TEMA 2

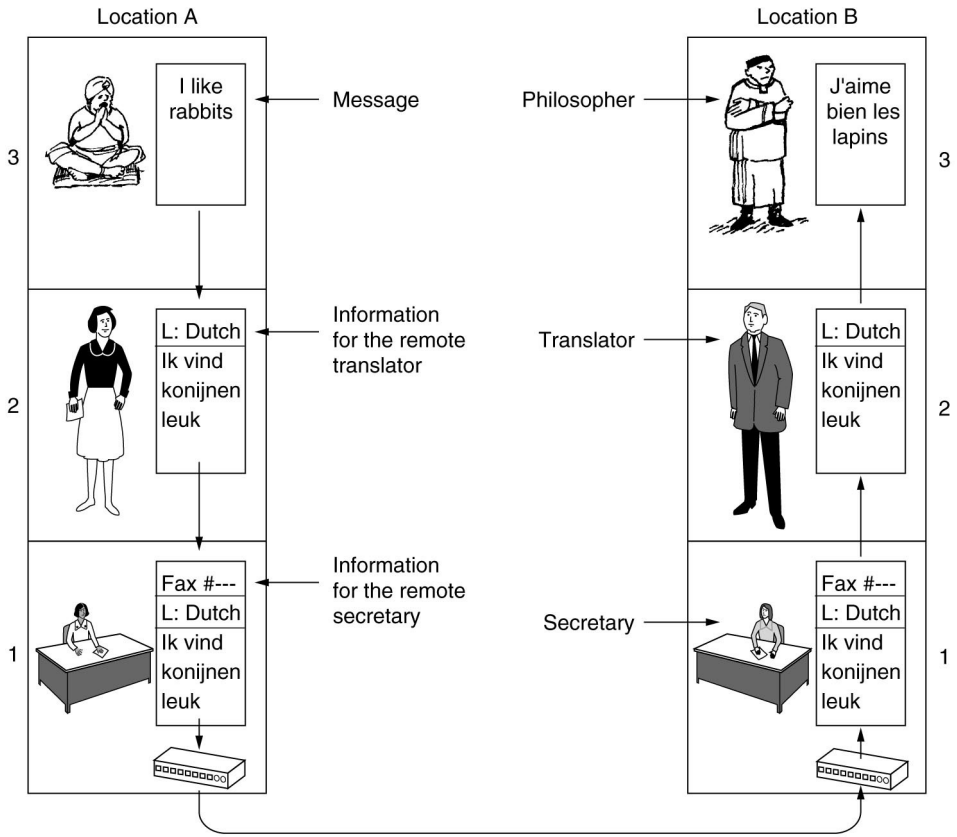
ARQUITECTURA DE RED

José Ángel Berná Galiano. DFISTS. Universidad de Alicante.

2.1 Modelo de capas

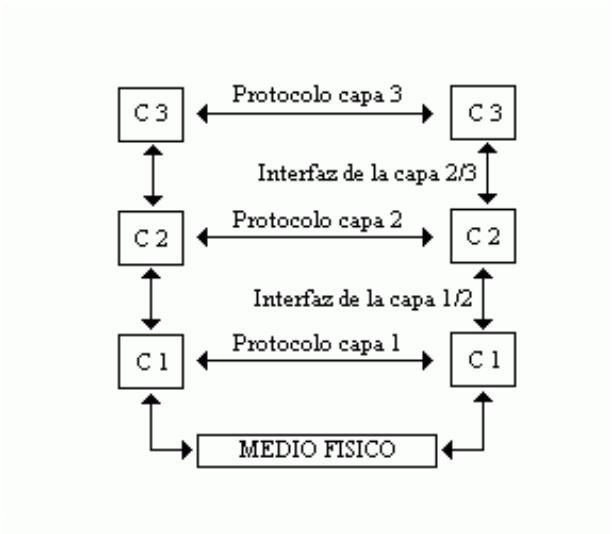
Arquitectura de red

Conjunto de protocolos perfectamente definidos e implementados que caracterizan cómo se realiza el intercambio de información en una red de comunicaciones



2.1 Modelo de capas

Modelo de capas



Capa o nivel de una arquitectura de red: Cada uno de los niveles de abstracción definidos en la comunicación.

Entidades pares: Las instancias de una capa en cada extremo de la comunicación.

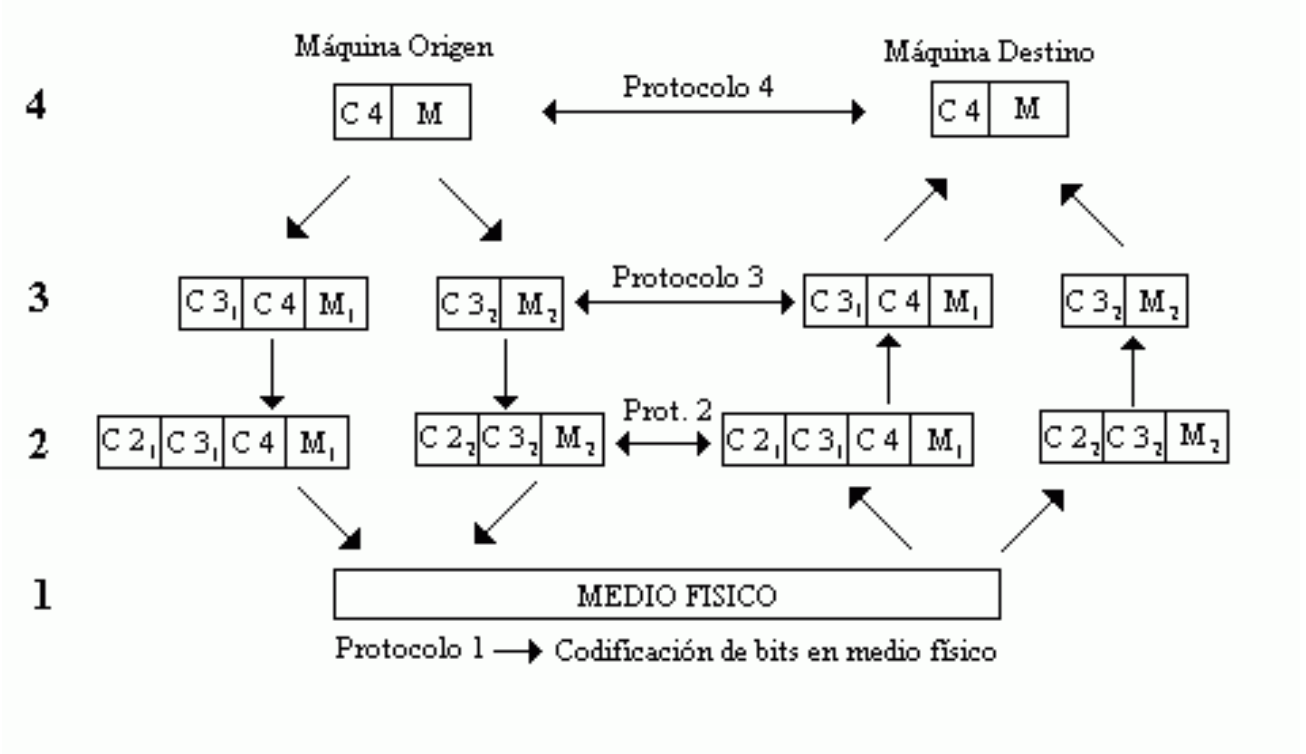
Protocolo: Conjunto de normas para la comunicación entre entidades pares

Servicios: Conjunto de funciones que una capa ofrece a su capa superior

Interfaz: Conjunto de normas para la comunicación entre capas adyacentes

2.1 Modelo de capas

Ejemplo de arquitectura de red



Protocolo 4: Definición del tipo de mensaje a intercambiar: e-mail, página web, fichero, etc.

Protocolo 3: Fragmentación del mensaje en trozos para evitar el retardo debido a errores.

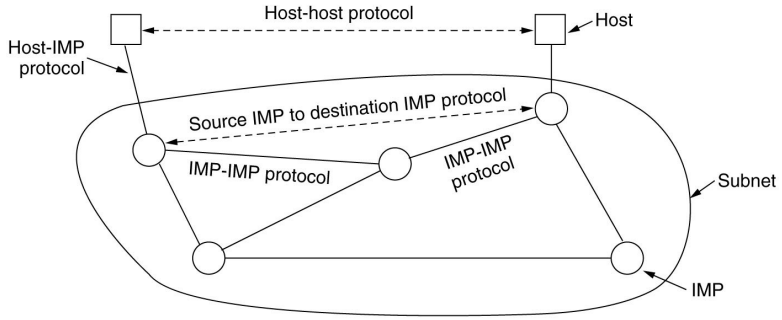
Protocolo 2: Identificación del destinatario del mensaje en la red.

Protocolo 1: Codificación de los bits en señales eléctricas.

