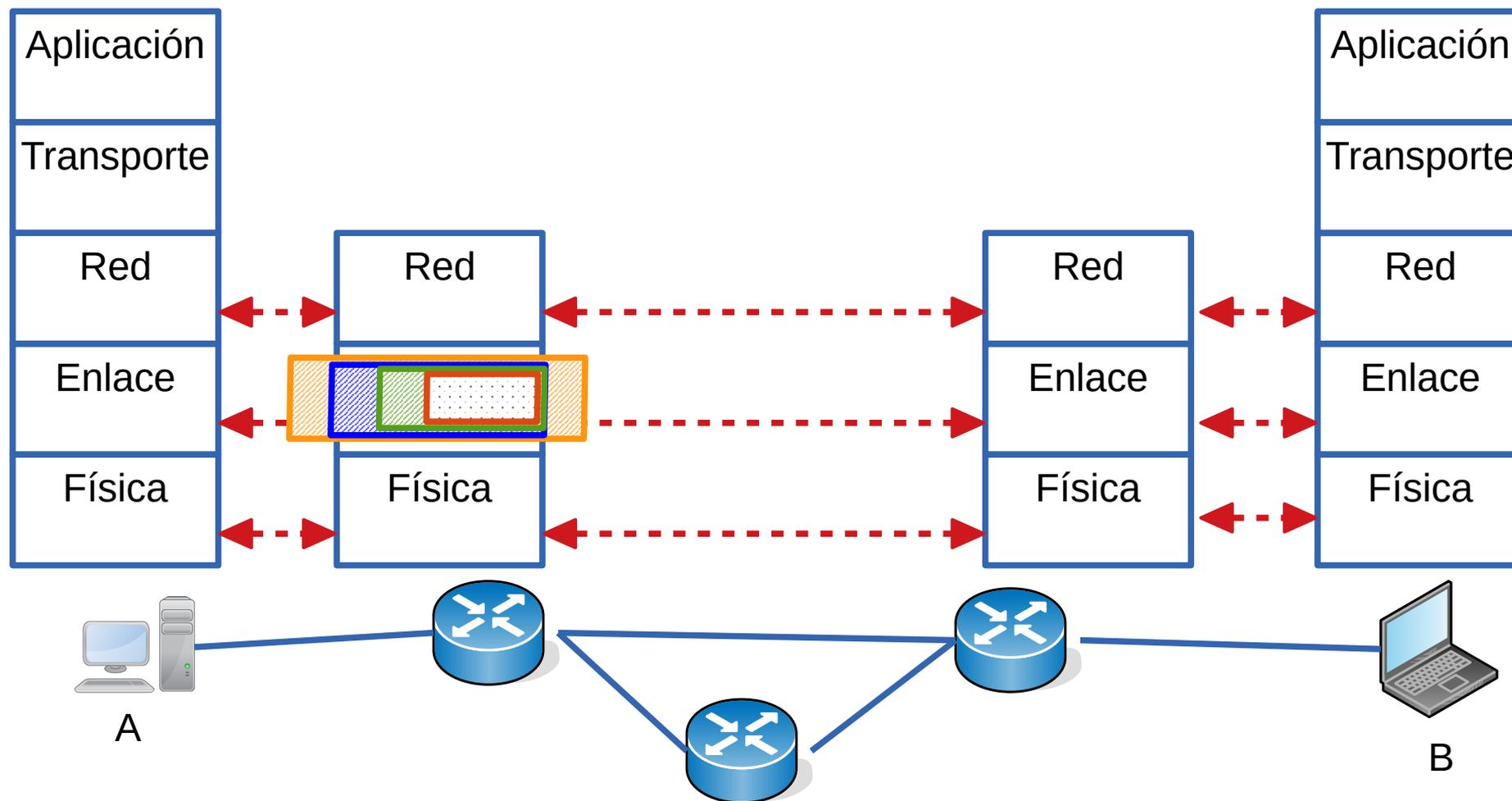
# Encapsulamiento y niveles de abstracción



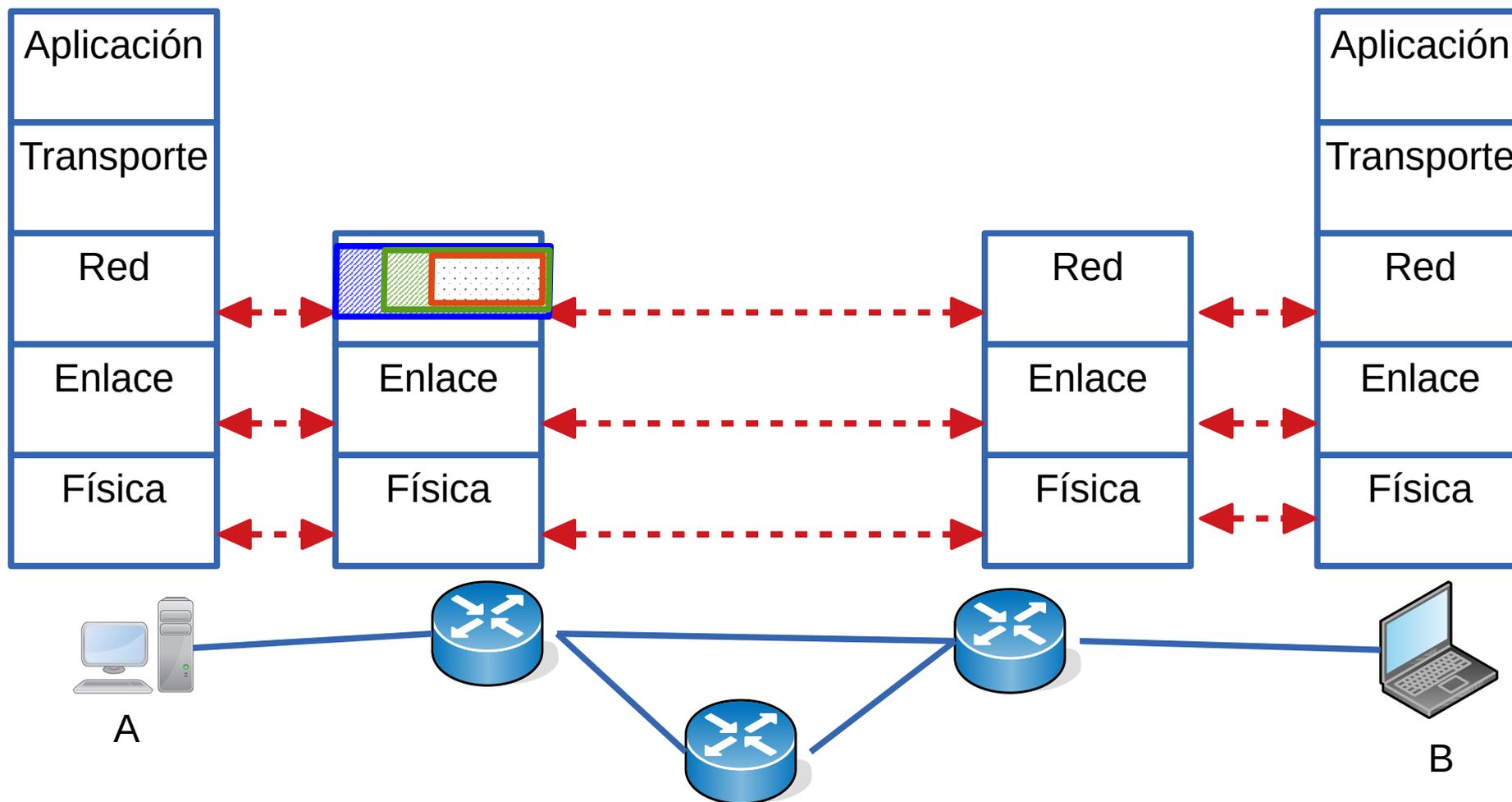
# Encapsulamiento y niveles de abstracción



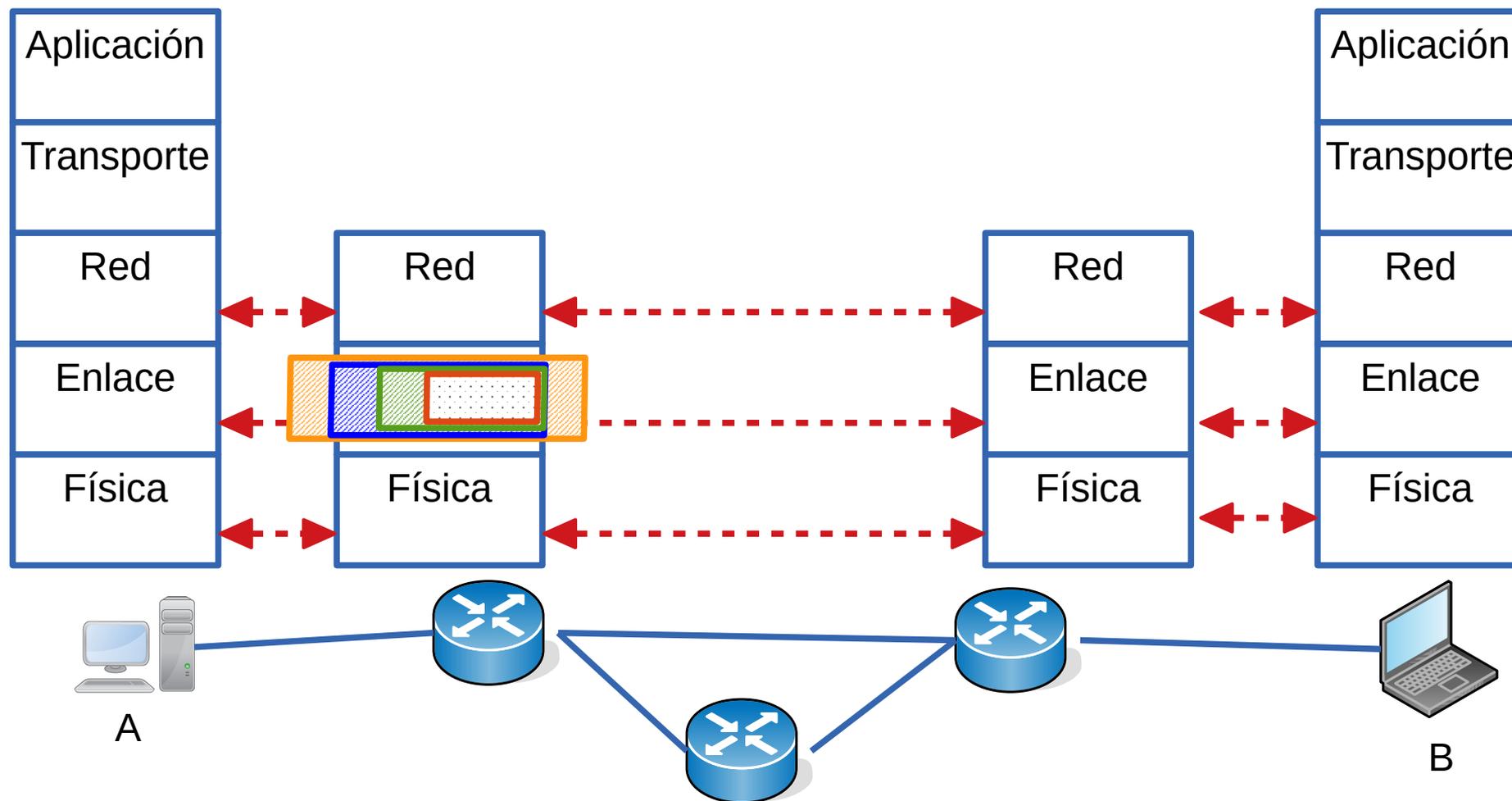
# Encapsulamiento y niveles de abstracción



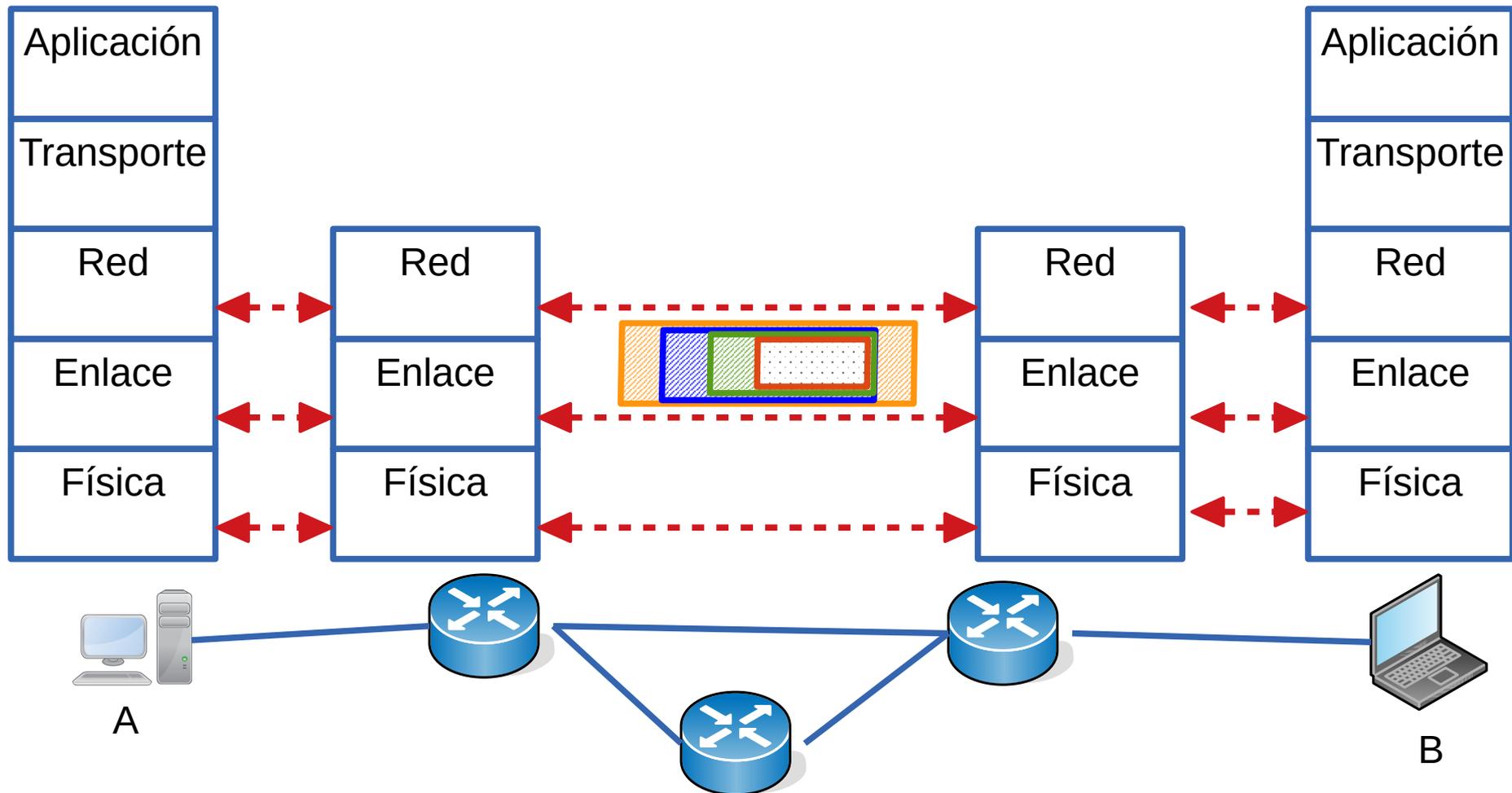
# Encapsulamiento y niveles de abstracción



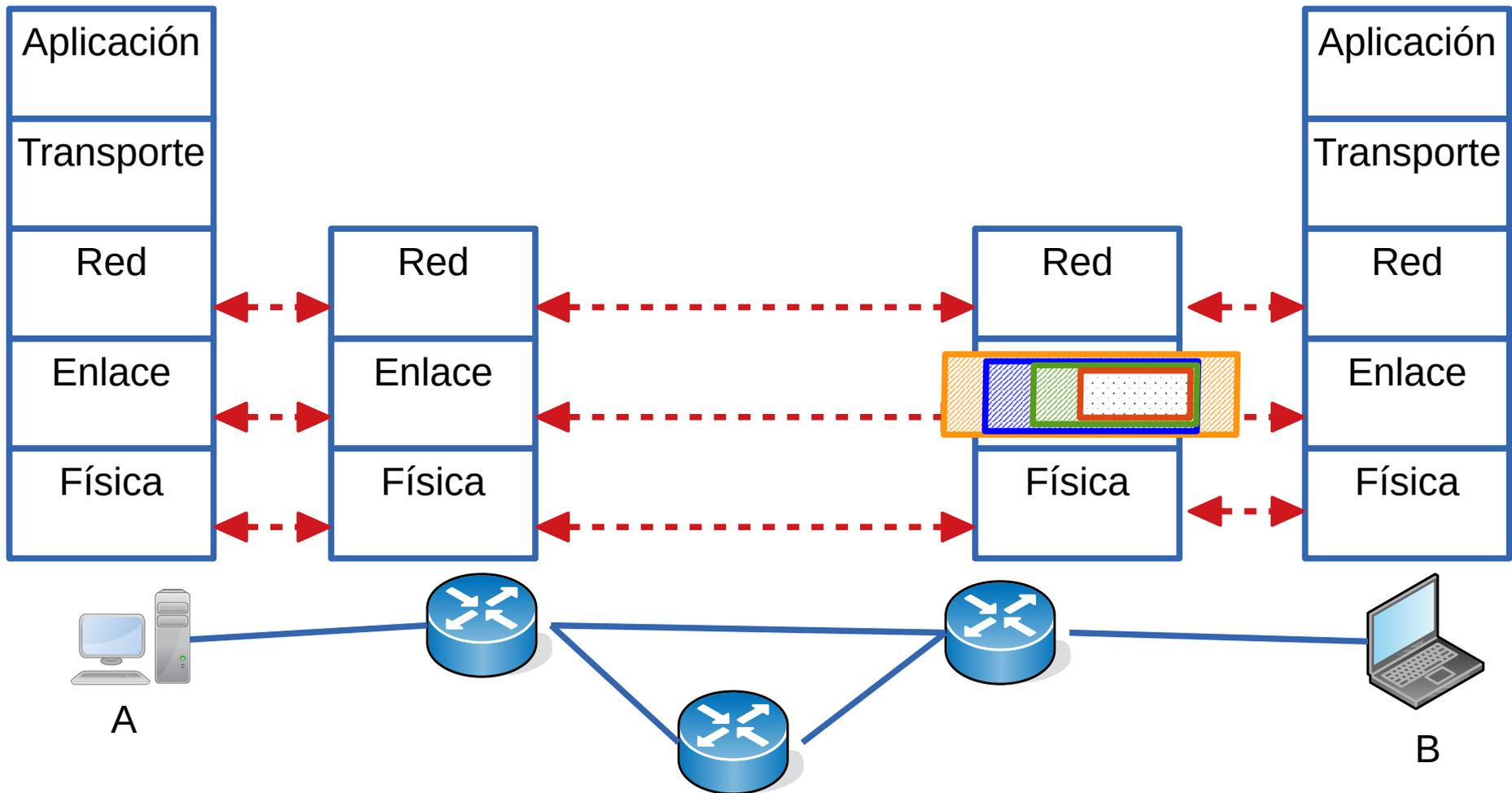
# Encapsulamiento y niveles de abstracción



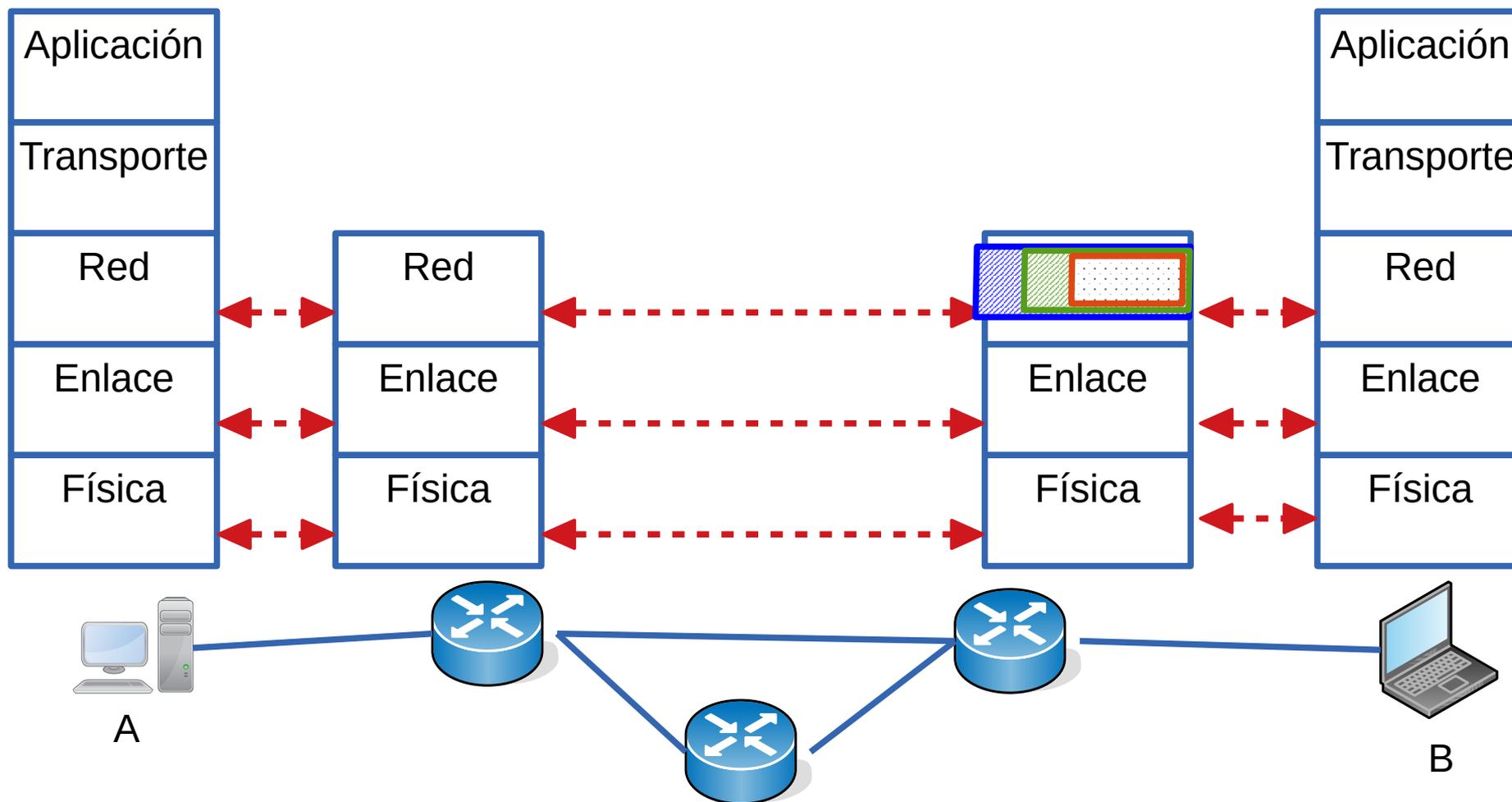
# Encapsulamiento y niveles de abstracción



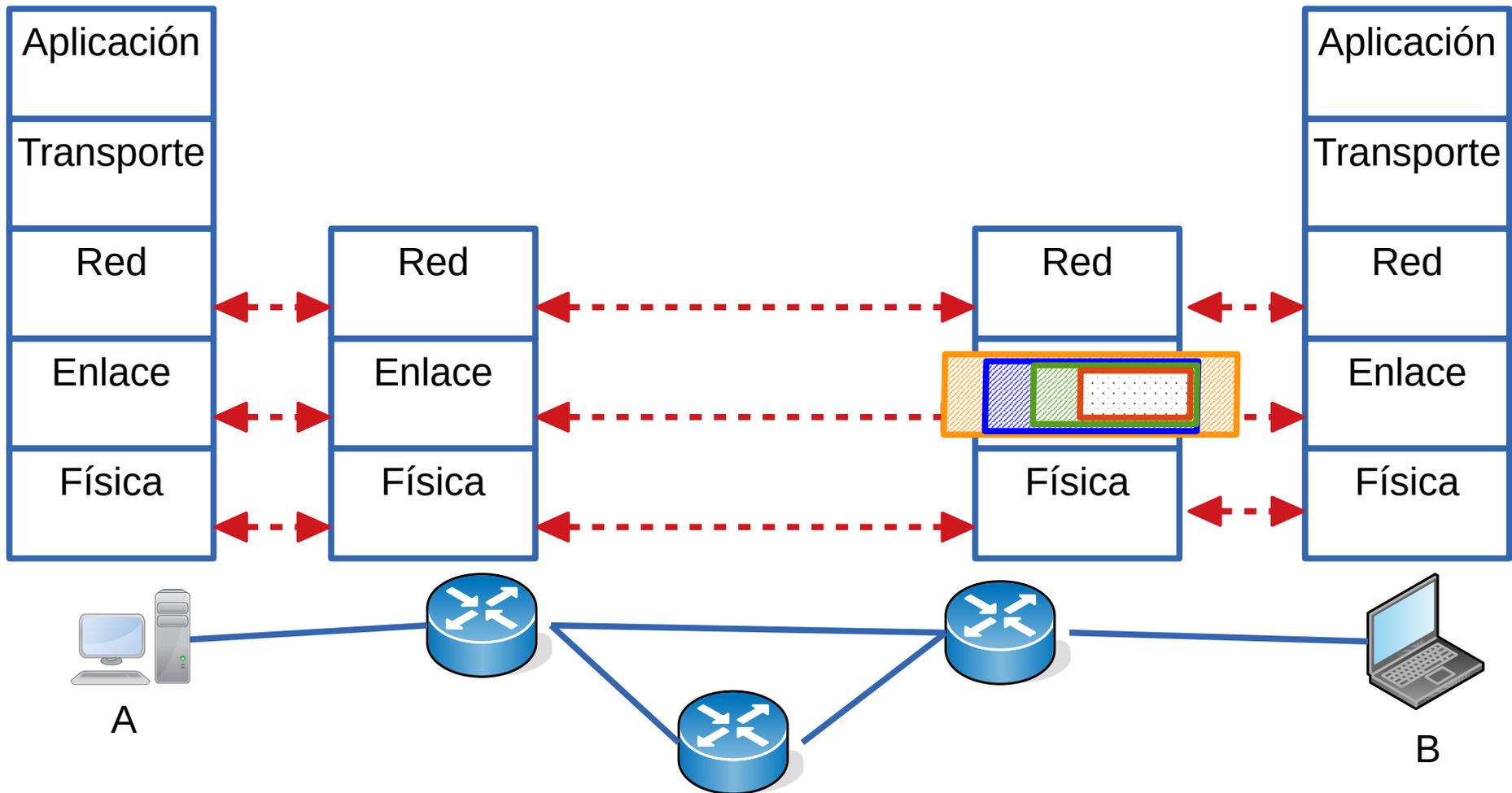
# Encapsulamiento y niveles de abstracción



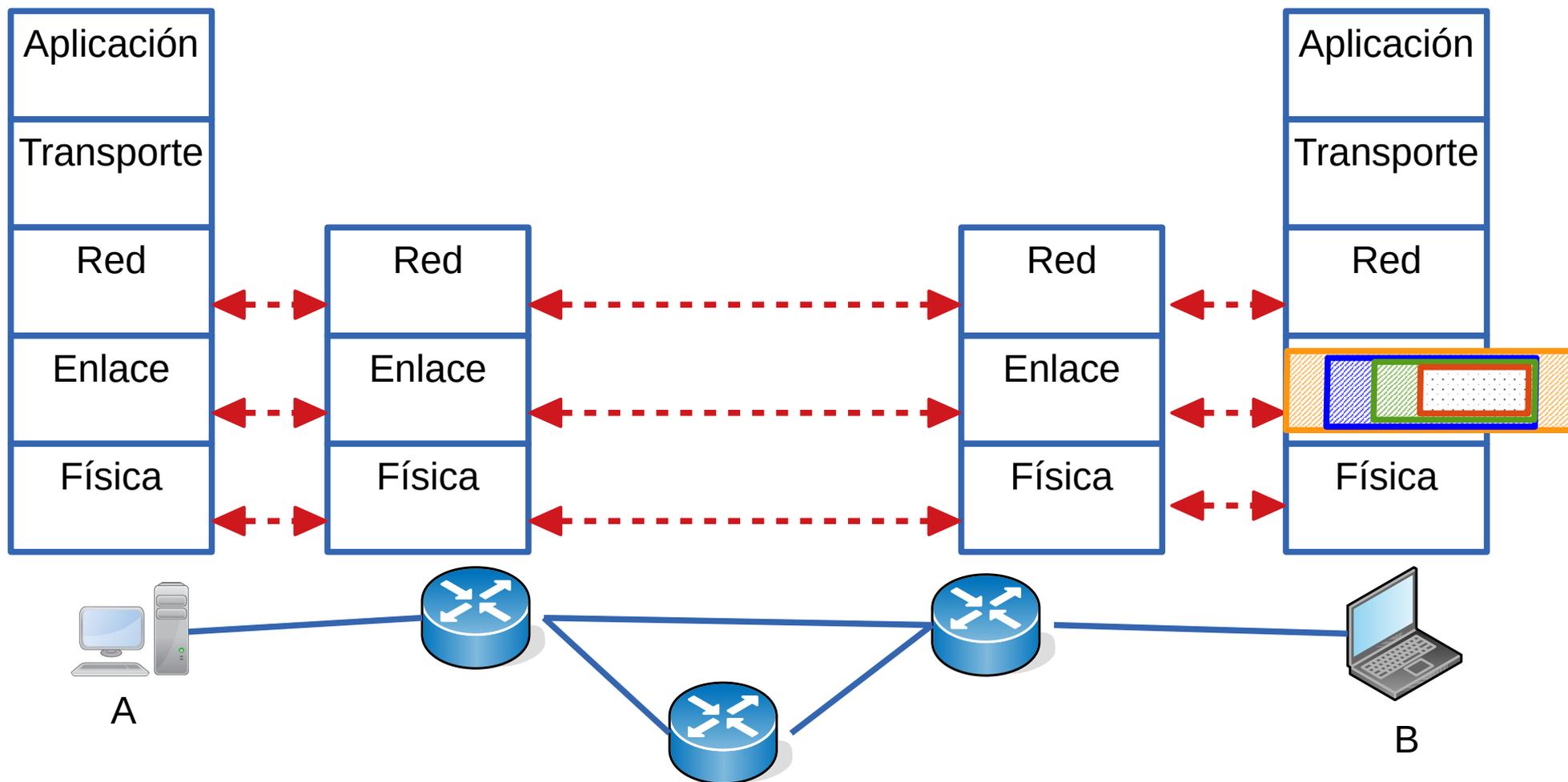
# Encapsulamiento y niveles de abstracción



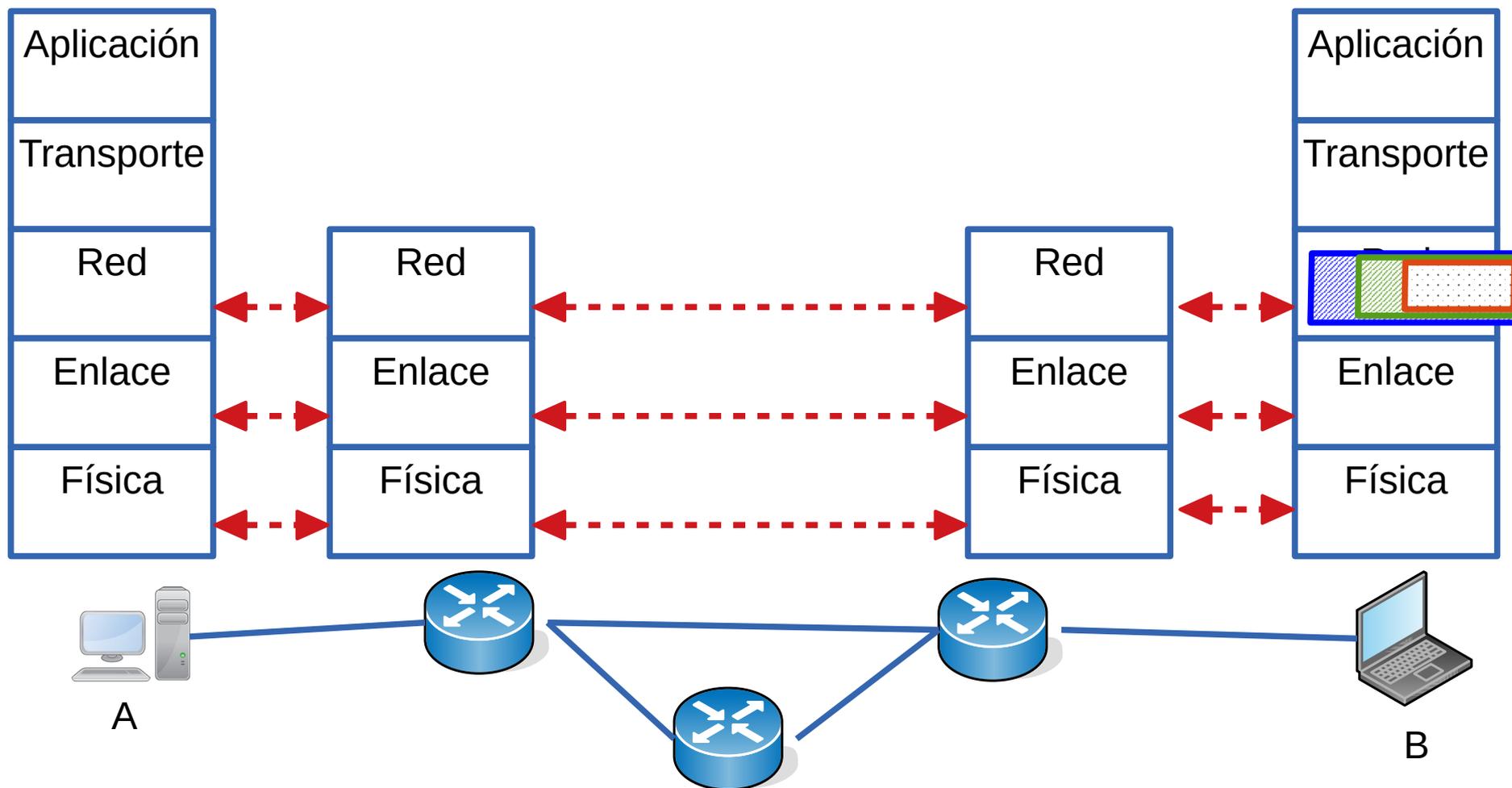
# Encapsulamiento y niveles de abstracción



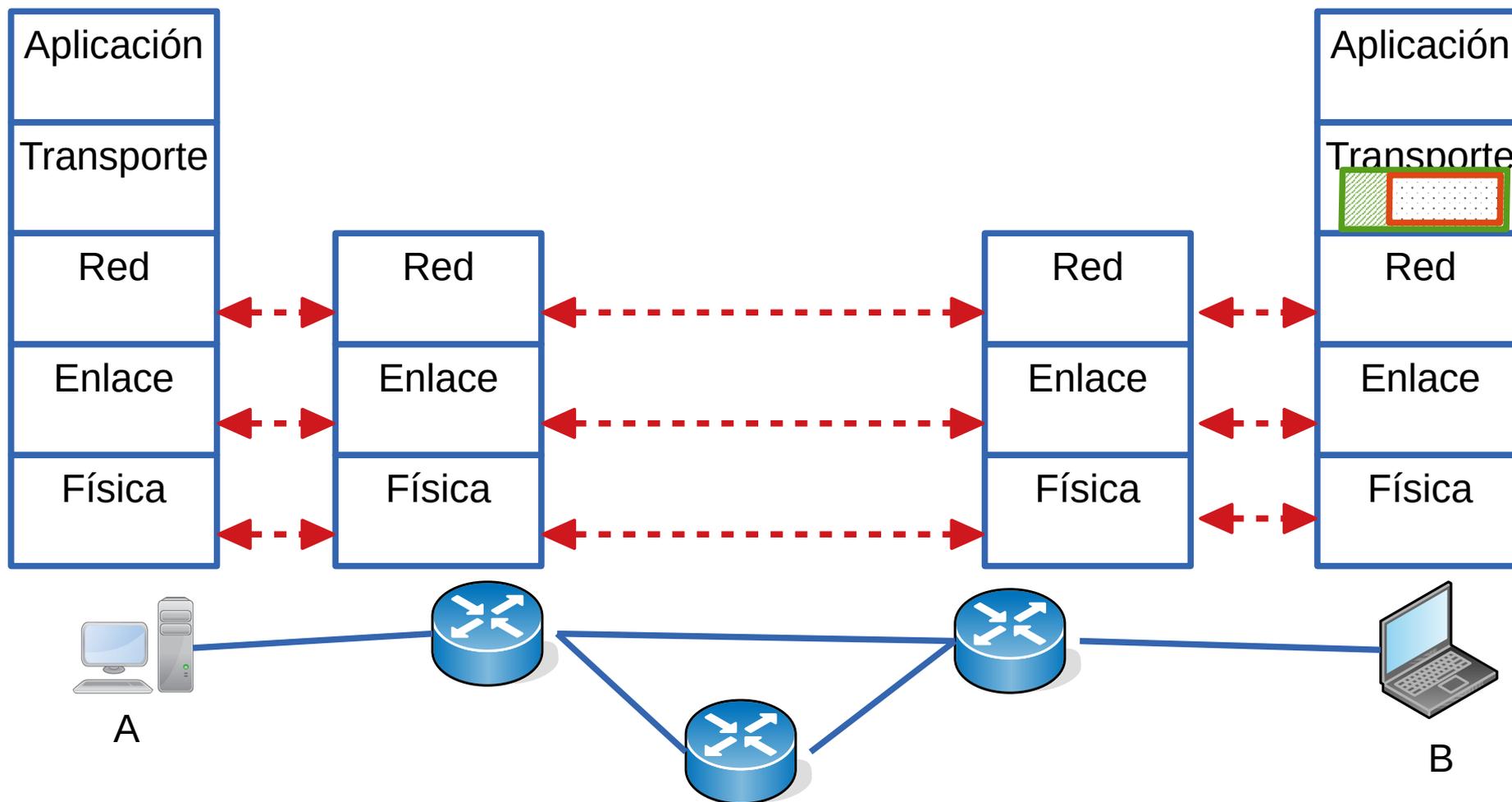
# Encapsulamiento y niveles de abstracción



