**Resumen Redes de Información 2018**

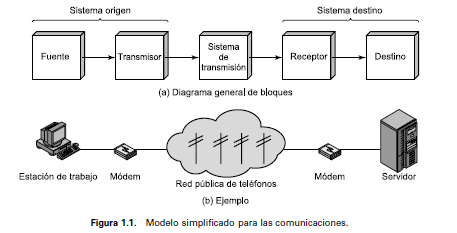
**Un Modelo para las comunicaciones:**

La **transmisión de datos** es el **intercambio de datos entre dos dispositivos** a través de alguna forma de medio de transmisión, como un cable. Para que la transmisión de datos sea posible, los dispositivos de comunicación deben ser parte de un sistema de comunicación formado por hardware (equipo físico) y software (programas).

El **objetivo** principal de todo sistema de comunicación es **intercambiar información entre dos entidades.**

L**os elementos claves de un sistema de comunicación son:**

* **La fuente:** este dispositivo genera los datos a transmitir
* **El transmisor:** el transmisor transforma y codifica la información, generando señales electromagnéticas susceptibles de ser transmitidas a través de algún sistema de transmisión.
* **El sistema de transmisión**: puede ser desde una sencilla línea de transmisión hasta una compleja red que conecte a la fuente con el destino.
* **El receptor**: el receptor acepta la señal proveniente del sistema de transmisión y la transforma de tal manera que pueda ser manejada por el dispositivo de destino. Por ejemplo, un modem captará la señal analógica de la red o línea de transmisión y la convertirá en una cadena de bits.
* **El destino:** toma los datos del receptor.



**Tareas en los sistemas de comunicación:**

Utilización del sistema de transmisión: se refiere a la necesidad de hacer un **uso eficaz de los recursos utilizados en la transmisión**, los cuales se suelen compartir habitualmente entre una serie de dispositivos de comunicación. La capacidad total del medio de transmisión se reparte entre los distintos usuarios haciendo uso de técnicas denominadas de multiplexación.

Implementación de la interfaz: Para que un dispositivo pueda transmitir información tendrá que hacerlo a través de la **interfaz con el medio de transmisión**. Todas las técnicas de transmisión dependen de la utilización de señales electromagnéticas que se transmiten a través del medio.

Generación de la señal: una vez que la interfaz está establecida, será necesaria la generación de la señal. Las características de la señal, como la forma y la intensidad, deben ser tales que permitan que la señal se propague a través del medio de transmisión y que se interprete en el receptor como datos.

Sincronización: Las señales se deben generar no solo considerando que deben cumplir los requisitos del sistema de transmisión y del receptor, sino que también deben permitir alguna forma de sincronizar el receptor y el emisor. El receptor debe ser capaz de determinar cuándo comienza y cuando acaba la señal recibida. Igualmente deberá conocer la duración de cada elemento de señal.

Gestión del intercambio: además se necesitará verificar un conjunto de requisitos que se pueden englobar bajo el término gestión del intercambio. Si se necesitan intercambiar datos durante un periodo de tiempo, las dos partes deben cooperar.

Detección y corrección de errores: En todos los sistemas de comunicación es posible que aparezcan errores, ya que la señal transmitida se distorsiona siempre antes de alcanzar su destino. Por lo tanto, se necesitan procedimientos para la detección y corrección de errores.

Control de flujo: Para evitar que la fuente no sature el destino transmitiendo datos más rápidamente de los que el receptor pueda procesar y absorber, se necesitan una serie de procedimientos denominados control de flujo.

Direccionamiento y encaminamiento: cuando cierto recurso de transmisión se comparte con mas de dos dispositivos, el sistema fuente deberá, de alguna manera, indicar la identidad del destino. El sistema de transmisión deberá garantizar que ese destino, y solo ese, recibe los datos. Es más, el sistema de transmisión puede ser una red en la que exista la posibilidad de usar mas de un camino para alcanzar el destino; en este caso se necesitará, por tanto, la elección de una entre las posibles rutas.

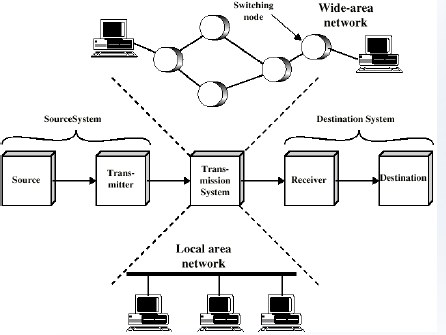
Recuperación: en ciertas ocasiones en las que el intercambio de información se ve interrumpido por algún fallo, se necesitará un mecanismo de recuperación. El objetivo será, o bien ser capaz de continuar transmitiendo desde donde se produjo la interrupción, o al menos recuperar el estado en el que se encontraban los sistemas antes de comenzar el intercambio.

Formato de Mensajes: esta relacionado con el acuerdo que debe existir entre las dos partes respecto al formato de los datos intercambiados, como, por ejemplo, el código binario usado para representar los caracteres.

Seguridad: es necesario dotar algunas medidas de seguridad. El emisor puede querer asegurarse de que sólo el destino deseado reciba los datos. Igualmente, el receptor querrá estar seguro de que los datos recibidos no se han alterado en la transmisión y que dichos datos realmente provienen del supuesto emisor.

Gestión de Red: se necesitan funcionalidades de gestión de red para configurar el sistema, monitorizar su estado, reaccionar ante fallos y sobrecargas y planificar con acierto los crecimientos futuros.

Modelo Simplificado de Redes:



**REDES DE ÁREA AMPLIA (WAN):**

Generalmente, se considera como redes de área amplia a todas aquellas que cubren una **extensa área geográfica**, requieren atravesar **rutas de acceso público** y utilizan, al menos parcialmente, circuitos proporcionados por una entidad proveedora de servicios de telecomunicación.

Generalmente, una WAN consiste en una serie de dispositivos de conmutación interconectados.

La **transmisión generada por cualquier dispositivo se encaminará a través de estos nodos internos hasta alcanzar el destino.** A estos nodos (incluyendo los situados en los contornos) no les concierne el contenido de los datos, al contrario, su función es proporcionar el servicio de conmutación, necesario para transmitir los datos de nodo en nodo hasta alcanzar su destino final.

Tradicionalmente, las WAN se han implementado usando una de las dos tecnologías siguientes: **conmutación de circuitos** y **conmutación de paquetes**. Últimamente, se está empleando como solución la técnica de **retransmisión de tramas (frame relay)**, así como las redes **ATM.**

**Conmutación de Circuitos:**

En las redes de conmutación de circuitos, para interconectar dos estaciones se **establece un camino dedicado a través de los nodos de la red**. El **camino es una secuencia conectada de enlaces físicos entre nodos**. En cada enlace, se dedica un canal lógico a cada conexión. **Los datos generados por la estación fuente se transmiten por el camino dedicado tan rápido como se pueda.** En cada nodo, los datos de entrada se encaminan o conmutan por el canal apropiado de salida sin retardos. El ejemplo más ilustrativo de la conmutación de circuitos es la **red de telefonía**.

**Conmutación de Paquetes:**

Un enfoque diferente al anterior es el adoptado en las redes de conmutación de paquetes. En este caso, **no es necesario hacer una asignación a priori de recursos** (capacidad de transmisión) en el camino (o sucesión de nodos). Por el contrario, **los datos se envían en secuencias de pequeñas unidades llamadas paquetes. Cada paquete se pasa de nodo en nodo en la red siguiendo algún camino entre la estación origen y la destino**. En cada nodo, el paquete se recibe completamente, se almacena durante un breve intervalo y posteriormente se retransmite al siguiente nodo. Las redes de conmutación de paquetes se usan fundamentalmente para las comunicaciones terminal-computador y computador-computador.

**Retransmisión de tramas (frame relay)**

La conmutación de paquetes se desarrolló en la época en la que los servicios de transmisión a larga distancia presentaban una tasa de error relativamente elevada, comparada con los servicios de los que se dispone actualmente. Por tanto, para compensar esos errores relativamente frecuentes, **en los esquemas de conmutación de paquetes se realiza un esfuerzo considerable, que se traduce en añadir información redundante en cada paquete**, así como en la realización de un procesamiento extra, tanto en el destino final como en los nodos intermedios de conmutación, **necesario para detectar los errores y, en su caso, corregirlos.**

Ahora bien, con **los modernos sistemas de telecomunicación de alta velocidad**, **este esfuerzo adicional es innecesario e incluso contraproducente**. Es innecesario ya **que la tasa de errores se ha reducido drásticamente** y los **escasos errores que aparecen se pueden tratar en el sistema fina**l mediante dispositivos que operan por encima del nivel de la lógica dedicada a la conmutación de paquetes.

A su vez, es contraproducente ya que los bits redundantes introducen un desaprovechamiento de parte de la capacidad proporcionada por la red. La tecnología de retransmisión de tramas se ha desarrollado teniendo presente que las velocidades de transmisión disponibles en la actualidad son mayores, así como que las tasas de error actuales son menores. Mientras que las redes originales de conmutación de paquetes se diseñaron para ofrecer una velocidad de transmisión al usuario final de 64 kbps, las redes con retransmisión de tramas están diseñadas para operar eficazmente a velocidades de transmisión de usuario de hasta 2 Mbps. La clave para conseguir estas velocidades reside en eliminar la mayor parte de la información redundante usada para el control de errores y, en consecuencia, el procesamiento asociado.

**ATM (Modo de Transferencia Asíncrono)**

El Modo de Transferencia Asíncrono (ATM, Asynchronous Transfer Mode), a veces denominado como modo de retransmisión de celdas (cell relay), es la culminación de todos los desarrollos en conmutación de circuitos y conmutación de paquetes. ATM se puede considerar como una **evolución de la retransmisión de tramas**. La diferencia más obvia entre retransmisión de tramas y ATM es que la primera usa paquetes de longitud variable, llamados «tramas», y ATM **usa paquetes de longitud fija denominados «celdas».** Al igual que en retransmisión de tramas, **ATM introduce poca información adicional para el control de errores**, confiando en la inherente robustez del medio de transmisión, así como en la lógica adicional localizada en el siste

ma destino para detectar y corregir errores. **Al utilizar paquetes de longitud fija, el esfuerzo adicional de procesamiento se reduce incluso todavía más que en retransmisión de tramas**. El resultado es que **ATM se ha diseñado para trabajar a velocidades de transmisión del orden de 10 a 100 Mbps, e incluso del orden de Gbps.**

ATM se puede considerar, a su vez, como una **evolución de la conmutación de circuitos**. En la conmutación de circuitos se dispone solamente de circuitos a velocidad fija de transmisión entre los sistemas finales**. ATM permite la definición de múltiples canales virtuales con velocidades de transmisión que se definen dinámicamente en el instante en el que se crea el canal virtual**. Al utilizar celdas de tamaño fijo, **ATM es tan eficaz que puede ofrecer un canal a velocidad de transmisión constante**, aunque esté usando una técnica de conmutación de paquetes. Por tanto, en este sentido, ATM es una generalización de la conmutación de circuitos en la que se ofrecen varios canales, en los que la velocidad de transmisión se fija dinámicamente para cada canal según las necesidades.

**REDES DE ÁREA LOCAL (LAN):**

Al igual que las redes WAN, una LAN es una **red de comunicaciones que interconecta varios dispositivos y proporciona un medio para el intercambio de información entre ellos**. No obstante, hay algunas diferencias entre las LAN y las WAN que se enumeran a continuación:

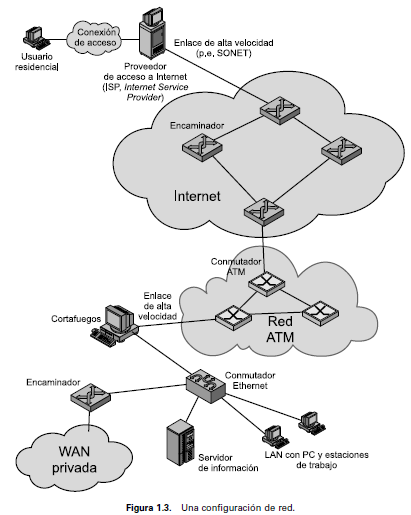
1. **La cobertura de una LAN es pequeña**, generalmente un edificio o, a lo sumo, un conjunto de edificios próximos. Como se verá más adelante, esta diferencia en cuanto a la cobertura geográfica condicionará la solución técnica finalmente adoptada.