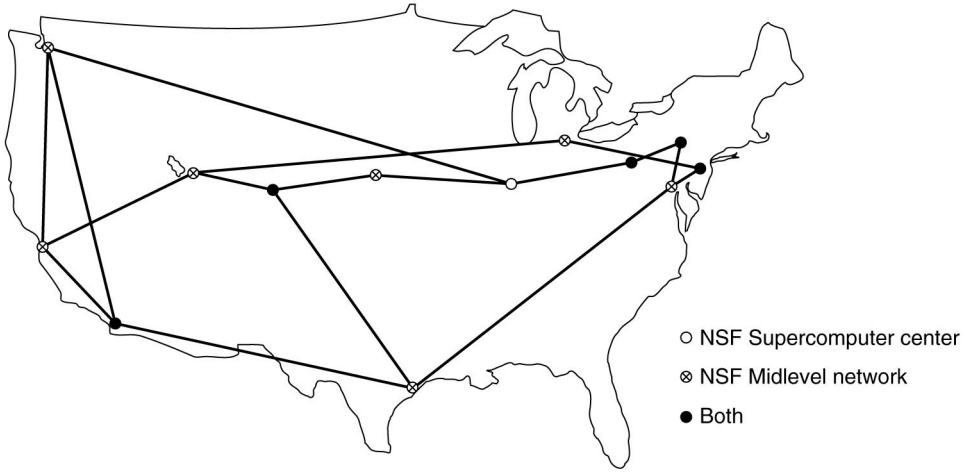
2.2 Modelo de Arquitectura TCP/IP (Internet)

El origen y desarrollo de Internet

Década de 1970: ARPANET. Red militar (DoD) en EEUU con objetivos de defensa.



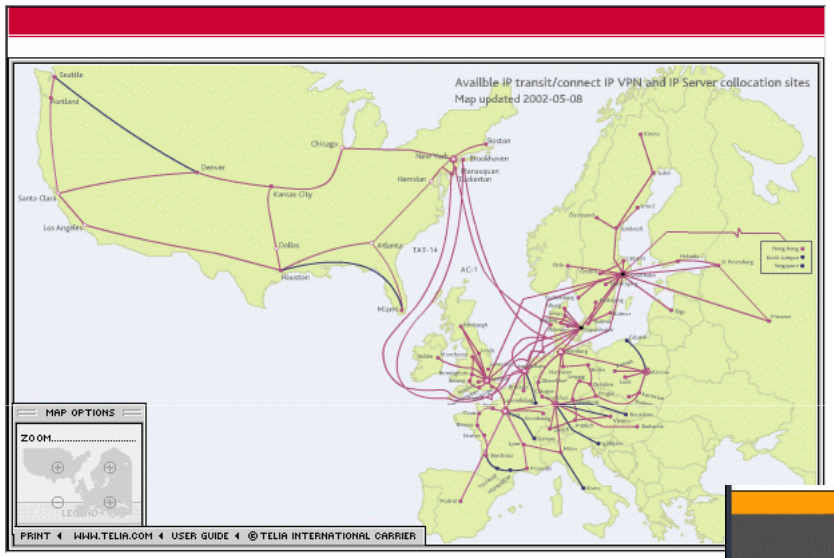
Década de 1980: ARPANET/MILNET. Separación en red de investigación y militar. Expansión de ARPANET en Universidades y centros de investigación EEUU y Europa. Unix de Berkeley.

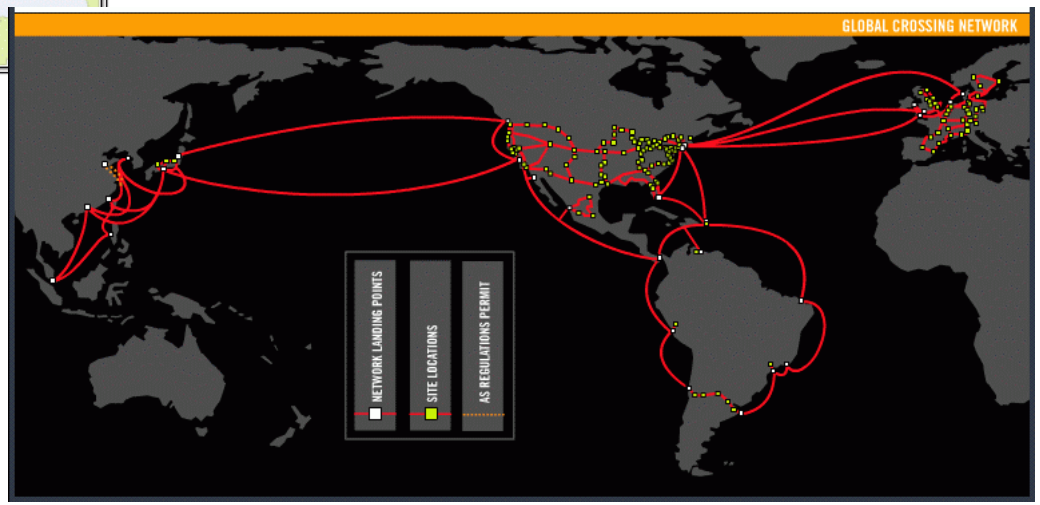


2.2 Modelo de Arquitectura TCP/IP (Internet)

El origen y desarrollo de Internet

Década 1990: Expansión de ARPANET en empresas de todo el mundo: conexión a Internet o adopción de protocolos de Internet.