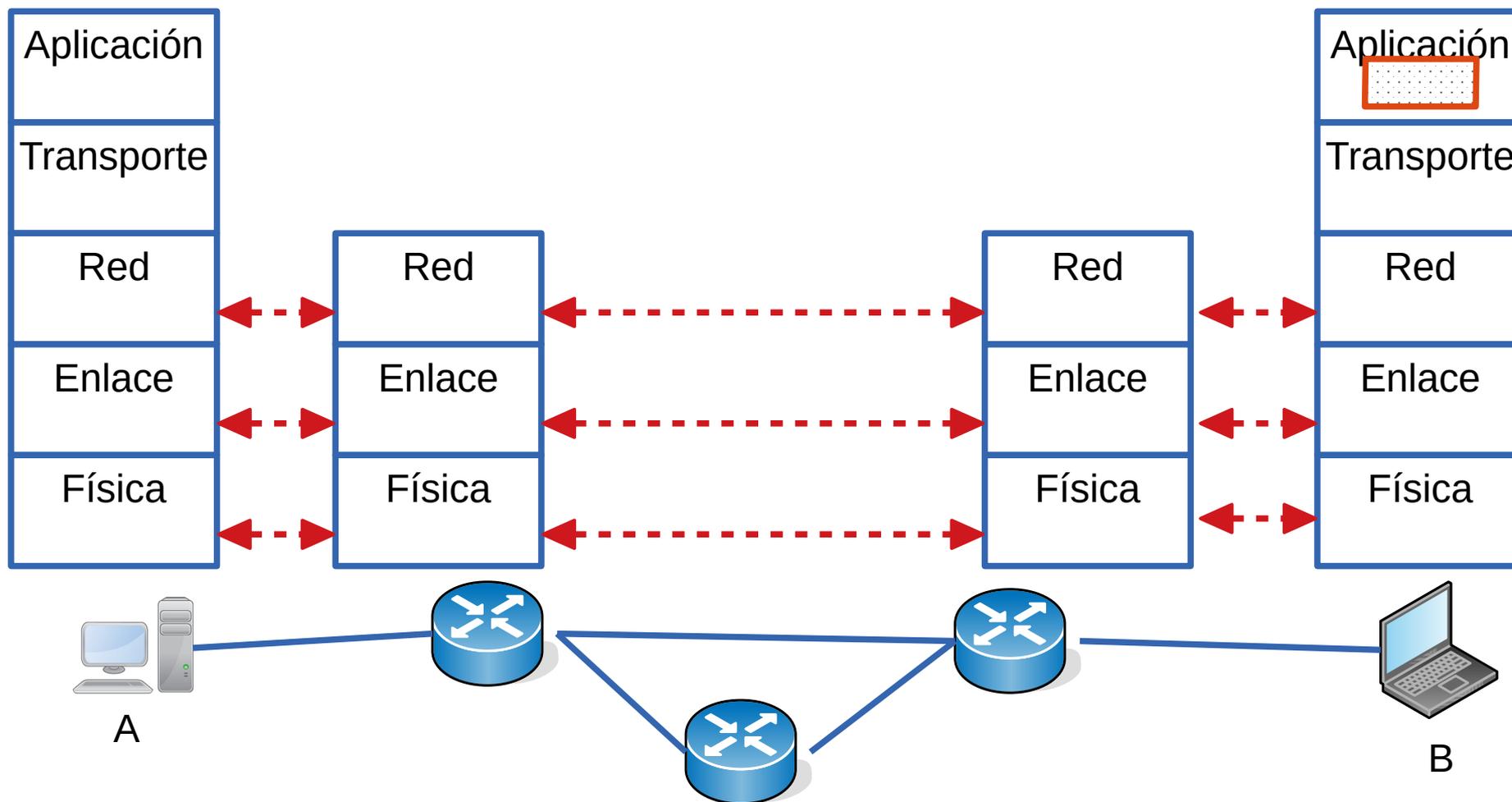
# Encapsulamiento y niveles de abstracción



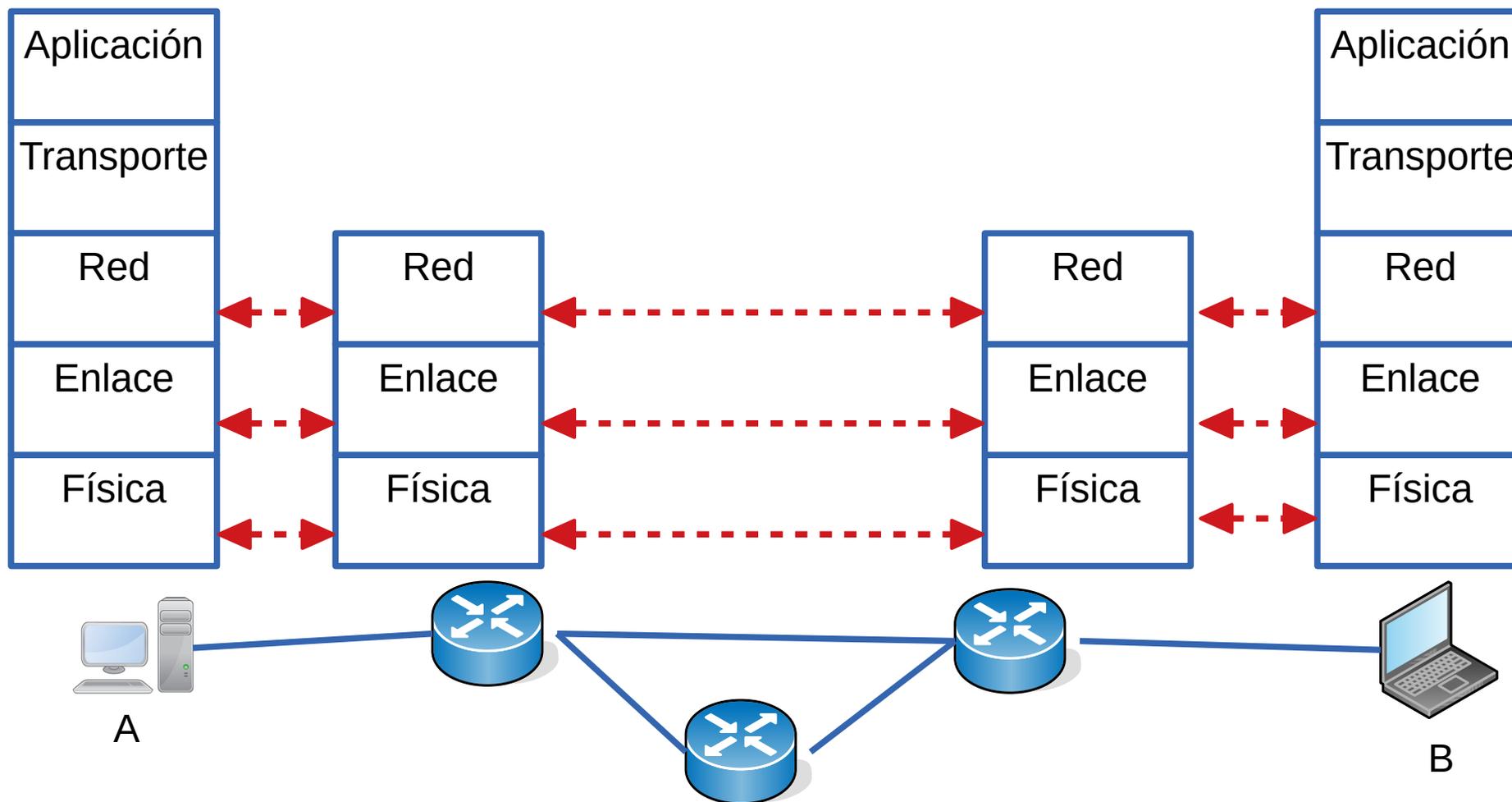
# Encapsulamiento y niveles de abstracción



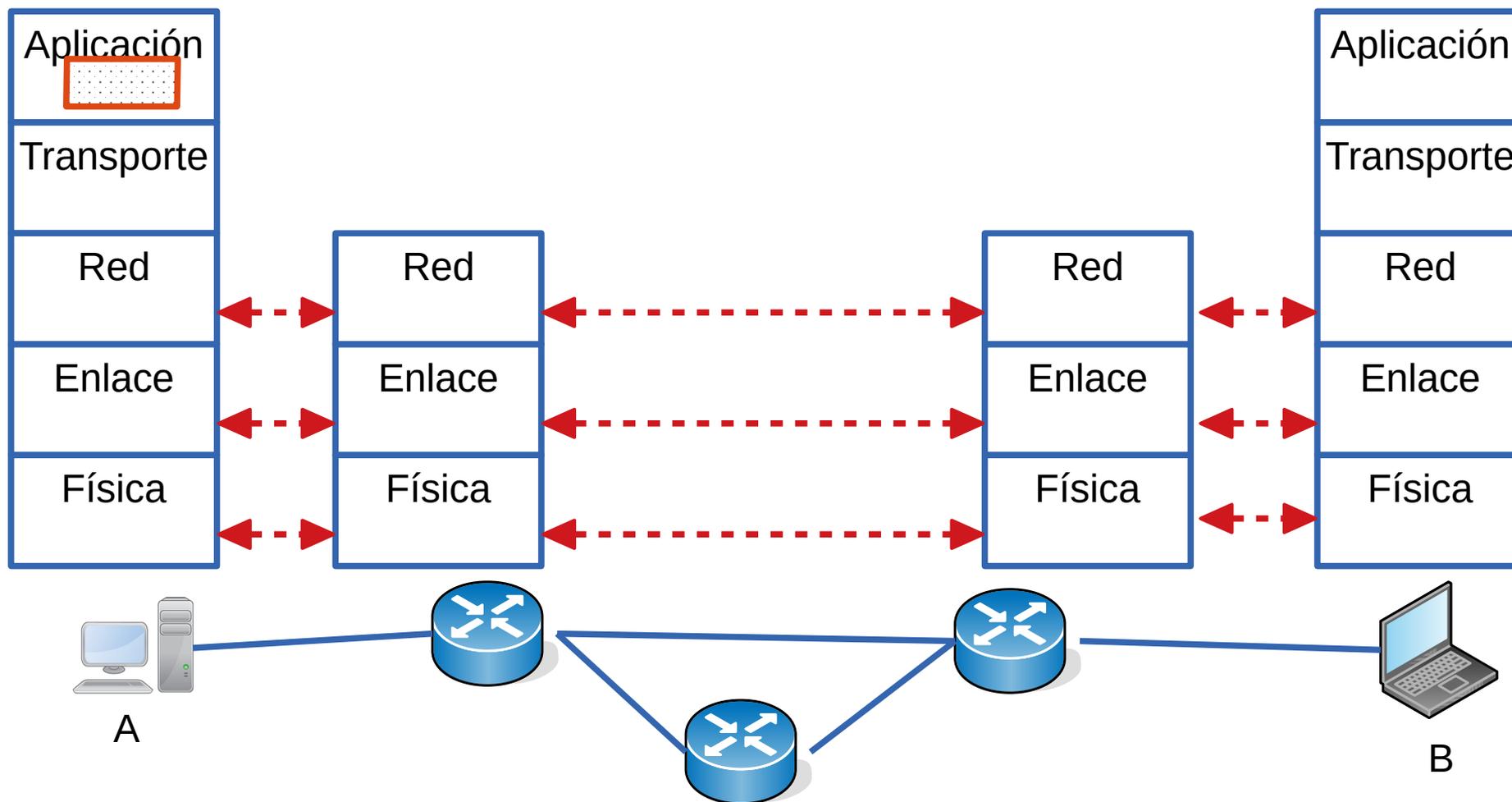
# Encapsulamiento y niveles de abstracción



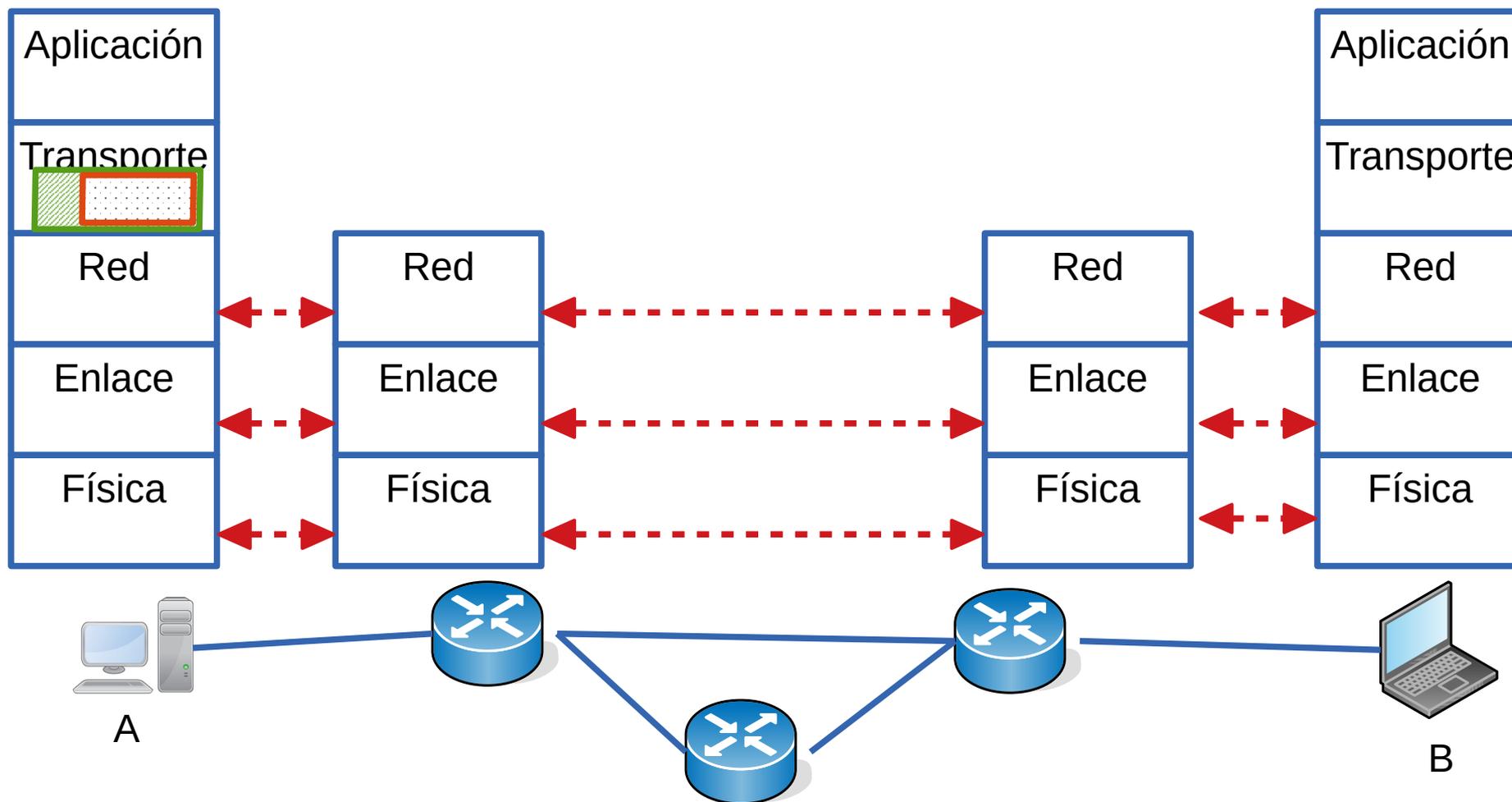
# Encapsulamiento y niveles de abstracción



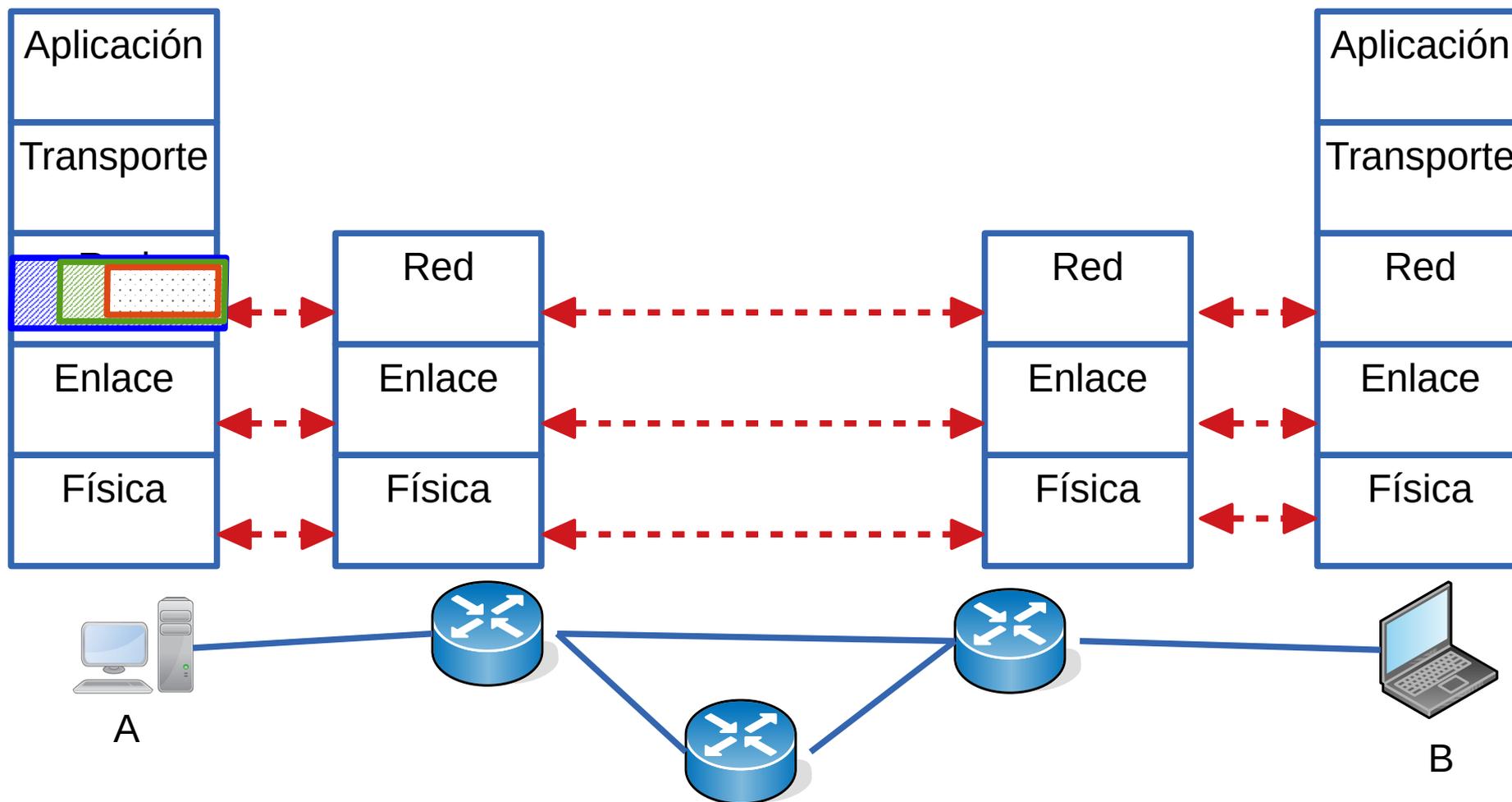
# Encapsulamiento y niveles de abstracción



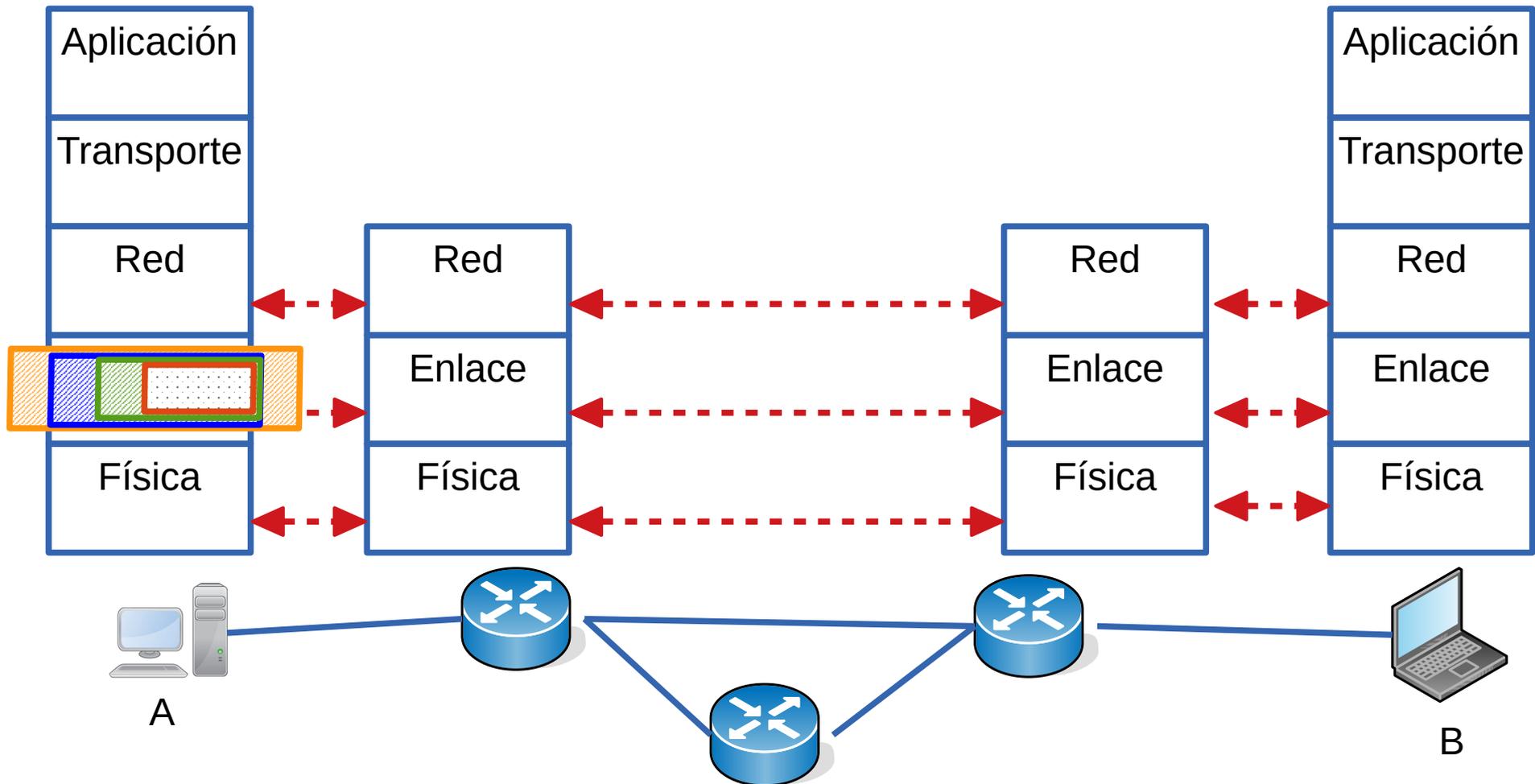
# Encapsulamiento y niveles de abstracción



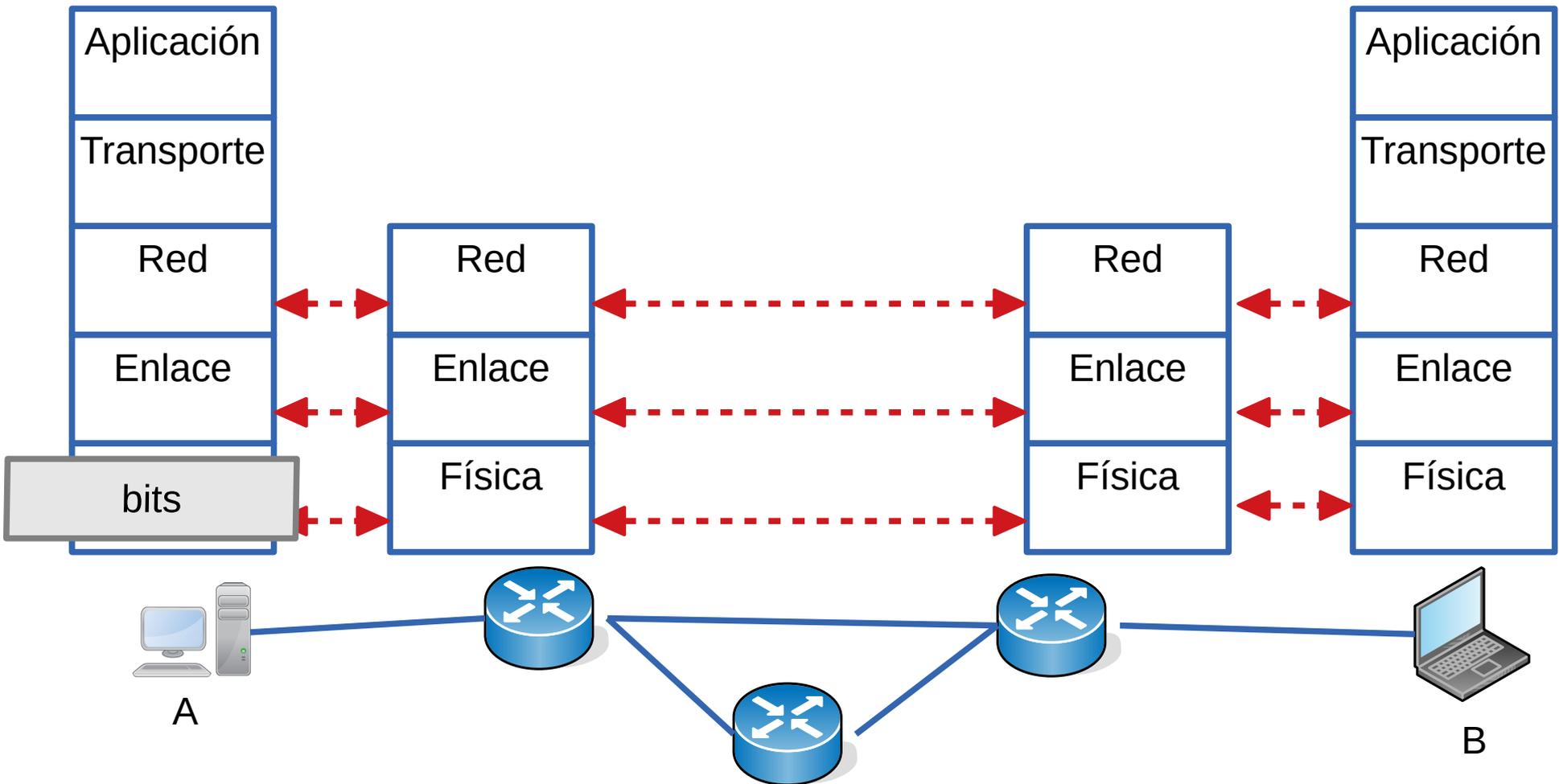
# Encapsulamiento y niveles de abstracción



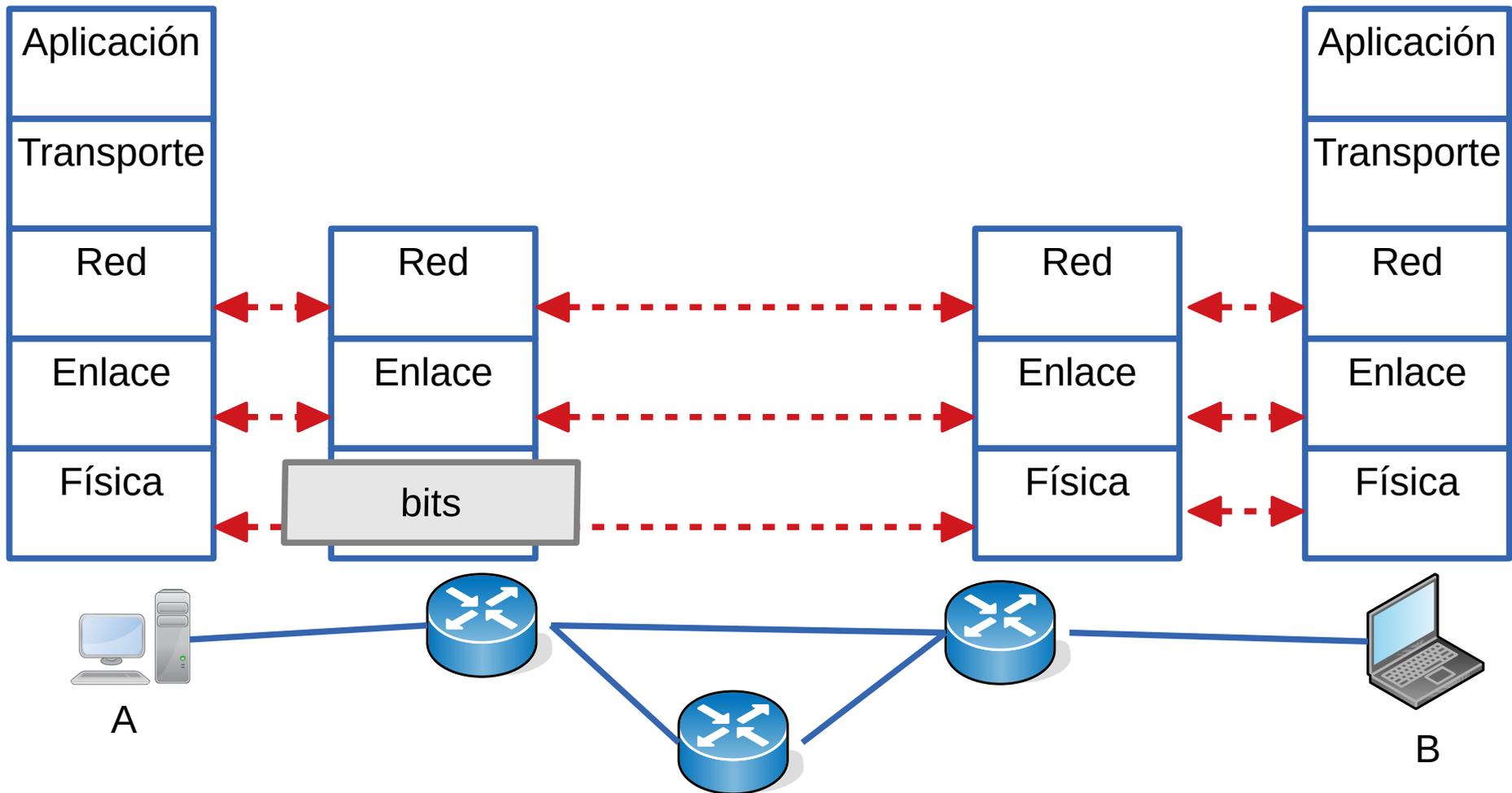
# Encapsulamiento y niveles de abstracción



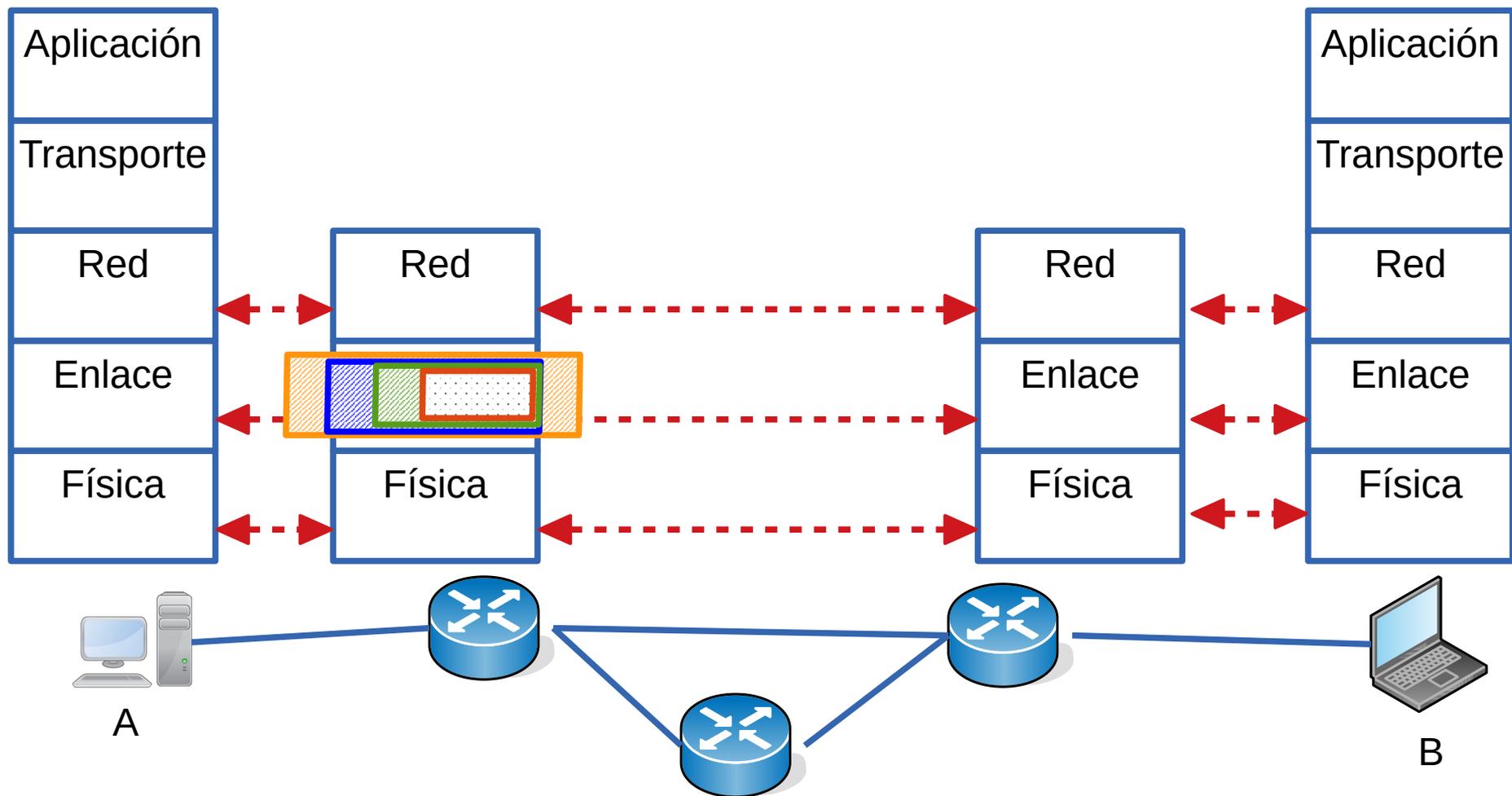
# Encapsulamiento y niveles de abstracción



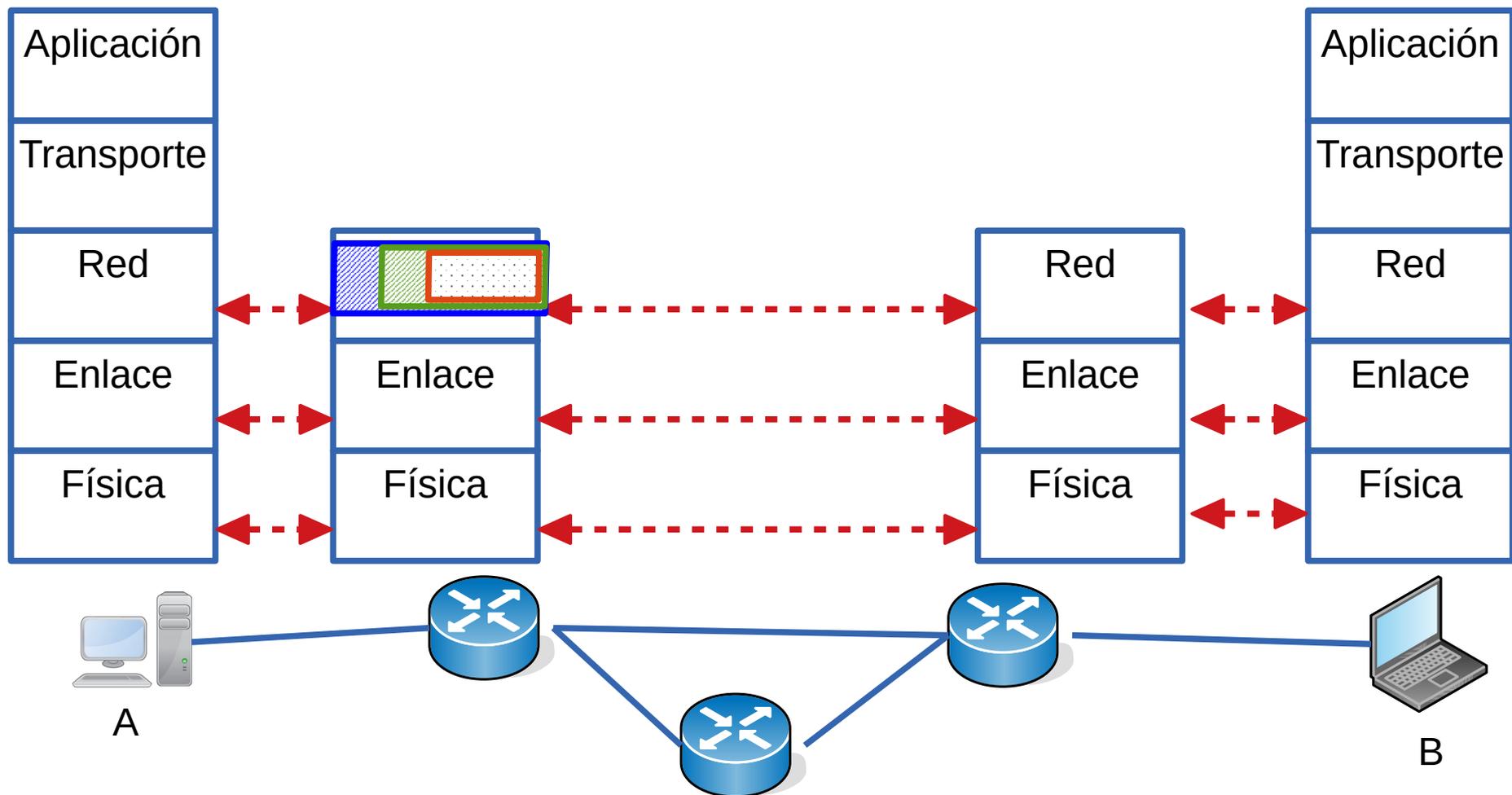
# Encapsulamiento y niveles de abstracción