2. Es habitual que la **LAN sea propiedad de la misma entidad propietaria de los dispositivos conectados a la red**. En WAN, esto no es tan habitual o, al menos, una fracción significativa de recursos de la red son ajenos. Esto tiene dos implicaciones. La primera es que se debe cuidar mucho la elección de la LAN, ya que, evidentemente, lleva acarreada una inversión sustancial de capital (comparada con los gastos de conexión o alquiler de líneas en redes de área amplia) tanto en la adquisición como en el mantenimiento. Segunda, la responsabilidad de la gestión de la red local recae solamente en el usuario.

3. Por lo general, **las velocidades de transmisión internas en una LAN son mucho mayores que en una WAN. Para las LAN hay muy diversas configuraciones**. De entre ellas, las más habituales son las **LAN conmutadas** y las **LAN inalámbricas**. **Dentro de las conmutadas, las más populares son las LAN Ethernet**, constituidas por un único conmutador, o, alternativamente, implementadas mediante un conjunto de conmutadores interconectados entre sí. Otro ejemplo muy relevante son las **LAN ATM**, caracterizadas por utilizar tecnología de red ATM en un entorno local. Por último, son también destacables las **LAN con canal de fibra (Fiber Channel)**. En **las LAN inalámbricas (Wireless)** se utilizan diversos tipos de tecnologías de transmisión y distintos tipos de configuraciones, permitiendo una mayor movilidad, facilidad en la instalación y configuración y fiabilidad.

Ejemplo de Configuración:

Esquina superior izquierda, se puede encontrar un usuario residencial conectado a Internet a través de un proveedor de acceso a internet (ISP) mediante algún tipo de conexión de abonado. Por ejemplo, una red pública de telefonía utilizando un módem, una línea digital de abonado DSL, tecnología que proporciona un enlace de alta velocidad a través de líneas de telefonía mediante el uso de un modem especial, o un acceso a TV por cable, la cual requerirá un cable modem.



**Arquitectura de Protocolos**

Una arquitectura de protocolos es una estructura en capas de elementos hardware y software que facilita el intercambio de datos entre sistemas y posibilita aplicaciones distribuidas.

En los sistemas de comunicación, en cada una de las capas de la arquitectura de protocolos se implementa uno o más protocolos comunes. Cada protocolo proporciona un conjunto de reglas para el intercambio de datos entre sistemas.

La arquitectura de protocolos más utilizada es TCP/IP, constituida por las siguientes capas: física, acceso a la red, internet, transporte y aplicación.

En una arquitectura de protocolos, los distintos módulos se disponen formando una pila vertical. Cada capa de la pila realiza el subconjunto de tareas relacionadas entre sí que son necesarias para comunicar con el otro sistema. Las funciones más básicas se dejan a la capa inferior, olvidándose en la capa actual de los detalles de estas funciones. Además, cada capa proporciona un conjunto de servicios a la capa superior. Las capas deberían definirse de forma que los cambios en una capa no provoquen cambios en las otras.

Para que haya comunicación se necesitan dos entidades, por lo que debe existir el mismo conjunto de funciones en capas en los dos sistemas. La comunicación se consigue haciendo que las capas correspondientes, o **pares**, intercambien información. Las capas pares se comunican intercambiando bloques de datos que verifican una serie de reglas o convenciones denominadas **protocolo.**

Un **protocolo** es un conjunto de reglas que gobiernan la comunicación de datos. Un protocolo define qué se comunica, cómo se comunica y cuándo se comunica.

Los **elementos clave de un protocolo** son su sintaxis, su semántica y su temporización.

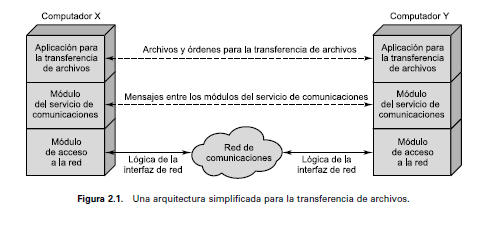
La sintaxis: establece cuestiones relacionadas con el formato de los bloques de datos.

La semántica: incluye información de control para la coordinación y la gestión de errores.

La temporización: considera aspectos relativos a la sintonización de velocidades y secuenciación.

**Arquitectura de Protocolos Simple:**

En vez de disponer de un solo modulo que realice todas las tareas involucradas en la comunicación, se considera una estructura consistente en un conjunto de módulos que realizarán todas las funciones. Esta estructura es denominada arquitectura de protocolos.

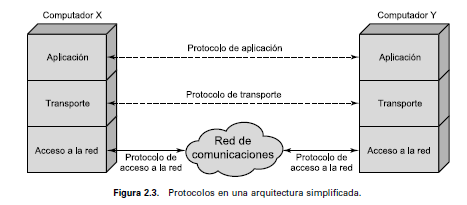


**Arquitectura simplificada de protocolos**

En términos muy generales, las comunicaciones involucran a tres agentes: aplicaciones, computadoras y redes.

Las aplicaciones se ejecutan en computadoras que, generalmente, permiten múltiples aplicaciones simultáneas. Las computadoras se conectan a redes y los datos a intercambiar se transfieren por la red de un computador a otro. Por lo tanto, la transferencia de datos desde una aplicación a otra implica, en primer lugar, la obtención de los mismos y, posteriormente, hacerlos llegar a la aplicación destino en el computador remoto.

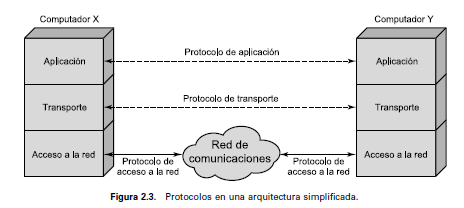
De esta forma, se pueden estructurar las tareas de las comunicaciones en tres capas relativamente independientes: La capa de acceso a la red, la capa de transporte y la capa de aplicación.



La **capa de acceso a la red** esta relacionada con el intercambio de datos entre el computador y la red a la que está conectado. El computador emisor debe proporcionar a la red la dirección del destino, de tal forma que la red pueda encaminar los datos al destino apropiado.

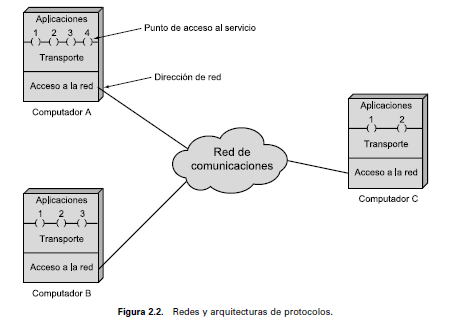
Independientemente de la naturaleza de las aplicaciones que están intercambiando datos, es un requisito que los datos se intercambien de manera fiable. Esto es, estar seguros de que los datos llegan a la aplicación de destino y llegan en el mismo orden en que fueron enviados. Los mecanismos que proporcionan dicha fiabilidad se encuentran en una capa común denominada **capa de transporte**.

Por último, la **capa de aplicación** contiene la lógica necesaria para admitir varias aplicaciones de usuarios. Para cada tipo distinto de aplicación, se necesita un modulo independiente y con características bien diferenciadas.



En esta figura se muestra cómo se comunican, mediante un protocolo, los módulos en el mismo nivel de computadoras diferentes.

Arquitectura de protocolos y redes.



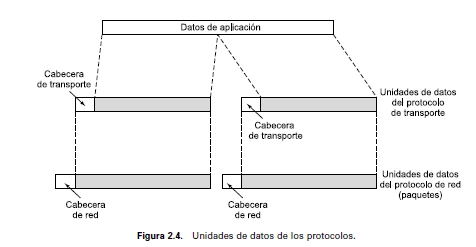
En esta figura se muestran tres computadoras conectadas a la red. Cada computador contiene software en las capas de acceso a la red, de transporte y aplicación para una o mas aplicaciones. Cada computador en la red debe tener una dirección de red. A su vez, cada aplicación en el computador debe tener una dirección que sea única dentro del propio computador.

**Unidad de Datos del Protocolo (PDU)**

Para controlar la comunicación, se debe transmitir información de control junto a los datos del usuario.

Si la aplicación emisora genera un bloque datos y se lo pasa a la capa de transporte, esta puede fraccionar el bloque en unidades más pequeñas para hacerlas mas manejables. A cada una de estas pequeñas unidades, la capa de transporte le añadirá una cabecera, que contendrá información de control de acuerdo con el protocolo.

La unión de los datos generados por la capa superior, junto con la información de control de la capa actual, se denomina unidad de datos del protocolo (PDU), en este caso PDU de transporte.



**La Unidad de Datos del Protocolo** es la porción de datos de un paquete en el nivel N-1 que transporta el paquete completo del nivel N. La unión de los datos generados por la capa superior junto con la información de control de la capa actual se denomina unidad de dato del protocolo o PDU.

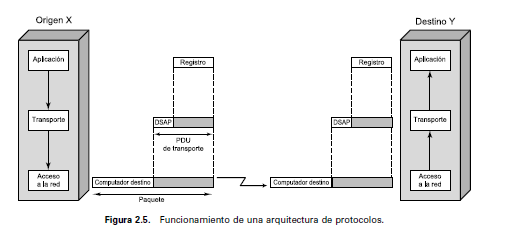
La cabecera en cada PDU de transporte contiene información de control que será utilizada por el protocolo de transporte en el computador par.

La información que se debe incluir en la cabecera puede ser, por ejemplo:

**SAP destino:** cuando la capa de transporte destino reciba la PDU de transporte, deberá saber a quien van destinados los datos.

**Numero de secuencia:** ya que el protocolo de transporte esta enviando una secuencia de PDU, estas se numerarán secuencialmente para que, si llegan desordenadas, la entidad de transporte destino sea capaz de ordenarlas

**Código de detección de errores**: la entidad de transporte emisora debe incluir un código obtenido en función del resto de la PDU. El protocolo de transporte receptor realiza el mismo cálculo y compara los resultados con el código recibido. Si hay discrepancia se concluirá que ha habido un error en la transmisión y, en ese caso, el receptor podrá descartar la PDU y adoptar las acciones oportunas para su corrección.

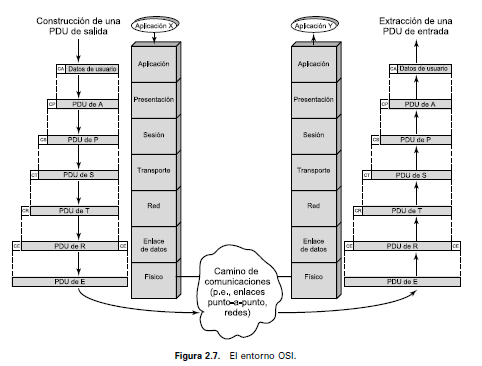


**Arquitecturas de Protocolos normalizadas: OSI Y TCP / IP:**

**Modelo OSI**

El modelo Interconexión de Sistemas Abiertos (OSI). Un **sistema abierto** es un modelo que permite que dos sistemas diferentes se puedan comunicar independientemente de sus arquitecturas. Sin que sea necesario cambiar la lógica del hardware o el software subyacente.

El modelo OSI no es un protocolo, es un modelo para comprender y diseñar una arquitectura de red flexible, robusta e interoperable, este modelo es una arquitectura por niveles para el diseño de sistemas de red que permiten la comunicación entre todos los tipos de computadoras. Está compuesto por 7 niveles relacionados, cada uno de los cuales define un segmento del proceso necesario para mover la información a través de una red.



**LAS CAPAS DE OSI:**

El modelo OSI está compuesto por 7 niveles ordenados, cada nivel define un conjunto de funciones distintas a los otros niveles y en cada capa se utilizan protocolos para comunicarse, cada nivel llama a los servicios del nivel que está por debajo, entre maquinas, el nivel x de un maquina se comunica con el nivel x de la otra. La comunicación se gobierna por protocolos.

El paso de los datos y la información a atreves de los distintos niveles es posible porque hay una interfaz entre cada par de niveles adyacentes. Cada interfaz define qué información y servicios debe proporcionar un nivel al superior.

|  |  |
| --- | --- |
| Nivel | Nombre de los datos |
| 7-aplicación | Datos |
| 6-presentacion. | Datos. |
| 5-sesion | Datos. |
| Transporte | segmento |
| Red | Paquete o datagrama |
| Enlace de datos | Trama |
| Físico | bits |

**Organización de los niveles:**

Los niveles superiores casi siempre se implementan casi siempre en software, los niveles inferiores son combinación de software y hardware y el físico es puro hardware.

La **transmisión de datos**, es un proceso de 2 partes:

**Encapsular:** empieza en el nivel 7 y se mueve de nivel a nivel en orden secuencial descendiente. En cada nivel (menos 1) se añade una **cabecera** a la unidad de datos. En el nivel 2 también se añade una **cola.**  Cuando las unidades de datos pasan a través del nivel físico se transforman en señales electromagnéticas y se transportan por el enlace físico.