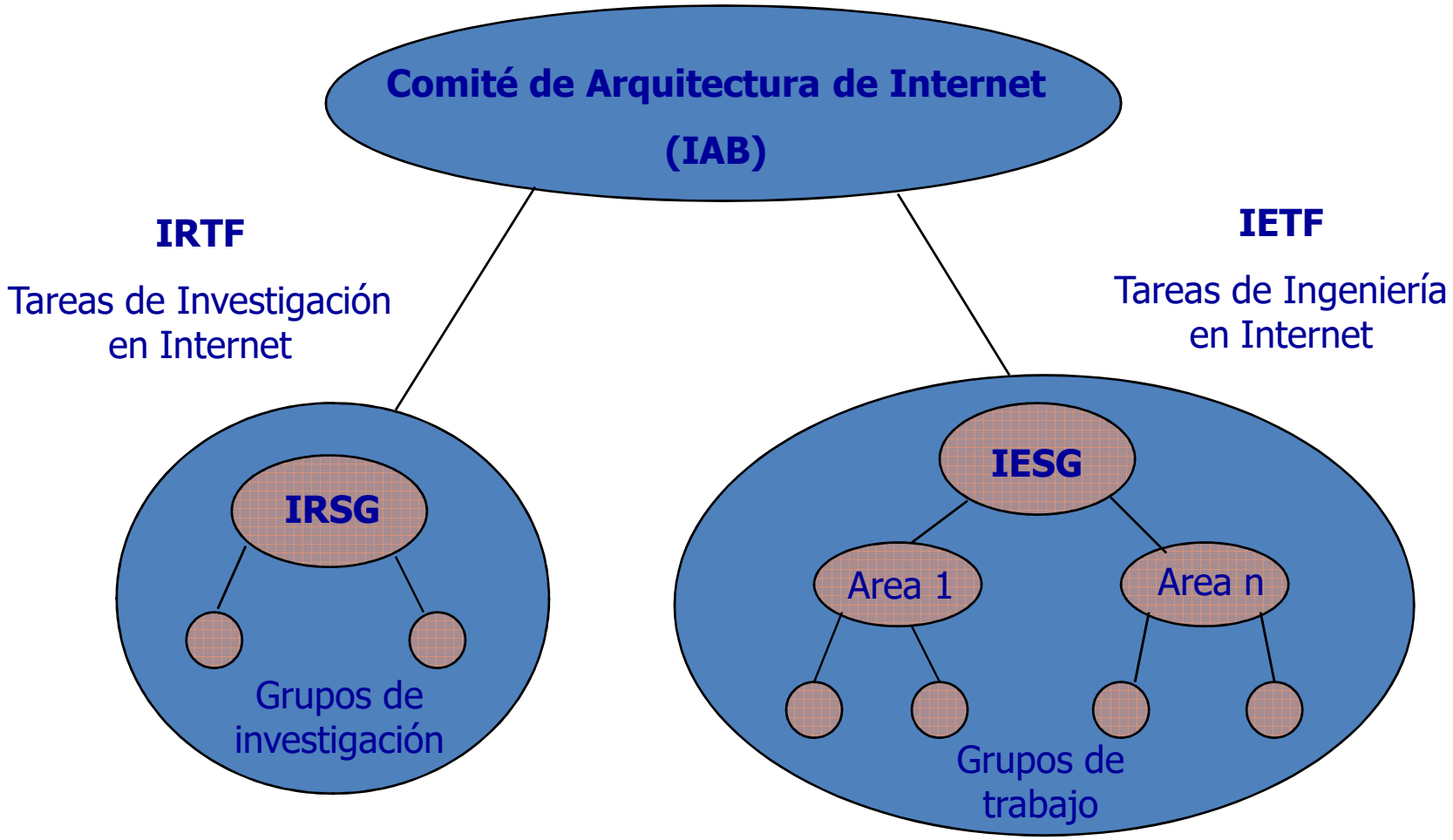




2.2 Modelo de Arquitectura TCP/IP (Internet)

El origen y desarrollo de Internet

Estructura organizativa en Internet



RFC: Request for comments

2.2 Modelo de Arquitectura TCP/IP (Internet)

Modelo de capas de TCP/IP

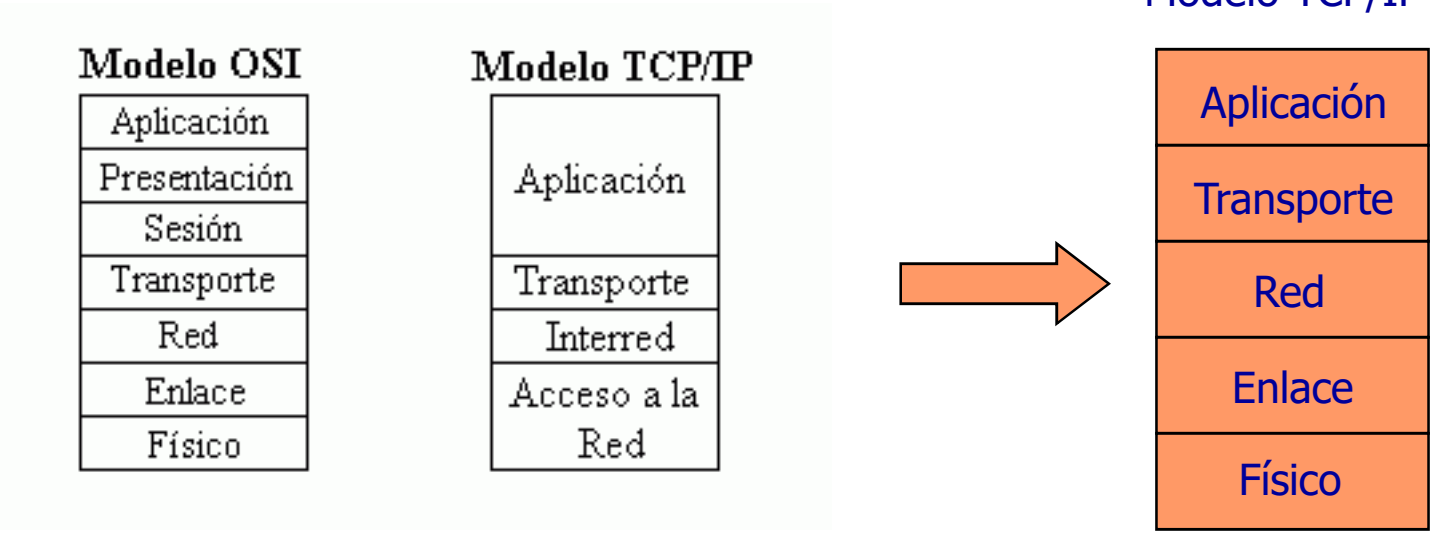
Aplicación
Transporte
Interred (Red)
Acceso a la red

Capa de aplicación: Define el conjunto de aplicaciones que ofrece Internet para la comunicación.

Capa de transporte: Permite el control de la comunicación extremo a extremo en Internet.

Capa de interred (red): Permite el encaminamiento de paquetes de información entre dos equipos de la red.

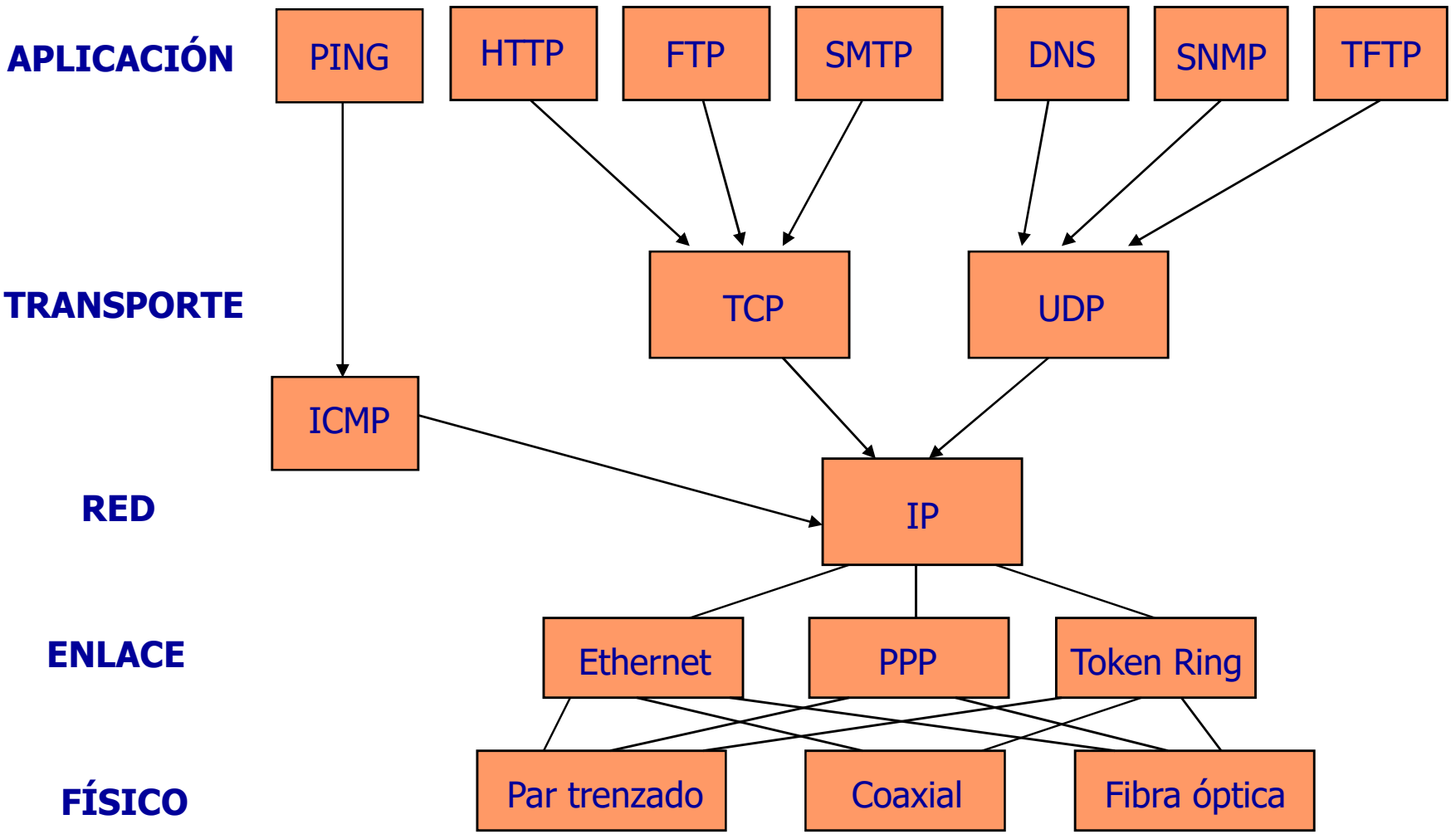
Capa de acceso al medio: Permite el envío de un paquete procedente de la capa de red (paquete IP) a través de un medio físico de comunicación



2.2 Modelo de Arquitectura TCP/IP (Internet)

Modelo de capas de TCP/IP

Protocolos de la arquitectura TCP/IP



2.2 Modelo de Arquitectura TCP/IP (Internet)

Modelo de capas de TCP/IP

Ejemplo protocolo de aplicación: DNS (Domain Name System) RFC 1035

Cada equipo de Internet está identificado con un valor único de dirección IP (x.x.x.x).

Memorizar los valores de direcciones IP es complejo y una máquina puede modificar el valor de su dirección IP, por lo que se ha desarrollado un sistema para asociar un mnemotécnico a una máquina de Internet.

Estos mnemotécnicos se denominan nombres de dominio y tienen la estructura maquina.organizacion.dominio. Por ejemplo, www.ua.es significa: servidor web de la Universidad de Alicante localizado en España.

Existe una jerarquía, donde el término más a la derecha indica el dominio de más alto nivel (.es, .fr, .org) definidos por la entidad raíz de Internet (IANA) y dentro de ese dominio se establecen subdominios gestionados por las organizaciones que lo gestionan (el dominio .es lo gestiona el gobierno de España y asigna subdominios a otras organizaciones, el subdominio ua.es lo gestiona la Universidad de Alicante).

El nombre de dominio permite identificar una máquina en Internet de manera más sencilla, pero el intercambio de datos precisa de conocer la dirección IP de esa máquina.

Aquí interviene el protocolo DNS, un mecanismo de pregunta y respuesta con el que cualquier máquina de Internet pregunta cuál es la dirección IP de un nombre de dominio.

Esta información (asociación nombre de dominio – dirección IP, www.ua.es tiene asociada la dirección 193.145.235.30) está almacenada en cientos de miles de máquina de Internet denominados servidores DNS. Para evitar la congestión en Internet esta información se almacena de forma distribuida y en cada máquina de Internet se indica las direcciones IP de los servidores DNS donde puede realizar las consultas (la Universidad de Alicante dispone como servidores DNS los equipos con direcciones IP 193.145.233.5 y 193.145.233.6).