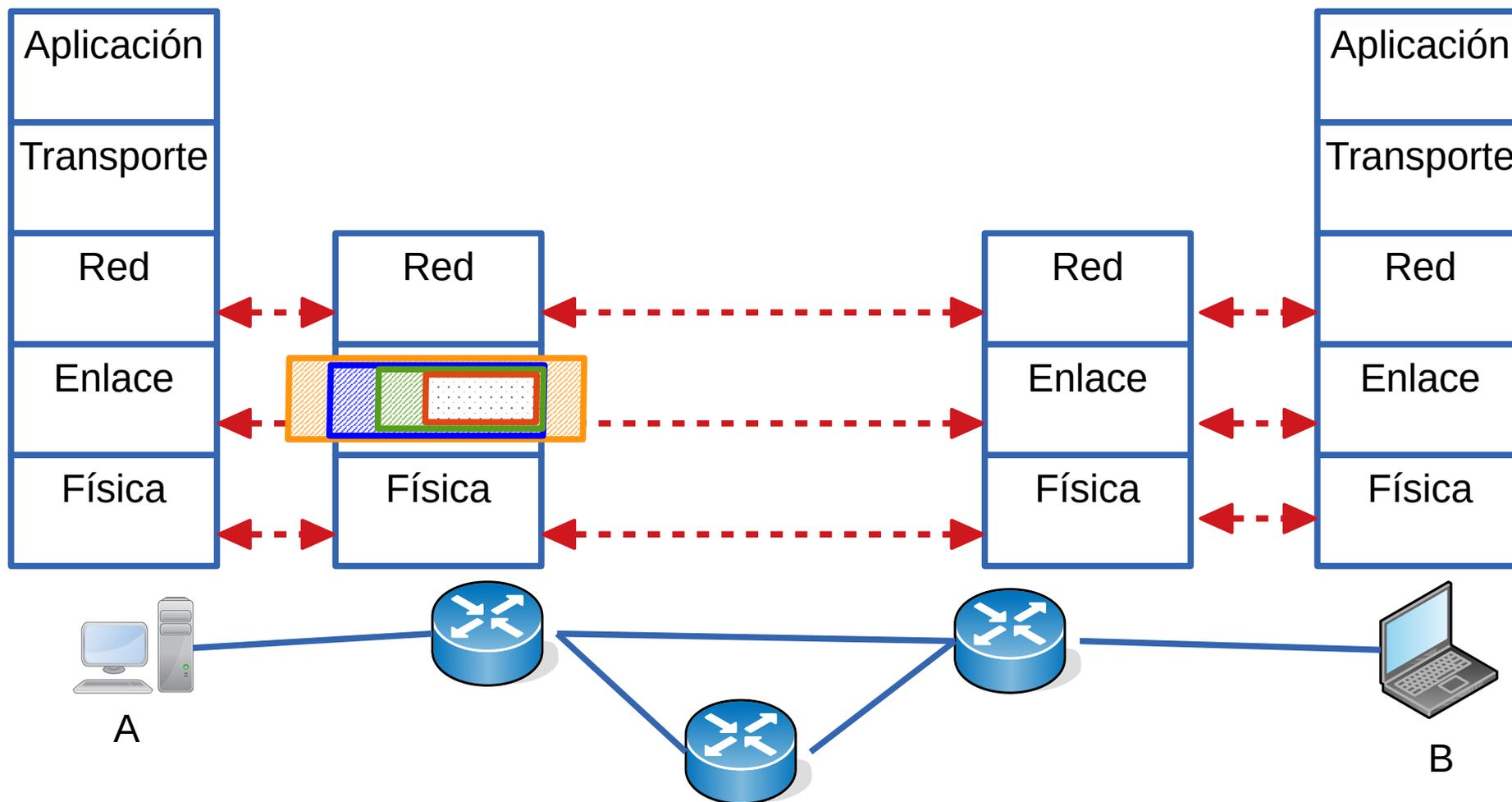
# Encapsulamiento y niveles de abstracción



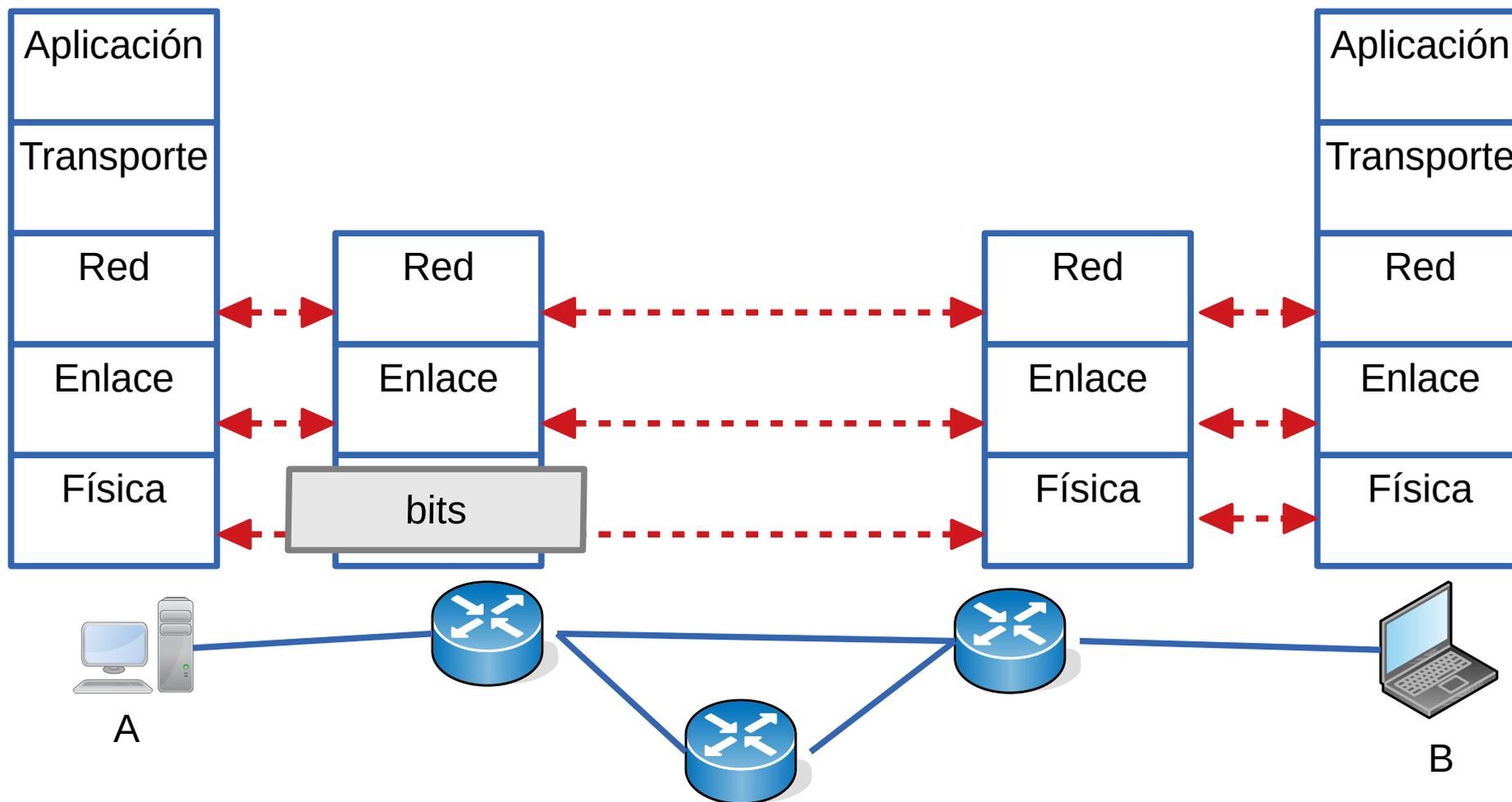
# Encapsulamiento y niveles de abstracción



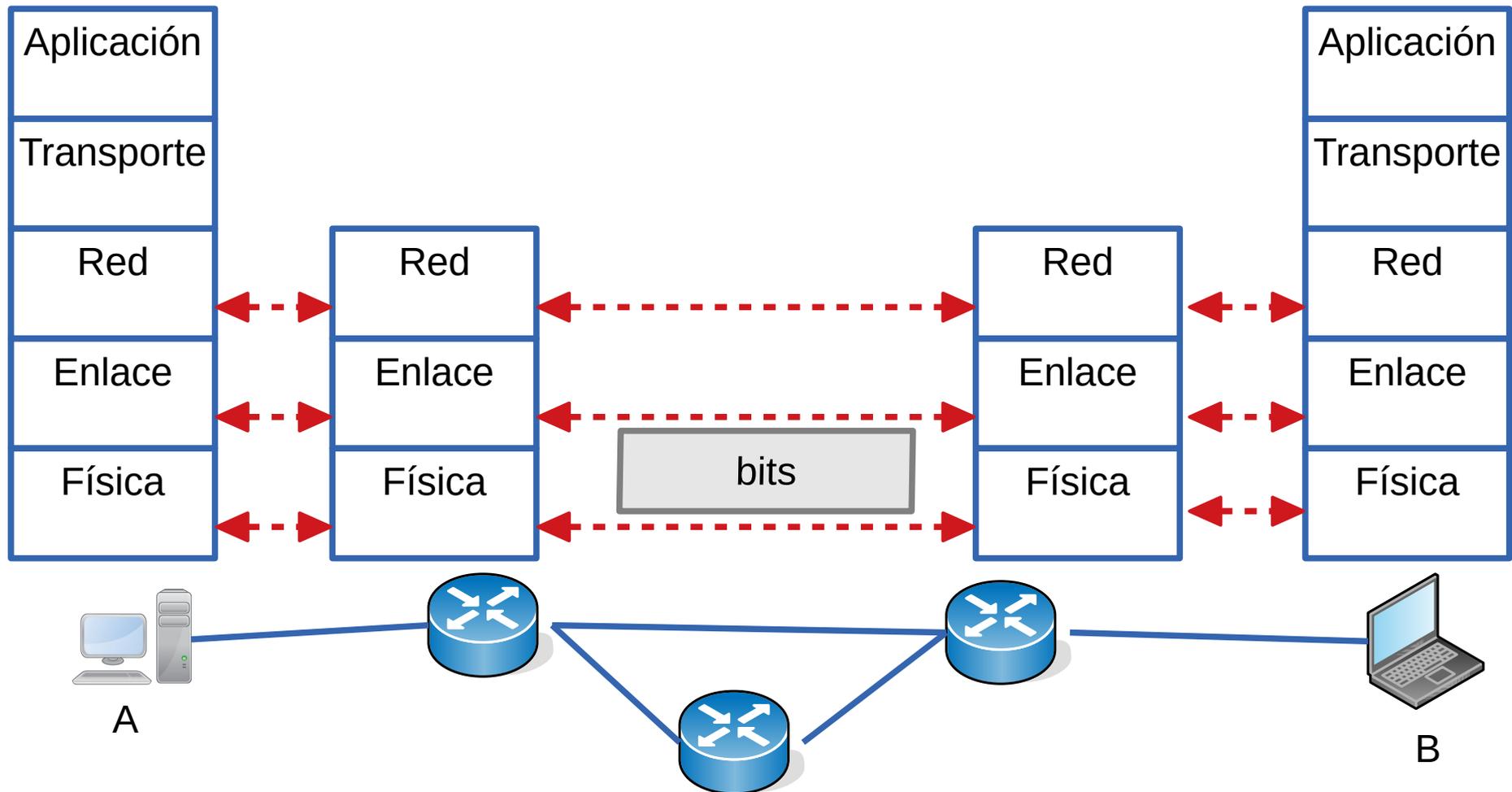
# Encapsulamiento y niveles de abstracción



# Encapsulamiento y niveles de abstracción

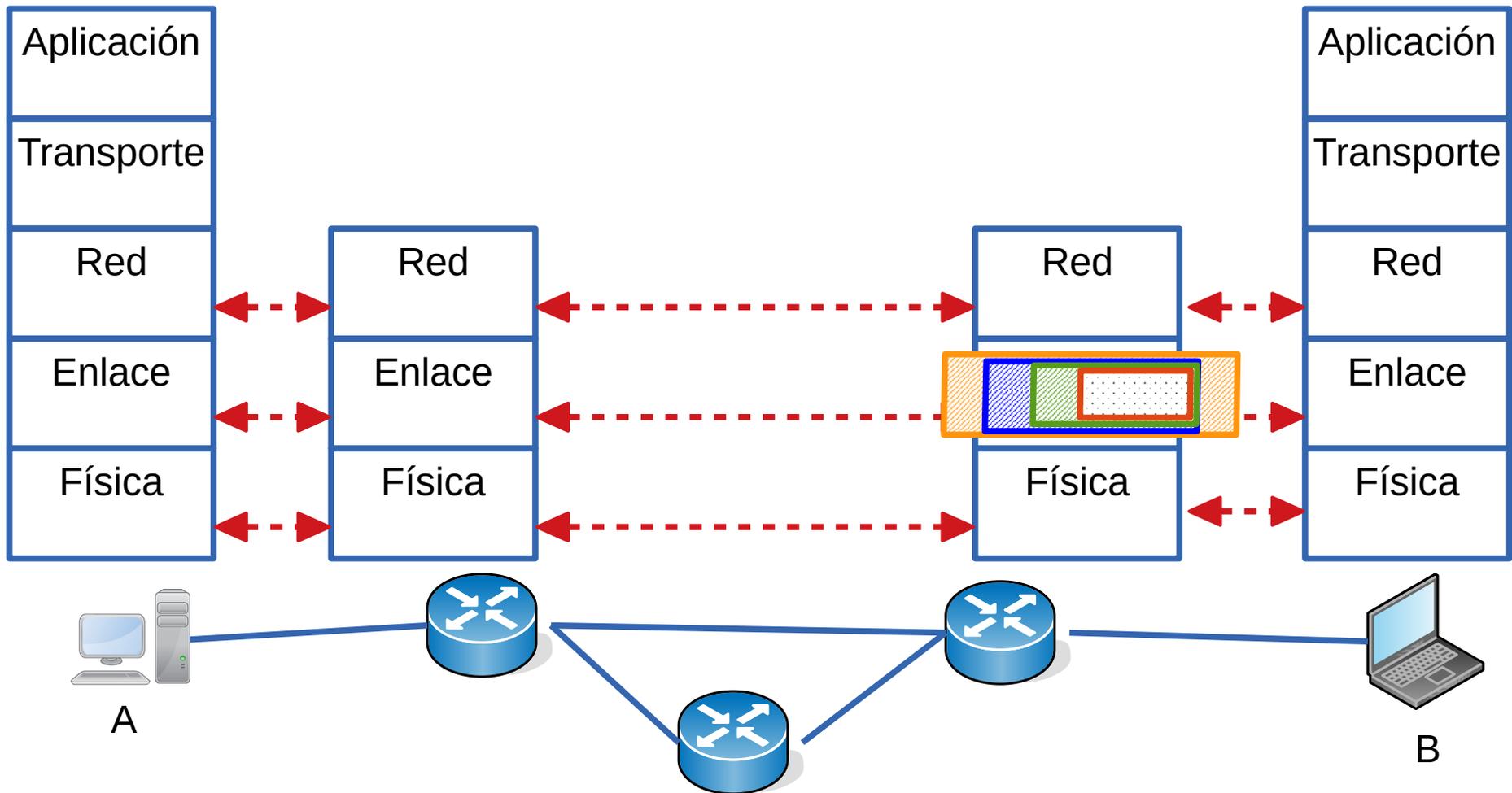


# Encapsulamiento y niveles de abstracción

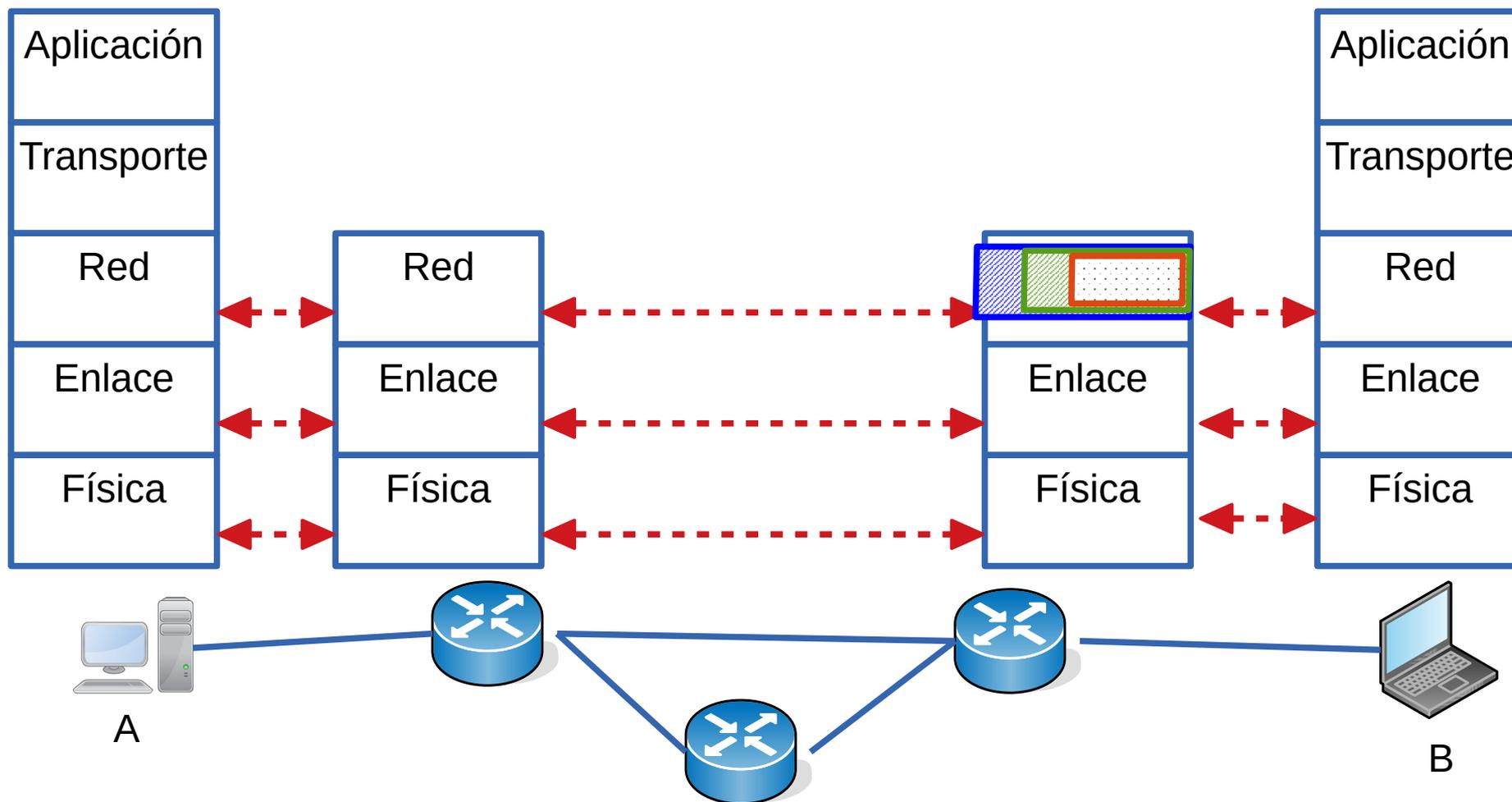




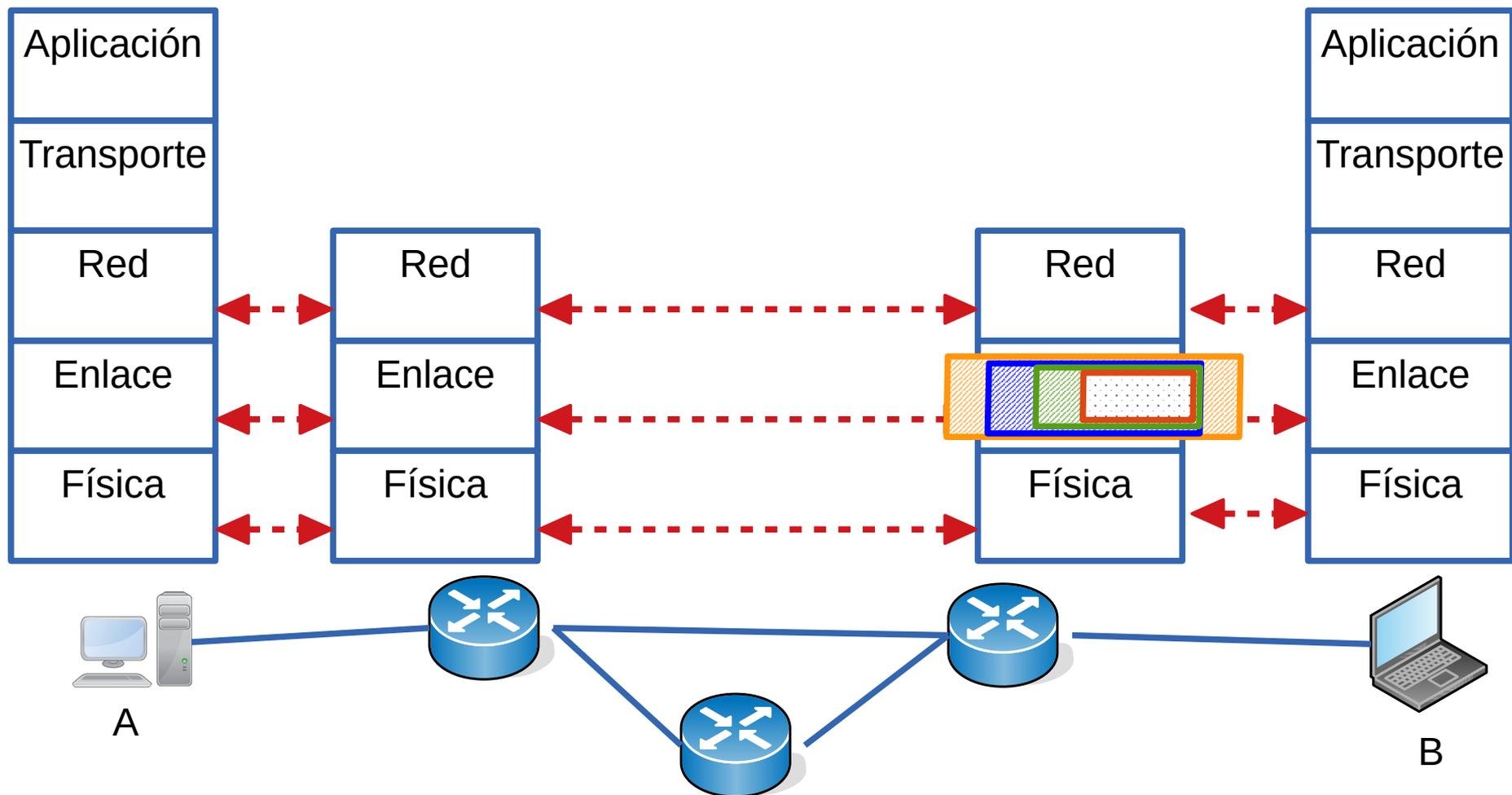
# Encapsulamiento y niveles de abstracción



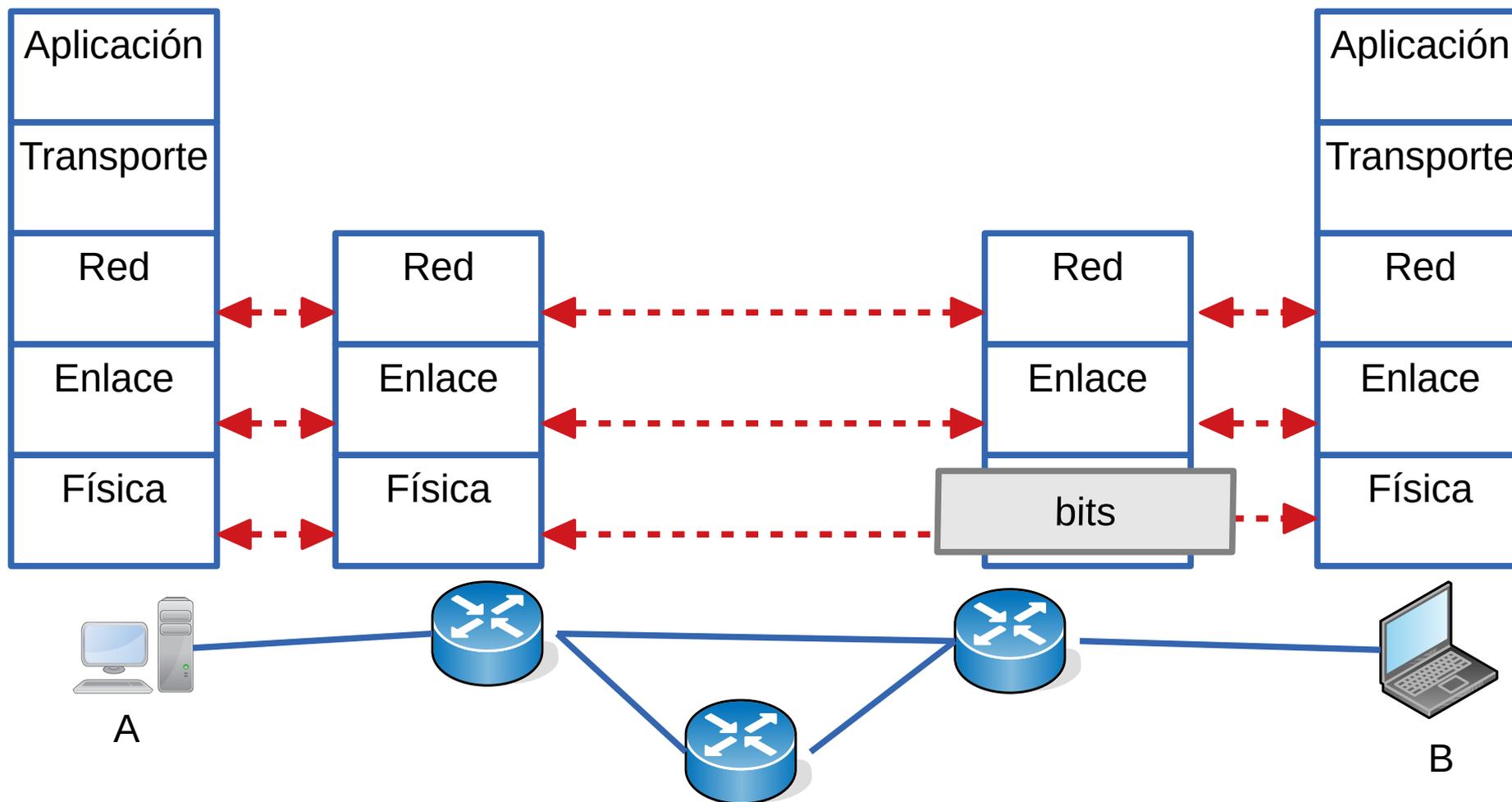
# Encapsulamiento y niveles de abstracción



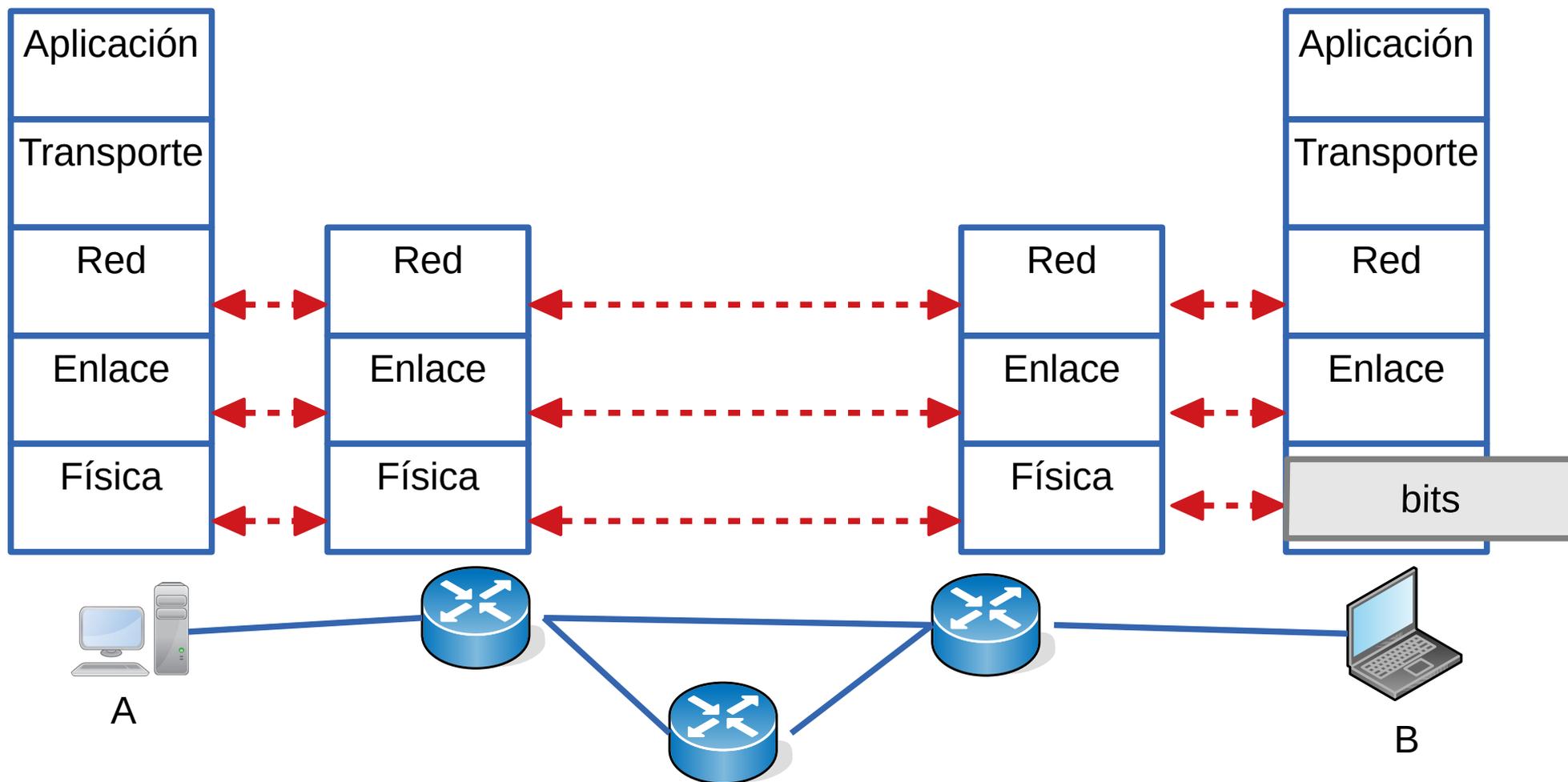
# Encapsulamiento y niveles de abstracción



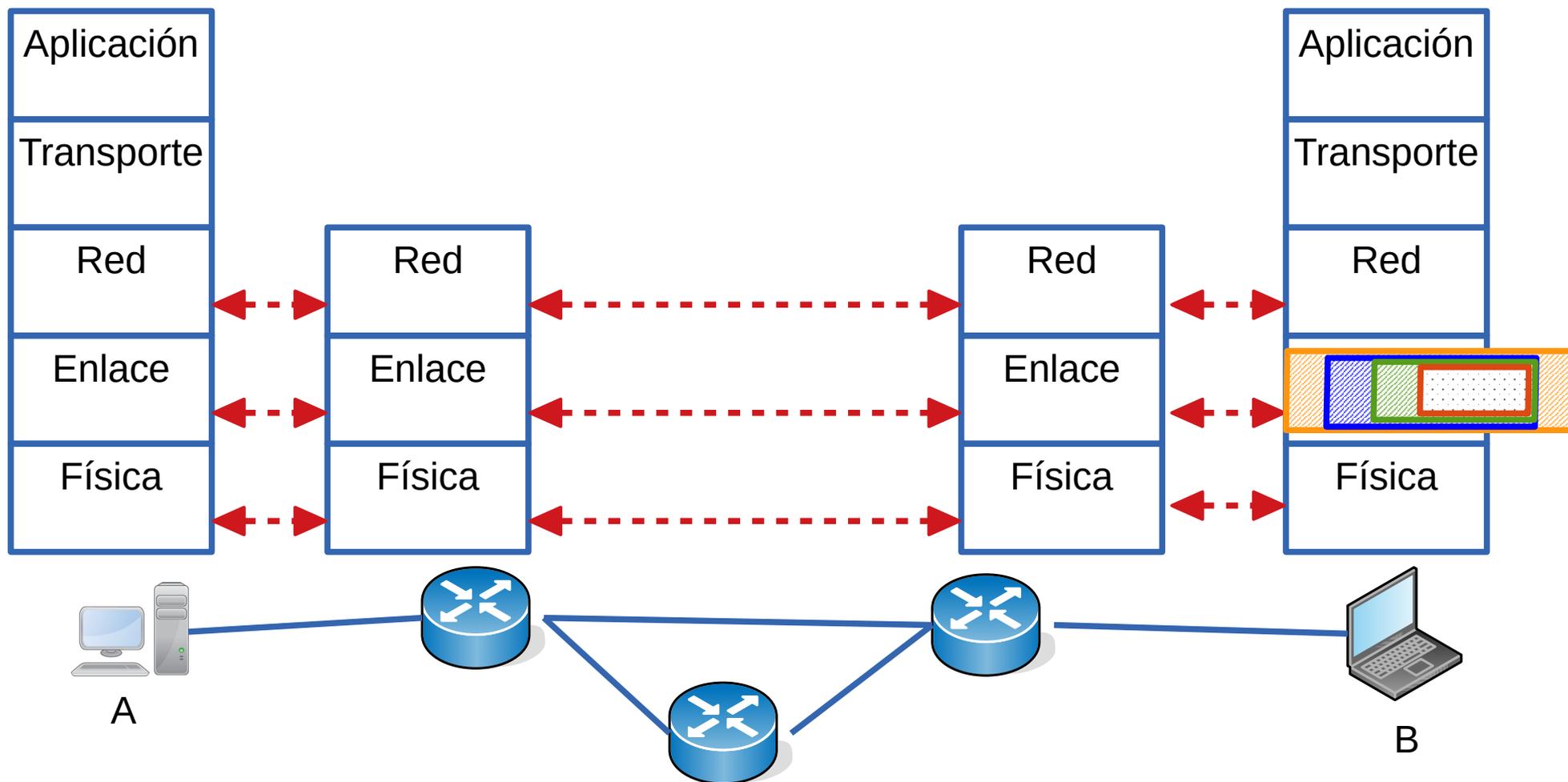
# Encapsulamiento y niveles de abstracción