**Desencapsular:** al alcanzar su destino, la señal pasa al nivel 1 y se transforma de nuevo en bits, luego las unidades de datos ascienden a través de los niveles OSI. Cuando cada bloque de datos alcanza el nivel superior siguiente, las cabeceras y las colas asociadas a dicho nivel se eliminan y efectúan las acciones apropiadas de ese nivel. Cuando se alcanza el nivel 7 el mensaje esta otra vez en el formato para la aplicación y se puede poner a disposición del receptor.

**CAPA FÍSICA**

Es responsable del movimiento de bits individuales desde un nodo al siguiente, define las reglas que rigen en la transmisión de los bits y se encarga de la interfaz física entre los dispositivos. La capa física tiene cuatro características importantes:

Mecánicas: relacionadas con las propiedades físicas de la interfaz con el medio de transmisión.

Normalmente, dentro de estas características se incluye la especificación del conector

que transmite las señales a través de conductores. A estos últimos se les denominan circuitos.

Eléctricas: especifican cómo se representan los bits (por ejemplo, en términos de niveles de

tensión), así como su velocidad de transmisión.

Funcionales: especifican las funciones que realiza cada uno de los circuitos de la interfaz

física entre el sistema y el medio de transmisión.

De procedimiento: especifican la secuencia de eventos que se llevan a cabo en el intercambio

del flujo de bits a través del medio físico.

**Funciones:**

* **Características físicas de la interfaz y el medio:** define las características de la interfaz entre los dispositivos y el medio de transmisión. Define el tipo de medio de transmisión.
* **Representación de los bits:** codifica los bits en señales.
* **Tasa de datos:** define el número de bits enviados por segundo.
* **Sincronización de los bits:** el emisor y el receptor deben estar sincronizados a nivel de bit. Los relojes del emisor y receptor deben estar sincronizados.
* **Configuración de la línea:** define la conexión del dispositivo al medio ya sea punto a punto o multipunto.
* **Topología física:** define como están conectados los dispositivos para formar una red.
* **Modo de transmisión:** define la dirección de la transmisión entre 2 dispositivos: simplex, semi dúplex o full-dúplex.

**ENLACE DE DATOS**

Transforma el nivel físico en un enlace fiable, hace que el nivel físico aparezca ante el nivel superior como un medio libre de errores. Es responsable del movimiento de tramas desde un nodo al siguiente. Activa, mantiene y desactiva el enlace. Funciones:

* **Tramado:** divide los paquetes recibidos del nivel de red en unidades de datos denominadas tramas.
* **Direccionamiento físico:** añade una cabecera a la trama para definir la dirección física del emisor y receptor de la trama.
* **Control de flujo:** si la velocidad a la que el receptor recibe los datos es menor que la velocidad de transmisión del emisor, el nivel de enlace de datos impone un mecanismo de control de flujo para prevenir el desbordamiento del receptor.
* **Control de errores:** incluir mecanismos para detectar y retransmitir las tramas defectuosas o perdidas, uno para prevenir duplicación de tramas. Esto se consigue añadiendo la cola al final de la trama.
* **Control de acceso:** cuando se conectan 2 o más dispositivos al mismo enlace, los protocolos de nivel de enlace deben determinar que dispositivo tiene el control del enlace.

**NIVEL DE RED**

Es responsable de la entrega de un paquete desde un host origen al destino, asegura que cada paquete va del origen al destino sean los adecuados. Libera a las capas superiores acerca de la tecnología de conmutación. Se encarga del direccionamiento y solicitudes de servicios. Funciones:

* **Direccionamiento lógico:** añade una cabecera al paquete que viene del nivel superior que incluye las direcciones lógicas del emisor y receptor.
* **Encaminamiento:** cuando un conjunto de redes se conecta para formar una más grande, los dispositivos de conexión encaminan los paquetes hasta su destino final, el nivel de red proporciona estos mecanismos.

Si se tiene un enlace punto a punto no es necesario esta capa, sus funciones la realiza la de enlace de datos. En cambio, se tiene una red, las 3 capas inferiores de OSI están relacionadas con la conexión y comunicación a la red, los paquetes creados pasan por uno o más nodos de la red que los retransmiten, por ende, cada nodo debe implementar los primeros 3 niveles.

**TRANSPORTE**

Es responsable de la entrega de origen a destinó de todo el mensaje, proporciona un mecanismo para intercambiar datos entre sistemas finales, asegura que todo el mensaje llega intacto y en orden, sin errores, pérdidas o duplicados, supervisando el control de errores y de flujo a nivel origen y destino. Funciones:

* **Direccionamiento en punto de servicio:** la cabecera del nivel de transporte debe incluir un tipo de dirección denominado de punto de servicio, para enviar el mensaje entero al proceso adecuado dentro de esa computadora.
* **Segmentación y reensamblado:** un mensaje se divide en segmentos transmisibles, cada cual contiene un numero de secuencia, estos números permiten al nivel de transporte reensamblar el mensaje a su llegada a destino e identificar y reemplazar paquetes que se han perdido.
* **Control de conexión:** el nivel de transporte puede estar orientado a conexión o no. Si es no orientado a conexión trata cada segmento como un paquete independiente y lo pasa al nivel de transporte de la maquina destino. Si es orientado a conexión, establece una conexión con el nivel de transporte del destino antes de enviar algún paquete, a la conexión se corta después de enviar todos los paquetes.
* **Control de flujo:** es responsable del control de flujo de extremo a extremo.
* **Control de errores:** lleva a cabo el control de errores de extremo a extremo para corregir errores mediante retransmisión.

**SESIÓN**

Es el controlador de dialogo de la red. Establece, mantiene y sincroniza la interacción entre sistemas de comunicación. Funciones:

* **Control de dialogo:** permite que dos sistemas establezcan un dialogo. Permite que la comunicación entre 2 procesos tenga lugar en semi o full dúplex.
* **Sincronización:** permite que un proceso pueda añadir puntos de sincronización en un flujo de datos.

**PRESENTACIÓN**

Es responsable de la traducción, compresión y cifrado. Define el formato de los datos que se van a intercambiar y ofrece los servicios de transformación de datos, define la sintaxis y permite modificar la representación de los datos. Funciones:

* **Traducción:** es responsable de la interoperabilidad entre los distintos métodos de codificación. El nivel de presentación en el emisor cambia la información del formato dependiente del emisor a un formato común. El nivel de presentación en la maquina receptora cambia el formato común en el formato específico del receptor.
* **Cifrado:** transforma la información original a otro formato y envía el mensaje por la red. El descifrado ejecuta lo inverso al recibir el mensaje.
* **Compresión:** reduce el número de bits a transmitir.
* **APLICACIÓN**

Es responsable de ofrecer los servicios y programas de aplicación a los usuarios, le permite acceder al entorno OSI y a la red. Ofrece programas como correo, transferencia de archivos, etc.

**Arquitectura de protocolo TCP/IP**



La arquitectura de protocolos TCP/IP es resultado de la investigación y desarrollo en la red de conmutación de paquetes ARPANET, financiada por la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA, *Defense Ad*v*anced Research Projects* *Agency*), y se denomina familia de protocolos TCP/IP que consiste en una extensa colección de protocolos que se han especificado como estándares de Internet.

**Capa física**

Define la interfaz física entre el dispositivo de transmisión de datos y el medio de transmisión o red. Esta capa se encarga de la especificación de las características del medio de transmisión, las señales, la velocidad de datos, etc.

**Capa de Enlace:**

Es responsable del intercambio de datos entre el sistema final y la red a la cual está conectado. El emisor debe proporcionar a la red la dirección del destino, de tal manera que ésta pueda encaminar los datos hasta el destino apropiado. El emisor puede requerir ciertos servicios que pueden ser proporcionados por el nivel de red. El software en particular que se use en esta capa dependerá del tipo de red que se disponga. El software de comunicaciones situado por encima de la capa de acceso a la red no tendrá que ocuparse de los detalles específicos de la red a utilizar. El software de las capas superiores debería, por tanto, funcionar correctamente con independencia de la red a la que el computador esté conectado.

**Capa Red/Internet(IP):**

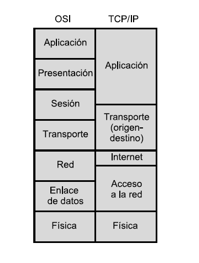
Si dos dispositivos estén conectados a redes diferentes, se necesitarán procedimientos para que los datos atraviesen las distintas redes interconectadas. Ésta es la función de la **capa red**. El protocolo internet (IP, *Internet Protocol*) se utiliza en esta capa para ofrecer el servicio de encaminamiento a través de varias redes. Este protocolo se implementa tanto en los sistemas finales como en los encaminadores intermedios. Un **encaminador** es un procesador que conecta dos redes y cuya función principal es retransmitir datos desde una red a otra siguiendo la ruta adecuada para alcanzar al destino.

**Capa extremo-a-extremo o de transporte.**

Esta capa se encarga que los datos se intercambien en forma fiable y del ordenamiento en la entrega, es decir asegurar que llegan a la aplicación destino y en el orden que se enviaron. Esta capa el protocolo para el control de la transmisión, TCP (*Transmission Control Protocol*), es el más utilizado para proporcionar esta funcionalidad.

**Capa de aplicación:**

Contiene toda la lógica necesaria para posibilitar las distintas aplicaciones del usuario.

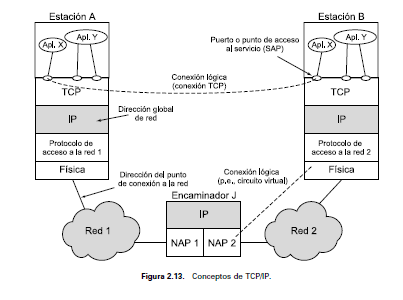


**TCP Y UDP**

Cada PDU de TCP, denominada **segmento TCP**, contiene en la cabecera la identificación de los puertos origen y destino, los cuales corresponden con los puntos de acceso al servicio (SAP) de la arquitectura OSI. Los valores de los puertos identifican a los respectivos usuarios (aplicaciones) de las dos entidades TCP. Durante la conexión, cada entidad seguirá la pista de los segmentos TCP que vengan y vayan hacia la otra entidad, para así regular el flujo de segmentos y recuperar aquellos que se pierdan o dañen.

Además del protocolo TCP, la arquitectura TCP/IP usa otro protocolo de transporte: el protocolo de datagramas de usuario, UDP (*User Datagram Protocol*). UDP no garantiza la entrega, la conservación del orden secuencial, ni la protección frente a duplicados. UDP posibilita el envío de mensajes entre aplicaciones con la complejidad mínima.

FUNCIONAMIENTO DE TCP E IP



El conjunto total de recursos para la comunicación puede estar formado por varias redes, a dichas redes constituyentes se les denomina **subredes**. Para conectar un computador a una subred se utiliza algún tipo de protocolo de acceso, por ejemplo, Ethernet. Este protocolo permite al computador enviar datos a través de la subred a otro computador o, en caso de que el destino final esté en otra subred, a un dispositivo de encaminamiento que los retransmitirá. **IP** se implementa en todos los sistemas finales y dispositivos de encaminamiento. Actúa como un porteador que transportara bloques de datos desde un computador hasta otro, a través de uno o varios dispositivos de encaminamiento. **TCP** se implementa solamente en los sistemas finales, donde supervisa los bloques de datos para asegurar que todos se entregan de forma fiable a la aplicación apropiada.

Cada entidad en el sistema global debe tener una única dirección. En realidad, se necesitan dos niveles de direccionamiento. Cada computador en una subred dada debe tener una dirección de internet única que permita enviar los datos al computador adecuado. Además, cada proceso que se ejecute dentro de un computador dado debe tener, a su vez, una dirección que sea única dentro del mismo. Esto permite al protocolo extremo-a-extremo (TCP) entregar los datos al proceso adecuado. Estas últimas direcciones se denominan puertos.

Supóngase que un proceso, asociado al puerto 1 en el computador A, desea enviar un mensaje a otro proceso, asociado al puerto 2 del computador B. El proceso en A pasa el mensaje a TCP con la instrucción de enviarlo al puerto 2 del computador B. TCP pasa el mensaje a IP con la instrucción de enviarlo al computador B. Luego, IP pasa el mensaje a la capa de acceso a la red con la orden de enviarlo al dispositivo de encaminamiento J (el primer salto en el camino hacia B).