2.2 Modelo de Arquitectura TCP/IP (Internet)

Modelo de capas de TCP/IP

Capa Física

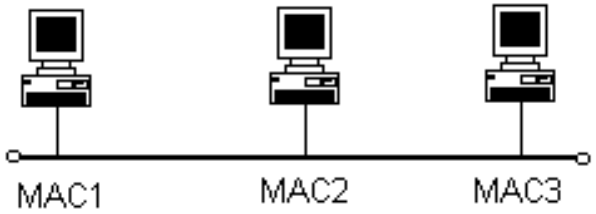
- Especificación de los medios físicos empleados en la comunicación
- Especificación de la señalización de la información en el medio físico

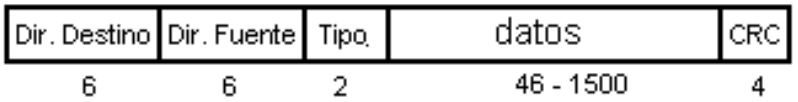
Ejemplo: cables pares trenzados, cable coaxial, fibra óptica

Capa de Enlace

- Especificación de los mecanismos para el intercambio de información en un medio físico

Ejemplo: Ethernet





2.2 Modelo de Arquitectura TCP/IP (Internet)

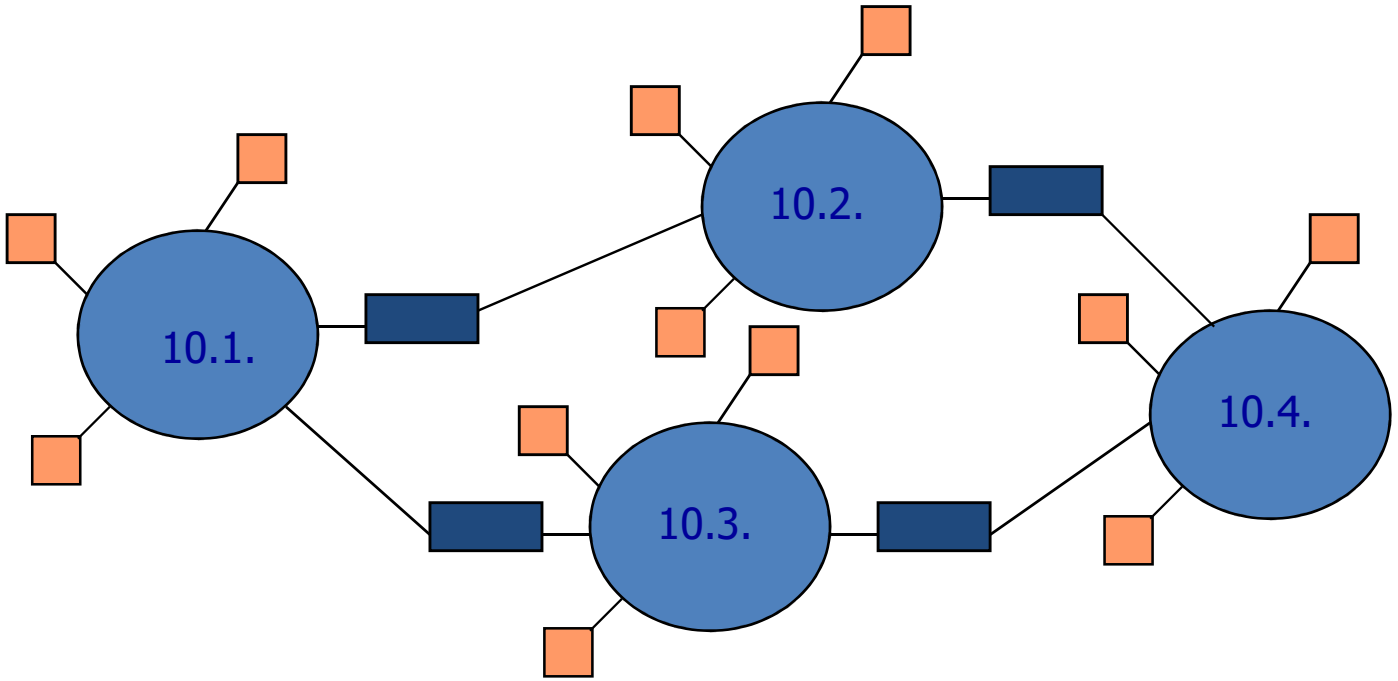
Modelo de capas de TCP/IP

Capa de red. Protocolo IP

- Identificación de equipos en una red formada por la interconexión de redes (Internet)
- Encaminamiento de paquetes en la red (Internet)

Direccionamiento IP

- Identificador de 32 bits ⇒ X . X . X . X ⇒ 0-255 . 0-255 . 0-255 . 0-255



2.2 Modelo de Arquitectura TCP/IP (Internet)

Modelo de capas de TCP/IP

- Dirección IP 192.168.17.23

¿ Identificador de red ? ⟹ Máscara de red de una red IP

Valor de 32 bits (X.X.X.X) ⟹ 111111111..10000000000000000

Máscara de red = 255.255.255.0 ⟹ 192.168.17.23 pertenece a la red 192.168.17.

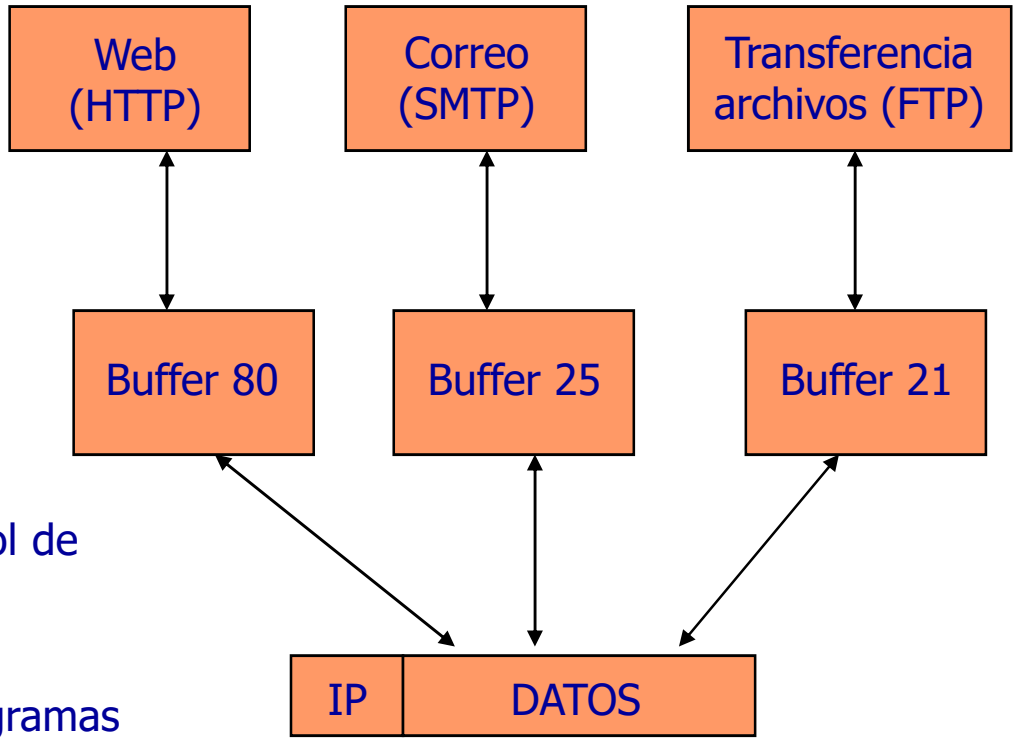
192.168.17.

192.168.17.0	Dirección de red
192.168.17.1	
192.168.17.2	
....	
....	
192.168.17.255	Dirección de broadcast

Modelo de capas de TCP/IP

Capa de transporte. Protocolos TCP y UDP

- Interfaz entre la capa de aplicación y red para la gestión de comunicaciones extremo a extremo (conexiones) entre equipos de Internet.



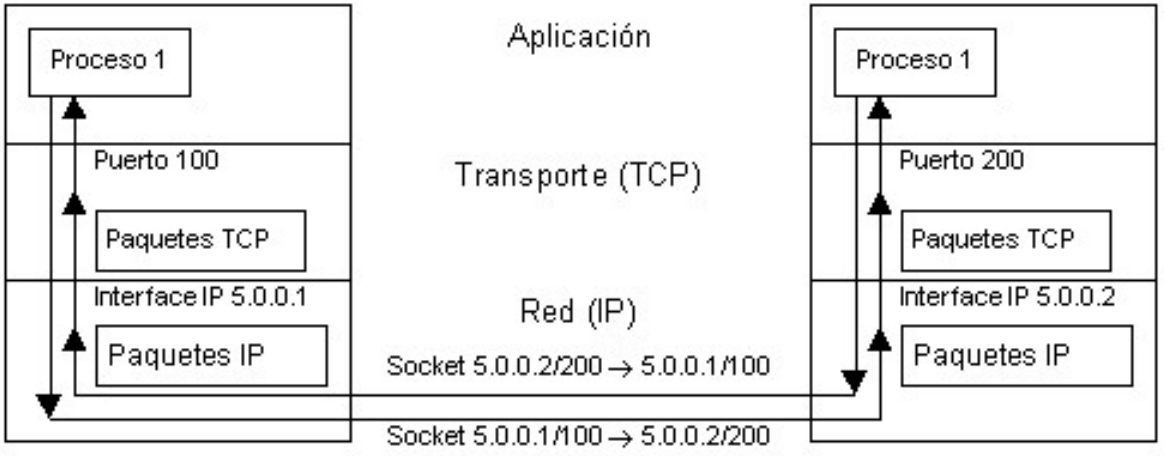
TCP: Protocolo de control de la transmisión (fiable)