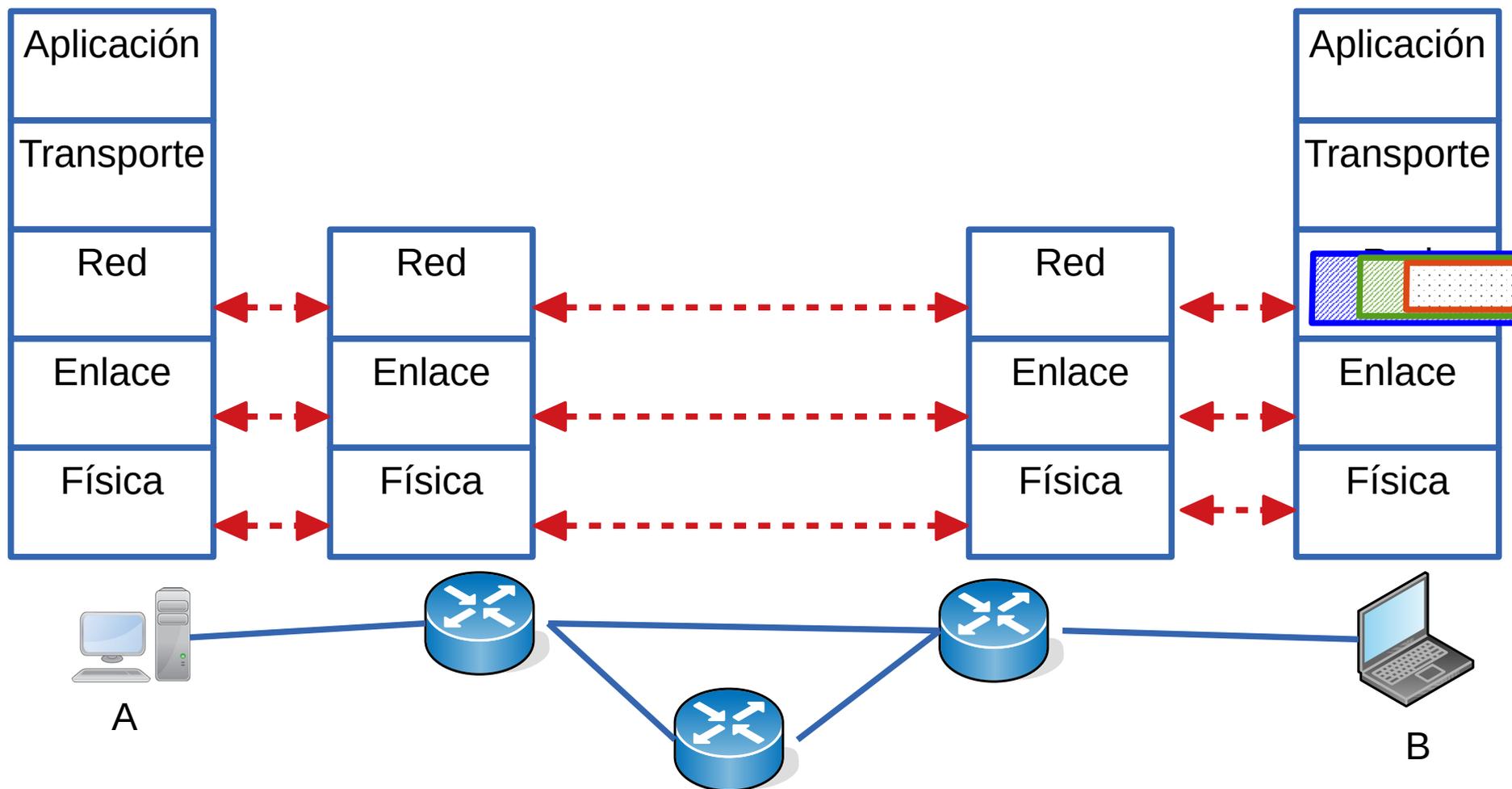
# Encapsulamiento y niveles de abstracción



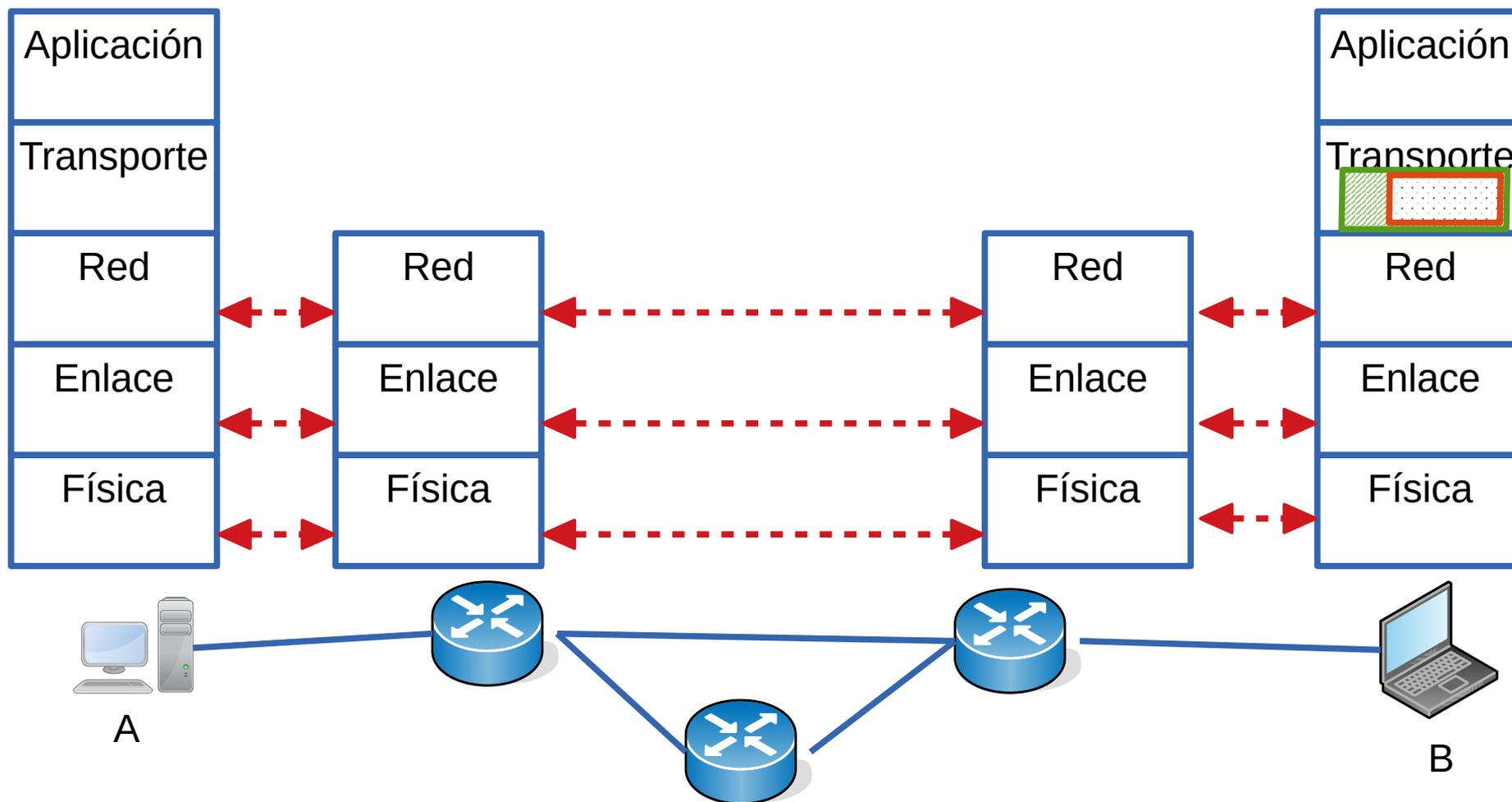
# Encapsulamiento y niveles de abstracción



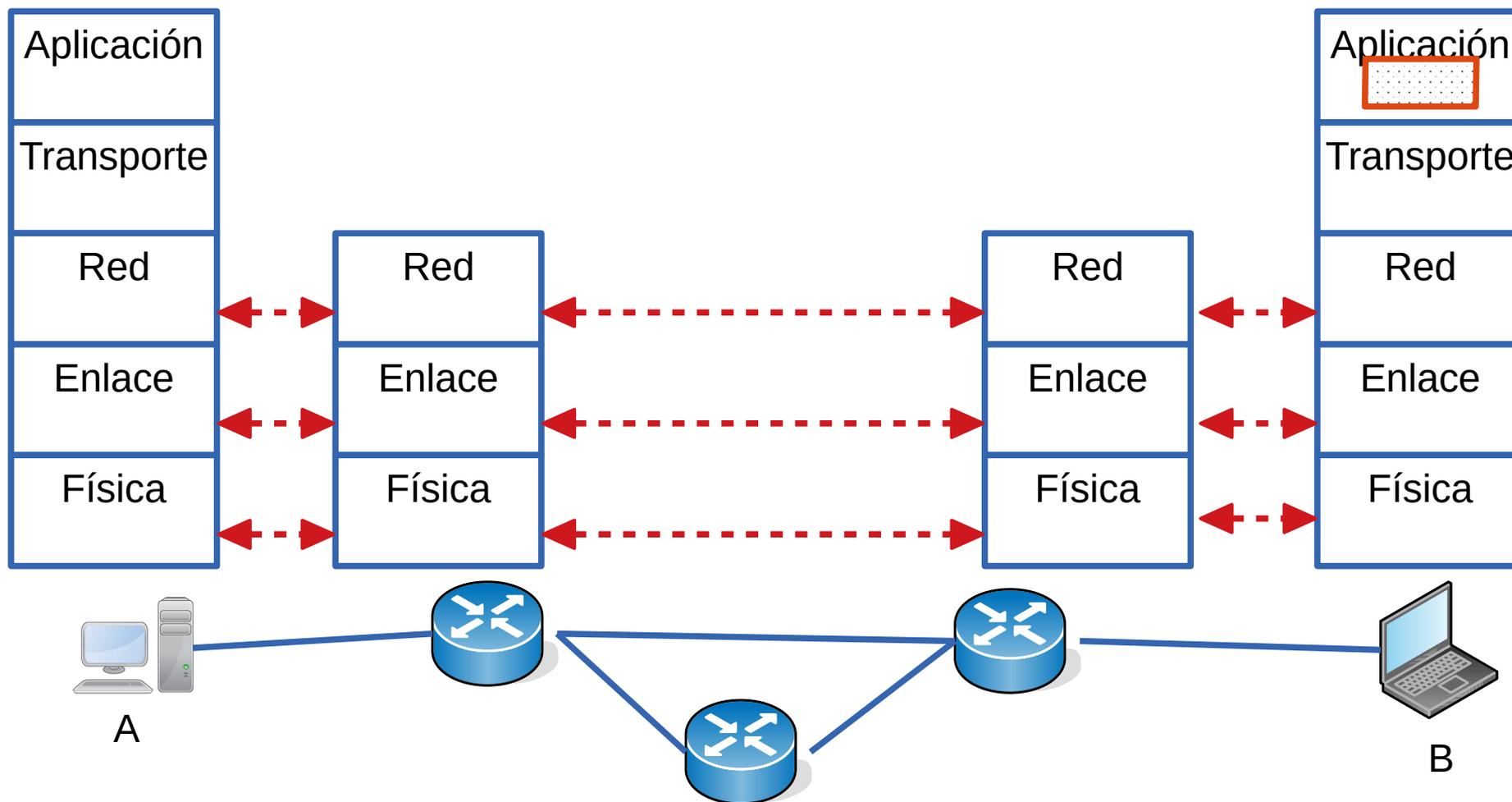
# Encapsulamiento y niveles de abstracción



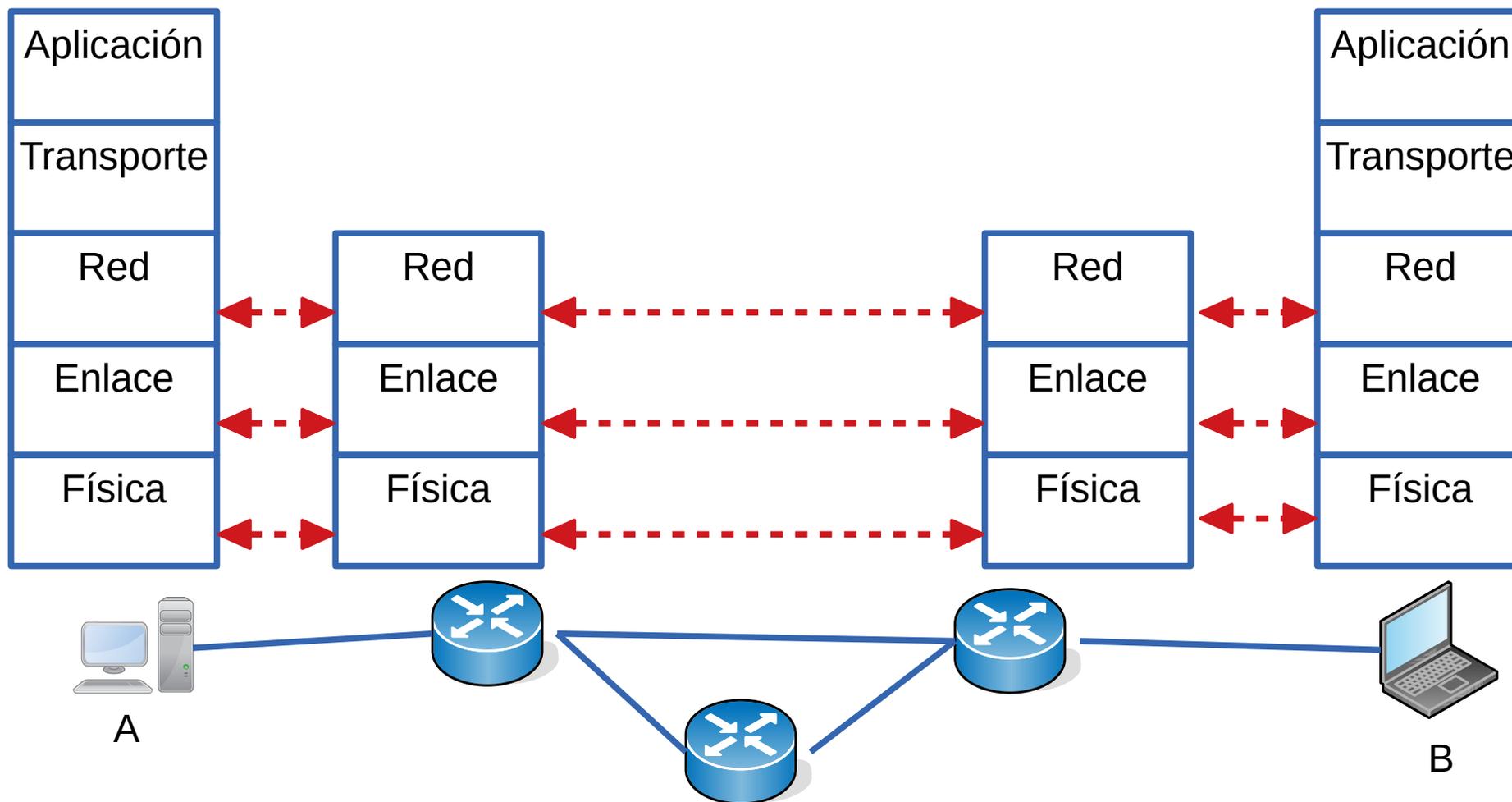
# Encapsulamiento y niveles de abstracción



# Encapsulamiento y niveles de abstracción



# Encapsulamiento y niveles de abstracción

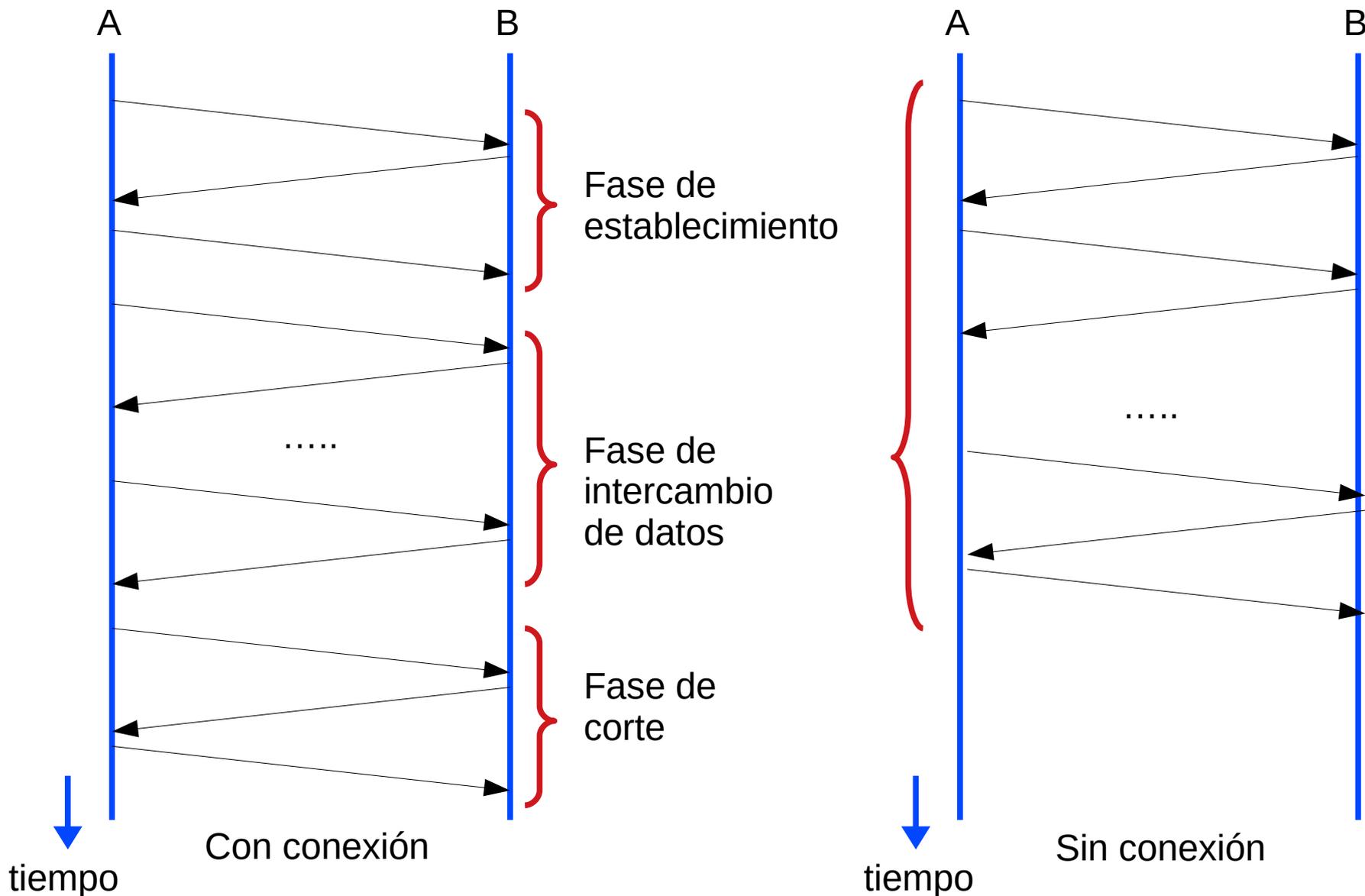


# Protocolo orientado a conexión o no orientado a conexión

- Un protocolo es **orientado a conexión** si para intercambiar datos entre A y B se requiere previamente **establecer una conexión**
  - En un protocolo orientado a conexión se identifican **3 fases** en el intercambio entre A y B
    - Fase de **establecimiento** de conexión:
      - Se intercambian mensajes de control para ponerse de acuerdo
      - En esta fase no se intercambian datos del usuario
      - Se pueden acordar parámetros que regulen el intercambio
    - Fase de **datos**
      - Se intercambian los datos del usuario
    - Fase de **corte**
      - En esta fase se intercambian mensajes de control para cortar la conexión y liberar los recursos asignados a la misma
- Un protocolo es **no orientado a conexión** si no se requiere trámite previo para intercambiar datos
  - En un protocolo no orientado a conexión, cuando un transmisor quiere enviar algo a un potencial receptor, simplemente se lo envía

# Intercambio con conexión y sin conexión

- En orientado a conexión hay 3 fases
- En no orientado a conexión solamente una fase de datos



# Confiable y no confiable

- La confiabilidad de un protocolo está asociado a la calidad de servicio que ofrece
- La **calidad de un servicio** puede medirse en función de parámetros como:
  - La probabilidad de **pérdida** o de **duplicación** de la información
    - Lo que envío, llega al destino? Llega al destino sin errores?
  - El **orden** en que llega la información al destino
    - Los datos llegan al destino en el mismo orden en que se enviaron?
  - El **retardo** entre A y B
    - Cuánto demoran los datos en llegar de A a B. Es predecible esa demora?
  - La **variación del retardo (jitter)**
    - Si envío datos con cierta cadencia, llegan a destino con la misma cadencia?
- Un protocolo es confiable si garantiza que los datos que se envían llegan todos en el mismo orden al receptor, de lo contrario el servicio será no confiable
- Para lograr esto requieren cierta complejidad (para controlar que todo llega, para ordenar, para descartar datos duplicados) y pueden implicar retardos adicionales (para reenviar las cosas que se hayan perdido pierdan)
- Se requiere un **reconocimiento del receptor** (ACK, “acknowledgement”)

# Pérdidas, duplicados y desorden de paquetes

- Los paquetes se pueden perder por:
  - Errores de bits que el control de error detecta y el paquete se descarta
  - Descarte por congestión en equipos intermedios
  - Problemas de ruteo (loops)
  - etc
- En caso de protocolos confiables, si los paquetes se pierden o demoran más de lo previsto, serán retransmitidos
  - Esto puede generar duplicados
- Los paquetes pueden no llegar a destino en el mismo orden en que se transmitieron por varias causas
  - Pérdida y posterior retransmisión
  - Caminos diferentes con diferentes retardos