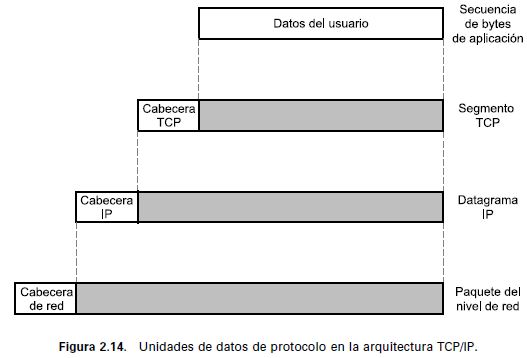
Para controlar esta operación se debe transmitir información de control junto con los datos de usuario, como en la Figura 2.14. El proceso emisor genera un bloque de datos y lo pasa a TCP. TCP puede que divida este bloque en fragmentos más pequeños para hacerlos más manejables. A cada uno de estos fragmentos le añade información de control, denominada cabecera TCP, formando un **segmento TCP**. La información de control la utilizará la entidad par TCP en el computador B. Entre otros, en la cabecera se incluyen los siguientes campos:

**Puerto destino:** cuando la entidad TCP en B recibe el segmento, debe conocer a quién se le deben entregar los datos.

**Número de secuencia:** TCP numera secuencialmente los segmentos que envía a un puerto destino dado para que, si llegan desordenados, la entidad TCP en B pueda reordenarlos.

**Suma de comprobación:** la entidad emisora TCP incluye un código calculado en función del resto del segmento. La entidad receptora TCP realiza el mismo cálculo y compara el resultado con el código recibido.

A continuación, TCP pasa cada segmento a IP con instrucciones para que los transmita a B. Estos segmentos se transmitirán a través de una o varias subredes y serán retransmitidos en uno o más dispositivos de encaminamiento intermedios. Esta operación también requiere el uso de información de control. Así, IP añade una cabecera de información de control a cada segmento para formar lo que se denomina un **datagrama IP**. En la cabecera IP, además de otros campos, se incluirá la dirección del computador destino (en nuestro ejemplo B).



Finalmente, cada datagrama IP se pasa a la capa de acceso a la red para que se envíe a través de la primera subred. La capa de acceso a la red añade su propia cabecera, creando un paquete, o trama. El paquete se transmite a través de la subred al dispositivo de encaminamiento J. La cabecera del paquete contiene la información que la subred necesita para transferir los datos. La cabecera puede contener, entre otros, los siguientes campos:

* **Dirección de la subred destino**: la subred debe conocer a qué dispositivo se debe entregar el paquete.
* **Funciones solicitadas**: el protocolo de acceso a la red puede solicitar la utilización de ciertas funciones ofrecidas por la subred.

En el dispositivo de encaminamiento J, la cabecera del paquete se elimina y, posteriormente, se examina la cabecera IP. El módulo IP del dispositivo de encaminamiento direcciona el paquete a través de la subred 2 hacia B basándose en la dirección destino que contenga la cabecera IP. Para hacer esto, se le añade al datagrama una cabecera de acceso a la red.

Cuando se reciben los datos en B, ocurre el proceso inverso. En cada capa se elimina la cabecera correspondiente y el resto se pasa a la capa inmediatamente superior, hasta que los datos de usuario originales alcancen al proceso destino.

Como nota final, recuérdese que el nombre genérico del bloque de datos intercambiado en cualquier nivel se denomina **unidad de datos del protocolo** (PDU, *Protocol Data Unit*). El segmento TCP es la PDU del protocolo TCP.

APLICACIONES TCP/IP

● SMTP (Simple Mail Transfer Protocol)

● FTP (File Transfer Protocol)

● SSH (Secure Shell)

● SNMP (Simple Network Management Protocol)

● HTTP (HyperText Transfer Protocol)

● IMAP (Internet Message Access Protocol)

● POP3 (Post Office Protocol)

● SNMP (Simple Network Management Protocol)

● DNS (Domain Name System)

● RIP (Routing Information Protocol)

● OSPF (Open Shortest Path First)



**Estándares**

● Requeridos para la interoperatibilidad entre equipamientos

de diferentes fabricantes

● Ventajas

● Asegura mercado de equipamiento y software

● Permite que los productos de diferentes fabricantes se

puedan comunicar

● Desventajas

● Congela la tecnología

● Pueden existir múltiples estándares de una misma cosa

**Organizaciones de Estándares**

● Sociedad de Internet (Internet Society)

● ISO

● ITU (formalmente CCITT)

● Foro ATM

**PROTOCOLOS DE CONTROL DEL ENLACE DE DATOS O DE LA CAPA DE DATOS EN REDES WAN**

**CUESTIONES BÁSICAS**

Las técnicas de sincronización y gestión de la interfaz son insuficientes para dar respuesta a errores en una transmisión y a la necesidad de regular la velocidad de datos por parte del receptor. Es necesario, por tanto, incluir en cada dispositivo de comunicación una capa de control que regule el flujo de información, además de detectar y controlar los errores. Esta capa se denomina **protocolo de control del enlace de datos**.

**El control de flujo** permite al receptor regular el flujo de los datos enviados por el emisor, de manera que la memoria temporal del primero no se desborde.

En un protocolo de control del enlace de datos, **el control de errores** se lleva a cabo mediante la retransmisión de las tramas dañadas que no hayan sido confirmadas o de aquellas para las que el otro extremo solicite su retransmisión.

**Control del enlace de datos o protocolo de control del enlace de datos**

Capa lógica adicional por encima de la interfaz física, la necesidad del control del enlace se deriva a partir de los siguientes requisitos y objetivos para la comunicación efectiva entre 2 estaciones:

**Sincronización de trama:** los datos se envían en bloques denominados tramas, cuyo principio y fin deben ser identificables.

**Control de flujo:** la estación emisora no debe enviar tramas a una velocidad superior a la que la estación receptora pueda absorberlas.

**Control de errores:** se debe corregir cualquier error en los bits provocado por el sistema de transmisión.

**Direccionamiento:** en una línea multipunto se debe identificar a las dos estaciones involucradas en una transmisión.

**Datos y control sobre el mismo enlace:** no se desea tener un canal de comunicaciones independiente para la información de control. El receptor deberá ser capaz de diferenciar entre la información de control y los datos.

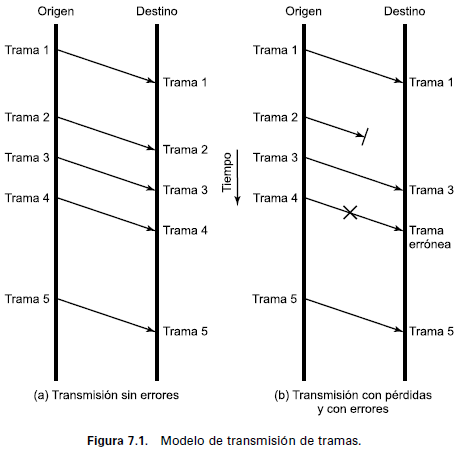
**Gestión del enlace:** el inicio, mantenimiento y finalización de un intercambio de datos precisa un alto grado de coordinación y cooperación entre las estaciones.

**CONTROL DE FLUJO**

Es una técnica para asegurar que un emisor no sobrecargue a un receptor con una excesiva cantidad de datos. Generalmente, el receptor reserva memoria temporal para la transferencia. Cuando se reciben los datos, el receptor debe realizar procesamiento antes de pasar los datos al software de las capas superiores. En ausencia de procedimientos para el control de flujo, la memoria temporal del receptor se podría llenar y desbordarse mientras éste se encuentra procesando datos previos.

Existen 2 consideraciones a la hora del control de flujo:

* **Tiempo de transmisión:** tiempo que una estación tarda emitir todos los bits sobre el medio.
* **Tiempo de propagación:** tiempo empleado por un bit en atravesar el medio de transmisión del origen al destino.



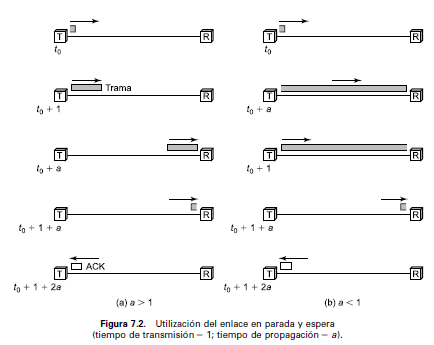
Este modelo consiste en un diagrama donde el tiempo se representa sobre la vertical. Cada flecha representa una única trama que transita por el enlace de datos establecido entre dos estaciones. Los datos se envían en base a una secuencia de tramas, en la que cada una de ellas contiene un campo de datos más información de control.

**Control de Flujo mediante parada y espera**

Procedimiento para controlar el flujo denominado control de flujo mediante parada y espera.

Una entidad origen transmite una trama. Tras la recepción, la entidad destino indica su deseo de aceptar otra trama mediante el envío de una confirmación de la trama que acaba de recibir, llamado ACK (acknowledgement). El origen debe esperar a recibir el ACK antes de proceder a la transmisión de la trama siguiente. El destino puede parar el flujo de los datos sin más que retener las confirmaciones. Este procedimiento funciona bien cuando el mensaje se envía usando un número reducido de tramas de gran tamaño. El origen puede dividir la información en bloques pequeños, transmitiendo los datos en varias tramas. Razones:

* El tamaño de la memoria temporal del receptor puede ser limitado.
* Los errores se detectarán antes y se necesitará retransmitir una cantidad de datos menor.
* En un medio compartido previene que una estación ocupe el medio durante un periodo de tiempo largo.



Si se usan varias tramas para un solo mensaje, el método puede ser inadecuado ya que sólo puede haber una trama en tránsito en un instante de tiempo dado.

Considerando el tiempo de transmisión igual a 1 y representando el tiempo de propagación como **a**, se tiene que:

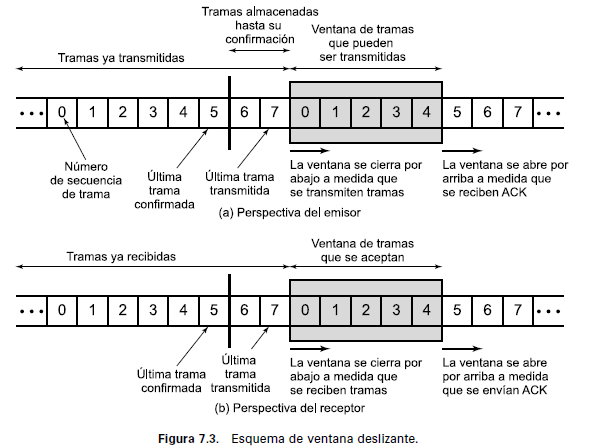
* Cuando *a* <1, el tiempo de propagación es menor que el de transmisión. En este caso, la trama es lo suficientemente larga para que los primeros bits de la misma lleguen al destino antes de que el origen haya concluido la transmisión de dicha trama.
* Cuando a >1, el tiempo de propagación es mayor que el de transmisión. En este caso, el emisor completa la transmisión de toda la trama antes que el primer bit de la misma llegue al receptor. Para velocidades de transmisión o distancias grandes, es adecuado valores grandes de a.

**Control de flujo mediante ventana deslizante:**

Se permite que transiten varias tramas al mismo tiempo sobre el enlace, la eficiencia mejora significativamente. Este procedimiento para dos estaciones, el receptor reserva memoria temporal para almacenar W tramas. El receptor puede aceptar W tramas, por lo que el emisor puede enviar este mismo número de tramas sin tener que esperar confirmación (ACK). Para saber qué tramas se han confirmado, cada una de ellas se etiqueta con un número de secuencia. El receptor confirma una trama mediante el envío de una confirmación (ACK) que incluye el número de secuencia de la siguiente trama que se espera recibir. Esta confirmación informa también que el receptor está preparado para recibir las W tramas siguientes, comenzando por la de número especificado. Este esquema se puede utilizar también para confirmar varias tramas simultáneamente. El emisor mantiene una lista con los números de secuencia que se le permite transmitir y el receptor mantiene una lista con los números de secuencia que está esperando recibir. Cada una de estas listas se puede considerar como una ventana de tramas, de ahí el nombre.

La numeración de las tramas ocupa un campo en las mismas, dicha numeración tiene un tamaño limitado. Para un campo de *k* bits:

* El rango de números de secuencia irá desde 0 hasta 2^ (k-1), numerándose las tramas módulo 2*k*.
* El tamaño máximo de la ventana es 2*k*.1.
* Ej: un campo de 3 bits, los números de secuencia pueden variar entre 0 y 7, las tramas se numerarán módulo 8; después del número 7 vendrá el 0.