UDP: Protocolo de datagramas de usuario (NO fiable)

2.2 Modelo de Arquitectura TCP/IP (Internet)

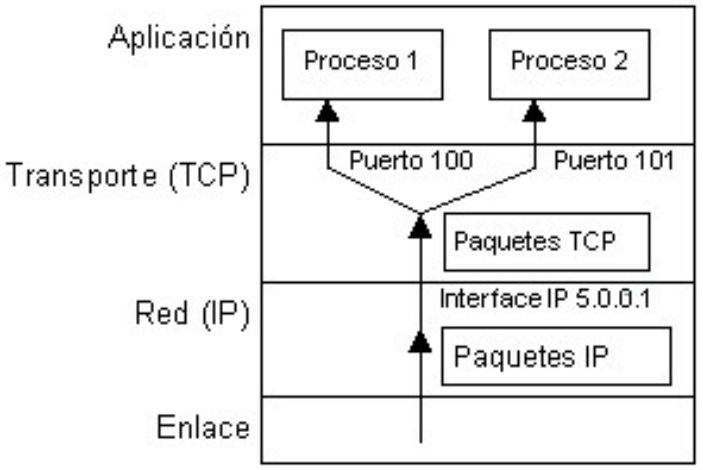
Modelo de capas de TCP/IP

Gestión de conexiones. Sockets



IP	TCP	
5.0.0.1 -> 5.0.0.2	100 -> 200	DATOS

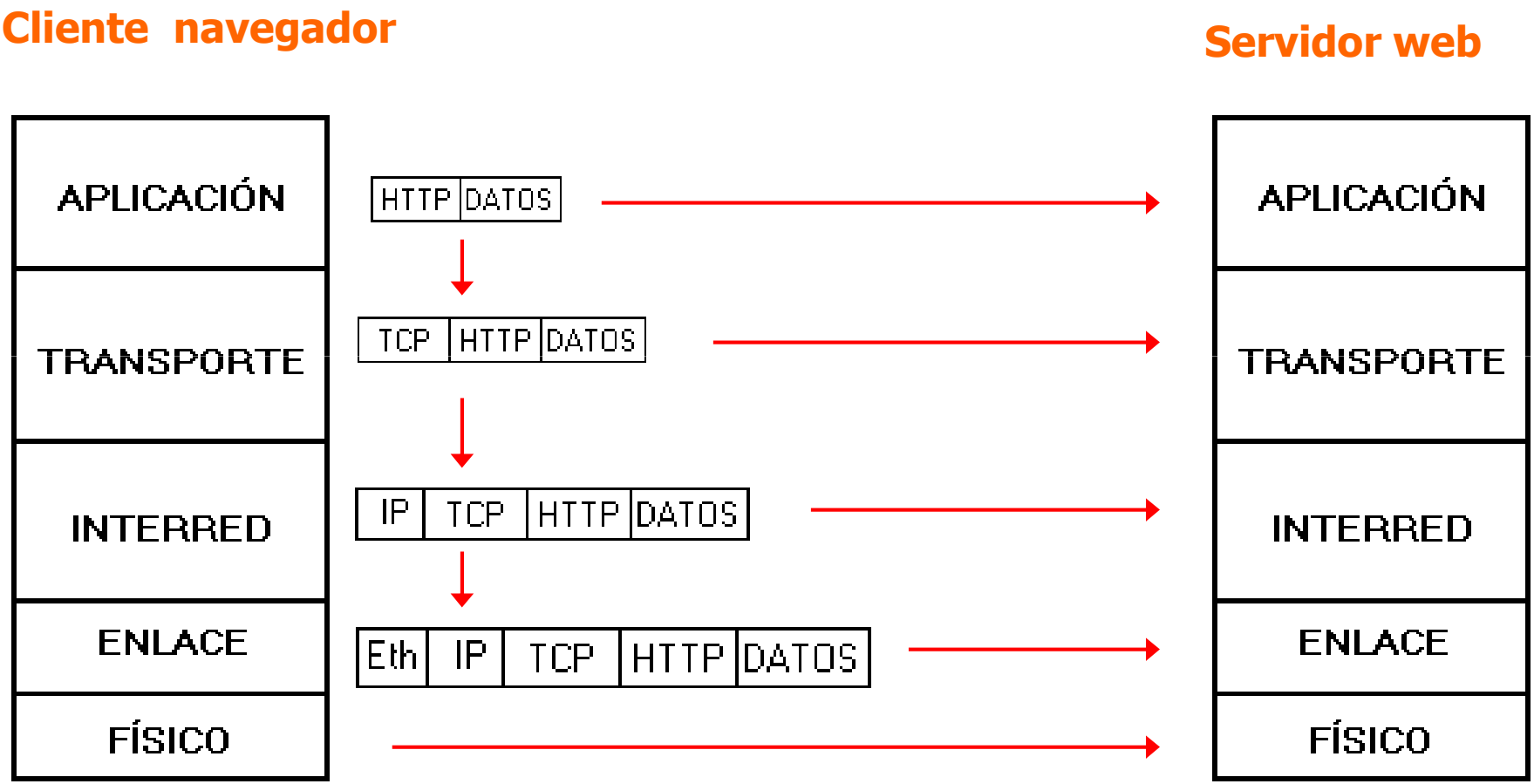
IP	TCP	
5.0.0.2 -> 5.0.0.1	200 -> 100	DATOS



2.2 Modelo de Arquitectura TCP/IP (Internet)

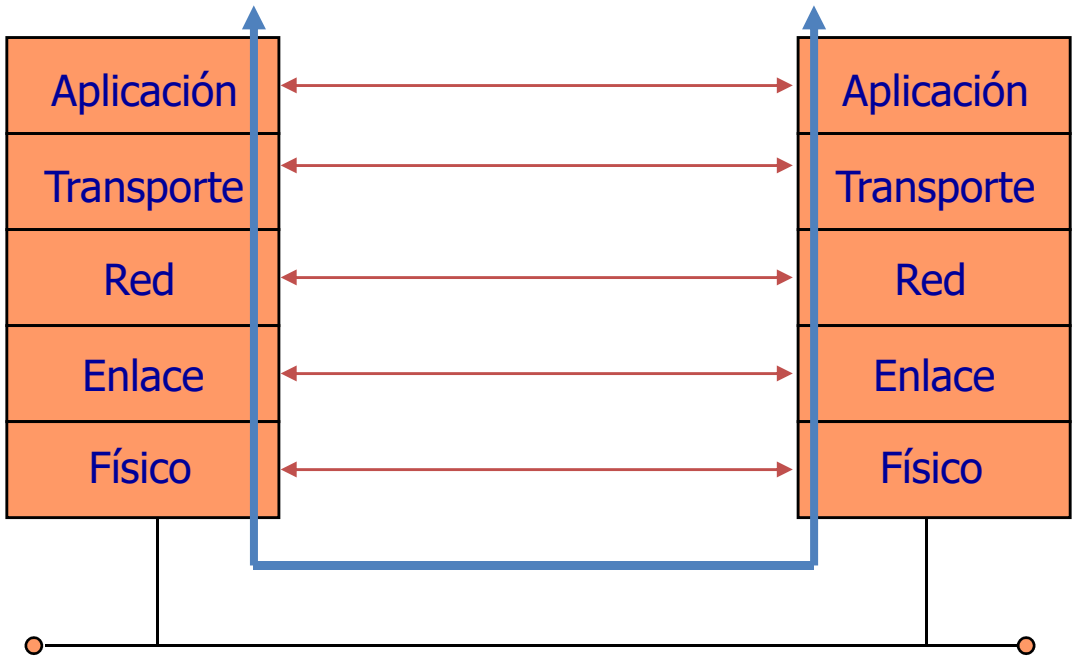
Modelo de capas de TCP/IP

Capa de aplicación. Protocolo HTTP



2.3 Interconexión de redes

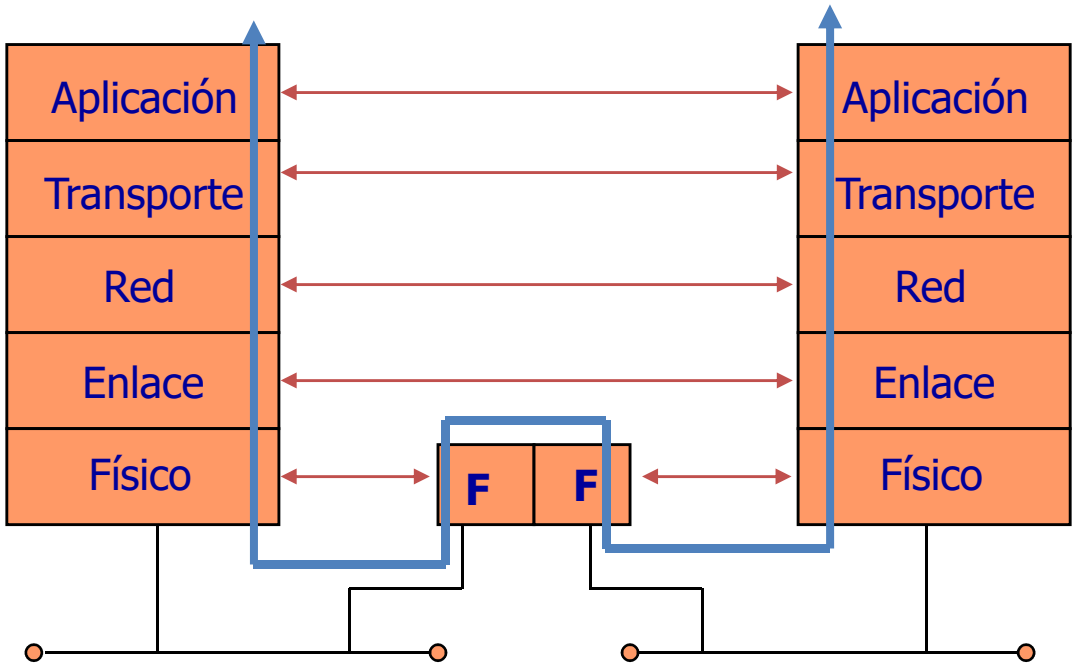
Modelo de comunicación entre capas en una red