# Escenarios de duplicados por pérdidas

Transmisión sin errores

A

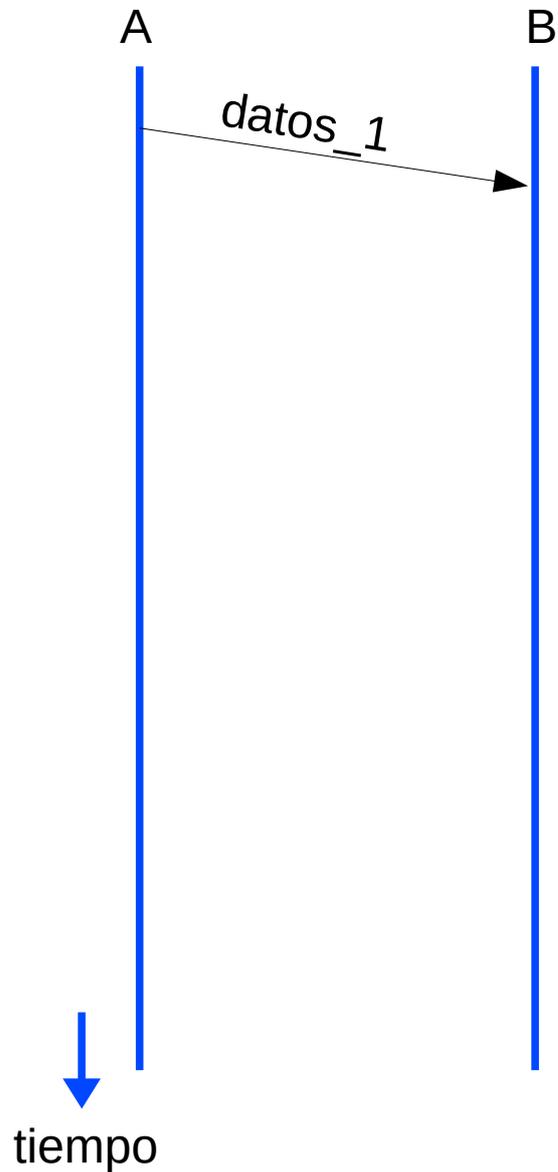
B



tiempo

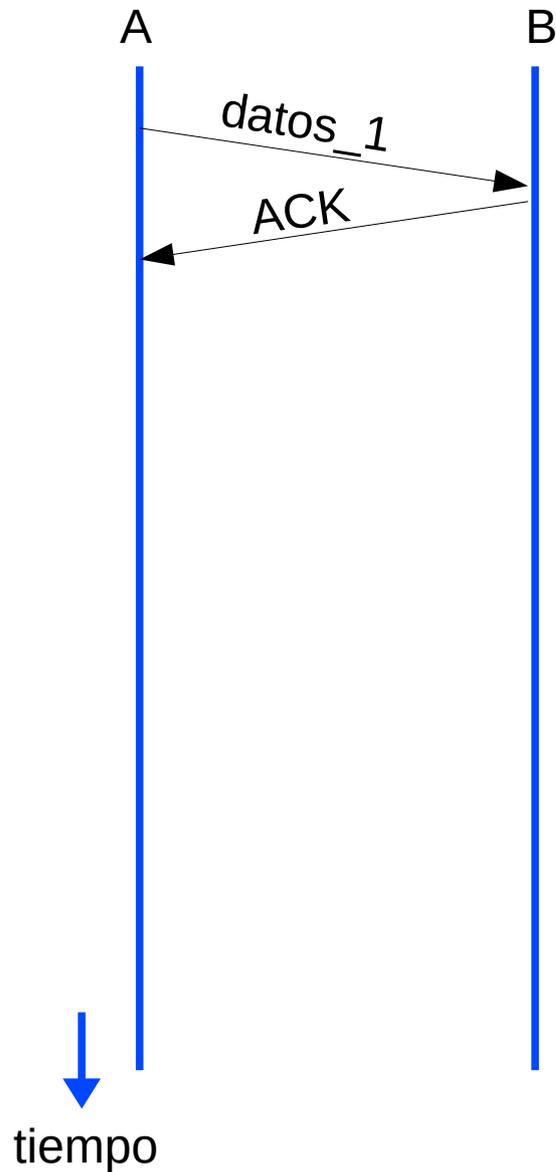
# Escenarios de duplicados por pérdidas

Transmisión sin errores



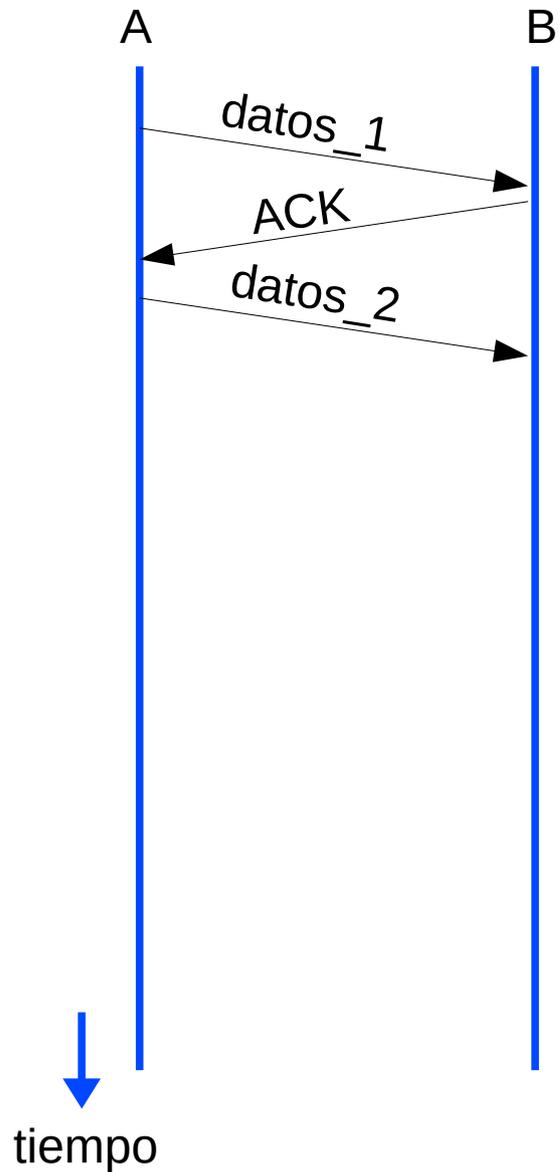
# Escenarios de duplicados por pérdidas

Transmisión sin errores



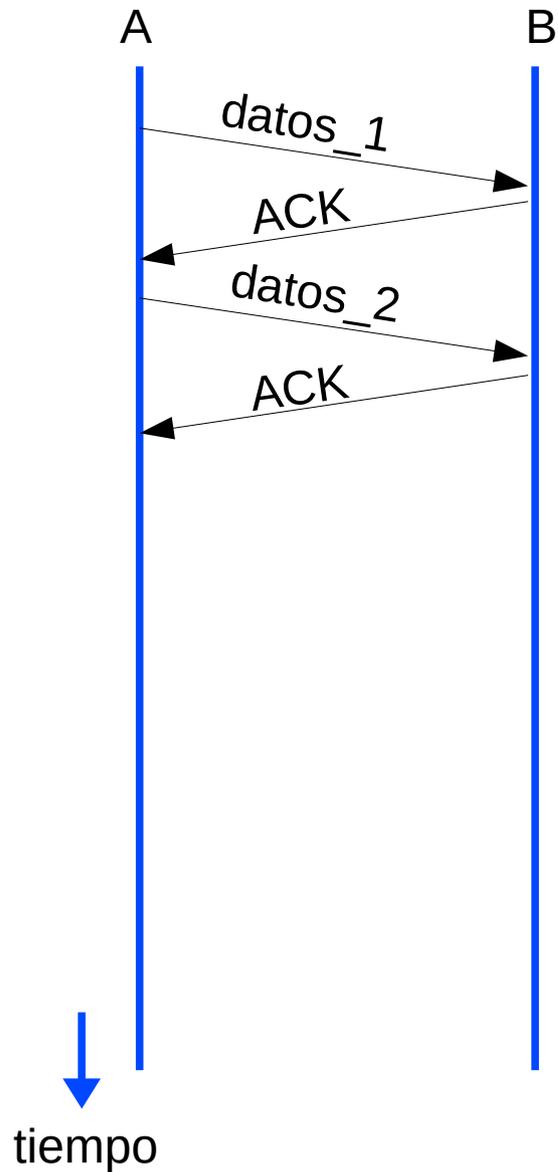
# Escenarios de duplicados por pérdidas

Transmisión sin errores



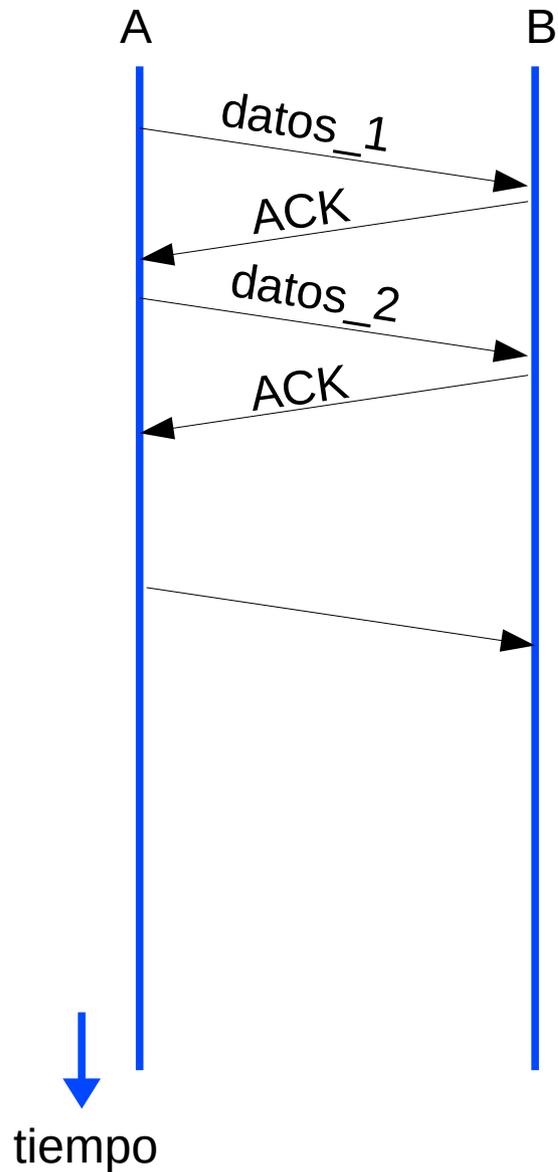
# Escenarios de duplicados por pérdidas

Transmisión sin errores



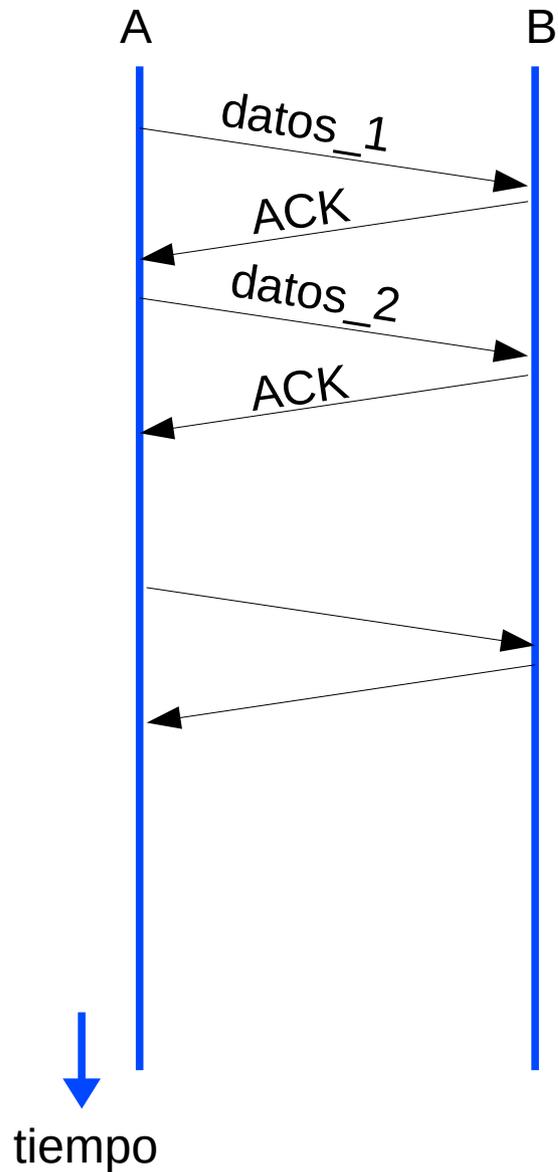
# Escenarios de duplicados por pérdidas

Transmisión sin errores



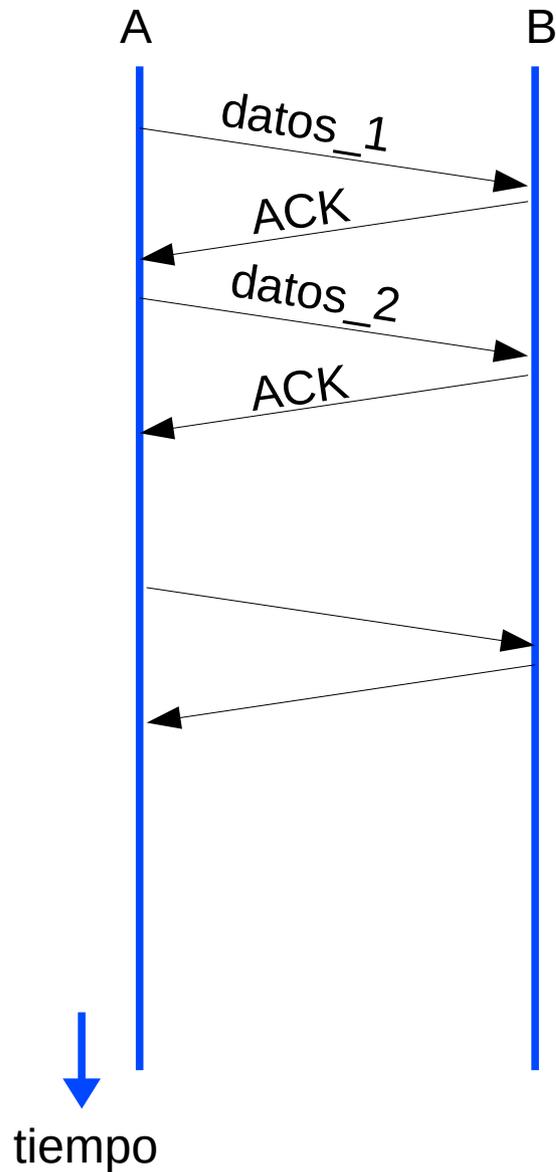
# Escenarios de duplicados por pérdidas

Transmisión sin errores

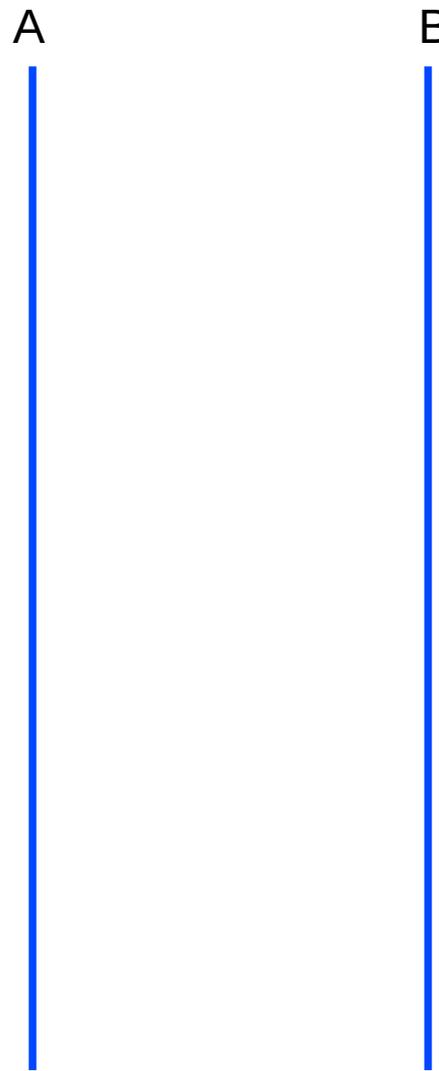


# Escenarios de duplicados por pérdidas

Transmisión sin errores

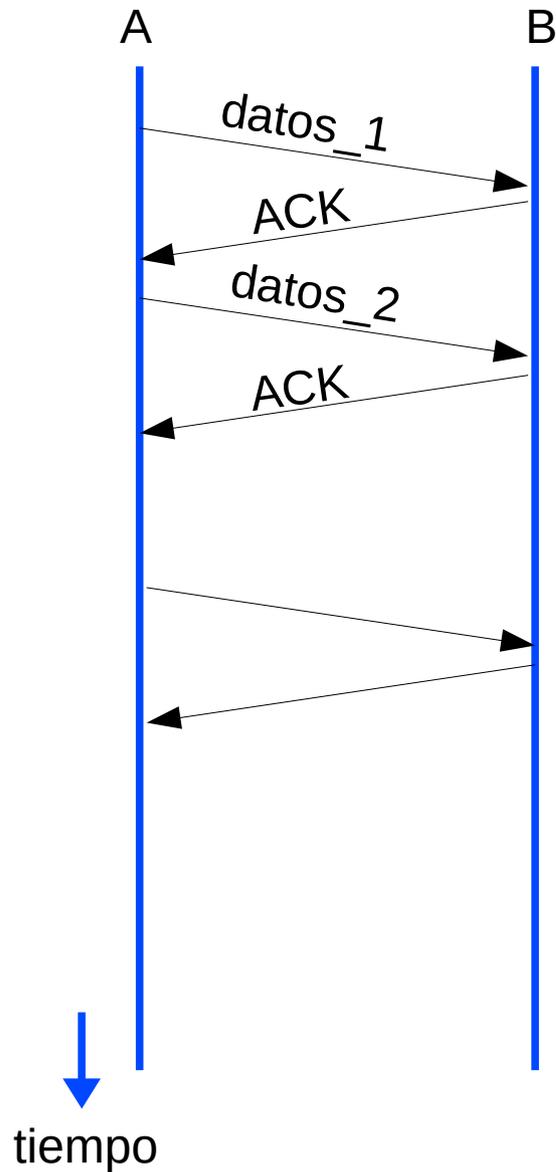


Errores y temporizadores

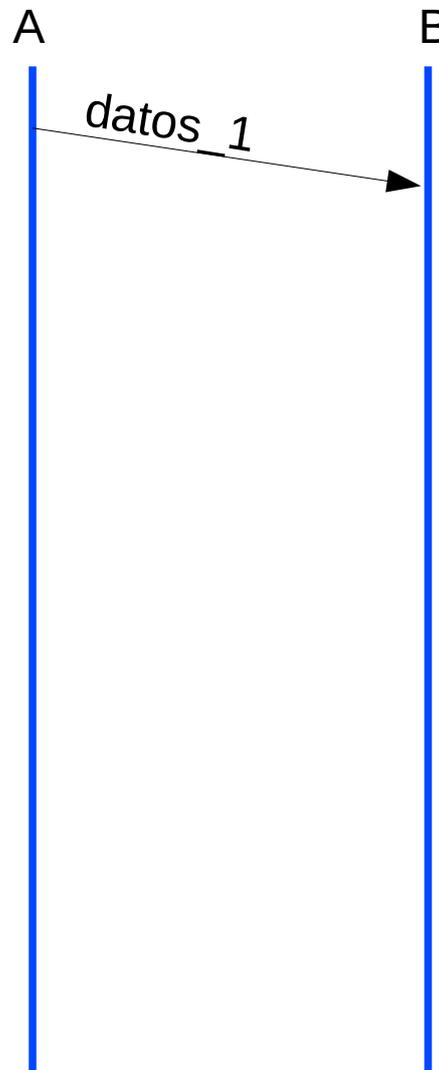


# Escenarios de duplicados por pérdidas

Transmisión sin errores

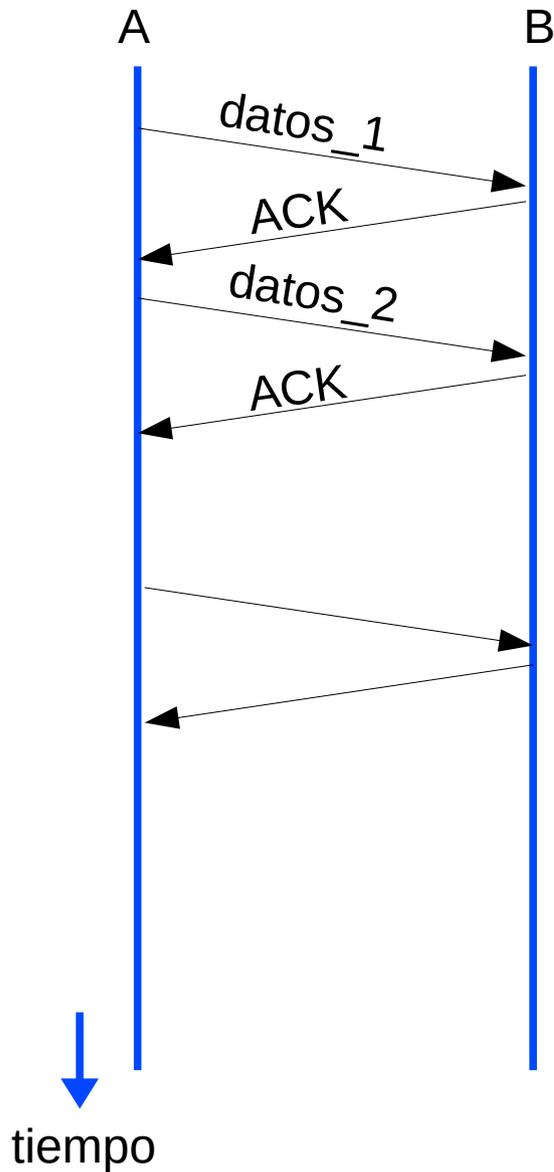


Errores y temporizadores

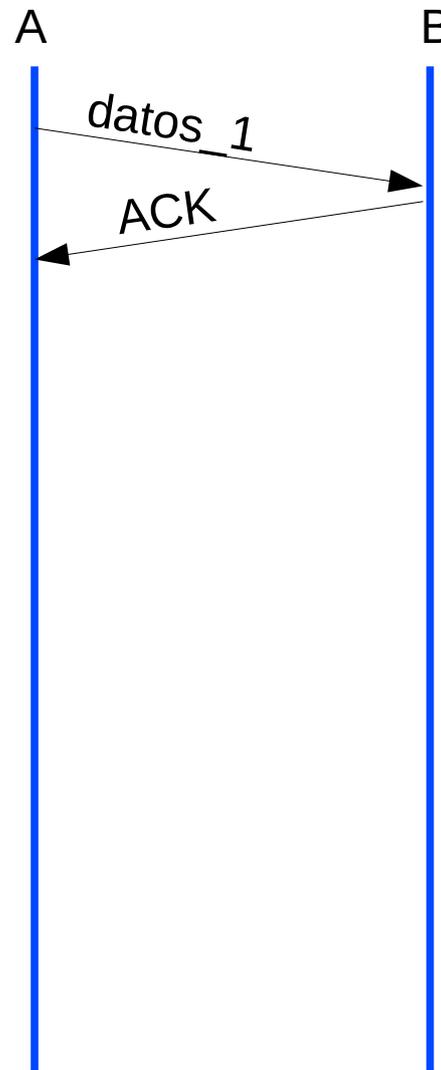


# Escenarios de duplicados por pérdidas

Transmisión sin errores

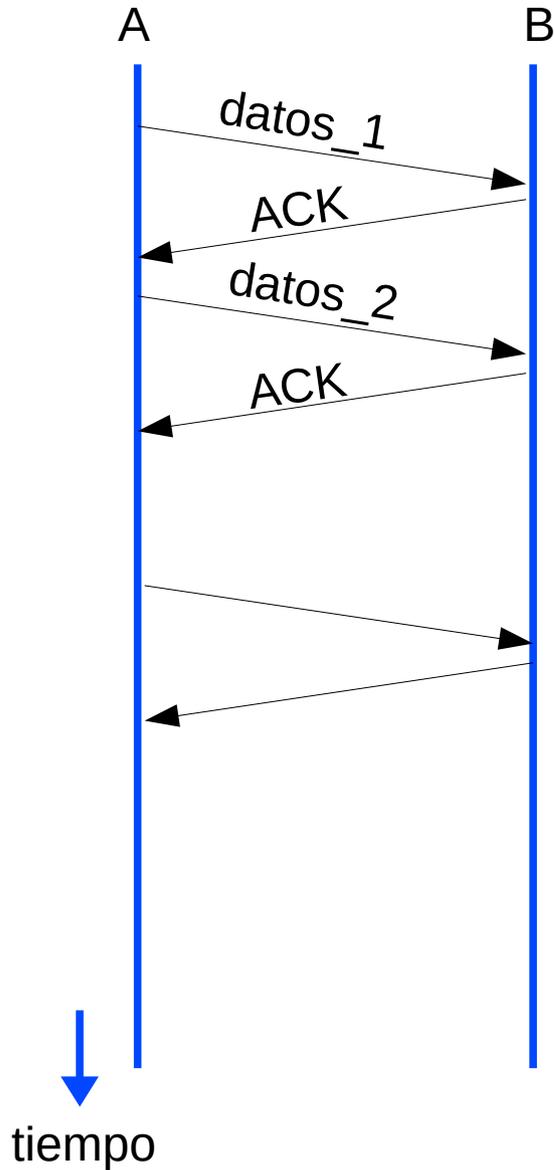


Errores y temporizadores

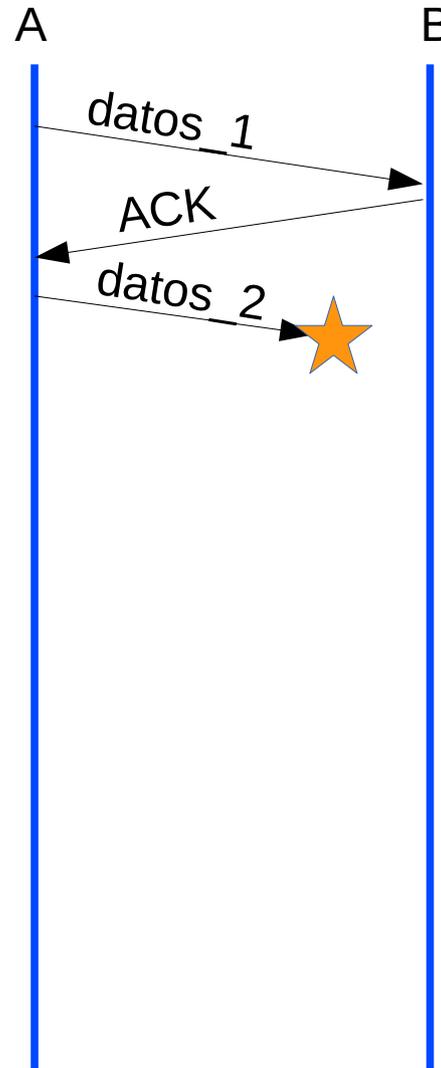


# Escenarios de duplicados por pérdidas

Transmisión sin errores

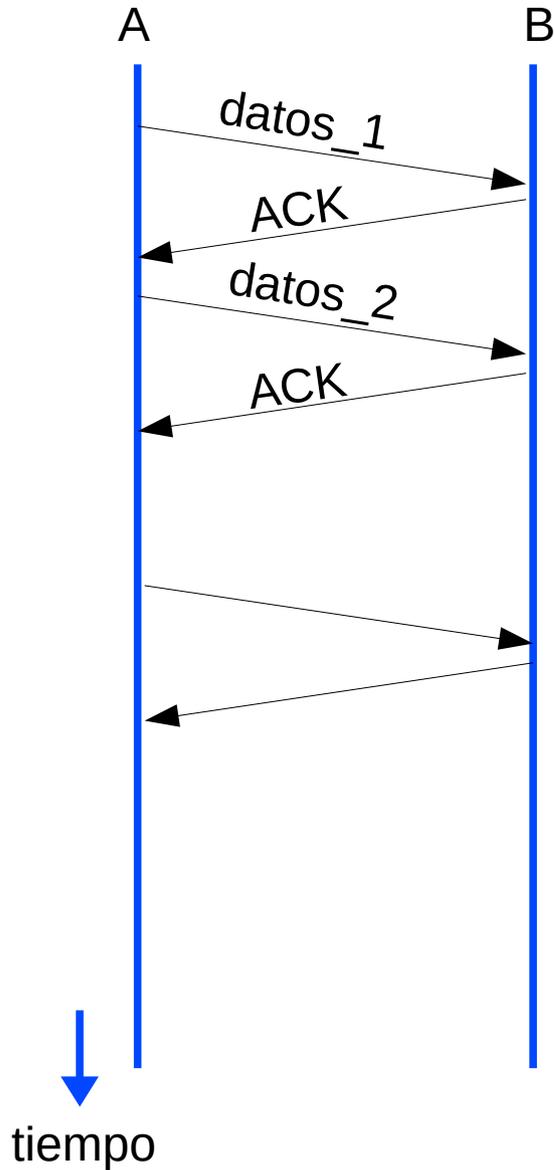


Errores y temporizadores

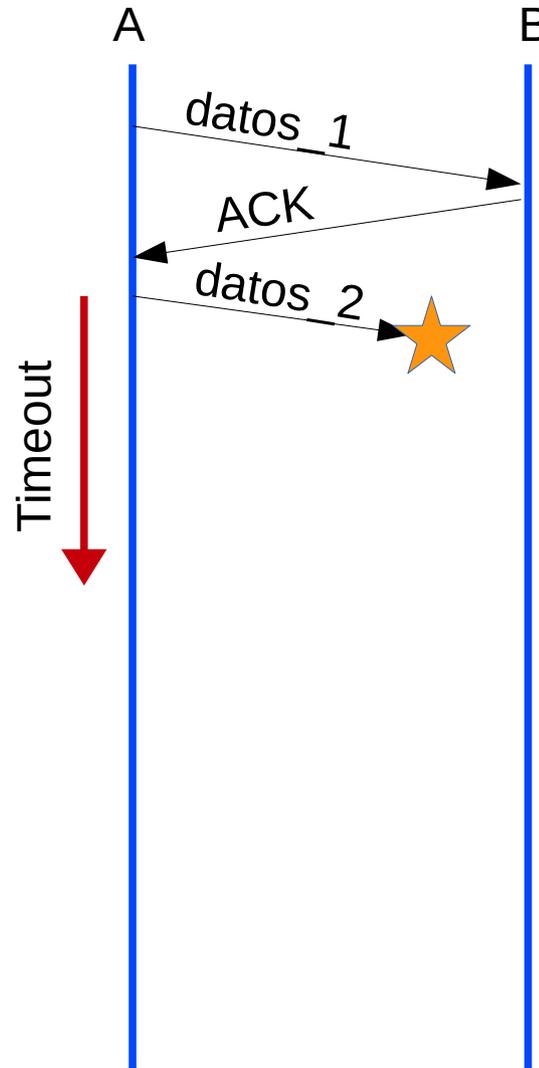


# Escenarios de duplicados por pérdidas

Transmisión sin errores

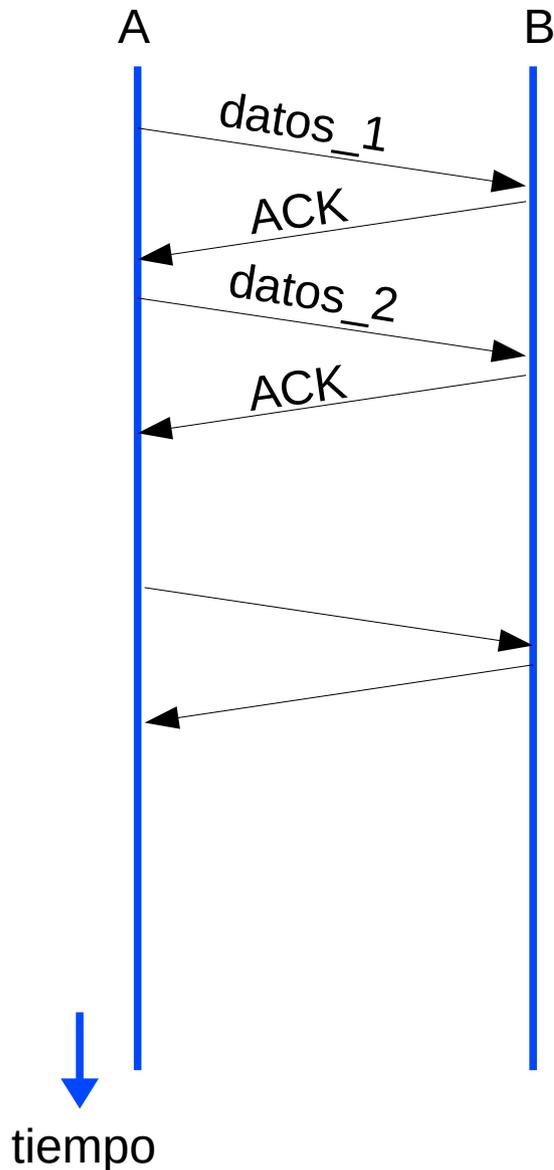


Errores y temporizadores

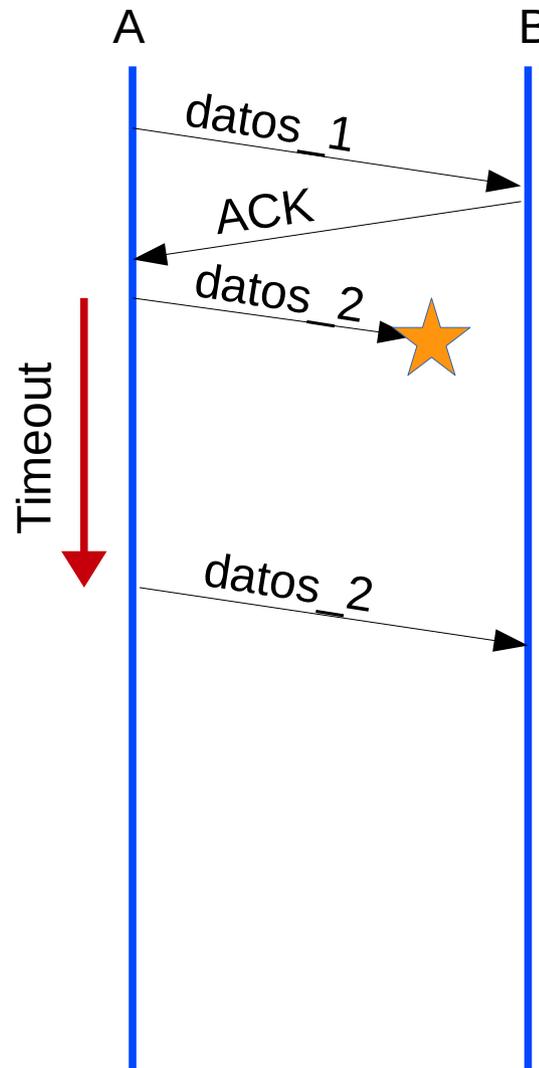


# Escenarios de duplicados por pérdidas

Transmisión sin errores

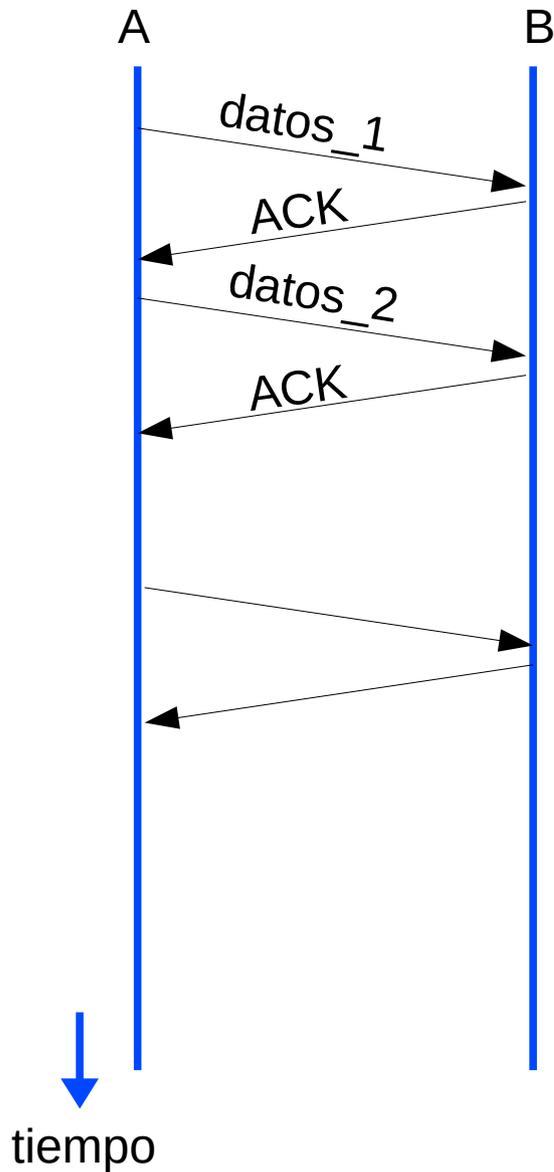


Errores y temporizadores

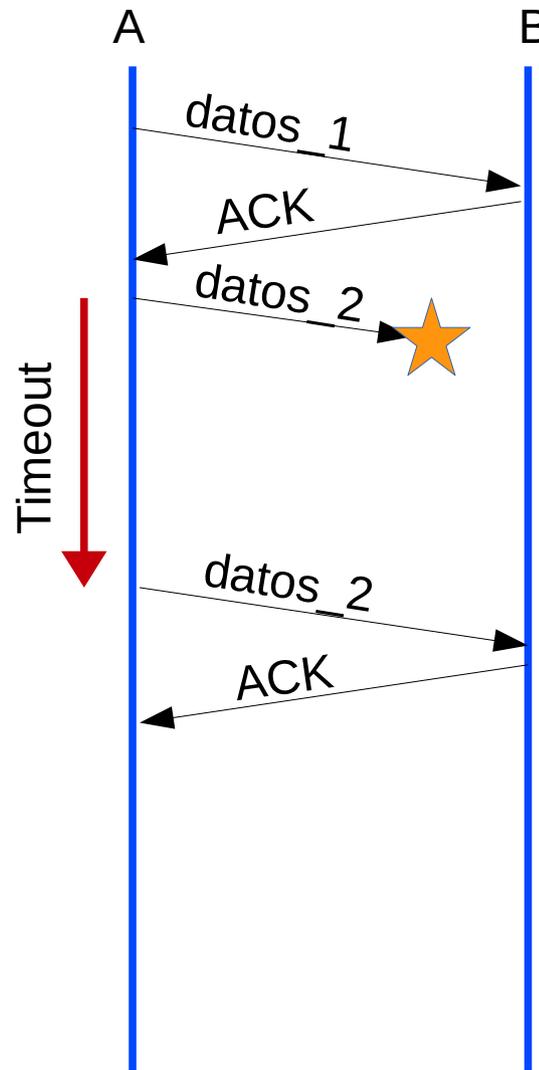


# Escenarios de duplicados por pérdidas

Transmisión sin errores

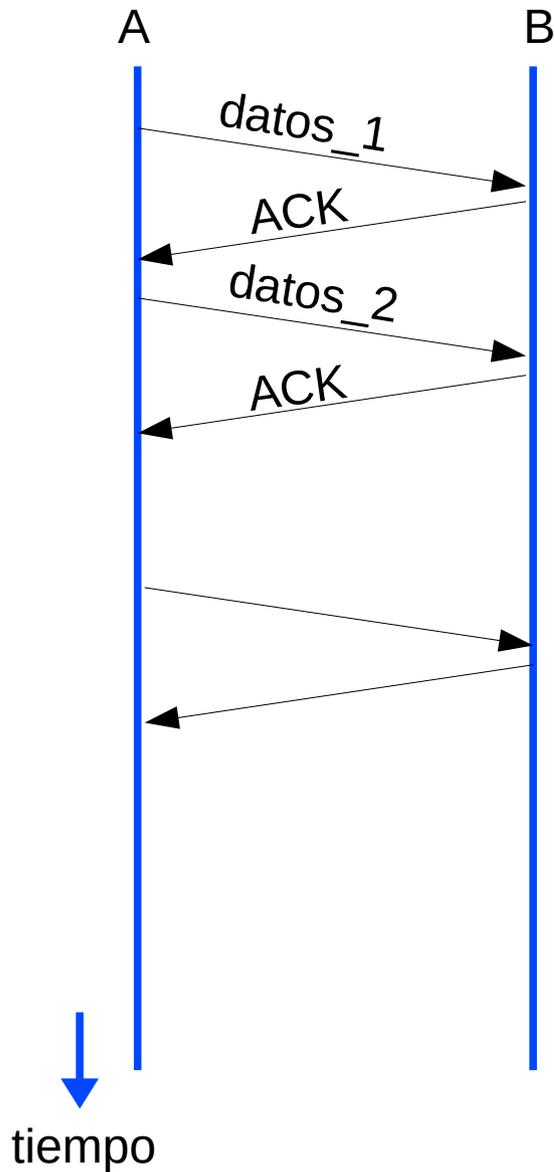


Errores y temporizadores

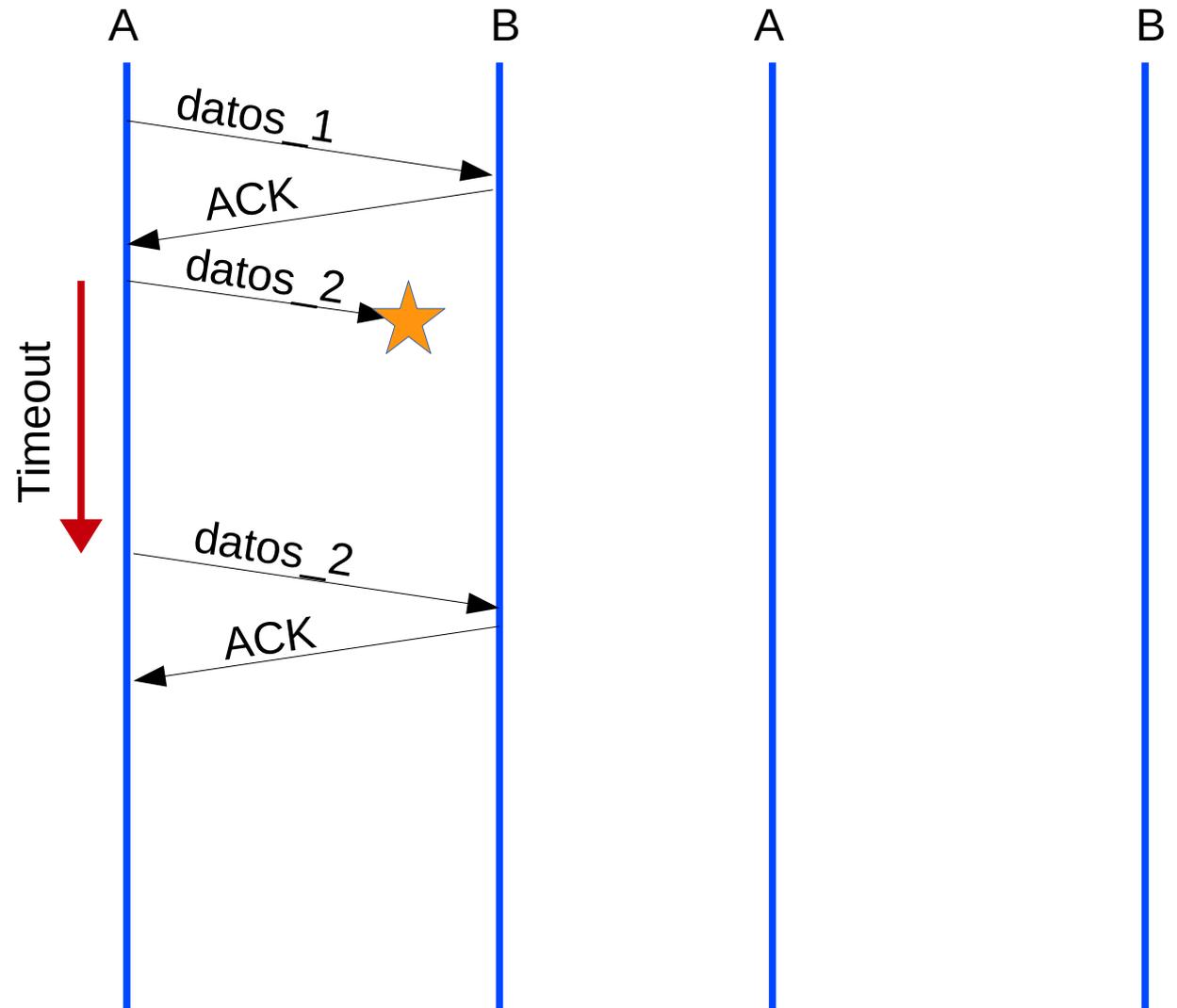


# Escenarios de duplicados por pérdidas

Transmisión sin errores

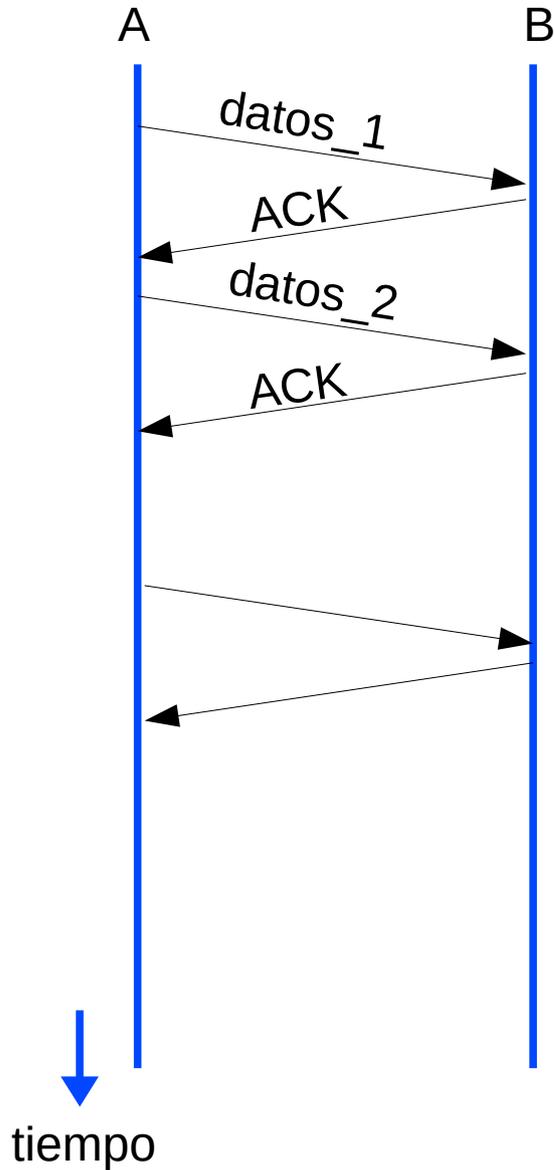


Errores y temporizadores

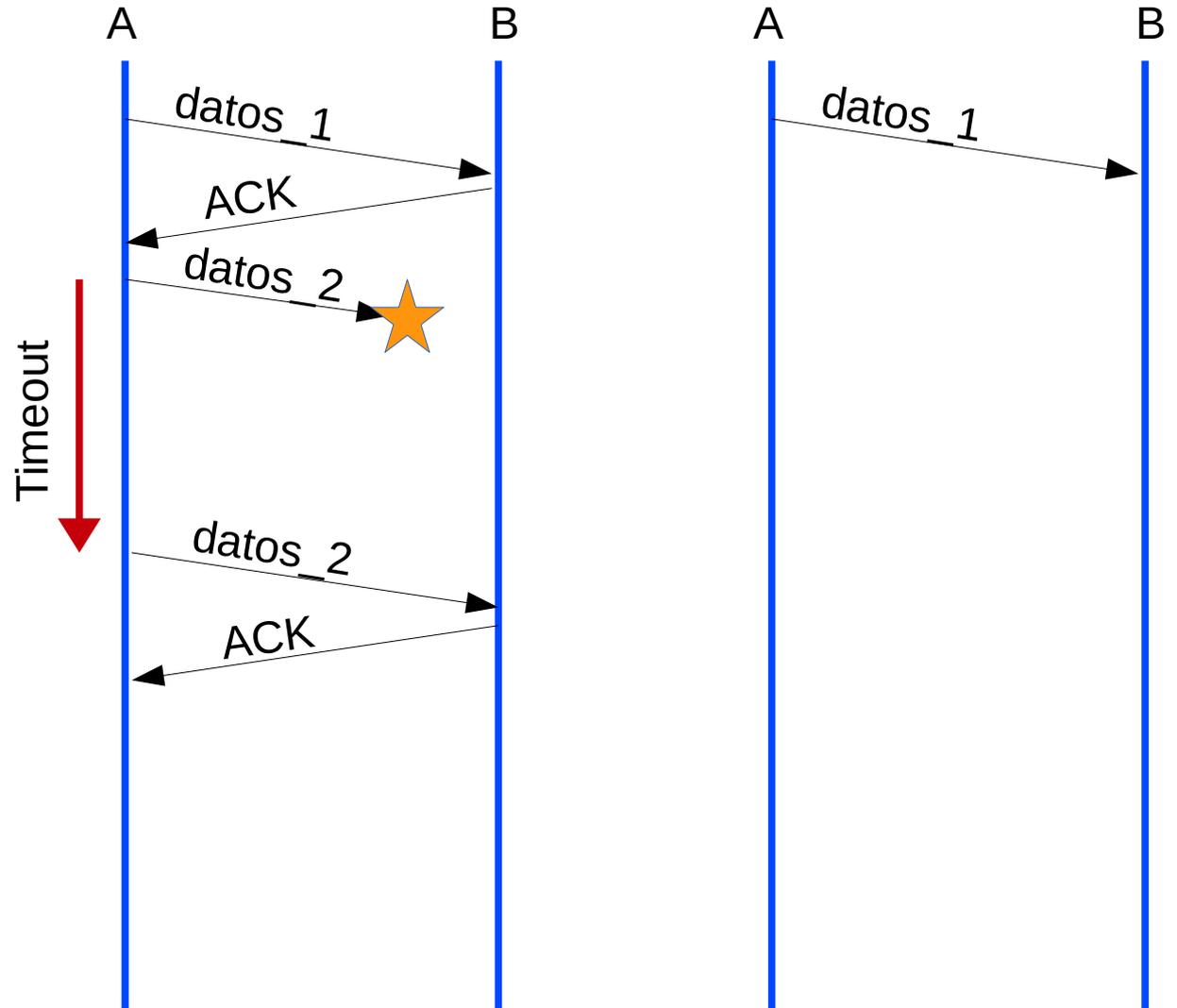


# Escenarios de duplicados por pérdidas

Transmisión sin errores

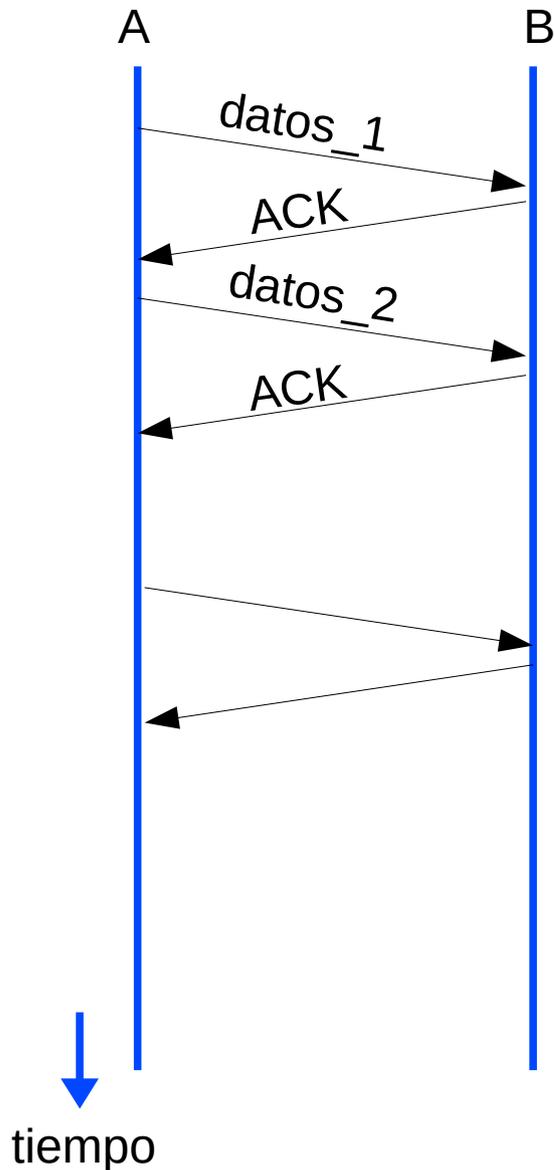


Errores y temporizadores

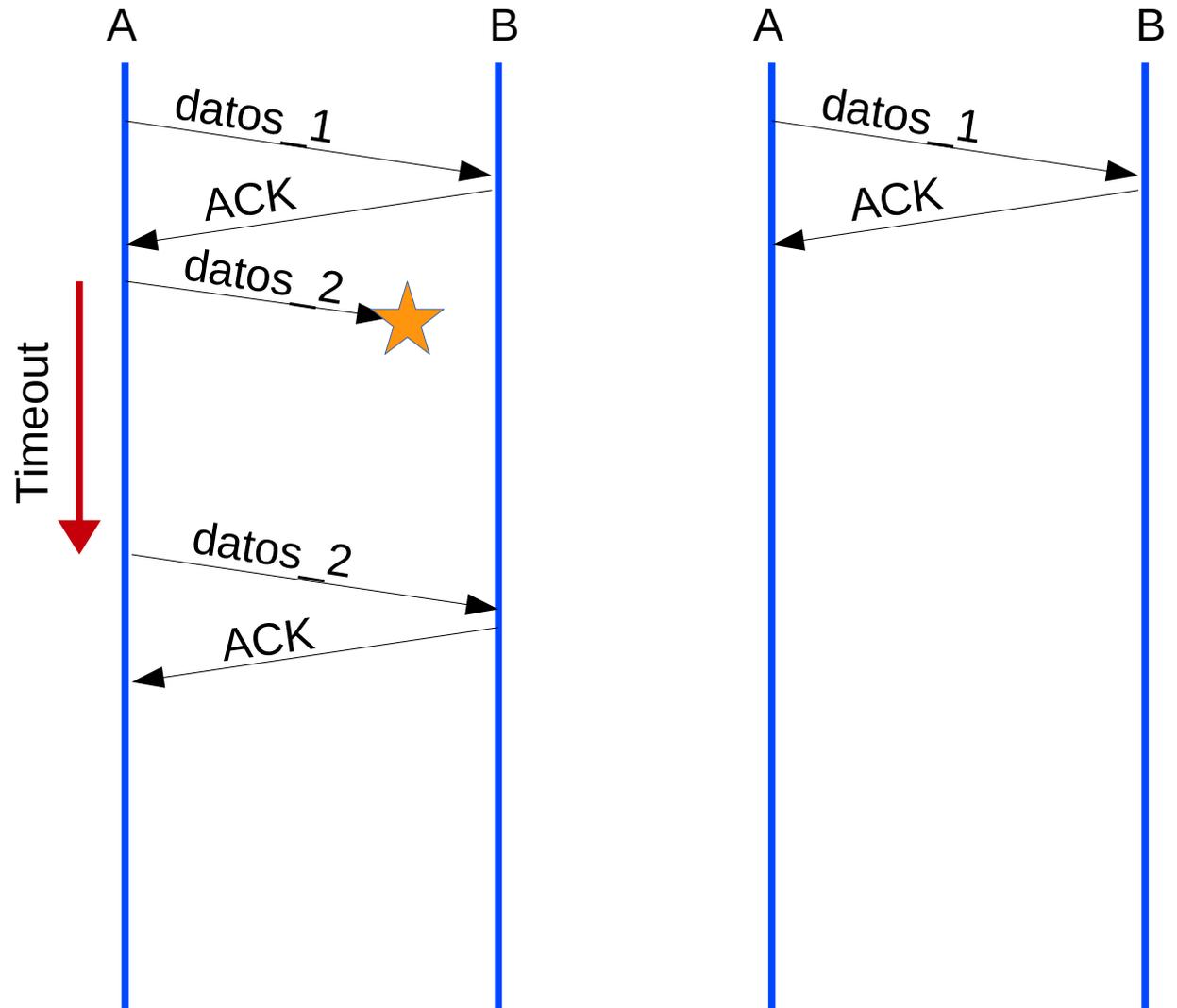


# Escenarios de duplicados por pérdidas

Transmisión sin errores

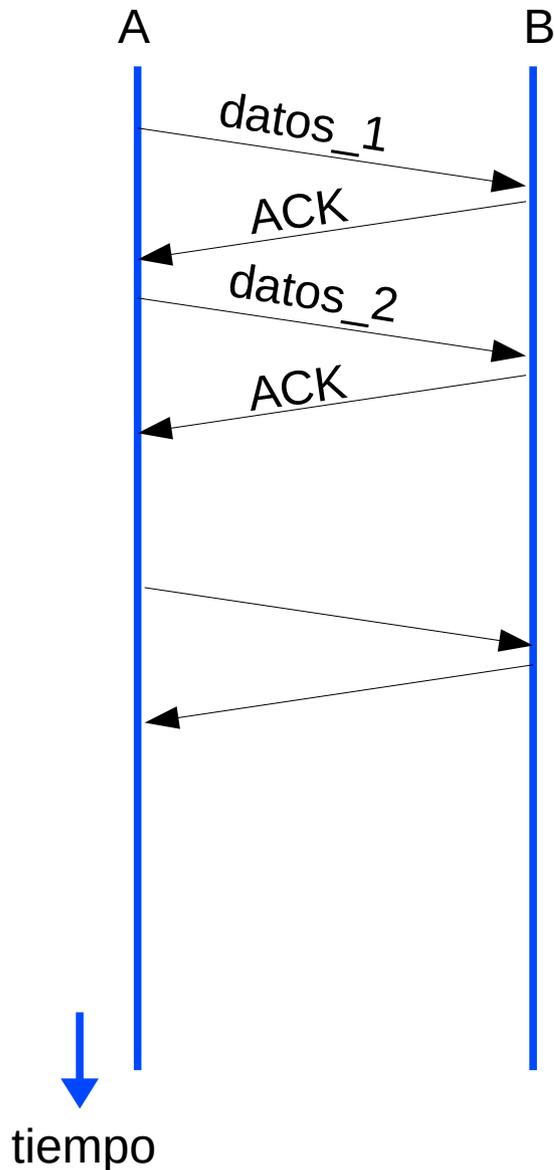


Errores y temporizadores

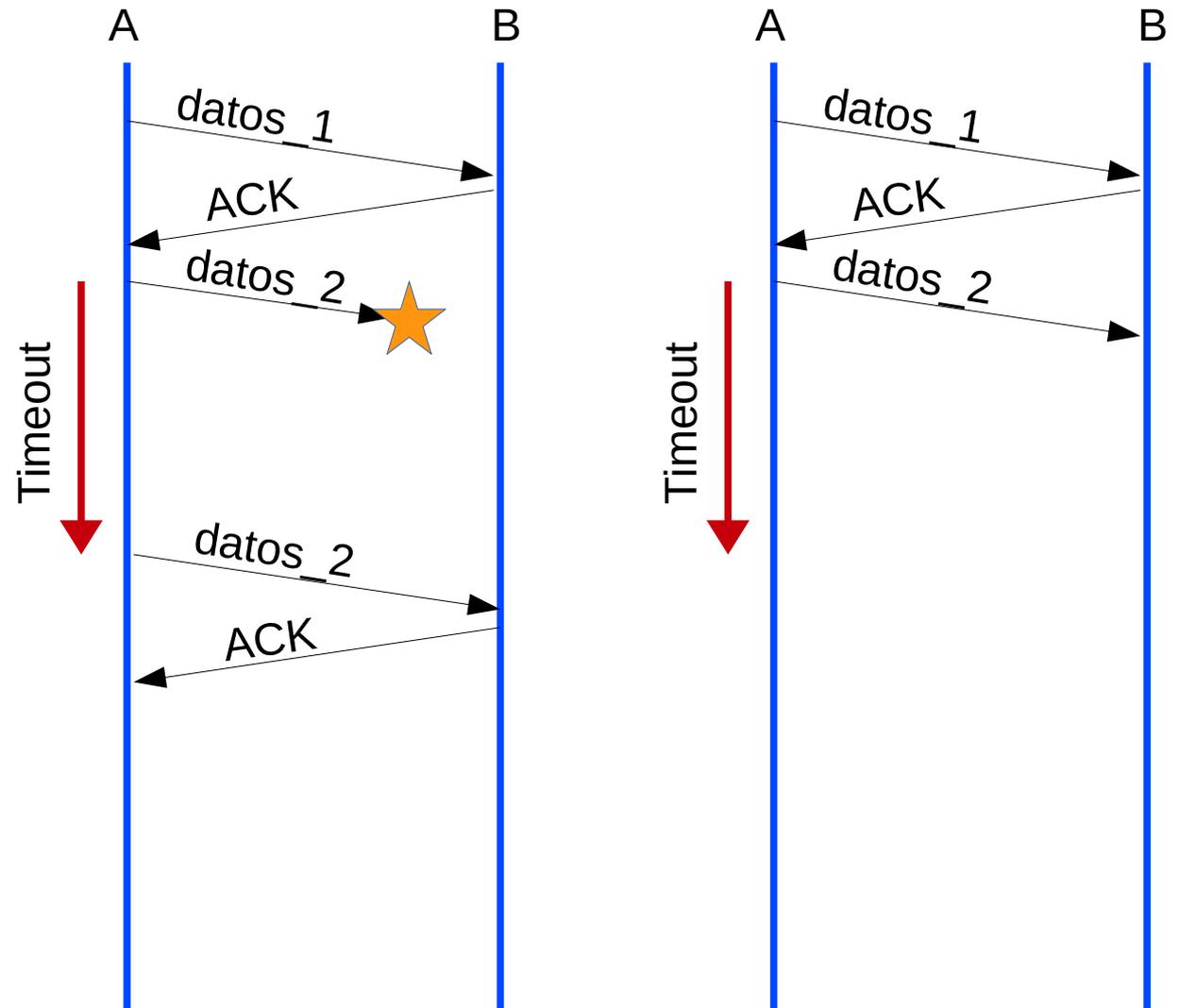


# Escenarios de duplicados por pérdidas

Transmisión sin errores

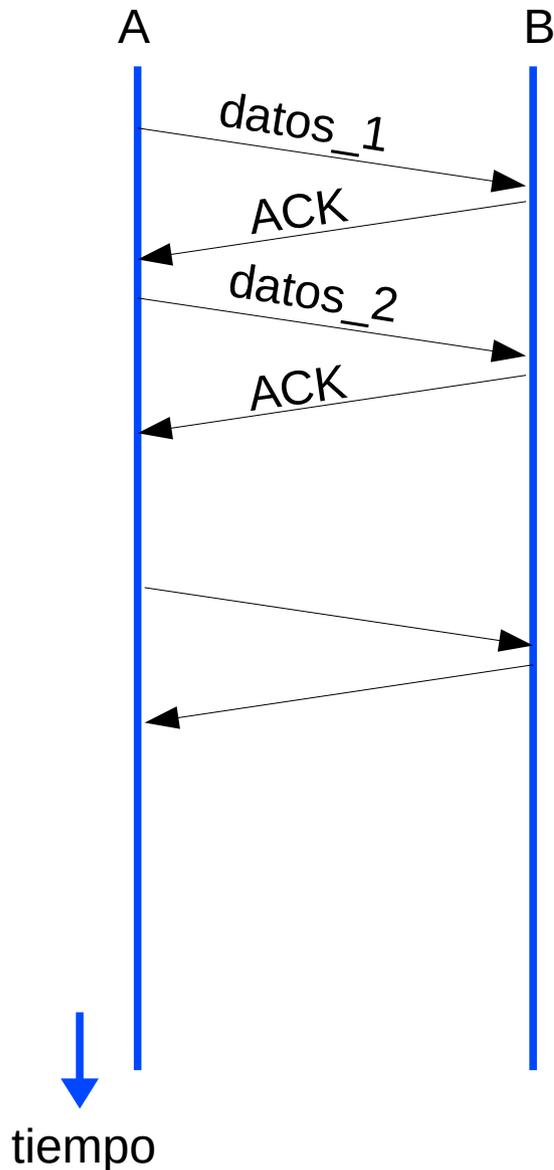


Errores y temporizadores

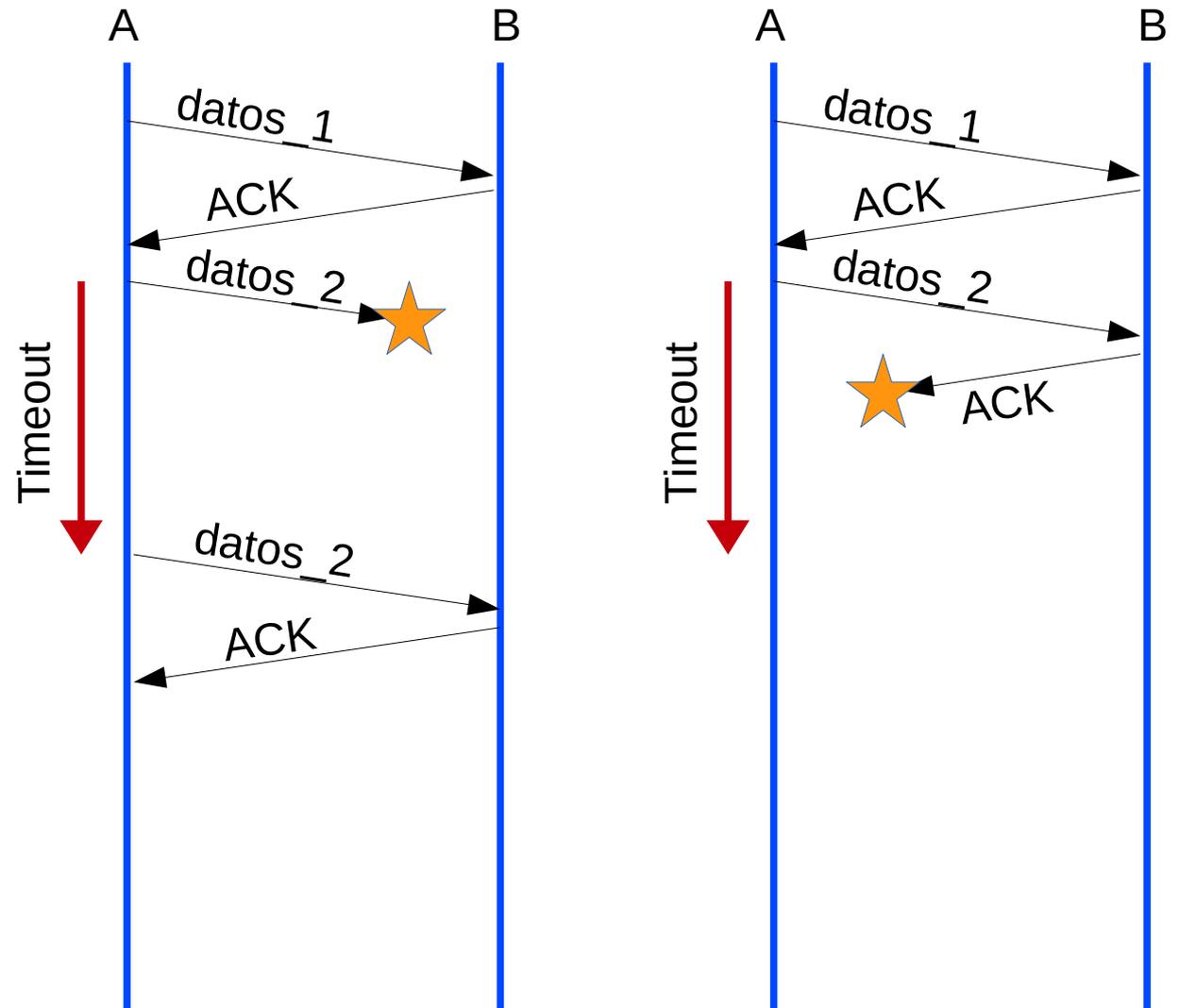


# Escenarios de duplicados por pérdidas

Transmisión sin errores

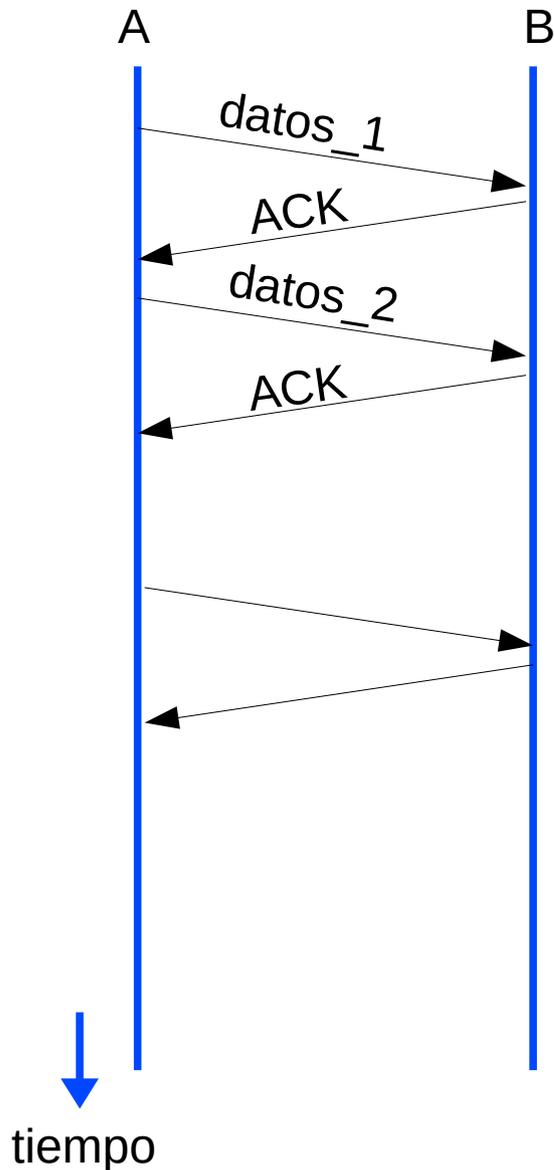


Errores y temporizadores

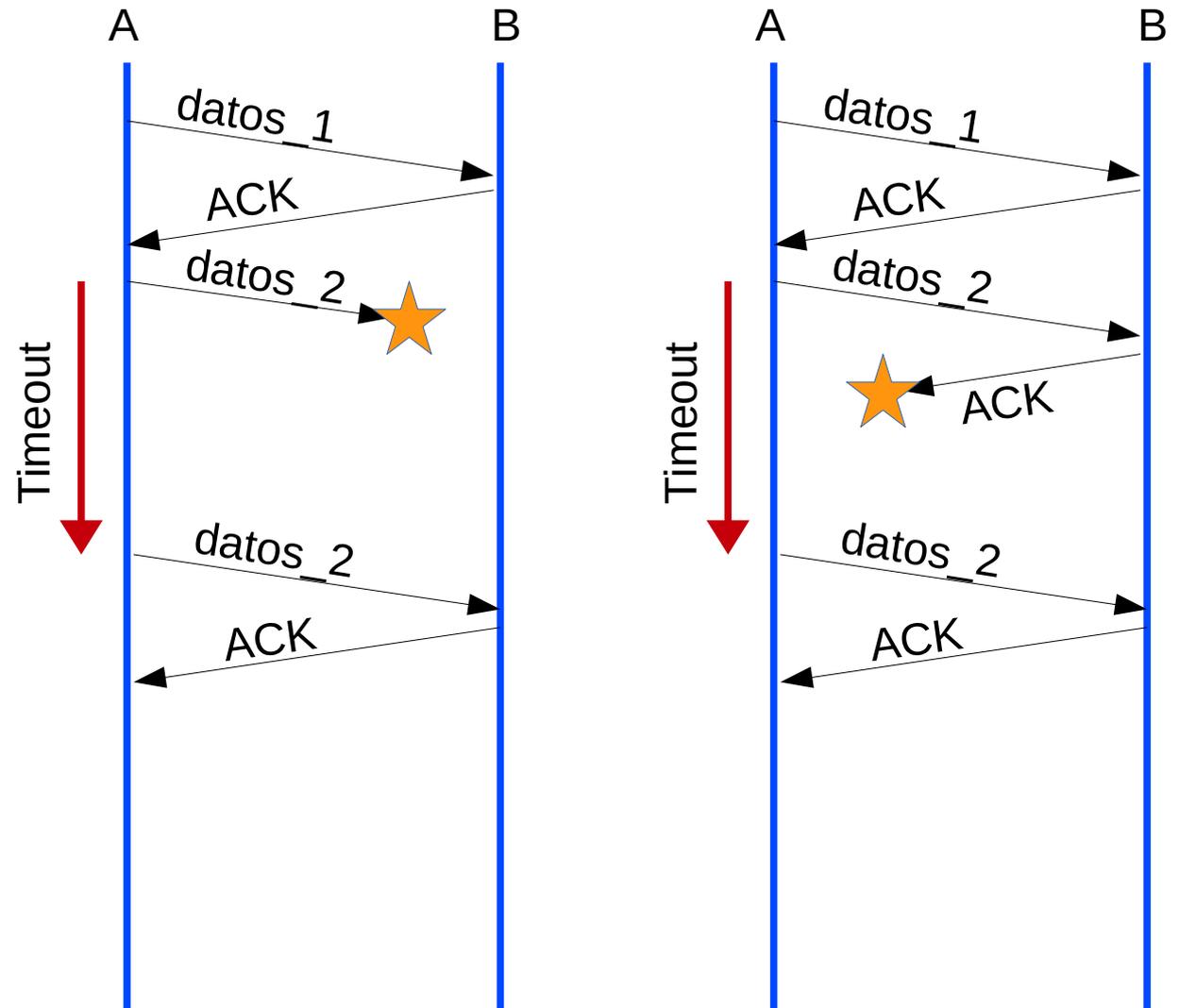


# Escenarios de duplicados por pérdidas

Transmisión sin errores

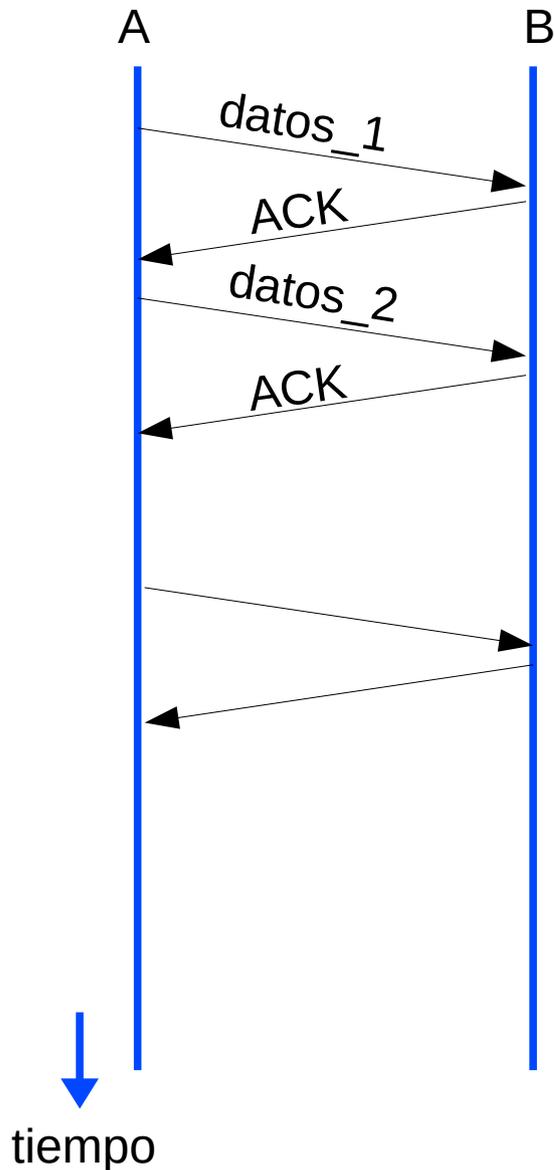


Errores y temporizadores

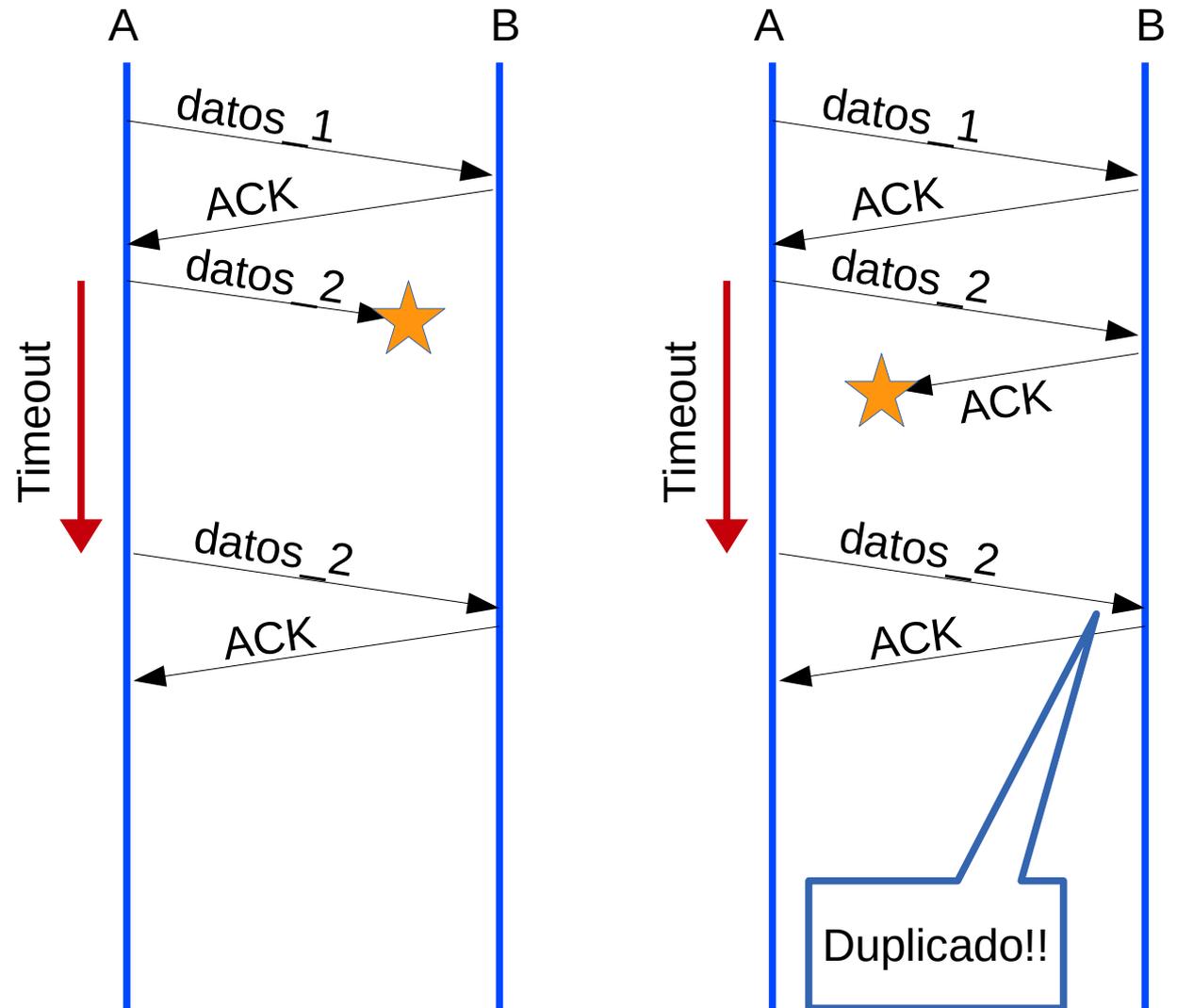


# Escenarios de duplicados por pérdidas

Transmisión sin errores

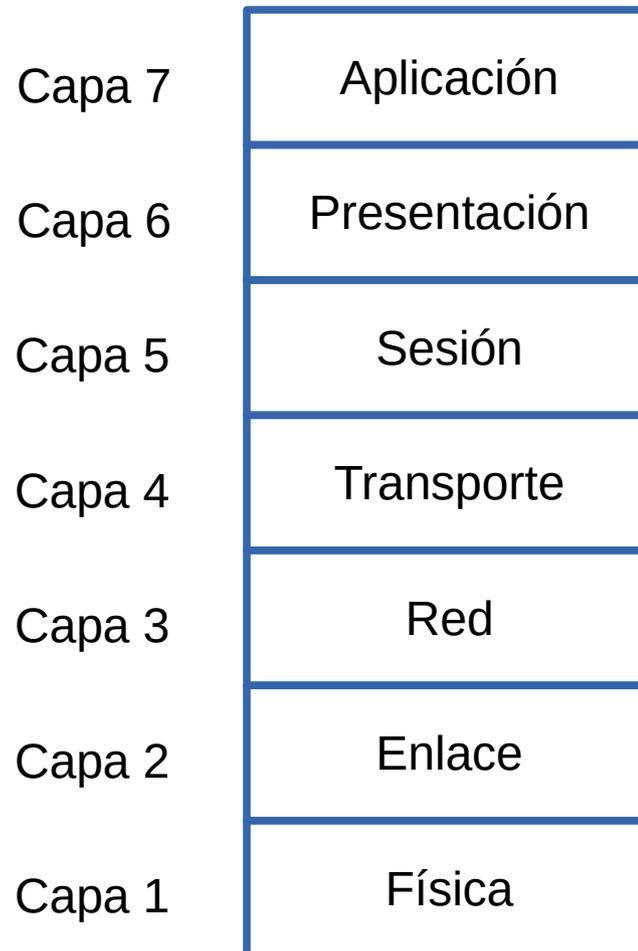


Errores y temporizadores



# Modelo OSI

- El modelo de 5 capas que usamos en el curso no es el único
- En los 70's se estandarizó por parte de la ISO (International Organization for Standardization) el modelo OSI (Open Systems Interconnection)
- Las funciones de capa 5 y 6 del modelo OSI las resuelve la capa de aplicación en el modelo de 5 capas



# Ejemplo de navegación en Internet

- Cuando abro el navegador y escribo en la barra de direcciones por ejemplo:
  - www.fing.edu.uy
  - En general un URL (Uniform Resource Locator) que puede ser más complicado. En este caso hay algunos campos no especificados que toman valores por defecto: http://www.fing.edu.uy:80/