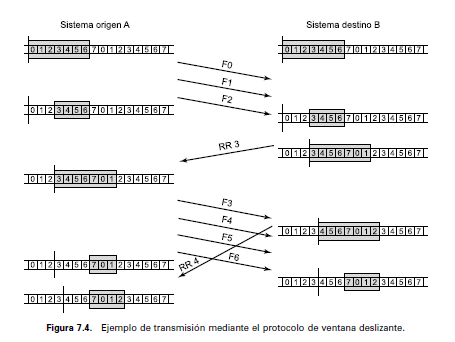
Representación del ejemplo anterior, campo de 3 bits, tramas numeradas en secuencia de 0 a 7 usándolos cíclicamente.

Desde el emisor el rectángulo sombreado indica las tramas que se pueden transmitir;

Cada vez que se envíe una trama, la ventana sombreada se cerrará, reduciendo su tamaño; cada vez que se reciba una confirmación, la ventana sombreada se abrirá.

Las tramas comprendidas entre la barra vertical y la ventana sombreada han sido ya enviadas, pero aún no han sido confirmadas. El emisor debe almacenar estas tramas en la memoria temporal por si hubiera que retransmitirlas.

Dada una longitud para los números de secuencia, el tamaño de la ventana real no necesita ser el máximo posible.



En este esquema se muestra un esquema en el que se supone un campo de 3 bits para los números de secuencia y un tamaño máximo para la ventana igual a siete tramas

Inicialmente, A y B tienen las ventanas indicando que A puede transmitir siete tramas, comenzando con la 0.

Tras transmitir tres tramas sin confirmación, A habrá cerrado su ventana hasta tener un tamaño de cuatro tramas, manteniendo una copia de las tres tramas transmitidas. La ventana indica que A puede transmitir cuatro tramas, comenzando por la numero 3.

El receptor luego de recibir tramas transmite una trama **RR + número de secuencia de trama** (receptor preparado, *Recei*v*e Ready*), ej. RR 3, lo que significa «he recibido todas las tramas hasta la número 2 y estoy preparado para recibir la número 3; de hecho, estoy preparado para recibir el número de tramas igual al tamaño de la ventana (7) , comenzando por la número 3».

Tras ello, al emisor se le permite transmitir un número de tramas (7) de hasta el tamaño de la ventana, comenzando por la trama 3; también puede descartar las tramas almacenadas en la memoria temporal que acaban de ser confirmadas.

Se permite que el receptor pueda interrumpir la transmisión de tramas mediante el envío de un mensaje **RNR (receptor no preparado, *Recei*v*e* *Not Ready*),** con el que se confirman las tramas anteriores, pero se prohíbe la transmisión de tramasadicionales. Así, RNR 5 significa: «he recibido todas las tramas hasta la número 4 pero no aceptomás». En algún momento posterior, la estación deberá transmitir una confirmación normal que«reabra» la ventana.

Si dos estaciones intercambian datos, cada una de ellas deberá mantener dos ventanas, una para transmitir y otra para recibir, y cada extremo deberá enviar hacia el otro tanto datos como confirmaciones. Se utiliza un procedimiento denominado **incorporación de confirmación** (***piggybacking* en inglés**)

**Cada trama de datos** incluye un campo en el que se indica el número de secuencia de dicha trama más un campo que indica el número de secuencia que se confirma.

Si una estación tiene para enviar una confirmación además de datos, lo hará conjuntamente utilizando una sola trama, ahorrando así capacidad del canal.

Si una estación tiene que enviar una confirmación, pero no tiene datos, se enviará una trama de confirmación, como por ejemplo una RR o una RNR.

Si la estación tiene datos para enviar, pero nada que confirmar, deberá repetir el último número de secuencia de confirmación enviado con anterioridad. Esto se debe a que en la trama de datos se prevé un campo para el número de confirmación y, por tanto, habrá que especificar algún valor en el mismo. Cuando una estación reciba una confirmación repetida, simplemente la ignorará.

El control de flujo mediante ventana deslizante es potencialmente mucho más eficiente que el control de flujo mediante un procedimiento de parada y espera. La razón se debe a que, con un control de flujo mediante ventana deslizante, el enlace de transmisión se trata como si fuese una tubería que se puede rellenar con tramas en tránsito.

**CONTROL DE ERRORES:**

Son los mecanismos necesarios para la detección y la corrección de errores que aparecen en una transmisión de tramas. Hay dos tipos de errores potenciales:

* **Tramas pérdidas:** se produce cuando una trama enviada no llega al otro extremo.
* **Tramas dañadas:** ocurre cuando una trama se recibe con algunos bits erróneos.

Las técnicas más usuales para el control de errores se basan en algunas o todas las siguientes aproximaciones:

* **Detección de errores.**
* **Confirmaciones positivas:** el destino devuelve una confirmación positiva por cada trama recibida con éxito, libre de errores.
* **Retransmisión tras la expiración de un temporizador:** la fuente retransmite las tramas que no se han confirmado tras un periodo de tiempo predeterminado.
* **Confirmación negativa y retransmisión:** el destino devuelve una confirmación negativa para aquellas tramas en las que se detecta la ocurrencia de errores. El origen retransmitirá de nuevo dichas tramas.

Estos mecanismos se denominan **solicitud de repetición automática** (ARQ, *Automatic Repeat reQuest*); el objetivo de un esquema ARQ es convertir un enlace de datos no fiable en fiable. Hay tres variantes ARQ estandarizadas:

* ARQ con parada y espera.
* ARQ con vuelta atrás N.
* ARQ con rechazo selectivo.

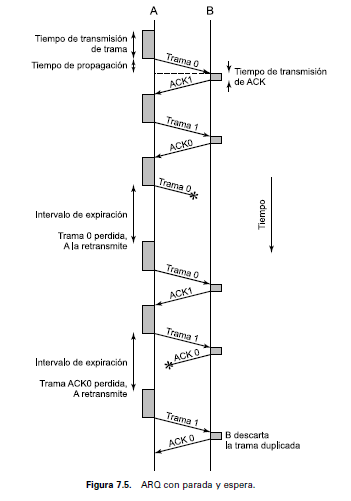
Todos estos procedimientos se basan en el empleo de las técnicas de control de flujo.

**ARQ con parada y espera.**

El origen transmite una única trama y debe esperar la confirmación (ACK). No se podrá enviar ninguna otra trama hasta que la respuesta de la estación destino llegue al emisor.

Pueden ocurrir dos tipos de error:

* **La trama que llega al receptor puede estar dañada**. El receptor descartará la trama, el emisor usa un temporizador. Tras el envío de una trama, el emisor espera la confirmación; si no se recibe ninguna confirmación antes de que el temporizador expire, se procederá a reenviar la misma trama. El emisor conserva una copia de la trama transmitida hasta que se reciba la confirmación correspondiente.
* **El ACK se pierde o daña y el emisor no puede reconocerlo.** La trama ACK se deteriora en el camino, no es identificable por el emisor, por lo que se producirá una expiración del temporizador y se reenviará la misma trama de datos. Esta trama duplicada llega y se acepta por el receptor; que ha aceptado dos copias de la misma trama como si fueran distintas. Para evitar este problema, las tramas se pueden etiquetar de forma alternada con 0 o 1, siendo las confirmaciones positivas de la forma ACK0 y ACK1. Para mantener las convenciones adoptadas en el procedimiento de ventana deslizante, una trama ACK0 confirma la recepción de la trama numerada como 1 e indica que el receptor está preparado para aceptar la trama numerada como 0.

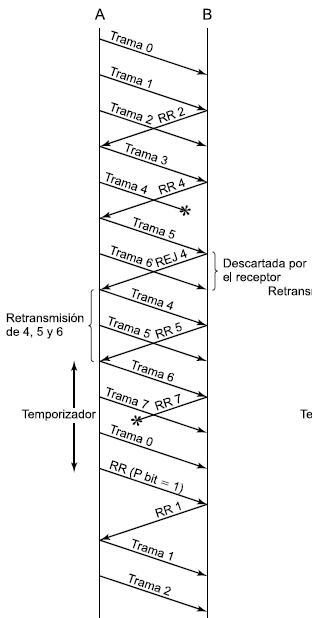


En el esquema se muestra un ejemplo acerca de la utilización de un esquema ARQ con parada y espera en la que se ilustra la transmisión de una secuencia de tramas desde un origen A a un destino B.

La tercera trama transmitida por A se daña o se pierde, por lo que B no devuelve ninguna trama ACK. En A se produce la expiración del temporizador y se retransmite la trama. Posteriormente, A transmite una trama etiquetada con 1 pero se pierde su correspondiente ACK0. El temporizador en A expira y se retransmite la trama. **Cuando** B **recibe dos tramas consecutivas con la misma etiqueta, descarta la segunda, pero devuelve una trama ACK0 para cada una de ellas.** Principal ventaja es la sencillez la desventaja es ineficiente.

**ARQ con vuelta atrás N**

La técnica de control de errores basada en el control de flujo mediante ventana deslizante. Una estación puede enviar una serie de tramas numeradas secuencialmente, el número de tramas pendientes de confirmar se determina mediante el tamaño de la ventana. Mientras no se produzcan errores, el receptor confirmará las tramas recibidas (ACK por ejemplo con RR). Si el receptor detecta un error en una trama, puede llevar a cabo una confirmación negativa (REJ, REJect) para dicha trama. El receptor descartará esa trama y todas las posteriores hasta que la trama errónea llegue correctamente. Así, cuando la estación origen reciba un REJ, deberá retransmitir la trama errónea además de todas las posteriores que hayan sido transmitidas tras ella.

Considérese que una estación A envía tramas a una estación B. Después de cada transmisión, A inicia un temporizador para la confirmación de la trama que se acaba de enviar. Supóngase que B ha recibido la trama (i + 1) sin errores y que A acaba de enviar la trama i. La técnica vuelta atrás N tiene en cuenta las siguientes contingencias:

1. **Trama deteriorada.** Si la trama recibida es no válida (es decir, B detecta un error en ella o la trama está tan dañada que B ni siquiera detecta su recepción), B descartará dicha trama sin más. Llegados a este punto se plantean dos posibilidades:

a) A envía la trama (i + 1) dentro de un periodo de tiempo razonable. B recibe la trama (i + 1) fuera de orden y envía una REJ i. A debe retransmitir la trama i y todas las posteriores.

b) A no envía tramas adicionales en un breve espacio de tiempo. B no recibe nada, por lo que ni devuelve una trama RR ni una REJ i. Cuando el temporizador de A expira, esta estación transmite una trama RR que incluye un bit denominado P, que estará puesto a 1. B interpreta la trama RR con el bit P igual a 1 como si fuera una orden que debe ser confirmada mediante el envío de una trama RR para indicar la siguiente trama que se espera recibir, la i. Cuando A recibe la trama RR, retransmite la trama i. Esta retransmisión por parte de A puede realizarse también ante la expiración de su temporizador.

**2. Trama RR deteriorada.** Existen dos casos posibles:

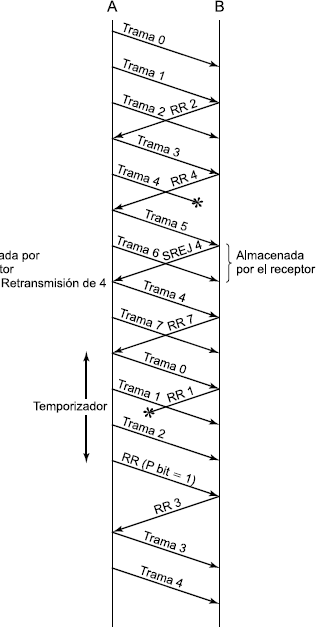
**a)** B recibe la trama *i* y envía una RR (i+1), que sufre un error en el camino. Dado que las confirmaciones son acumulativas (por ejemplo, RR 6 significa que se confirman todas las tramas hasta la 5), puede ocurrir que A reciba después una RR correspondiente a una trama posterior y que llegue antes de que el temporizador asociado a la trama *i* expire.

**b)** Si el temporizador de del transmisor expira, dicha transmite una orden RR, como en el caso 1.b. A inicia otro temporizador, denominado temporizador del bit P. Si B no responde a la orden RR, o si la respuesta se deteriora a lo largo de su transmisión, entonces el temporizador del bit P en A expirará. A lo intentará de nuevo enviando otra orden RR, reiniciando el temporizador del bit P. Este procedimiento se repite una serie de veces. Si A no recibe la confirmación tras un número máximo de intentos, comenzará un procedimiento de reinicio.