En base a este modelo de comunicación, se puede estudiar la necesidad de diferentes tipos de dispositivos para interconectar diferentes segmentos físicos de red.

2.3 Interconexión de redes

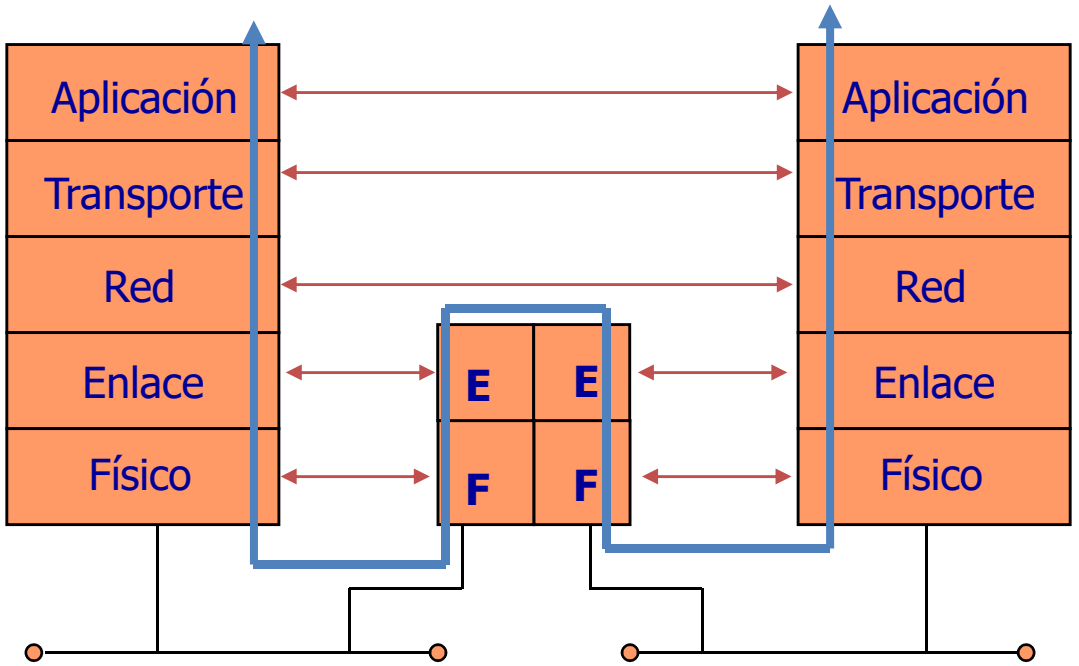
Interconexión de redes a nivel físico. Repetidor (Repeater)



Dispositivo sencillo y económico que proporciona muy poco rendimiento y situaciones de colisiones permanentes.

2.3 Interconexión de redes

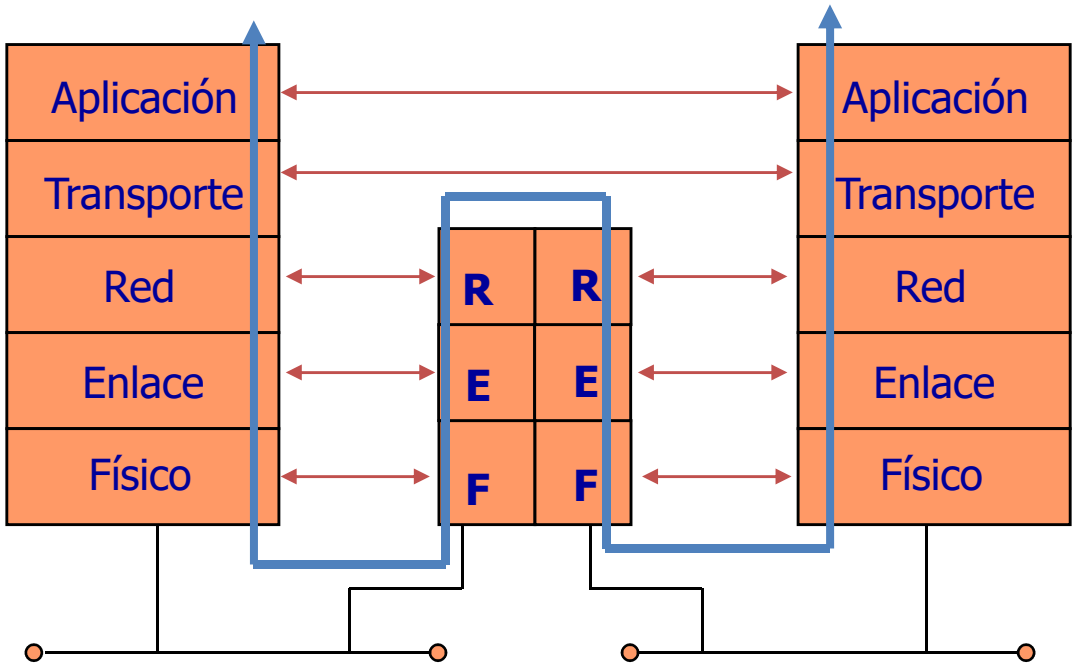
Interconexión de redes a nivel de enlace. Puente (Bridge)



Dispositivo que presenta un buen rendimiento al evitar transmisiones innecesarias.
Limitado en cuanto a los tipos de redes a interconectar.

2.3 Interconexión de redes

Interconexión de redes a nivel de red. Encaminador (Router)



Dispositivo con rendimiento de interconexión menor que los puentes, pero aplicable para la interconexión de cualesquiera segmentos de red que soporten un protocolo de red común (IP).

2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Especificación de un protocolo

Definición: Conjunto de reglas de utilización de las funciones suministradas por el nivel inferior (envío/recepción de bloques de datos) para llevar a cabo la comunicación a nivel horizontal.

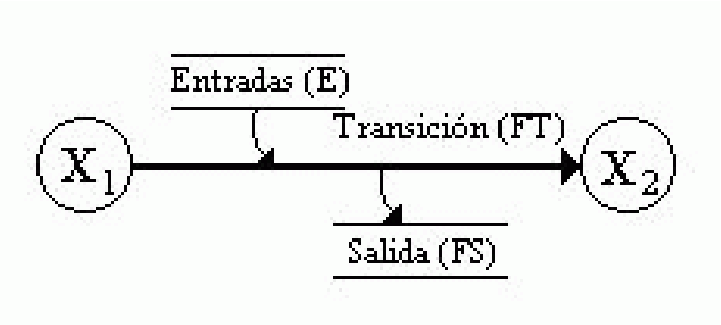
Elementos de una máquina de estado finito