- Se despliega en la pantalla un conjunto de texto con formato en conjunto con imágenes, animaciones, etc



# Ejemplo de navegación en Internet

- Cuando abro el navegador y escribo en la barra de direcciones por ejemplo:
  - www.fing.edu.uy
  - En general un URL (Uniform Resource Locator) que puede ser más complicado. En este caso hay algunos campos no especificados que toman valores por defecto: http://www.fing.edu.uy:80/
- Se despliega en la pantalla un conjunto de texto con formato en conjunto con imágenes, animaciones, etc
- ¿Cuáles son las funciones involucradas para que esto sea posible? ¿qué protocolos están involucrados?



# Capa de aplicación

- Hay una **aplicación en mi dispositivo**: Navegador
  - Ejemplos: Chrome, Firefox, Internet Explorer (Edge), Opera, Safari, etc
- Hay una **aplicación en el servidor** de FING: Servidor Web
  - Ejemplo: Apache http Server, Internet Information Services (IIS), etc
- Dialogan entre si usando un **protocolo** para entenderse
  - http (Hypertext Transfer Protocol) o https (Seguro)

# Capa de aplicación

- Hay una **aplicación en mi dispositivo**: Navegador
  - Ejemplos: Chrome, Firefox, Internet Explorer (Edge), Opera, Safari, etc
- Hay una **aplicación en el servidor** de FING: Servidor Web
  - Ejemplo: Apache http Server, Internet Information Services (IIS), etc
- Dialogan entre si usando un **protocolo** para entenderse
  - http (Hypertext Transfer Protocol) o https (Seguro)

Navegador



Mi dispositivo

# Capa de aplicación

- Hay una **aplicación en mi dispositivo**: Navegador
  - Ejemplos: Chrome, Firefox, Internet Explorer (Edge), Opera, Safari, etc
- Hay una **aplicación en el servidor** de FING: Servidor Web
  - Ejemplo: Apache http Server, Internet Information Services (IIS), etc
- Dialogan entre si usando un **protocolo** para entenderse
  - http (Hypertext Transfer Protocol) o https (Seguro)

Navegador

Servidor  
WEB



Mi dispositivo



[www.fing.edu.uy](http://www.fing.edu.uy)

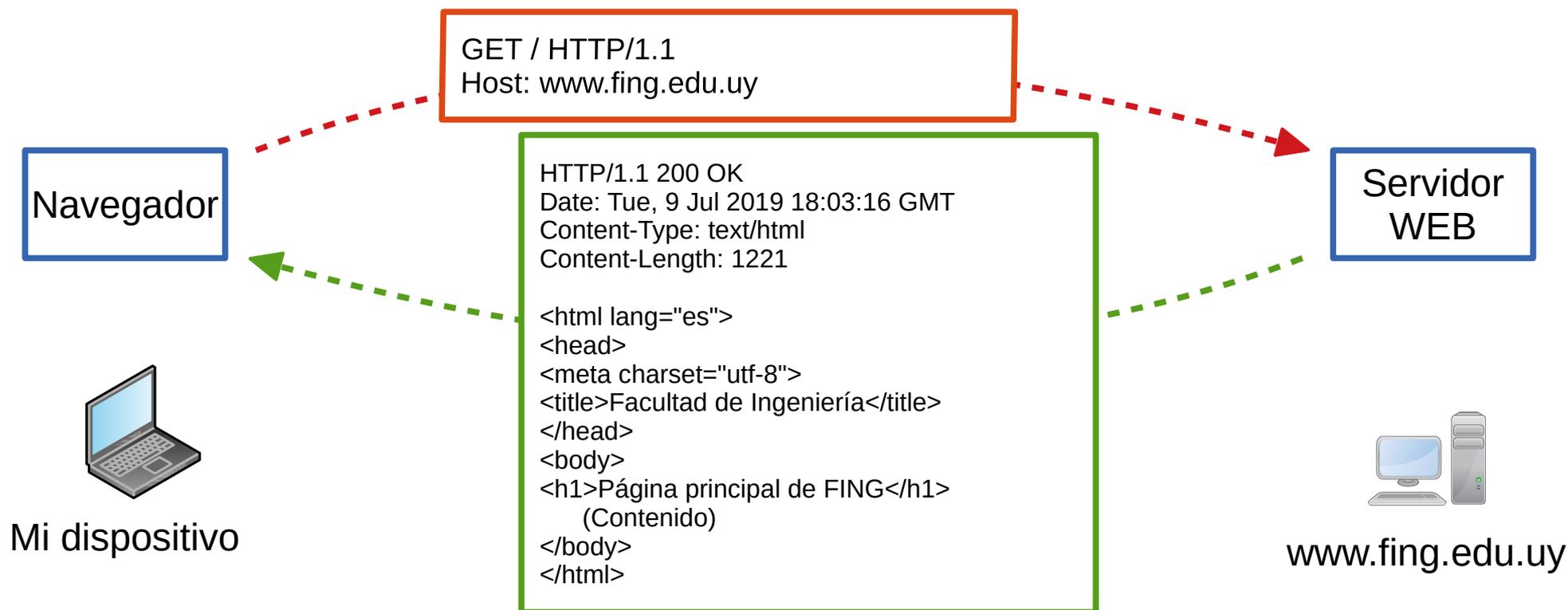
# Capa de aplicación

- Hay una **aplicación en mi dispositivo**: Navegador
  - Ejemplos: Chrome, Firefox, Internet Explorer (Edge), Opera, Safari, etc
- Hay una **aplicación en el servidor** de FING: Servidor Web
  - Ejemplo: Apache http Server, Internet Information Services (IIS), etc
- Dialogan entre si usando un **protocolo** para entenderse
  - http (Hypertext Transfer Protocol) o https (Seguro)



# Capa de aplicación

- Hay una **aplicación en mi dispositivo**: Navegador
  - Ejemplos: Chrome, Firefox, Internet Explorer (Edge), Opera, Safari, etc
- Hay una **aplicación en el servidor** de FING: Servidor Web
  - Ejemplo: Apache http Server, Internet Information Services (IIS), etc
- Dialogan entre si usando un **protocolo** para entenderse
  - http (Hypertext Transfer Protocol) o https (Seguro)



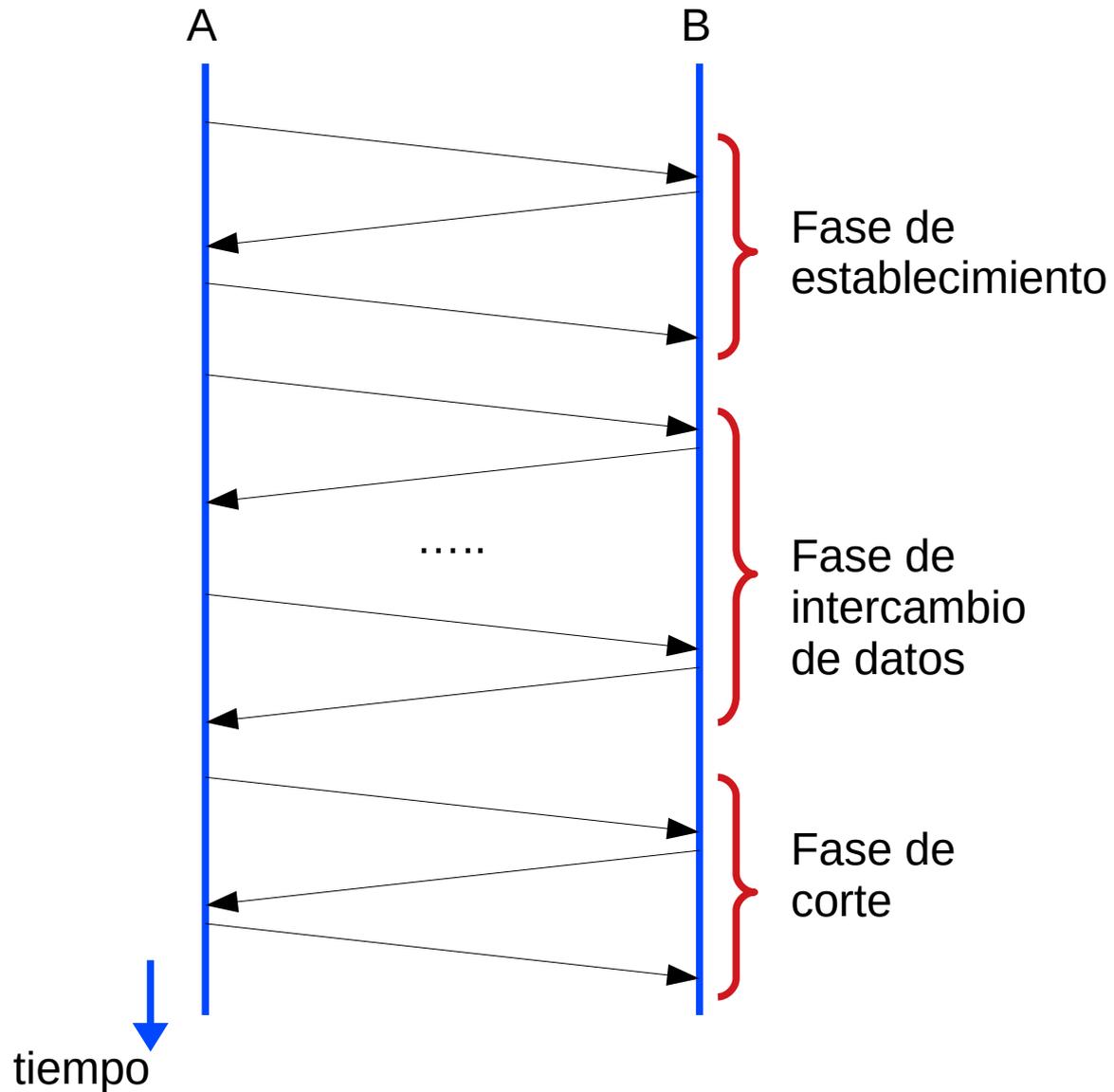
# Capa de aplicación

- ¿Dónde está el servidor `www.fing.edu.uy`?
- ¿Cómo llego hasta él para pedirle el contenido de su página principal?
- Las personas se manejan mejor con nombres para identificar los recursos que desean acceder (cadenas de caracteres como `www.fing.edu.uy`), pero las computadoras se manejan mejor con números
- Se necesita poder convertir un nombre como `www.fing.edu.uy` en un identificador numérico o dirección que permita identificar y localizar el servidor en la red
  - **Dirección IP** (Internet Protocol)
- Para resolver esto existe el Servicio de Nombres de Dominio (**DNS**, Domain Name System)
- Es una aplicación que brinda soporte a las demás aplicaciones en Internet y es esencial para que ésta funcione
- Se requiere como paso previo consultar al DNS para que resuelva cuál es la dirección IP asociada a `www.fing.edu.uy`
  - en este ejemplo es `164.73.32.5`