**3. Trama REJ deteriorada.** La pérdida de una trama REJ es equivalente al caso 1b.

**ARQ con rechazo selectivo o retransmisión selectiva**

Las únicas tramas que se retransmiten son las de confirmación negativa, denominada SREJ (*Selecti*v*e REJect*) ahora, o aquellas para las que el temporizador correspondiente expira. Ejemplo, Cuando la trama 5 se recibe fuera de orden, se envía una SREJ 4, indicando que la trama 4 no se ha recibido. Sin embargo, el receptor sigue aceptando tramas y las almacena en la memoria temporal hasta que se reciba correctamente la trama.

* minimiza el número de retransmisiones con respecto a vuelta atrás N.
* el receptor debe mantener una zona de memoria temporal lo suficientemente grande para almacenar las tramas recibidas tras el envío de una SREJ hasta que la trama errónea se retransmita
* debe tener lógica adicional para reinsertar la trama reenviada en la posición correspondiente.
* ****el transmisor necesita también una lógica más compleja para poder enviar tramas fuera de orden.

**Tamaño de la ventana**

La limitación en el tamaño máximo de la ventana es más restrictiva en el caso del esquema de rechazo selectivo que en el de vuelta atrás N.

Porque el tamaño de la secuencia en 2^k-1:

Con ventana deslizante:

* A transmite 0 y recibe un RR1.
* A transmite a B 1, 2, 3, 4, 5, 6, 7, 0 y A recibe un RR1.

Con rechazo selectivo que utiliza 3 bits para los números de secuencia. Permítase un tamaño de ventana igual a 7 y ténganse en cuenta las siguientes consideraciones:

**1.** La estación A envía las tramas desde la 0 hasta la 6 a la estación B.

**2.** La estación B recibe las siete tramas y las confirma acumulativamente mediante el envío de una trama RR 7.

**3.** Debido a una ráfaga de ruido, la trama RR 7 se pierde.

**4.** El temporizador de A expira y se retransmite la trama 0.

**5.** B ha desplazado su ventana de recepción indicando que acepta las tramas 7, 0, 1, 2, 3, 4 y 5. Al recibir la trama 0, supone que la 7 se ha perdido y que se trata de una trama 0 diferente, por lo que la acepta.

**Para evitar este problema, el tamaño máximo de la ventana no debería ser mayor que la mitad del rango de los números de secuencia. Para un campo de números de secuencia de *k* bits, es decir, para un rango igual a 2**^***k*, el tamaño máximo de la ventana se limita a 2**^***k*-1.**

**CONTROL DEL ENLACE DE DATOS DE ALTO NIVEL (HDLC)**

El protocolo de control del enlace de datos más importante, HDLC (*High-le*v*el Data Link Control*, ISO 3009, ISO 4335), es la base de otros importantes protocolos de control del enlace.

**CARACTERÍSTICAS BÁSICAS**

HDLC define tres tipos de estaciones, dos configuraciones del enlace y tres modos de operación para la transferencia de los datos. Los tres tipos de estaciones son:

**Estación primaria:** es la responsable de controlar el funcionamiento del enlace. Las tramas generadas por la estación primaria se denominan órdenes. Establece un enlace lógico independiente con cada estación secundaria de la línea.

**Estación secundaria**: funciona bajo el control de la estación primaria. Las tramas generadas por la estación secundaria se denominan respuestas.

**Estación combinada:** combina las características de las primarias y de las secundarias, pudiendo generar tanto órdenes como respuestas.

Las dos posibles configuraciones del enlace son:

**Configuración no balanceada:** está formada por una estación primaria y una o más secundarias. Permite tanto transmisión *full* como *half-duplex*.

**Configuración balanceada:** consiste en dos estaciones combinadas. Permite igualmente transmisión *full-duplex* y *half-duplex*.

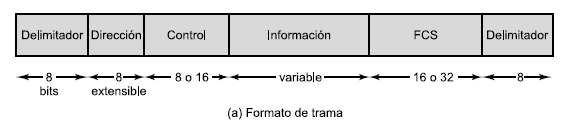
Los tres modos de transferencia de datos son:

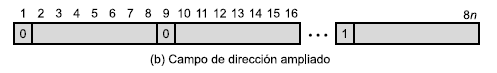
**Modo de respuesta normal** (NRM): se utiliza en la configuración no balanceada. La estación primaria puede iniciar la transferencia de datos hacia la secundaria, pero la secundaria sólo puede transmitir datos en base a respuestas a las órdenes emitidas por la primaria.

**Modo balanceado asíncrono** (ABM): se utiliza en la configuración balanceada. En este modo, cualquier estación combinada puede iniciar la transmisión sin necesidad de recibir permiso por parte de la otra estación combinada.

**Modo de respuesta asíncrono** (ARM): se utiliza en la configuración no balanceada. La estación secundaria puede iniciar la transmisión sin tener permiso explícito de la primaria. La estación primaria sigue teniendo la responsabilidad del funcionamiento de la línea, incluyendo la iniciación, la recuperación de errores y la desconexión lógica.

**ESTRUCTURA DE TRAMA**

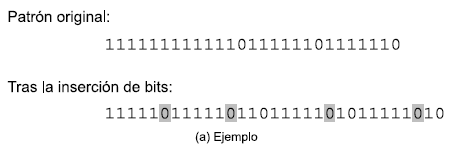
HDLC emplea transmisión síncrona. Todos los intercambios se realizan en base a tramas con un único formato. En la Figura se muestra la estructura de la trama HDLC. Los campos de delimitación, de dirección y de control, que preceden al campo de información, se denominan **cabecera**. Los campos FCS y de delimitación, que están a continuación del campo de datos, se denominan **cola**.



**Campos de delimitación**

Localizados en los dos extremos de la trama y ambos son el patrón de bits 01111110. Se puede usar un único delimitador como final de trama y comienzo de la siguiente simultáneamente. Los receptores están continuamente intentando detectar la secuencia para sincronizarse con el comienzo de la trama. Mientras se recibe una trama, la estación sigue intentando detectar esa misma secuencia para determinar el final de la trama. Como se permite cualquier combinación de bits para evitar confusión con el patrón 01111110 se utiliza un procedimiento denominado ***inserción de bits*.** En los bits existentes entre los delimitadores de comienzo y de fin:

* el emisor insertará un 0 extra siempre que se encuentre con la aparición de cinco 1 consecutivos.
* El receptor monitorizará la cadena de bits recibida entre los delimitadores, cuando aparezca una combinación de cinco 1 seguidos, el sexto bit se analiza como sigue:
  + Si dicho bit es 0, se eliminará sin más.
  + Si el sexto bit es un 1 y el séptimo es un 0, la combinación se considera como un delimitador.
  + Si los bits sexto y séptimo son ambos igual a 1, se interpreta como una indicación de cierre generada por el emisor.



Cuando se usa un solo delimitador para el final y el comienzo, un simple error en un bit causaría que las dos tramas se fundieran en una. Del mismo modo, la aparición de un error en un solo bit dentro de la trama podría partir ésta en dos.

**Campo de dirección**

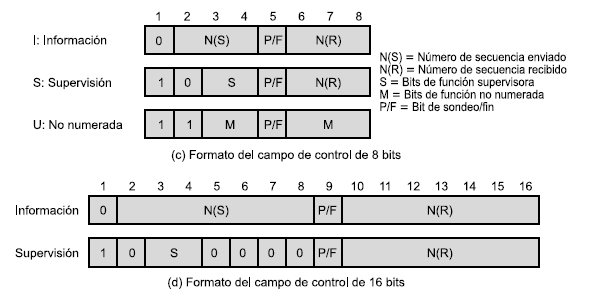
Identifica la estación secundaria que ha transmitido o va a recibir la trama. Consta de 8 bits, se puede utilizar un formato ampliado en el que la dirección es múltiplo de siete bits. El bit menos significativo de cada octeto será 1 o 0 en función de si es o no, respectivamente, el último octeto del campo de dirección. Los siete bits restantes de cada octeto constituyen la dirección propiamente dicha. Un octeto de la forma 11111111 se interpreta como una dirección que representa a todas las estaciones, tanto en el formato básico como en el ampliado. Se usa cuando la estación primaria quiere enviar una trama a todas las secundarias.

**Campo de control**

Se definen tres tipos de tramas, cada una de ellas con un formato diferente para el campo de control:

* Las **tramas de información (tramas-I)** transportan los datos generados por el usuario, se incluye información para el control ARQ de errores y de flujo.
* Las **tramas de** **supervisión (tramas-S)** proporcionan el mecanismo ARQ cuando no se usa la incorporación de las confirmaciones en las tramas de información (*piggybacking*).
* Las **tramas no numeradas (tramas-U)** proporcionan funciones complementarias para controlar el enlace.

El primero o los dos primeros bits del campo de control se utilizan para identificar el tipo de trama. Los bits restantes se organizan en subcampos. Todos los formatos posibles del campo de control contienen el bit sondeo/fin (P/F), en las tramas de órdenes se denomina bit P y se fija a valor 1 para solicitar (sondear) una trama de respuesta a la entidad HDLC par. En las tramas de respuesta, este bit se denomina F y se fija a valor 1 para identificar la trama de respuesta devuelta tras la recepción de una orden.



**Campo de información**

El campo de información sólo está presente en las tramas-I y en algunas tramas-U. Este campo puede contener cualquier secuencia de bits, con la única restricción de que el número de bits sea igual a un múltiplo entero de octetos. La longitud del campo de información es variable y siempre será menor que un valor máximo predefinido.

**Campo de secuencia de comprobación de trama**

La secuencia de comprobación de trama (FCS, *Frame Check Sequence*) es un código para la detección de errores calculado a partir de los bits de la trama, excluyendo los delimitadores. El código que se usa normalmente es el CRC-CCITT de 16 bits también se puede utilizar un campo FCS de 32 bits, que haga uso del polinomio CRC-32.

**FUNCIONAMIENTO**

Consiste en el intercambio de tramas-I, tramas-S y tramas-U entre dos estaciones. Implica tres fases. En primer lugar, uno de los dos extremos inicia el enlace de datos, de manera que las tramas se puedan intercambiar de una forma ordenada. Durante esta fase se acuerdan las opciones que se usarán en el intercambio posterior. Tras la iniciación, los dos extremos intercambian datos de usuario e información de control para llevar a cabo los procedimientos de control de flujo y de errores. Finalmente, uno de los dos extremos indicará la finalización de la transmisión.

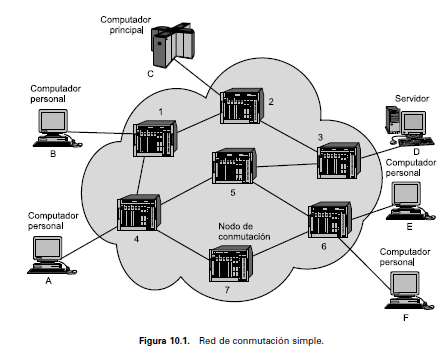
**CONMUTACIÓN DE CIRCUITOS Y DE PAQUETES**

**REDES CONMUTADAS**

Para la transmisión de datos, la comunicación se realiza mediante la transmisión de datos desde el origen hasta el destino a través de una red de nodos de conmutación intermedios. El contenido de los datos no es del interés de los nodos de conmutación, sino que el propósito de estos últimos es proporcionar un servicio de conmutación que permita el intercambio de datos entre nodos hasta alcanzar el destino deseado.

Los dispositivos finales que desean comunicarse se denominan *estaciones*, los dispositivos de conmutación cuyo objetivo es proporcionar la comunicación se les denomina *nodos*. Los nodos están conectados entre sí mediante enlaces de transmisión, formando una topología dada. Cada estación se conecta a un nodo, llamándose *red de comunicaciones* al conjunto de todos los nodos.

Estos tipos de redes se denominan *redes de* ***comunicación conmutadas***. Los datos que entran a la red procedentes de una estación se encaminan hacia el destino mediante su conmutación de nodo en nodo.



Consideraciones de las redes conmutadas:

* Algunos nodos sólo se conectan con otros nodos, siendo su única tarea la conmutación interna de los datos. Otros nodos tienen también conectadas una o más estaciones, de modo que, además de sus funciones de conmutación, estos nodos aceptan datos desde y hacia las estaciones conectadas a ellos.
* Los enlaces entre nodos están multiplexados, utilizándose multiplexación por división de frecuencias FDM o por división de tiempo TDM
* La red no está completamente conectada, no existe un enlace directo entre cada posible

En las redes conmutadas de área amplia se emplean **dos tecnologías diferentes**: **conmutación de circuitos y conmutación de paquetes,** las cuales difieren en la forma en que los nodos conmutan la información entre enlaces en el camino desde el origen hasta el destino.