Estados: Descripción de las situaciones de funcionamiento del protocolo

Entradas: Eventos que provocan cambios en el estado del protocolo

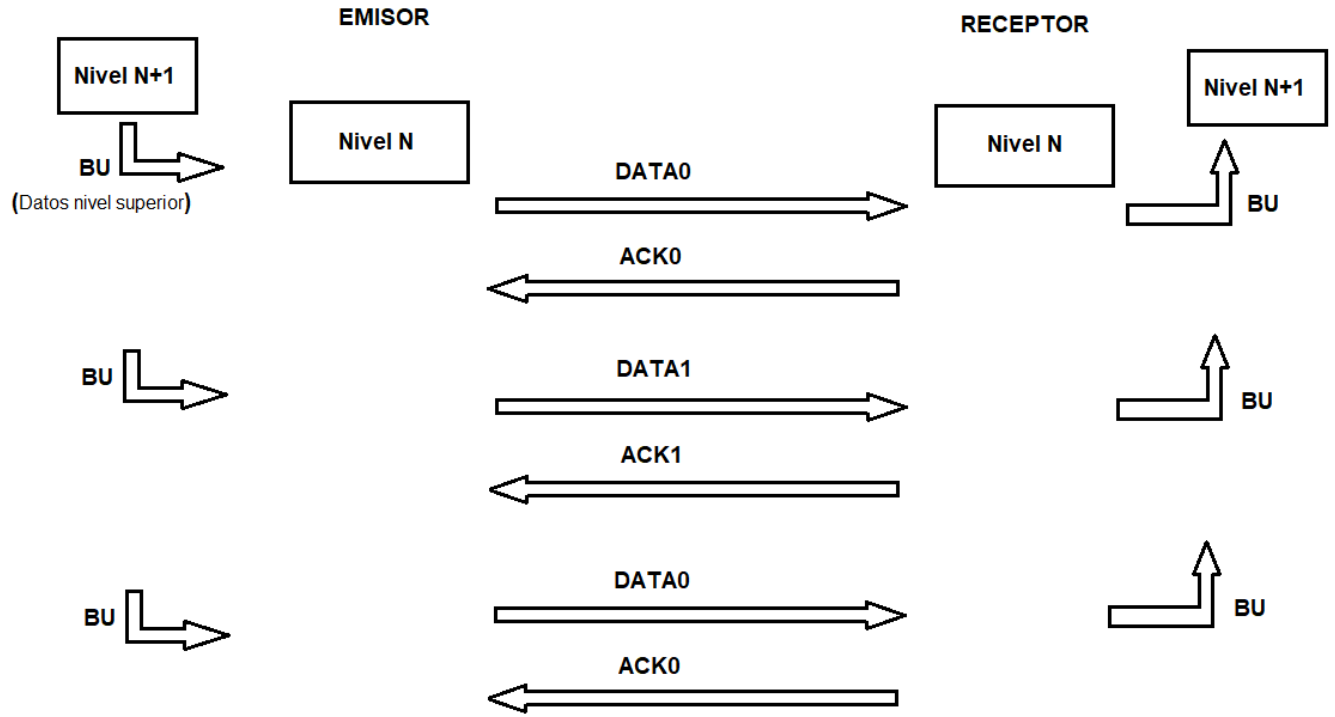
Salidas: Acciones como consecuencia de cambios en el estado del protocolo

Transición: Proceso por el cual un protocolo cambia de un estado de funcionamiento a otro.



2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Ejemplo de protocolo: Protocolo bilateral de parada y espera



ESTADOS EMISOR

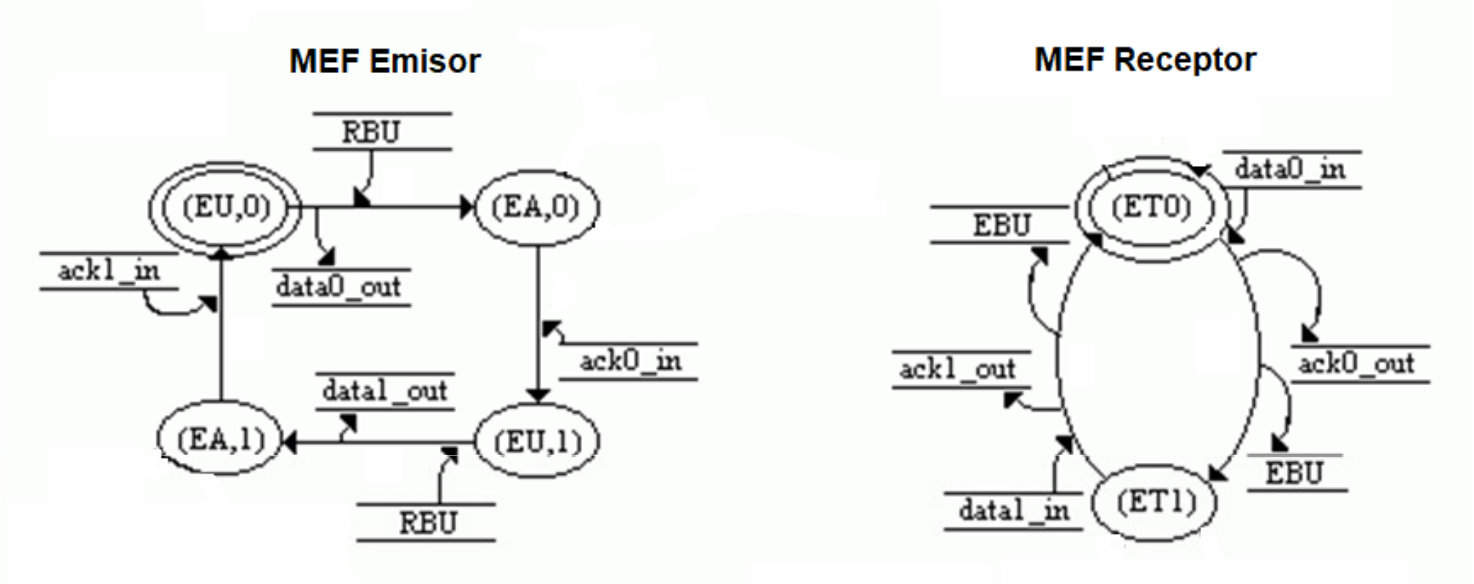
EU0 → El emisor espera un bloque de datos del nivel superior para numerar con 0.
EU1 → El emisor espera un bloque de datos del nivel superior para numerar con 1.
EA0 → El emisor espera un ACK0.
EA1 → El emisor espera un ACK1.

ESTADOS RECEPTOR

ET0 → El receptor espera datos con numeración 0.
ET1 → El receptor espera datos con numeración 1.

2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Ejemplo de protocolo: Protocolo bilateral de parada y espera



Eventos de entrada

RBU → Emisor recibe un bloque de datos del nivel superior.
ACK0_IN → Emisor recibe un ACK0.
ACK0_IN → Emisor recibe un ACK1.

DATA0_IN → Receptor recibe DATA0.
DATA1_IN → Receptor recibe DATA1.

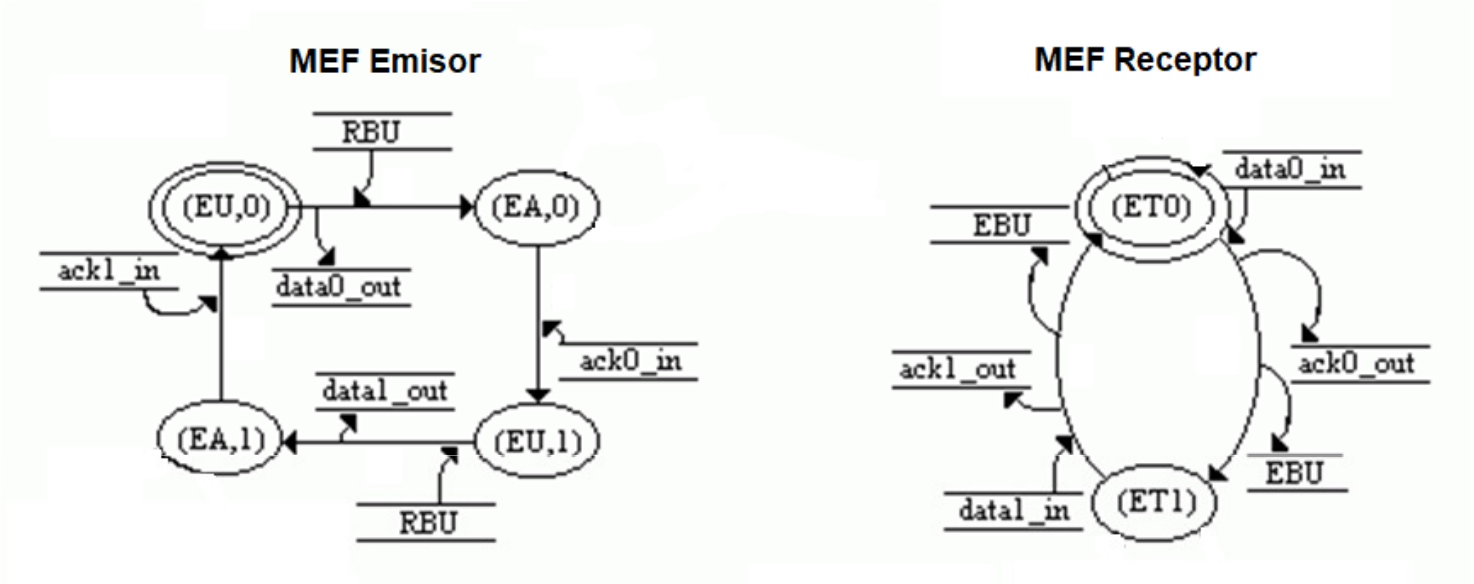
Eventos de salida

DATA0_OUT → Emisor envía DATA0.
DATA1_OUT → Emisor envía DATA1.

EBU → Receptor envía un bloque de datos al nivel superior.
ACK0_OUT → Receptor envía ACK0.
ACK1_OUT → Receptor envía ACK1.