# Capa de transporte

- Si bien usando el modelo de capas, las aplicaciones creen que dialogan entre si directamente, para que la capa de aplicación pueda enviar el mensaje, necesitará usar los servicios de capa de transporte
- En la capa de transporte de Internet están disponibles dos servicios:
  - TCP (Transmission Control Protocol)
    - Protocolo confiable y orientado a conexión
  - UDP (User Datagram Protocol)
    - Protocolo no confiable y no orientado a conexión
- En el caso de la navegación se utiliza tradicionalmente TCP
- ¿Cómo invocamos desde la aplicación los servicios de TCP?
  - Biblioteca “sockets” con métodos para
    - establecer la conexión
    - enviar y recibir datos
    - cortar o liberar la conexión
- Una vez establecida la conexión, TCP garantiza un canal confiable y bidireccional para intercambiar información entre A y B

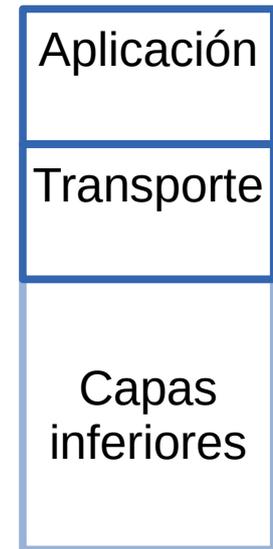
# Fases de una conexión TCP



# Interacción de capa de aplicación con TCP



A



B

# Interacción de capa de aplicación con TCP

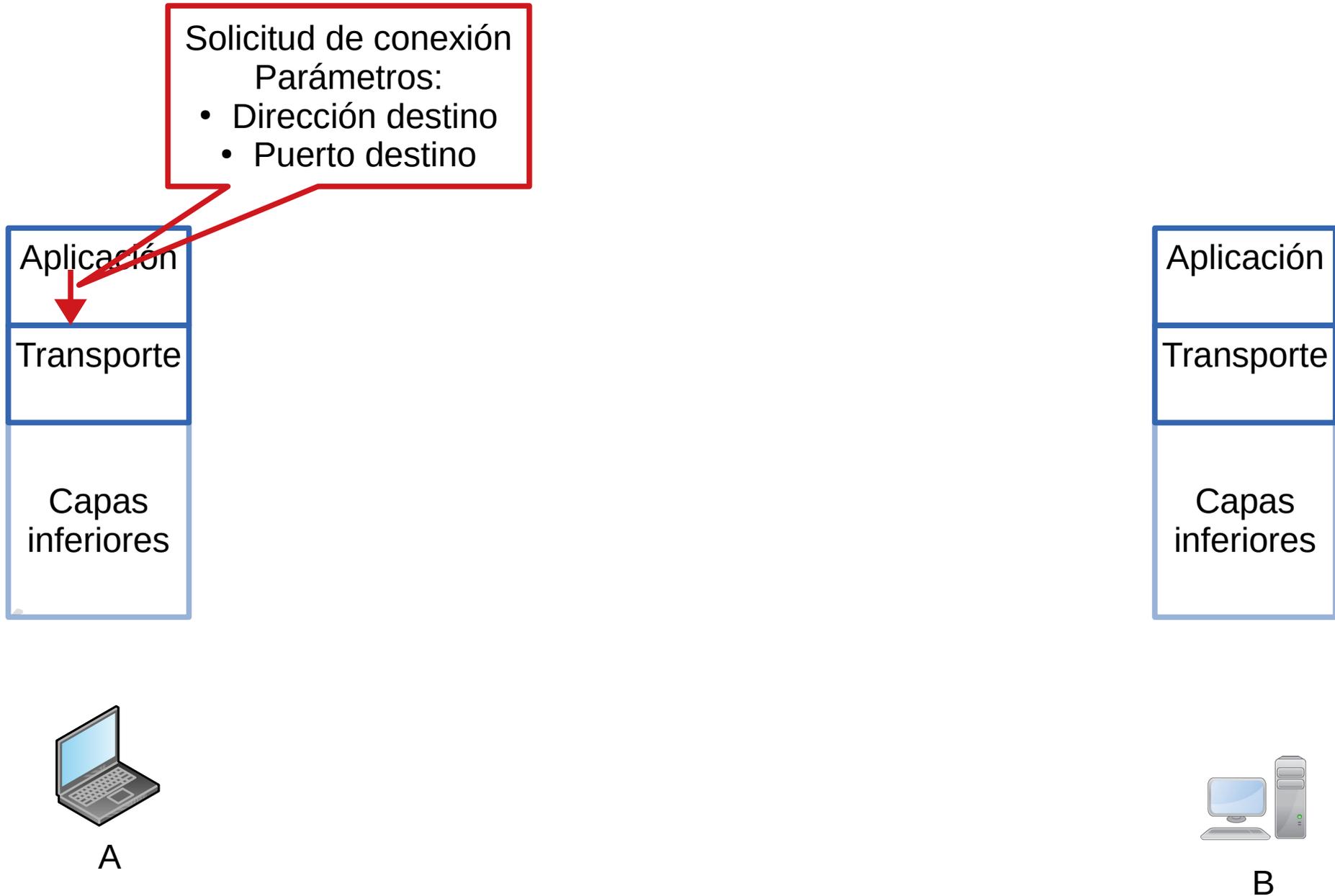


A

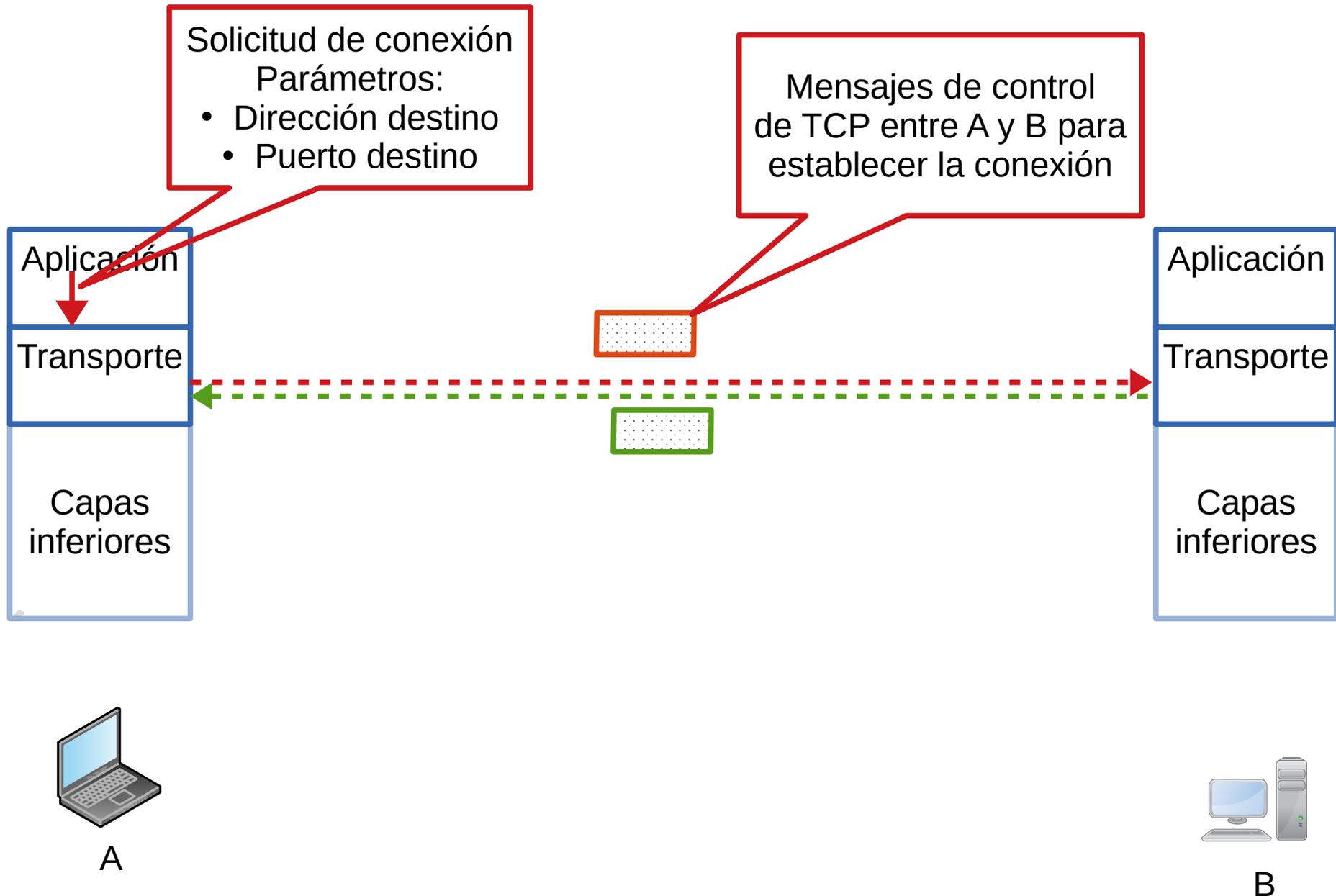


B

# Interacción de capa de aplicación con TCP



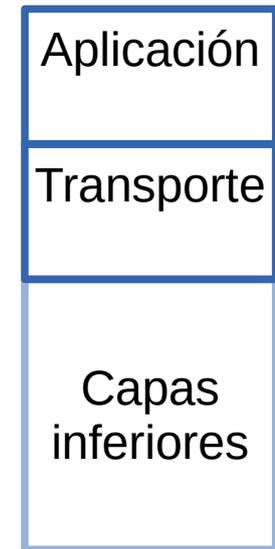
# Interacción de capa de aplicación con TCP



# Interacción de capa de aplicación con TCP



A

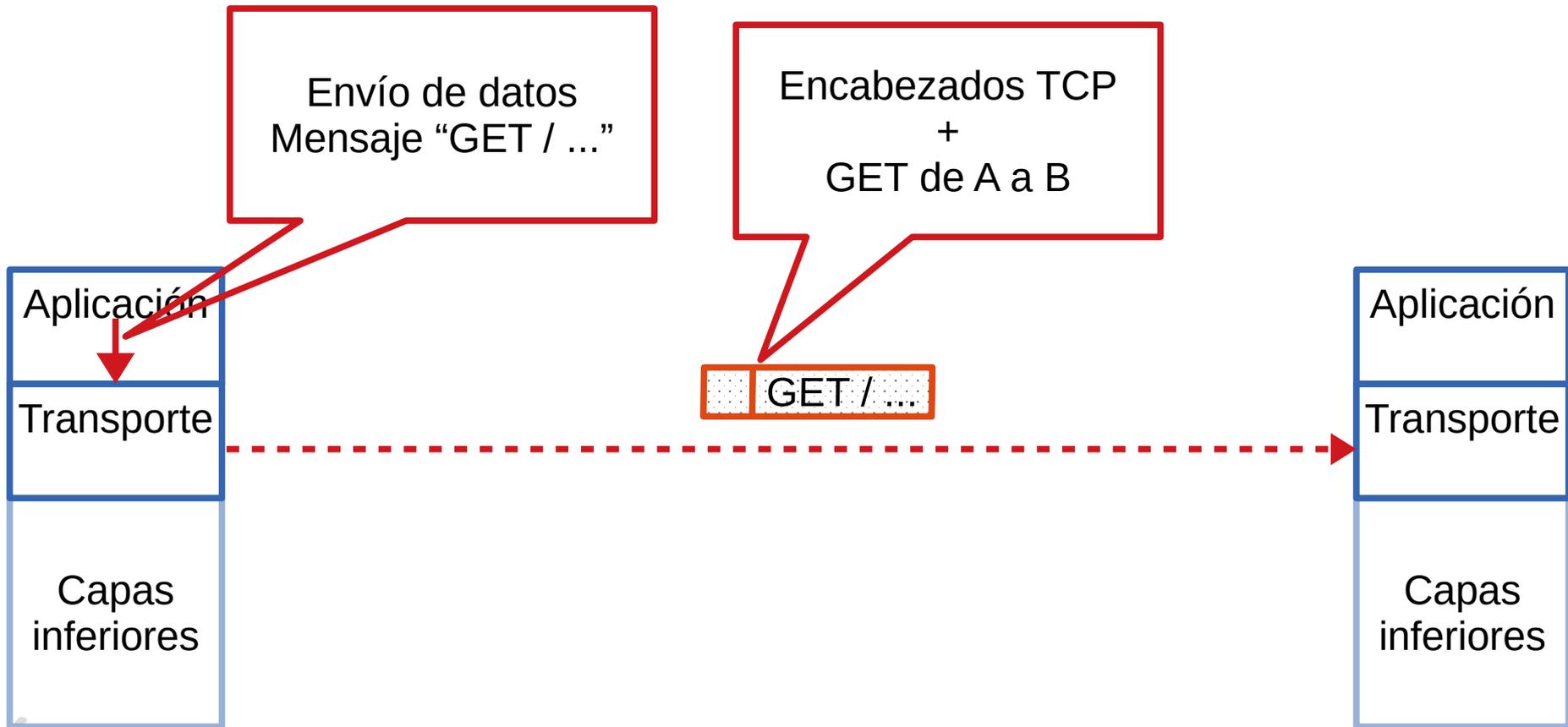


B

# Interacción de capa de aplicación con TCP



# Interacción de capa de aplicación con TCP

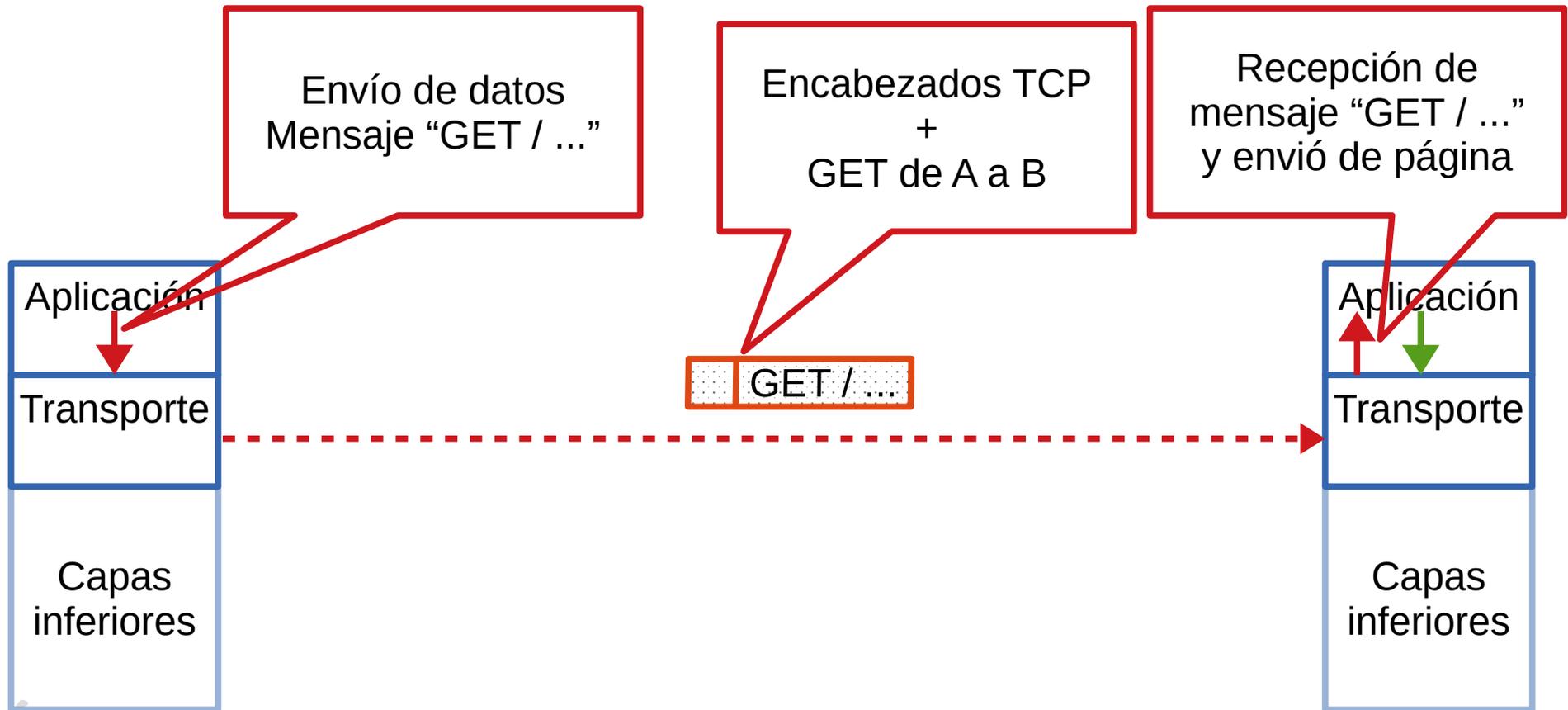


A



B

# Interacción de capa de aplicación con TCP

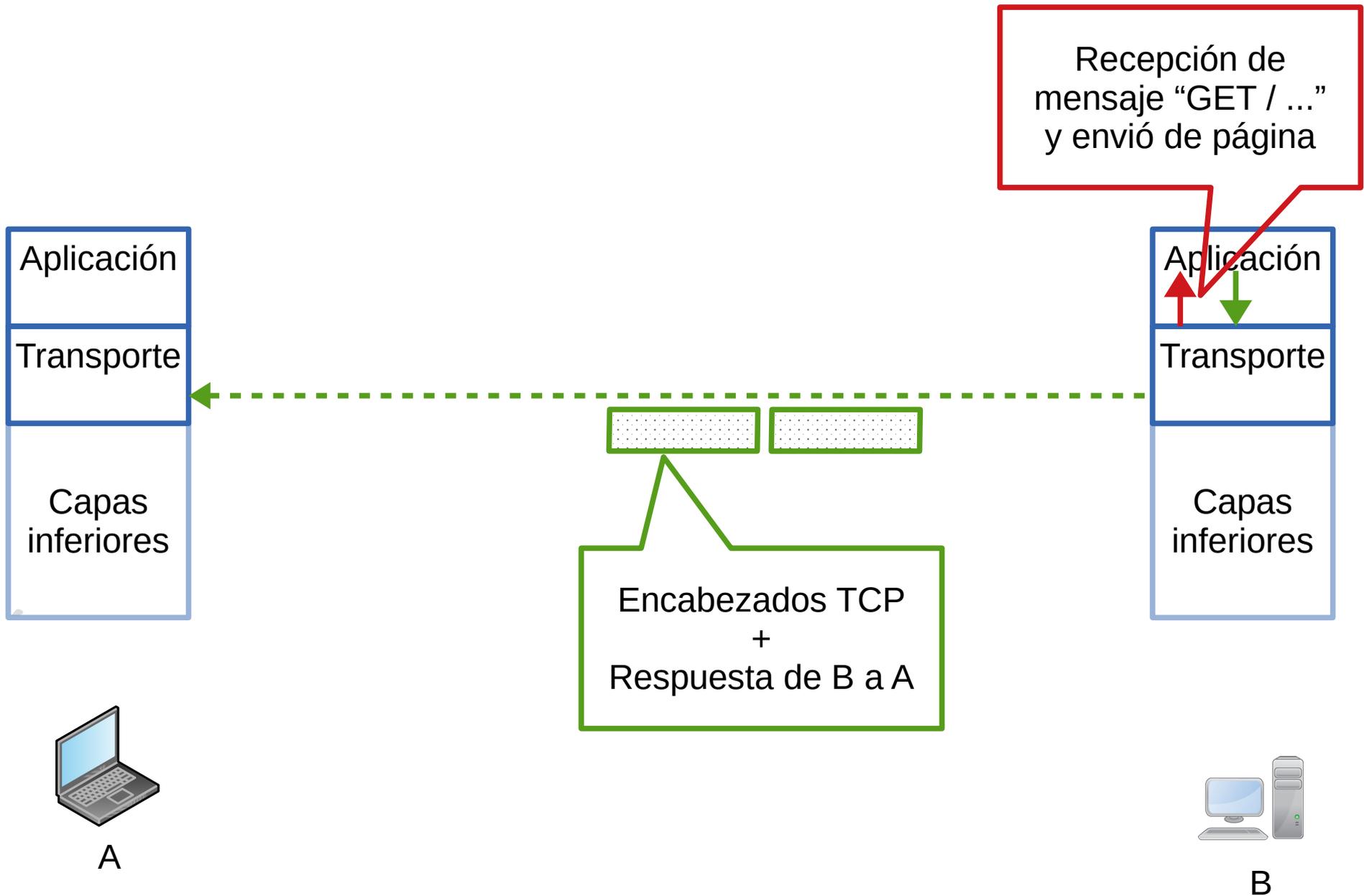


A

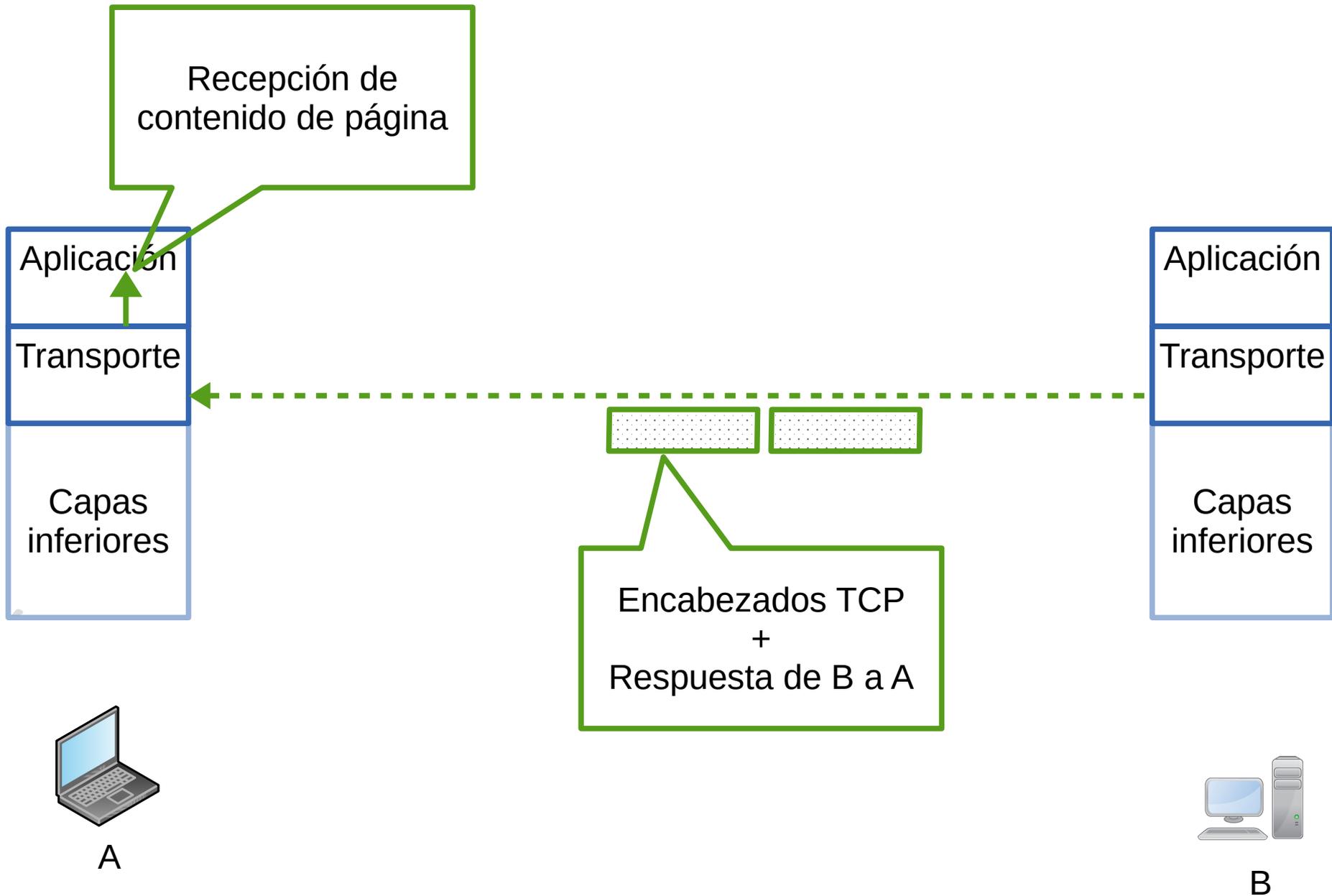


B

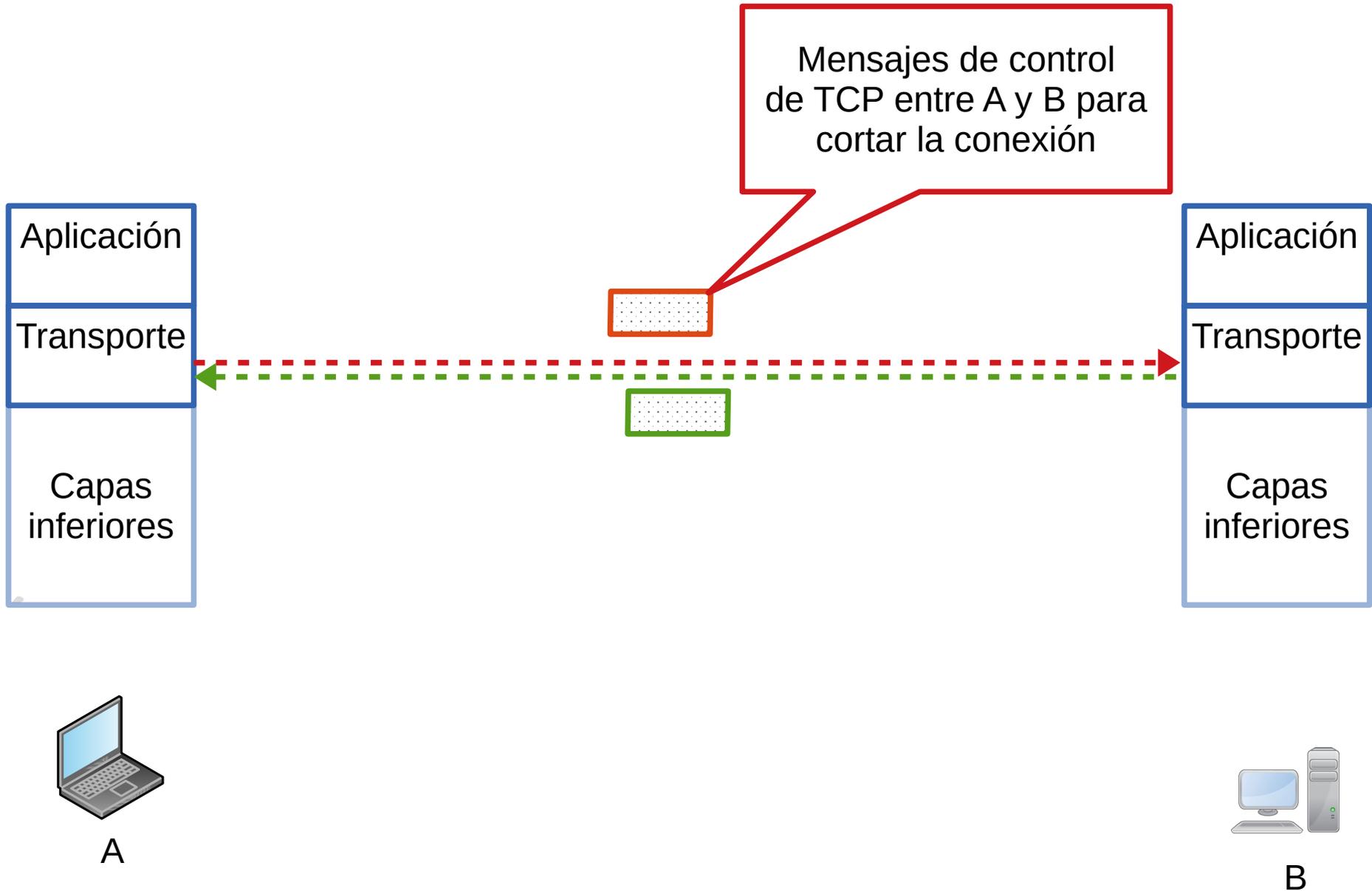
# Interacción de capa de aplicación con TCP



# Interacción de capa de aplicación con TCP



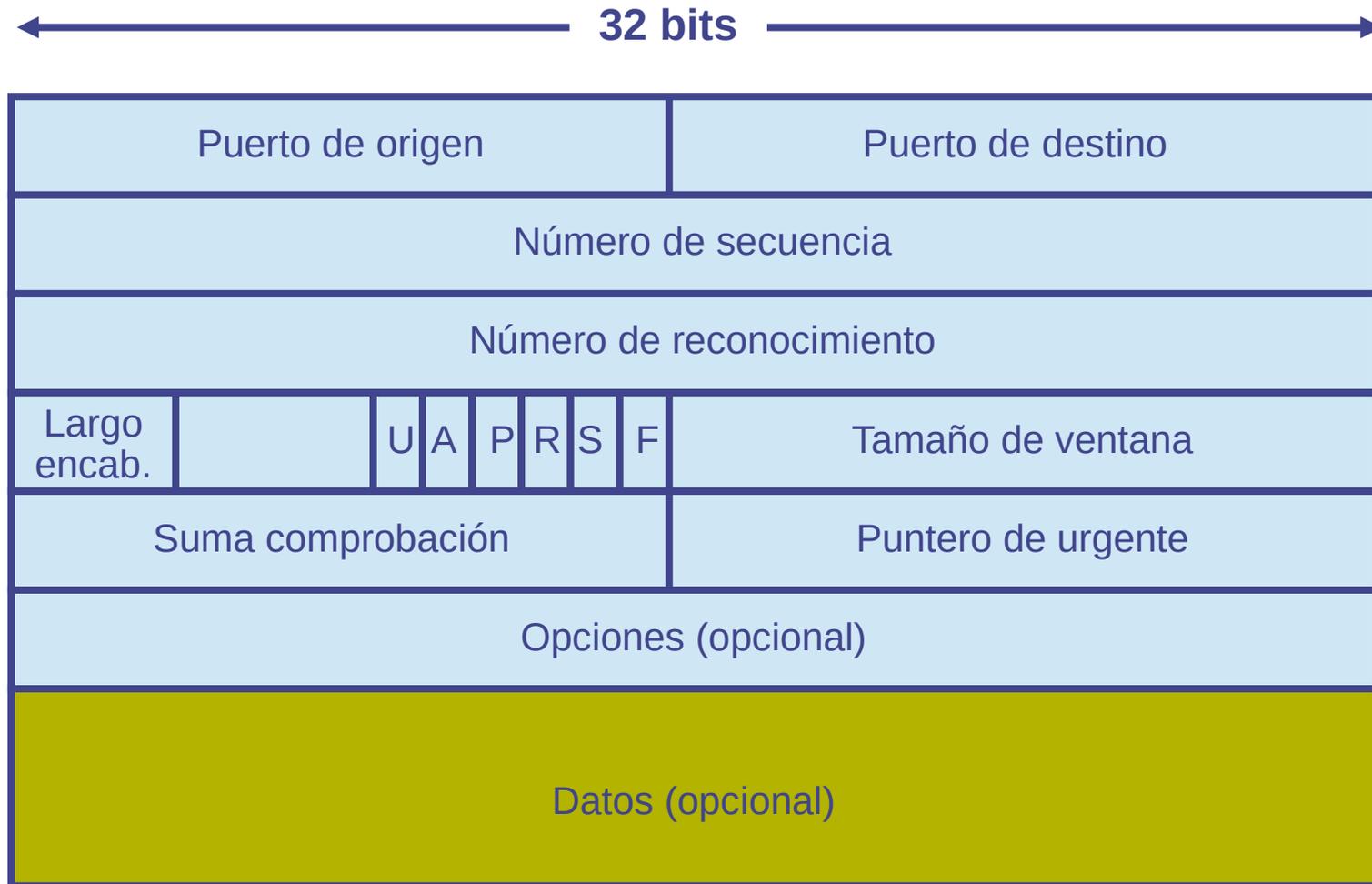
# Interacción de capa de aplicación con TCP



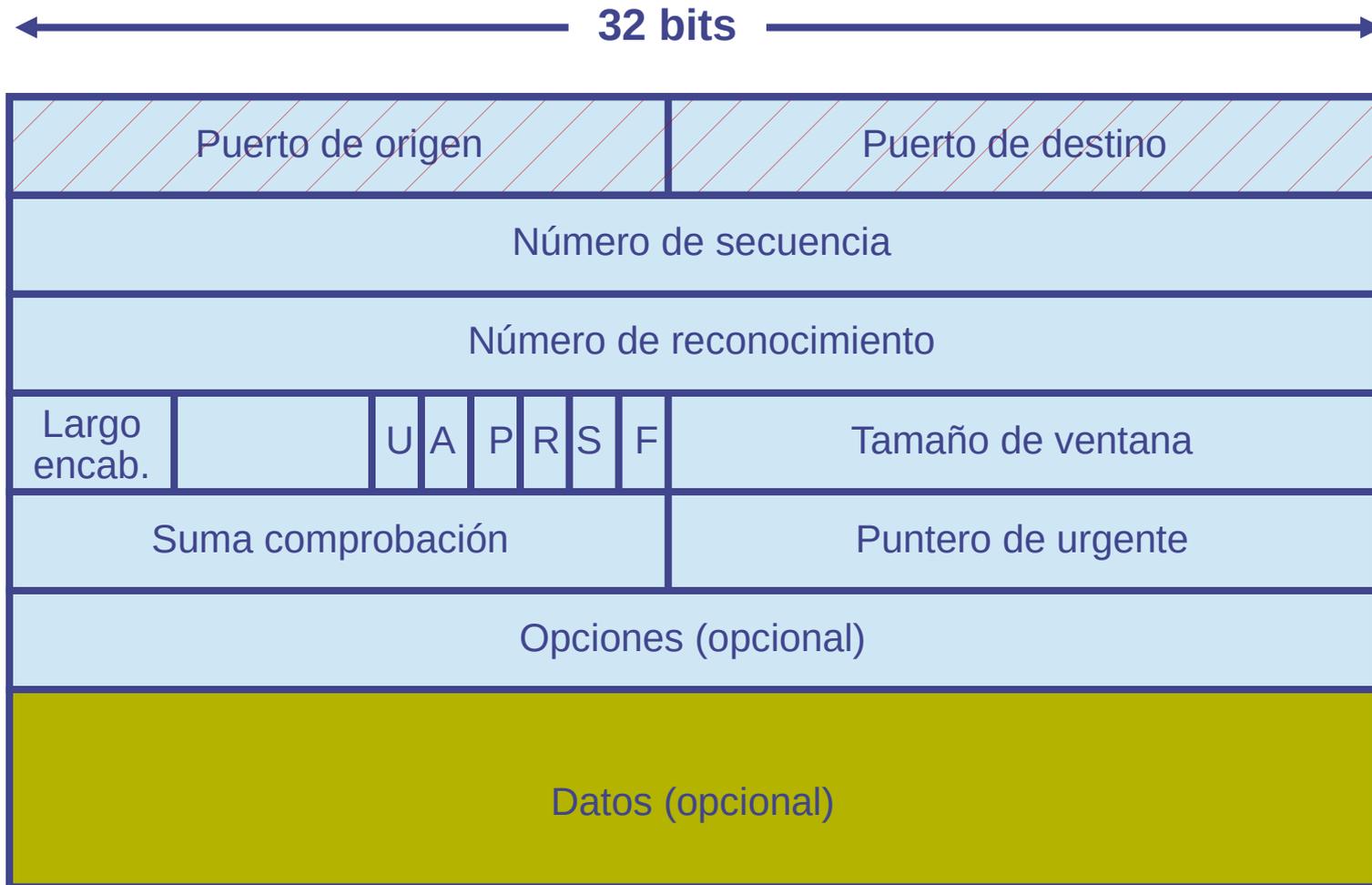
# Capa de transporte: TCP

- ¿Qué parámetros se requieren para establecer la conexión TCP?
- Las conexiones se identifican con:
  - La dirección IP del dispositivo del cliente
    - dato conocido por la aplicación del cliente
  - La dirección IP del servidor o destinatario
    - en el ejemplo 164.73.32.5, la asociada con [www.fing.edu.uy](http://www.fing.edu.uy)
  - El identificador (puerto) asociado a la aplicación que está iniciando la conexión
    - dato conocido por la aplicación del cliente
  - El identificador (puerto) de la aplicación con la que se quiere conectar
    - surge del URL (en el ejemplo el valor por defecto 80)
- Entonces una conexión queda identificada por:
  - **Protocolo:** TCP
  - **Dirección IP de origen**
  - **Puerto de origen**
  - **Dirección IP de destino**
  - **Puerto destino**