**REDES DE CONMUTACIÓN DE CIRCUITOS**

Las comunicaciones implican la existencia de un camino o canal de comunicación dedicado entre dos estaciones, el cual consiste en una secuencia de enlaces conectados entre nodos de la red; es decir, se establece un canal de comunicaciones dedicado entre dos estaciones. Se reservan recursos de transmisión y de conmutación de la red para su uso exclusivo en el circuito durante la conexión. En cada uno de los enlaces físicos se dedica un canal lógico para cada conexión establecida. Implica 3 fases:

**1. Establecimiento del circuito.** Antes de transmitir señal alguna, se debe establecer un circuito estación a estación, reservando los enlaces y los recursos necesarios en cada nodo.

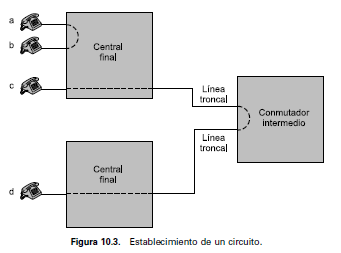
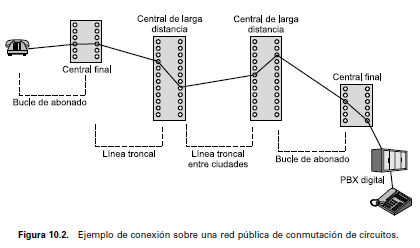
**2. Transferencia de datos.** Tras el establecimiento del circuito se puede transmitir la información desde A hasta E a través de la red.

**3. Desconexión del circuito.** Tras la fase de transferencia de datos, la conexión finaliza por orden de una de las dos estaciones involucradas. Las señales se deben propagar a los nodos involucrados para que éstos liberen los recursos dedicados a la conexión que se cierra.

El canal de conexión se establece antes de que comience la transmisión de datos, los conmutadores deben contar con la inteligencia necesaria para realizar estas reservas y establecer una ruta a través de la red.

La conmutación de circuitos puede ser bastante ineficiente. La capacidad del canal se dedica permanentemente a la conexión mientras dura ésta, incluso si no se transfieren datos. La utilización puede ser bastante alta para una conexión de voz. Por su parte, para comunicaciones entre un terminal y un computador, es posible que el canal esté libre durante la mayor parte de la conexión. Desde el punto de vista de las prestaciones, existe un retardo previo a la transferencia de las señales debido al establecimiento de la llamada; no obstante, una vez establecido el circuito, **la red es transparente para los usuarios.** La información se transmite a una velocidad fija sin otro retardo que el de propagación a través de los enlaces de transmisión, siendo despreciable el retardo introducido por cada nodo de la ruta.

La conmutación de circuitos fue desarrollada para el tráfico de voz, pero en la actualidad se usa también para el tráfico de datos. Ejemplos:

* Una red de conmutación de circuitos es el de la red telefónica pública.
* Otra aplicación son las centralitas privadas (PBX, *Pri*v*ate Branch eXchange*), usadas para conectar los teléfonos dentro de un edificio u oficina.
* También se utiliza la conmutación de circuitos en redes privadas. Este tipo de redes se utiliza usualmente en compañías u organizaciones para conectar sus diferentes delegaciones o sedes

La conmutación de circuitos es adecuada para la transmisión analógica de señales de voz. **Una de sus ventajas principales es la transparencia: una vez que se ha establecido el circuito, éste parece una conexión directa entre las dos estaciones conectadas, no siendo necesaria la inclusión de lógica de red especial en las estaciones.**

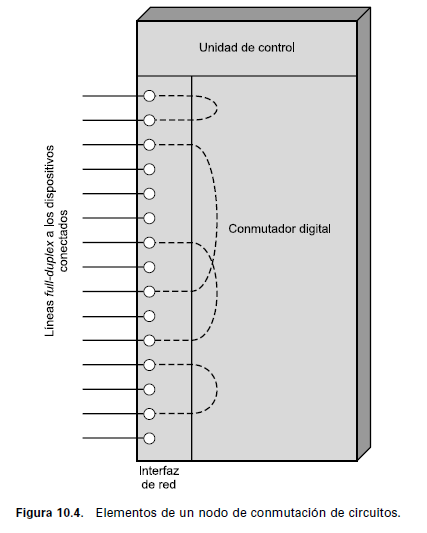
**CONCEPTOS DE CONMUTACIÓN DE CIRCUITOS**

Consideremos el funcionamiento de un solo nodo conmutado. El conmutador establecerá un canal dedicado entre dos dispositivos cualquieras que deseen comunicarse.

La parte central es el **conmutador digital**, su función es proporcionar una ruta transparente entre dos dispositivos conectados cualquiera en el sentido de que parece como si existiese una conexión directa entre los dispositivos.

El elemento de **interfaz de red** incluye las funciones y el hardware necesarios para conectar dispositivos digitales.

La **unidad de control** realiza tres tareas: establece conexiones, lo cual se realiza bajo demanda; es decir, ante la solicitud de un dispositivo conectado a la red. En segundo lugar, la unidad de control debe mantener la conexión. Por último, la unidad de control debe liberar la conexión.



Una característica importante de un dispositivo de conmutación de circuitos es si es ***bloqueante* o *no bloqueante***. El **bloqueo** se produce cuando la red no puede conectar dos estaciones debido aque todos los posibles caminos entre ellas están siendo ya utilizados.

**Una red bloqueante** es aquellaen la que es posible el bloqueo. Por su parte, una **red no bloqueante** se caracteriza porque permiteque todas las estaciones se conecten simultáneamente (por parejas) y garantiza el servicio a todaslas solicitudes de conexión posibles siempre que el destino esté libre.

Técnicas de conmutación internas a un nodo de conmutación de circuitos:

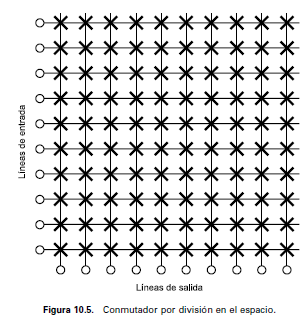
**CONMUTACIÓN POR DIVISIÓN EN EL ESPACIO**

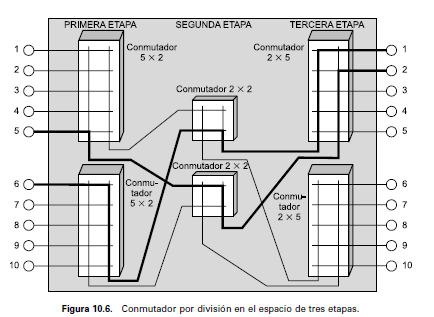
Se desarrolló para entornos analógicos, desplazándose luego para entornos digitales.

Un conmutador por división en el espacio es aquel en el que las rutas de señal que se establecen son físicamente independientes entre sí (separadas en el espacio). Cada conexión necesita del establecimiento de un camino físico a través del conmutador que se dedique únicamente a la transferencia de señales entre los dos extremos. El bloque básico de un conmutador consiste en una matriz de conexiones metálicas (o puntos de cruce) o puertas semiconductoras que una unidad de control puede habilitar o deshabilitar.

Ejemplo. La matriz tiene 10 entradas y 10 salidas; cada estación se conecta a la matriz a través de una línea de entrada y otra de salida. La conexión entre dos líneas cualesquiera es posible habilitando el punto de cruce correspondiente. Se necesitan 100 conexiones. Los conmutadores matriciales presentan varias limitaciones:

* El número de conexiones crece exponencialmente, lo cual resulta costoso para conmutadores grandes.
* La pérdida de un cruce impide la conexión entre los dos dispositivos cuyas líneas interseccionan en ese punto de cruce.
* Las conexiones se utilizan de forma ineficiente; incluso cuando todos los dispositivos conectados se encuentran activos, sólo está ocupada una pequeña fracción de los puntos de cruce.





Para superar estas limitaciones se emplean conmutadores multietapa. Esta solución presenta dos ventajas respecto a una matriz de una sola etapa:

* El número de conexiones se reduce, aumentando la utilización de las líneas de cruce.
* Existe más de una ruta a través de la red para conectar dos extremos, incrementándose así la fiabilidad de la misma.

Una red multietapa necesita un esquema de control más complejo, se debe determinar una ruta libre a través de las etapas.

Un conmutador por división en el espacio multietapa es que puede ser bloqueante. Una matriz de una sola etapa es no bloqueante; es decir, siempre hay un camino disponible para conectar una entrada con una salida. Un conmutador multietapa puede convertirse en no bloqueante aumentando el número o el tamaño de los conmutadores intermedios, si bien ello incrementará el costo.

**CONMUTACIÓN POR DIVISIÓN EN EL TIEMPO**

La conmutación por división en el tiempo involucra la fragmentación de una cadena de bits de menor velocidad en segmentos que compartirán una secuencia de velocidad superior con otras cadenas de bits. Los fragmentos individuales, o ranuras, se gestionan por parte de la lógica de control con el fin de encaminar los datos desde la entrada hacia la salida.

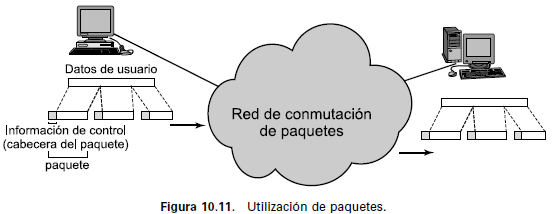
**REDES DE CONMUTACIÓN DE PAQUETES**

La conmutación de circuitos tiene 2 problemas:

* La línea puede estar desocupada la mayor parte del tiempo lo que resulta ineficiente.
* La conexión ofrece una velocidad de datos constante de modo que los dispositivos deben transmitir y recibir a la misma velocidad.

Para resolver esto, la conmutación de paquetes una estación transmite los datos en bloques pequeños llamados paquetes, siendo 1.000 octetos un límite típico de la longitud de los mismos. Si un emisor tiene que enviar un mensaje de mayor longitud, éste se segmenta en una serie de paquetes. Cada paquete contiene una parte de los datos de usuario más información de control. Esta información comprende la que necesita la red para encaminar el paquete a través de ella y alcanzar el destino deseado. En cada nodo de la ruta, el paquete se recibe, se almacena temporalmente y se envía al siguiente nodo.

Cada nodo almacena el paquete, determina el siguiente nodo en la ruta y pone en cola el paquete en ese enlace. Cuando el enlace está disponible, el paquete se transmite hacia el siguiente nodo.



Ventajas frente a la conmutación de circuitos:

* La eficiencia de la línea es superior, ya que un único enlace entre dos nodos se puede compartir dinámicamente en el tiempo entre varios paquetes. Los paquetes forman una cola y se transmiten sobre el enlace tan rápidamente como es posible.
* Una red de conmutación de paquetes puede realizar una conversión en la velocidad de los datos. Dos estaciones de diferente velocidad pueden intercambiar paquetes, ya que cada una se conecta a su nodo con una velocidad particular.
* Cuando aumenta el tráfico en una red de conmutación de circuitos algunas llamadas se bloquean, en una red de conmutación de paquetes éstos siguen aceptándose, si bien aumenta el retardo en la transmisión.
* Se puede hacer uso de prioridades, de modo que si un nodo tiene varios paquetes en cola para su transmisión.

**TÉCNICA DE CONMUTACIÓN DE PAQUETES**

La red tiene que gestionar la secuencia de paquetes para encaminarlos y entregarlos en el destino deseado. Existen 2 opciones: datagramas y circuitos virtuales.

En la técnica de **datagramas** cada paquete se trata de forma independiente, sin referencia alguna a los paquetes anteriores. Cada nodo elige el siguiente nodo en la ruta del paquete de acuerdo con información recibida de los nodos vecinos acerca de tráfico, fallo en las líneas, etc. De este modo, no todos los paquetes, aunque con el mismo destino, seguirán la misma ruta, pudiendo recibirse desordenados en el último nodo. Pudiendo ser el nodo final o el destino quien los reordena. Es posible que los paquetes se pierdan en la red, será responsabilidad del nodo final o del destino detectar la pérdida de un paquete y decidir cómo recuperarlo. En esta técnica, **cada paquete se denomina datagrama** y se trata de forma independiente del resto.

En la técnica de **circuitos virtuales** se establece una ruta previa al envío de los paquetes. Una vez establecida ésta, todos los paquetes intercambiados siguen ese camino a través de la red. Dado que el camino es fijo mientras dura la conexión lógica, éste es similar a un circuito en redes de conmutación de circuitos, por lo que se le llama circuito virtual. Además de los datos, cada paquete contiene un identificador de circuito virtual. Cada nodo de la ruta preestablecida sabe hacia dónde dirigir los paquetes, por lo que no necesita de la toma de decisiones de encaminamiento. En un instante de tiempo dado, cada estación puede tener más de un circuito virtual hacia otra u otras estaciones.