2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Ejemplo de protocolo: Protocolo bilateral de parada y espera



Situaciones de error pérdidas de información en el canal (errores de transmisión)

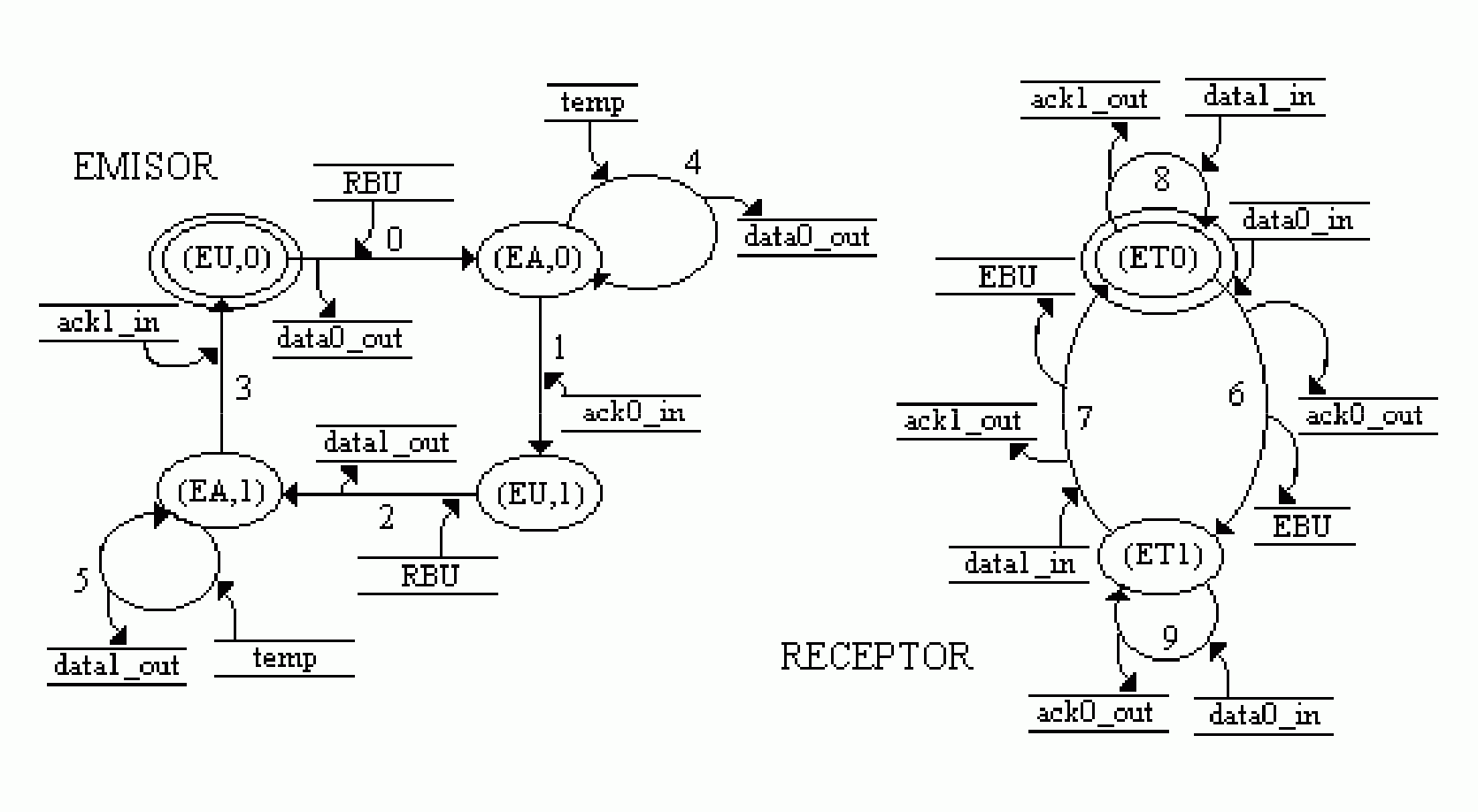
El emisor dispone de un temporizador de espera de ACK. Si en el estado de espera de ACK expira el temporizador, el emisor reenvía los últimos datos transmitidos (DATA0/DATA1).

Si un bloque de datos (DATA0/DATA1) no llega al receptor, el emisor no recibirá ACK y reenviará los datos correctamente.

Si el ACK0(1) no llega al emisor, el emisor reenviará DATA0(1), que ya han sido recibidos por el receptor. En ese caso (recepción de datos fuera de secuencia) el receptor reenvía el ACK0(1) y NO envía los datos al nivel superior.

2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Ejemplo de protocolo: Protocolo bilateral de parada y espera



Eventos de entrada

TEMP → Expira el temporizador de espera de ACK.

2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Protocolo para la actualización del software de un antivirus

Una empresa de desarrollo de software comercializa un programa antivirus. Este antivirus precisa de actualizaciones para los clientes que lo han adquirido. Para ello ha desarrollado un protocolo para la comunicación entre el programa antivirus y un centro de actualizaciones.

El programa antivirus dispone de un temporizador de 24 horas para comprobar la existencia de actualizaciones del programa. Cada 24 horas, el antivirus envía al centro de actualizaciones una petición de actualización con la versión del programa antivirus. Si el centro de actualizaciones contesta indicando que no existen actualizaciones, el programa antivirus activa el temporizador de 24 horas y espera a que expire de nuevo. Si el centro de actualizaciones contesta indicando que hay actualizaciones, el programa antivirus envía una clave de cifrado al centro de actualizaciones y pasa a esperar recibir el contenido de la actualización cifrada. Cuando el programa antivirus recibe la actualización, la instala y envía al centro de actualizaciones una confirmación de actualización. Además, inicia el temporizador de 24 horas y pasa a esperar su expiración.

Determina los estados, eventos de entrada y salida, y la MEF que describe el funcionamiento del programa antivirus en este protocolo.

ESTADOS

ETMP → El programa antivirus espera la expiración del temporizador de 24 horas.

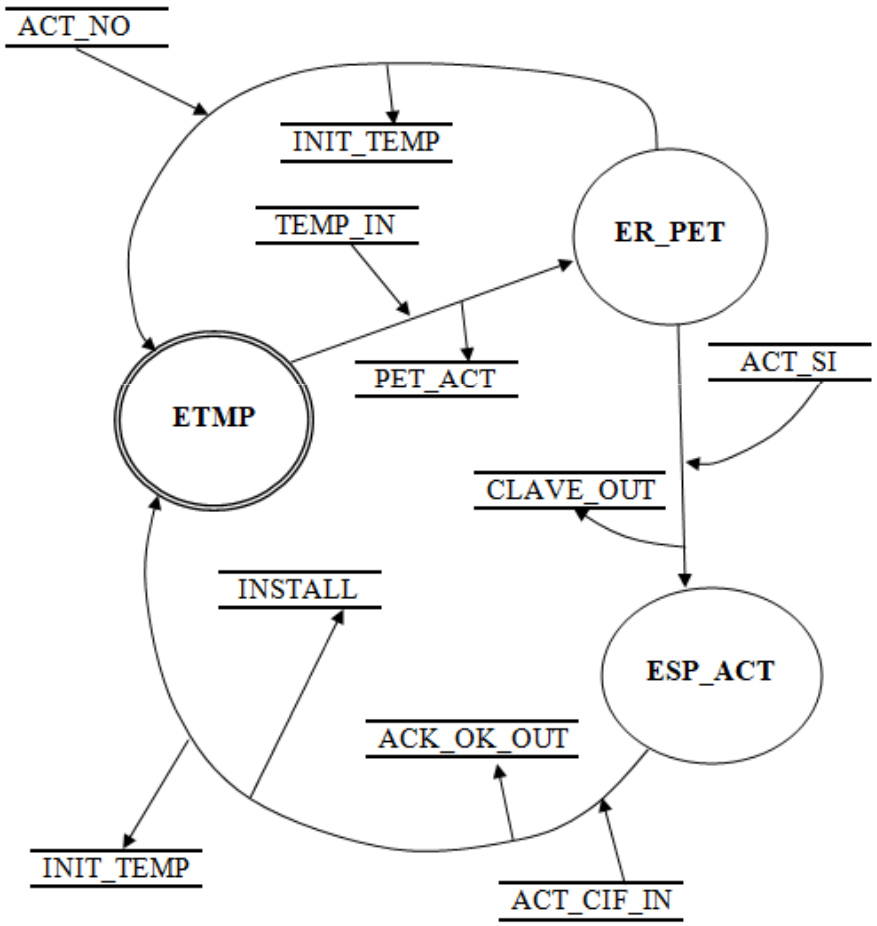
ER_PET → El programa antivirus espera la respuesta del centro de actualizaciones.

ESP_ACT → El programa antivirus espera la actualización del centro de actualizaciones.

2.4 Modelado de protocolos. Máquinas de estado finito (MEF)

Protocolo para la actualización del software de un antivirus

MEF SOFTWARE ANTIVIRUS



EVENTOS DE ENTRADA

TEMP_IN → Expira el temporizador de 24 horas.
ACT_SI → El programa antivirus recibe una respuesta con existencia de actualización
ACT_NO → El programa antivirus recibe una respuesta con NO existencia de actualización
ACT_CIF_IN → El programa antivirus recibe la actualización cifrada.

EVENTOS DE SALIDA

PET_ACT → El programa antivirus envía una petición de actualización.
INIT_TEMP → El programa antivirus inicia el temporizador de 24 horas.
CLAVE_OUT → El programa antivirus envía la clave de cifrado al centro de actualizaciones.
INSTALL → El programa antivirus instala la actualización.
ACT_OK_OUT → El programa antivirus envía la confirmación de actualización al centro de actualizaciones.