# Formato del segmento TCP

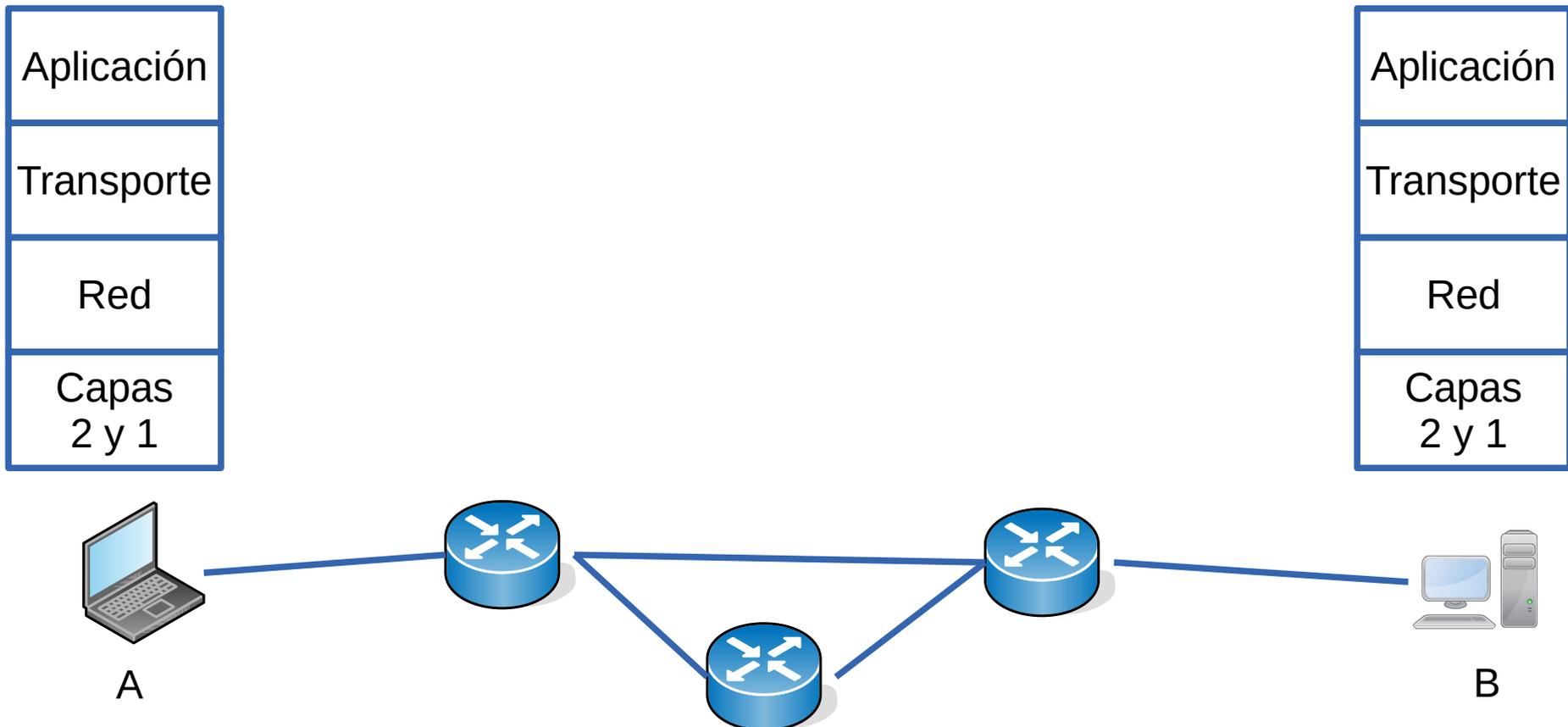


# Formato del segmento TCP



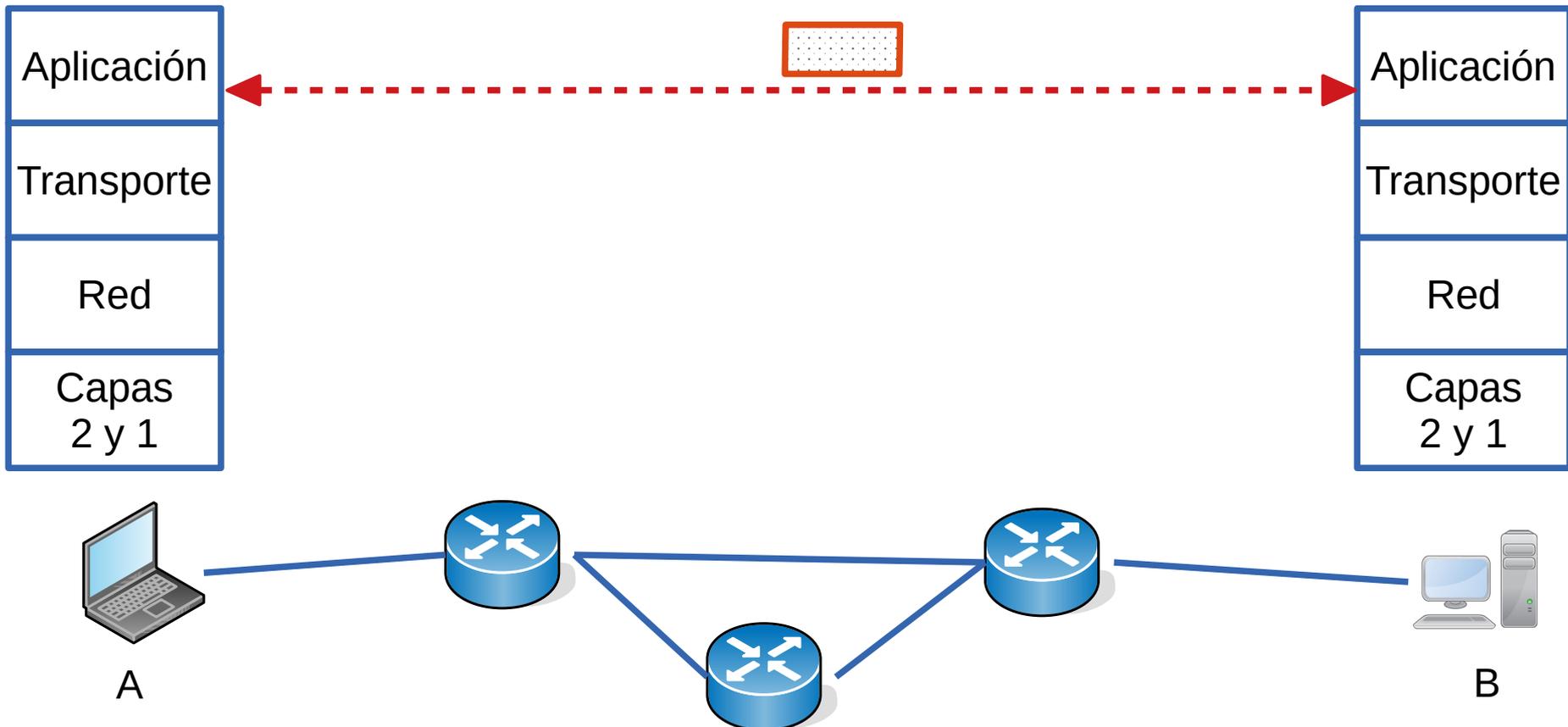
# Capa de red: IP

- Los segmentos TCP de control (establecimiento y corte de conexión) y de datos se enviarán al destino usando los servicios de la capa de red
- La capa de red de Internet ofrece un servicio: **IP (Internet Protocol)**
  - Protocolo no confiable y no orientado a conexión
- La capa de red tiene que encontrar un camino en base a los recursos disponibles en la red (enlaces y nodos) y encaminar los paquetes al destino



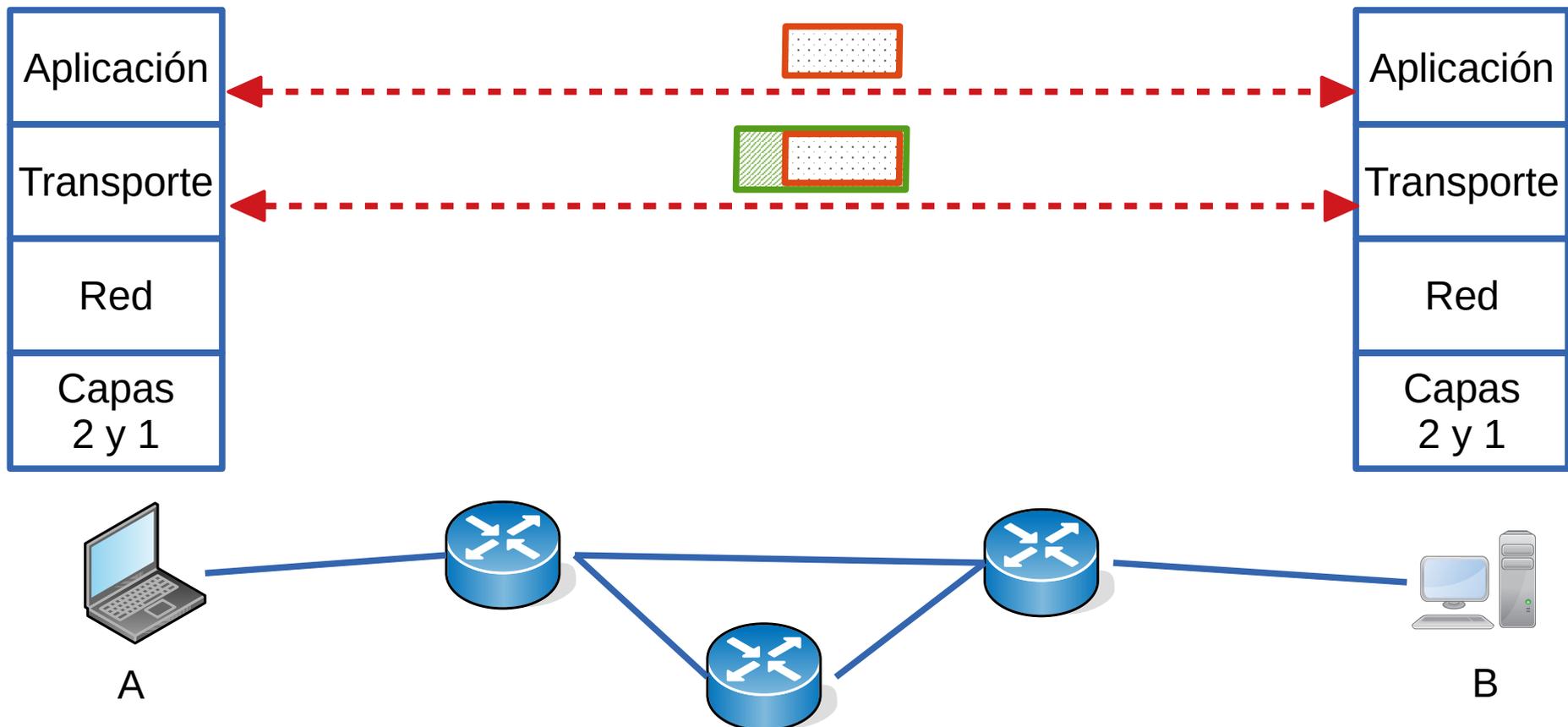
# Capa de red: IP

- Los segmentos TCP de control (establecimiento y corte de conexión) y de datos se enviarán al destino usando los servicios de la capa de red
- La capa de red de Internet ofrece un servicio: **IP (Internet Protocol)**
  - Protocolo no confiable y no orientado a conexión
- La capa de red tiene que encontrar un camino en base a los recursos disponibles en la red (enlaces y nodos) y encaminar los paquetes al destino



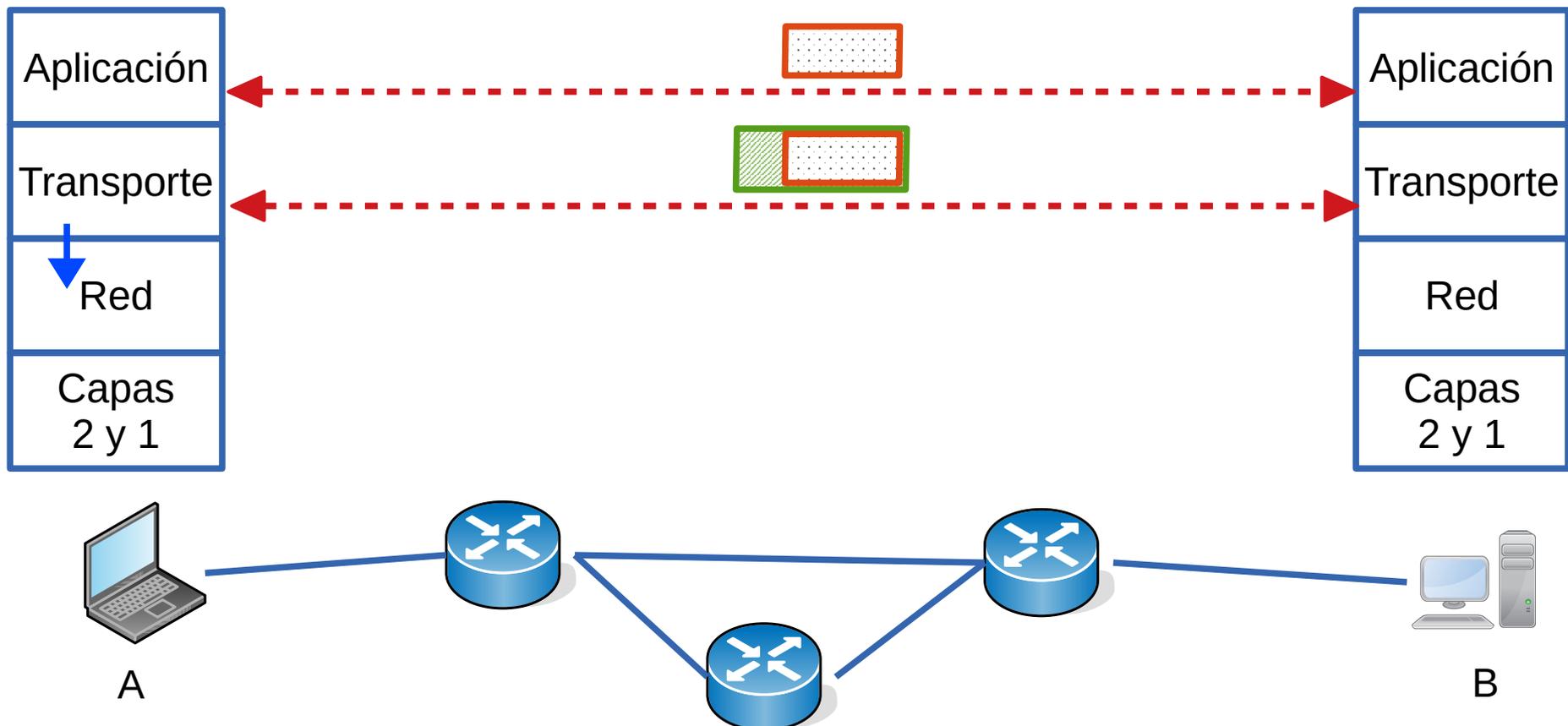
# Capa de red: IP

- Los segmentos TCP de control (establecimiento y corte de conexión) y de datos se enviarán al destino usando los servicios de la capa de red
- La capa de red de Internet ofrece un servicio: **IP (Internet Protocol)**
  - Protocolo no confiable y no orientado a conexión
- La capa de red tiene que encontrar un camino en base a los recursos disponibles en la red (enlaces y nodos) y encaminar los paquetes al destino



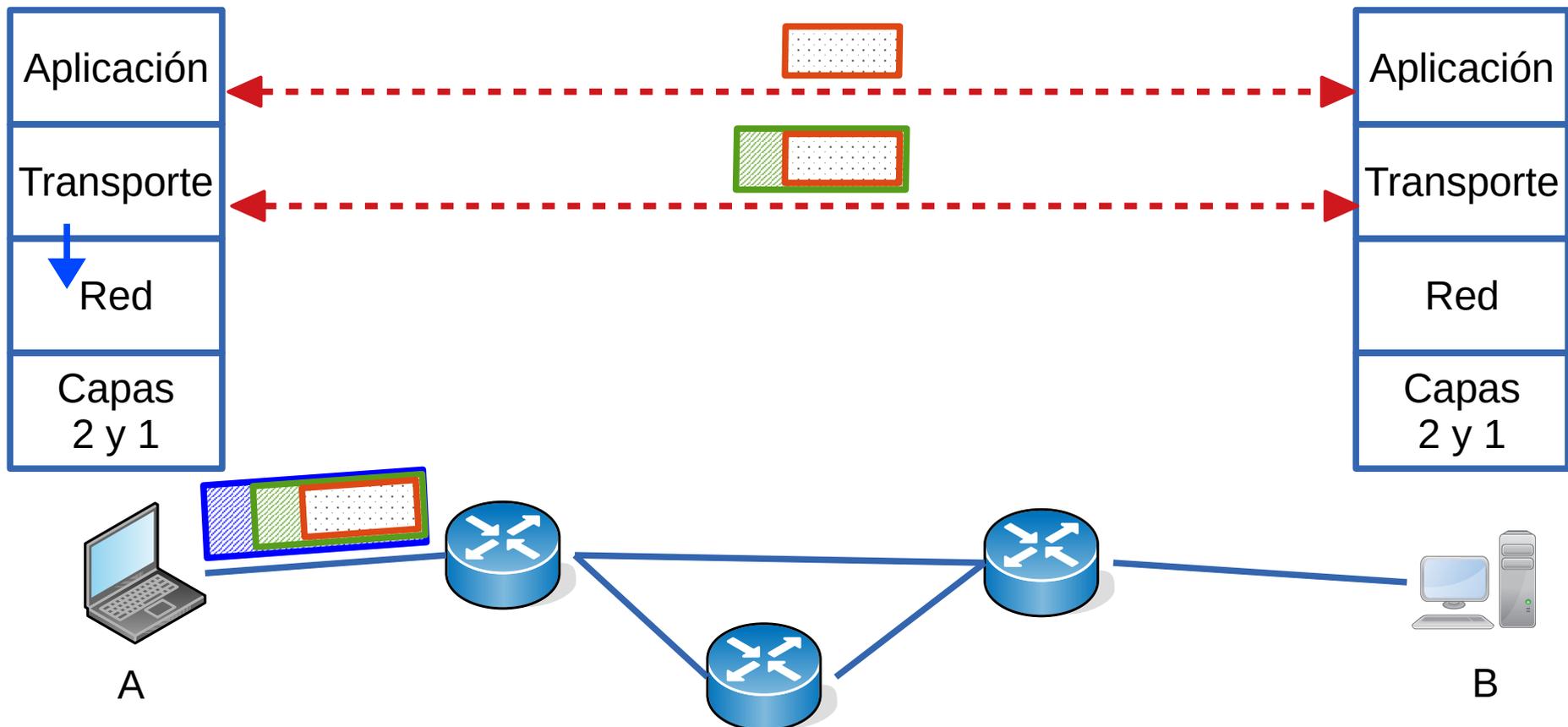
# Capa de red: IP

- Los segmentos TCP de control (establecimiento y corte de conexión) y de datos se enviarán al destino usando los servicios de la capa de red
- La capa de red de Internet ofrece un servicio: **IP (Internet Protocol)**
  - Protocolo no confiable y no orientado a conexión
- La capa de red tiene que encontrar un camino en base a los recursos disponibles en la red (enlaces y nodos) y encaminar los paquetes al destino



# Capa de red: IP

- Los segmentos TCP de control (establecimiento y corte de conexión) y de datos se enviarán al destino usando los servicios de la capa de red
- La capa de red de Internet ofrece un servicio: **IP (Internet Protocol)**
  - Protocolo no confiable y no orientado a conexión
- La capa de red tiene que encontrar un camino en base a los recursos disponibles en la red (enlaces y nodos) y encaminar los paquetes al destino

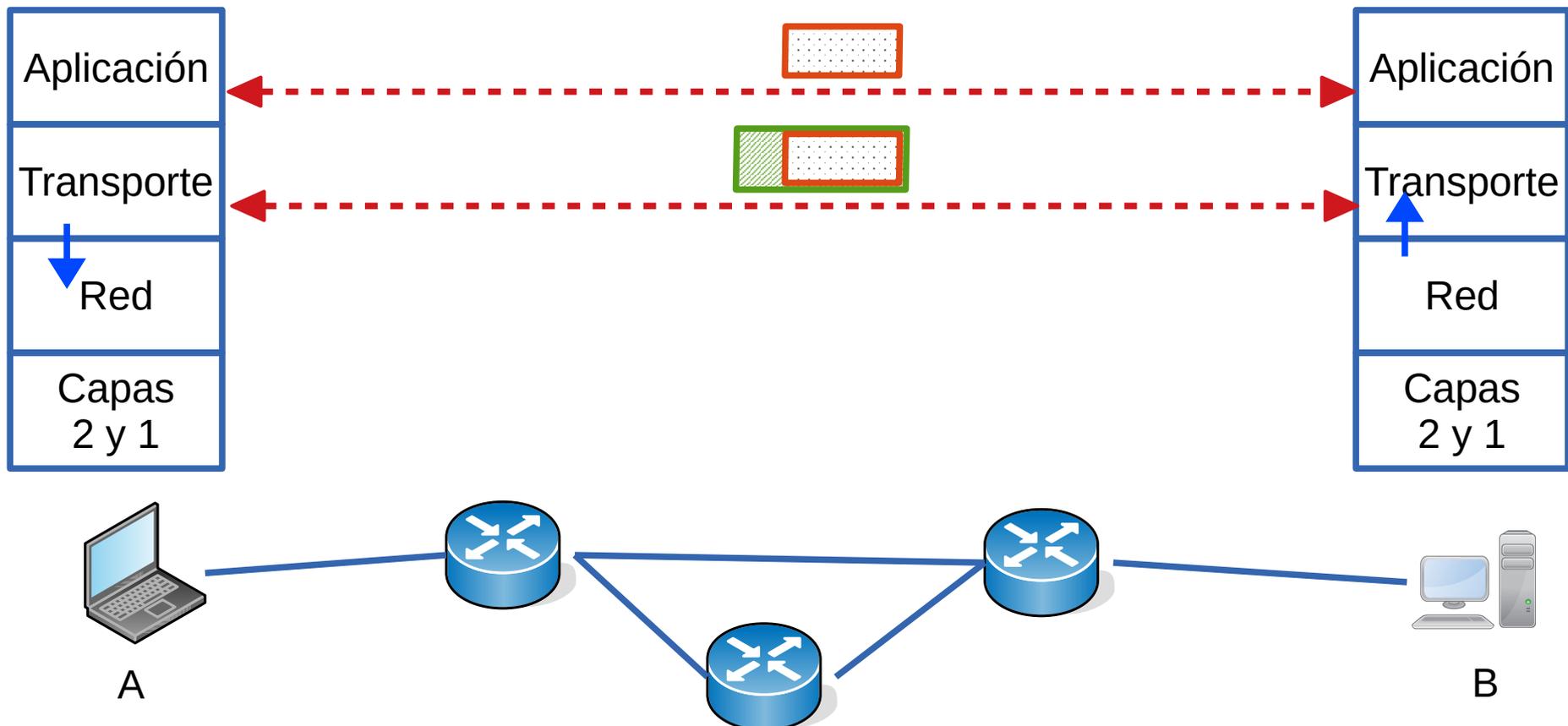






# Capa de red: IP

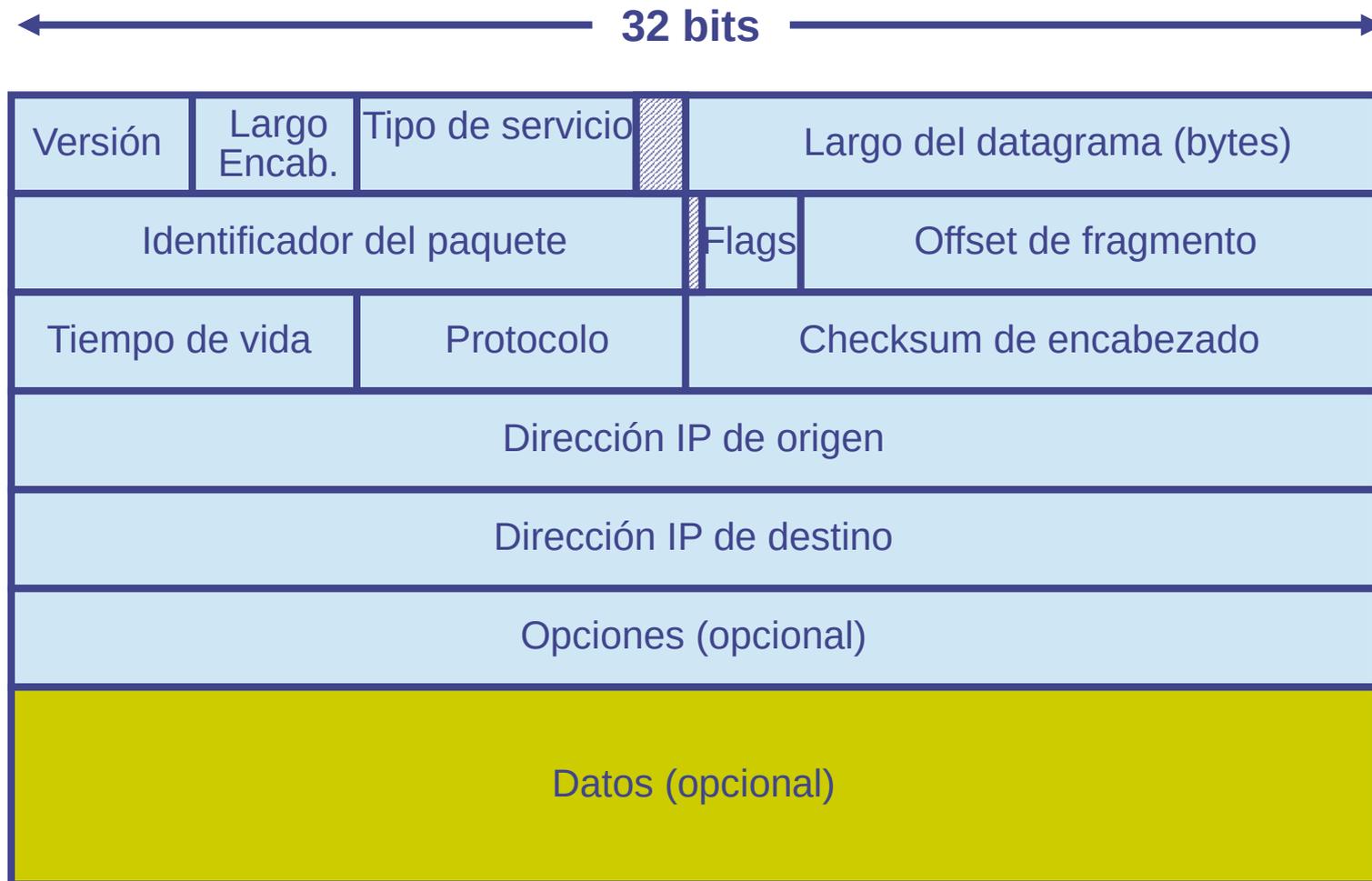
- Los segmentos TCP de control (establecimiento y corte de conexión) y de datos se enviarán al destino usando los servicios de la capa de red
- La capa de red de Internet ofrece un servicio: **IP (Internet Protocol)**
  - Protocolo no confiable y no orientado a conexión
- La capa de red tiene que encontrar un camino en base a los recursos disponibles en la red (enlaces y nodos) y encaminar los paquetes al destino



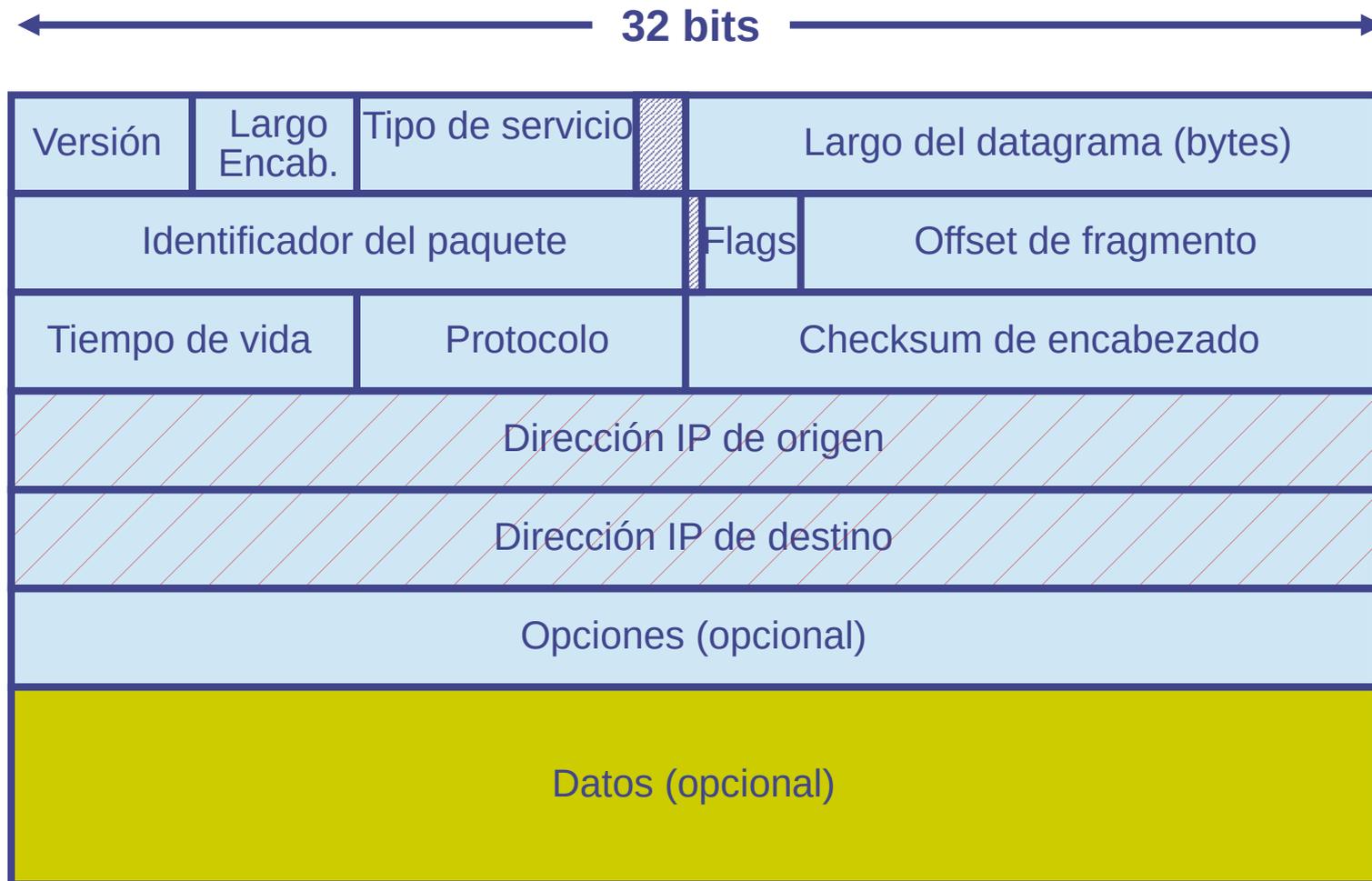
# Capa de red: IP

- Cada paquete se encamina por separado en base a la dirección completa del destino: **dirección IP**
- Hay dos versiones en uso:
  - **IPv4** (direcciones de 32 bits)
  - **IPv6** (direcciones de 128 bits)
- La versión 6 va a reemplazar a la versión 4 pero van a convivir por años
- Los paquetes se encaminan de a saltos en base a las tablas de rutas
- Brinda un servicio no orientado a conexión y no confiable
  - Servicio “best effort”
- La capa de red en Internet (IP) es muy simple y la complejidad está en las capas superiores (TCP por ejemplo)
  - Esto es importante porque la capa de red está implementada en todos los nodos (los extremos y los equipos intermedios), mientras que TCP está solamente en los equipos de los extremos
  - La complejidad de la red está en los nodos de los extremos (en el software o sistema operativo) y los enrutadores solo encaminan paquetes

# Formato del paquete IPv4

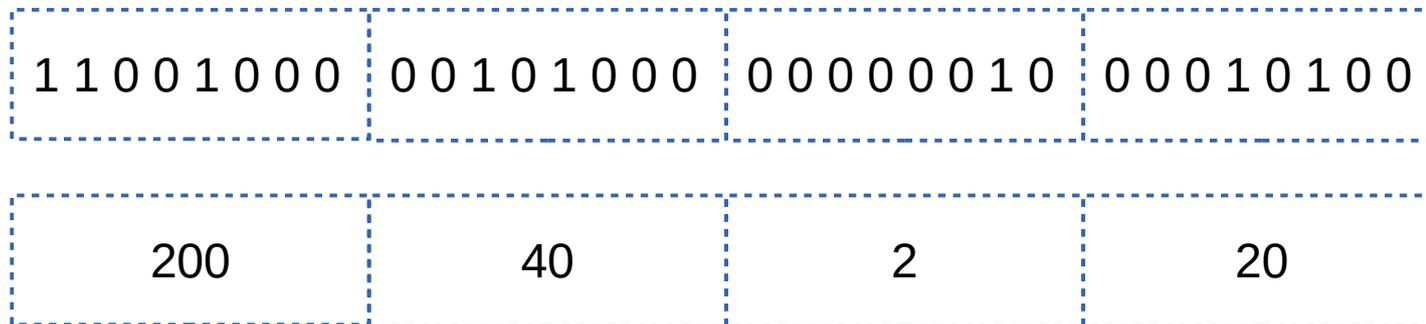


# Formato del paquete IPv4



# Direcciones IPv4

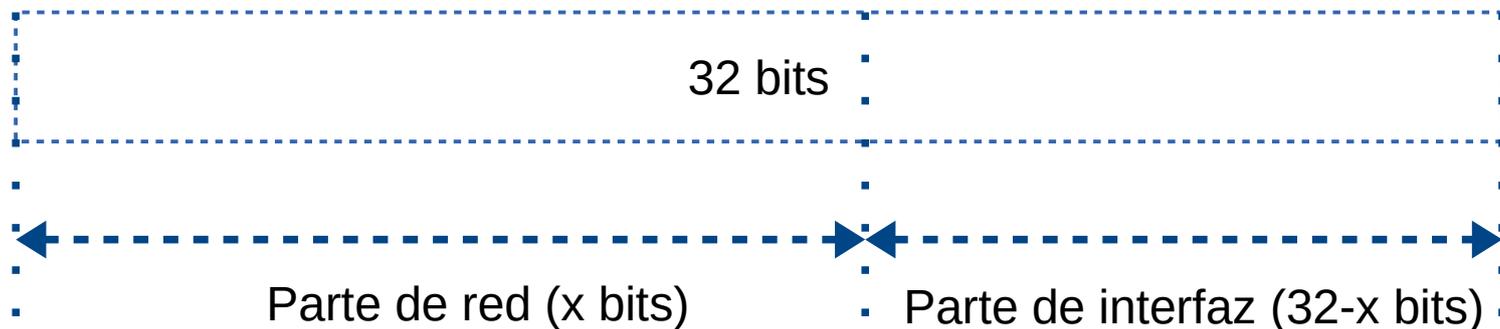
- Números de **32 bits** (4 bytes) que permiten identificar y localizar un dispositivo en la red
- Estrictamente las direcciones están asociadas a una interfaz de red
- Un equipo puede tener más de una interfaz si está conectado a varias redes
  - En particular los enrutadores normalmente tienen más de una interfaz porque interconectan al menos dos segmentos de red
- En general las direcciones IP deben ser únicas en la red (salvo en presencia del mecanismo de NAT que veremos luego)
- Se representan en el formato llamado: **dotted-decimal notation**



- Se escribe como: 200.40.2.20

# Rangos de direcciones

- Las direcciones se asignan por **rangos**
  - Si se asignaran direcciones individualmente, direcciones contiguas podrían ser asignadas a equipos de distintas partes del mundo y por tanto los enrutadores necesitarían tablas de forwarding más grandes
  - Tablas más grandes, implican más tiempo para buscar en ellas, más memoria
  - En las tablas de forwarding se utilizan esos rangos como destinos
- Un rango de direcciones está compuesto por el conjunto de direcciones que tienen un **prefijo** (los  $x$  primeros bits) en común
  - $x$  es el largo del prefijo
- Determina la **parte de red** y la **parte de interfaz (o de host)**





# Ejemplo de rangos de direcciones

- Las direcciones IP:
  - 223.1.1.0
  - 223.1.1.1
  - 223.1.1.2
  - ...
  - 223.1.1.255
- Tienen los primeros 3 bytes (24 bits) en común (223.1.1.x)
- Se puede referir este rango de direcciones como 223.1.1.0/24
  - Conjunto de direcciones consecutivas que empiezan en la 223.1.1.0 y mantienen los primeros 24 bits iguales
- Las **direcciones con todos los bits de la parte de interfaz en 0** no se pueden asignar a interfaces ya que se reservan para identificar la red
  - En el ejemplo 223.1.1.0
- Las **direcciones con todos los bits de la parte de interfaz en 1** no se pueden asignar a interfaces ya que se usan como dirección de difusión de la red
  - En el ejemplo 223.1.1.255

# Capa de enlace y Capa física

- Los protocolos de capa de enlace están generalmente vinculados o adaptados a los medios físicos que se estén usando
- Dependiendo de los retardos, las tasas de errores de los medios físicos, será necesario utilizar protocolos en capa de enlace más o menos complicados

# ¿Cómo sigue el curso?

- Una vez comprendido el modelo de capas, podemos empezar a estudiar los detalles de cada capa en cualquier orden
- En particular vamos a ir desde las aplicaciones hacia la capa física
- La capa física no la vamos a abordar con más detalle de lo que se vio antes
- Los temas que siguen para el teórico son:
  - Capa de aplicación (2 clases)
  - Capa de transporte (4 clases)
  - Capa de red (6 clases)
  - Capa de enlace (6 clases)
  - Conceptos de seguridad en redes (2 clases)
  - Conceptos de gestión de redes (2 clases)
- En los laboratorios se verán los detalles de la implementación de los protocolos en Internet