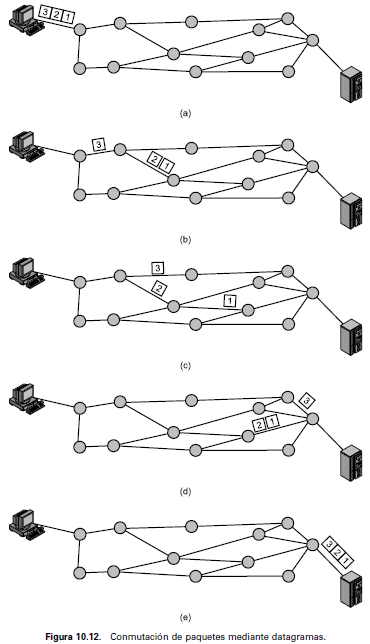
La ruta entre las estaciones se establece antes de la transferencia de los datos, pero no significa que sea una ruta dedicada como la conmutación de circuitos. Un paquete continúa siendo almacenado en cada nodo y puesto en cola sobre una línea de salida, mientras que otros paquetes en otros circuitos virtuales pueden compartir el uso de la línea. La diferencia con la técnica de datagramas es que, en circuitos virtuales, el nodo no necesita tomar decisiones de encaminamiento para cada paquete, sino que ésta se toma una sola vez para todos los paquetes que usan dicho circuito virtual.

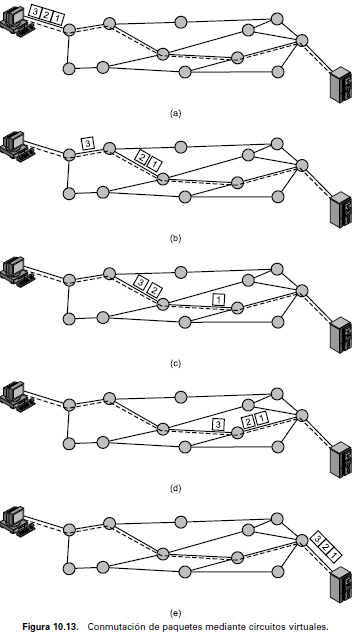
Ventajas circuito virtual:

* La red puede ofrecer servicios sobre el circuito virtual, incluyendo orden secuencial y control de errores.

Ventajas datagramas:

* No existe la fase de establecimiento de llamada, si una estación desea enviar sólo uno o pocos paquetes, el envío resultará más rápido.
* Como es más rudimentario, resulta más flexible. Si se congestiona una parte de la red esta se puede evitar en cambio en circuitos virtuales siguen la ruta predefinida.
* El envío datagrama es inherentemente más seguro. Con la utilización de circuitos virtuales, si un nodo falla se perderán todos los circuitos virtuales que atraviesan ese nodo.





**TAMAÑO DE PAQUETE**

Existe una relación importante entre el tamaño de paquete considerado y el tiempo de transmisión. Si se produce un solapamiento adecuado en la transmisión, el tiempo total de ésta disminuye.

**COMPARACIÓN DE LAS TÉCNICAS DE CONMUTACIÓN DE CIRCUITOS Y DE PAQUETES**

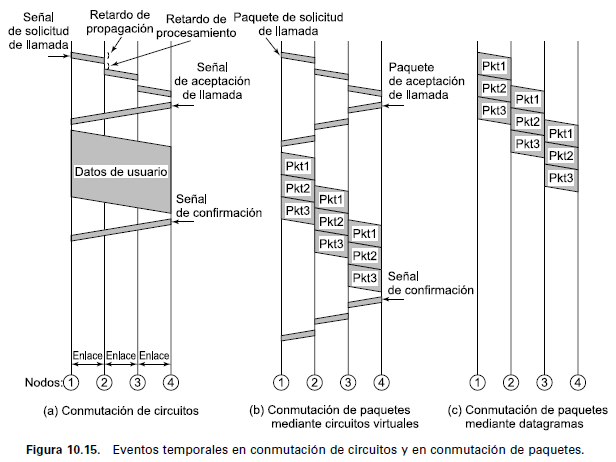
**Prestaciones**

Se ilustra la transmisión de un mensaje a través de cuatro nodos, desde una estación emisora conectada al nodo 1 hasta una estación de destino conectada al nodo 4. En esta figura se indican tres tipos de retardo:

* **Retardo de propagación:** tiempo que tarda la señal en propagarse desde un nodo hasta el siguiente.
* **Tiempo de transmisión:** tiempo que tarda un transmisor en enviar un bloque de datos.
* **Retardo de nodo:** tiempo que tarda un nodo en realizar los procesos necesarios para la conmutación de los datos.

En conmutación de circuitos existe un cierto retardo antes de que se pueda enviar el mensaje. Primero se envía una señal para establecer una conexión con el destino. Si la estación de destino no está ocupada, devuelve una señal. Obsérvese la aparición de un retardo de procesamiento en cada nodo durante la solicitud de llamada debido a la necesidad de establecer la ruta para la conexión. A la vuelta no se requiere procesamiento dado que la conexión está ya establecida. Una vez establecida la conexión, el mensaje se envía como un único bloque, sin retardos en los nodos de conmutación.

La técnica de conmutación de paquetes mediante circuitos virtuales parece muy similar a la de conmutación de circuitos. Un circuito virtual se solicita mediante el uso de un paquete, el cual sufre un retardo en cada nodo. El circuito virtual se acepta mediante un paquete. Al contrario que en el caso de conmutación de circuitos, la aceptación de llamada también experimenta retardos en los nodos, aunque la ruta del circuito virtual se encuentre ya establecida. Ya que el paquete se pone en cola en cada nodo y debe esperar turno para su transmisión.

La técnica de conmutación de paquetes mediante datagramas no precisa un establecimiento de llamada, de modo que para mensajes cortos resulta más rápida que la conmutación de paquetes mediante circuitos virtuales y, quizá, que la conmutación de circuitos. Sin embargo, dado que cada datagrama individual se encamina de forma independiente, el procesamiento de cada uno de ellos en cada nodo puede llegar a ser superior que en el caso de circuitos virtuales.

**X.25**

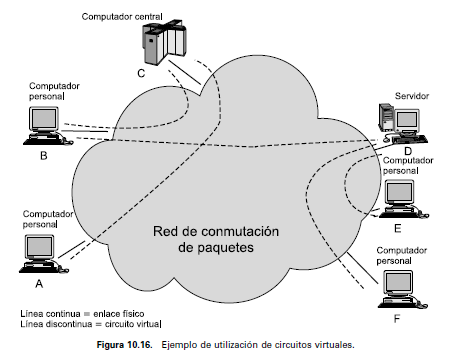
Un aspecto de las redes de conmutación de paquetes: es la interfaz entre los dispositivos conectados y la red. En el caso de las redes de conmutación de paquetes las estaciones conectadas deben organizar sus datos en paquetes para su transmisión. Esto requiere cierto grado de cooperación entre la red y las estaciones, cooperación definida en una norma de interfaz. El estándar usado casi universalmente con este fin es X.25.

X.25 es un estándar que especifica una interfaz entre una estación y una red de conmutación de paquetes. La funcionalidad de X.25 se especifica en tres niveles:

**La capa física** trata la interfaz física entre una estación y el enlace que la conecta con un nodo de conmutación de paquetes.

**La capa de enlace** se encarga de la transferencia fiable de datos a través del enlace físico mediante la transmisión de los datos como una secuencia de tramas. La capa de enlace estándar es el conocido como LAPB.

**El nivel o capa de paquete** proporciona un servicio de circuito virtual, lo que posibilita a un abonado de la red establecer conexiones lógicas, llamadas circuitos virtuales, con otros abonados.

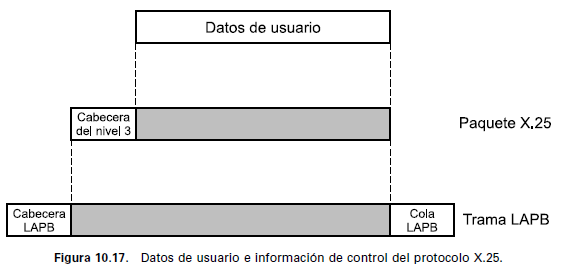


En este contexto, el término *circuito* v*irtual* se refiere a la conexión lógica entre dos estaciones a través de la red; a esto se le suele denominar circuito virtual externo. Antes, Se usó *circuito* v*irtual* para referirnos a una ruta específica predefinida a través de la red entre dos estaciones; es el denominado circuito virtual interno. Generalmente, existe una relación uno a uno entre los circuitos virtuales internos y externos. Lo importante en un circuito virtual externo es que se establece una relación lógica, o canal lógico, entre dos estaciones, considerándose todos los datos asociados a dicho canal lógico parte de una única secuencia de datos entre las estaciones.

F**uncionamiento del X.25**

Los datos de usuario se pasan hacia abajo al nivel 3 de X.25, el cual les añade una cabecera consistente en información de control, dando lugar a un paquete. Esta información de control tiene varios objetivos, entre los que se encuentran los siguientes:

**1.** Identificación de un circuito virtual dado mediante un número.

**2.** Definición de números de secuencia.

El paquete X.25 completo se pasa después a la entidad LAPB, que añade información de control al principio y al final del paquete, dando lugar a una trama LAPB.

El funcionamiento del nivel de paquete X.25, cada paquete de datos X.25 incluye números de secuencia de emisión y de recepción. El de emisión, P(S), se usa para numerar secuencialmente todos los paquetes de salida sobre un circuito virtual específico. El número de secuencia de recepción, P(R), es una confirmación de los paquetes recibidos sobre el circuito virtual en cuestión.

**RETRANSMISIÓN DE TRAMAS**

La técnica de retransmisión de tramas (*frame relay*) se diseñó para proporcionar un esquema de transmisión más eficiente que el proporcionado por X.25.

**FUNDAMENTOS**

Conmutación de paquetes hace uso de X.25, lo que no sólo determina la interfaz usuario-red, sino que también afecta al diseño interno de la red. Algunas de las características básicas de X.25 son:

* Tanto la capa 2 como la 3 incluyen mecanismos de control de flujo y de errores.

X.25 es muy costosa, ya que el protocolo de control de enlace intercambia tramas de datos y de confirmación en cada salto a través de la red. Además, cada nodo intermedio debe mantener tablas de estado para cada circuito virtual con objeto de abordar aspectos de gestión de llamadas y de control de flujo/errores del protocolo X.25. Este coste queda justificado en caso de que la probabilidad de error en los enlaces de la red sea significativa, por lo que esta técnica puede no ser la más apropiada para los servicios de comunicación modernos, dado que las redes actuales hacen uso de tecnologías de transmisión fiables sobre enlaces de transmisión de alta calidad. En este contexto, el coste de X.25 no sólo es innecesario, sino que además degrada la utilización efectiva de las altas velocidades de transmisión disponibles.

La retransmisión de tramas se ha diseñado para eliminar gran parte del costo de X.25 para el sistema final de usuario y para la red de conmutación de paquetes.

Las principales diferencias entre la técnica de retransmisión de tramas y un servicio convencional de conmutación de paquetes X.25 son:

* La señalización de control de llamadas se transmite a través de una conexión lógica distinta de la de los datos de usuario. De este modo, los nodos intermedios no necesitan mantener tablas de estado ni procesar mensajes relacionados con el control de llamadas individuales.
* La multiplicación y conmutación de conexiones lógicas tienen lugar en la capa 2 en lugar de en la capa 3, eliminándose así una capa completa de procesamiento.
* No existe control de flujo ni de errores a nivel de líneas individuales (salto a salto). Si se lleva a cabo este control, será extremo a extremo y responsabilidad de capas superiores.

En retransmisión de tramas sólo se envía una trama de datos de usuario desde el origen hasta el destino, devolviéndose al primero una trama de confirmación generada por una capa superior. En este caso no existe intercambio de tramas de datos y confirmaciones en cada uno de los enlaces del camino entre el origen y el destino.

Ventajas y desventajas en comparación con X.25

Desventajas

* No es posible el control de flujo y errores en cada enlace, pero no supone inconveniente ya que aumento la fiabilidad de los mismos.

Ventaja:

* Potencia del proceso de comunicación, reduce la funcionalidad del protocolo y el procesamiento interno de la red.
* Menor retardo
* Mayor rendimiento.