1) INTRODUCCIÓN:

- Los protocolos de las capas inferiores (física, enlace de datos y de red), están bien estandarizados y con funciones más entendibles y sencillas que las de la capa de transporte.
- Las capas superiores dependen en buena medida del protocolo de transporte, quien brinda el servicio básico extremo a extremo de la transferencia de datos entre usuarios, y aísla las aplicaciones y otros protocolos de un nivel más alto, de la necesidad de tratar con las características y servicios de la red de comunicaciones.

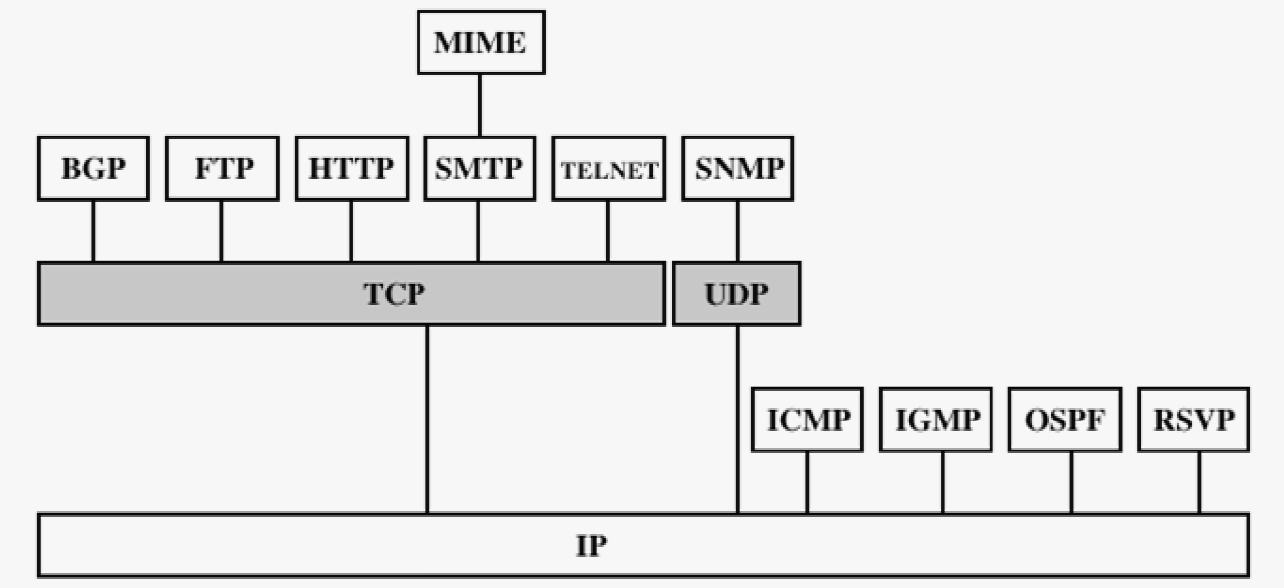


Ilustración 1: Contexto de los Protocolos del nivel de transporte.

2) SERVICIOS DE TRANSPORTE:

En los documentos de la ISO se denomina TS-"Transport Services" (Servicios de Transporte) a las clases de servicios que el protocolo de transporte puede o debería brindar a las capas superiores.

a) Tipo de servicio: puede ser orientado a conexión y no orientado a conexión

- Orientado a conexión:
  - Establecimiento, mantenimiento y cierre de la conexión lógica entre usuarios TS.
  - Servicio seguro: control de flujo, control de errores y transporte en secuencia.
  - Es el tipo de servicio más usual.
- No orientado a conexión:
  - Sin fases de establecimiento y desconexión.
  - Servicio no seguro.
  - Mayor robustez que el servicio orientado a conexión.
  - De mayor utilidad en muchos contextos: p.ej. recolección de datos de sensores, difusión de mensajes a los usuarios de red, transacciones del tipo petición/respuesta, aplicaciones de voz y telemedida en tiempo real, etc.

b) Calidad del servicio: la capa de transporte permite al usuario TS especificar la calidad de servicio de transmisión a ser suministrado, e intentará luego optimizar al máximo de sus posibilidades el uso del enlace, la red y los recursos de una colección de redes para proporcionar los servicios colectivos solicitados.

- Ejemplos de servicios:
  - Niveles de error y pérdida aceptables.
  - Retardo medio y máximo aceptable.
  - Rendimiento medio y mínimo deseado.
  - Niveles de prioridad.
- Condicionantes en la calidad del servicio:
  - Depende de las capacidades de los servicios de las capas inferiores.
  - Dependiendo de la naturaleza de la transmisión, la capa de transporte tendrá mayor o menor grado de éxito en la obtención del grado de servicio requerido.
  - Hay un compromiso entre seguridad, retardo, rendimiento y costo del servicio.
- Ejemplos de aplicaciones con calidad de servicio:
  - Protocolo de transferencia de ficheros: requiere gran rendimiento y seguridad.
  - Protocolo de transacción (p.ej. peticiones a bases de datos): requerirán bajo retardo.
  - Protocolo para correo electrónico: podría requerir diversos grados de prioridad.

c) Transferencia de datos: el principal propósito entre dos entidades de transporte, es la transferencia de datos (tanto de usuario como de control), por el mismo canal (o por canales separados), y en modalidad simplex, semiduplex o duplex, entre ambos usuarios TS.

d) Interfaz de usuario: se deben optimizar de acuerdo al entorno de la estación.

- Mecanismos para prevenir, tanto que el usuario TS inunde la entidad de transporte con datos, como para que ocurra la misma situación a la inversa.
- Mecanismos de sincronización y confirmación entre las entidades de transporte local y remota.

e) Supervisión de la conexión: en los servicios orientados a conexión, la entidad de transporte es responsable de establecer y dar fin a la conexión.

- Procedimiento de establecimiento simétrico, que permita a cualquier usuario TS iniciar la conexión.
- Procedimiento de finalización ordenada de una comunicación: evita la pérdida de datos en tránsito.

f) Transporte rápido: servicio similar al que proporcionan las clases de prioridad. En el receptor, la entidad de transporte enviará una interrupción al usuario TS para notificarle la recepción de datos urgentes. El servicio de datos urgentes es en esencia un mecanismo de interrupción, que se utiliza para transferir datos urgentes ocasionales, tales como un carácter de cancelación (break) proveniente de un terminal o una condición de alarma.

g) Informe de estado: un servicio de informe de estado permite al usuario TS obtener o conocer información relativa a la condición o a los atributos de la entidad de transporte o conexiones de transporte. Algunos ejemplos de información de estado son:

- Características de las prestaciones de una conexión: rendimiento, retardo medio, etc.
- Direcciones (de red, transporte)
- Tipo de protocolo en uso.
- Valores actuales de los temporizadores.
- Estado de la “máquina”de transporte que realiza la conexión.
- Degradación de la calidad del servicio requerido.

h) Seguridad: la entidad de transporte puede proporcionar una variedad de servicios de seguridad.

- Acceso de control mediante verificaciones locales del que envía o verificaciones remotas del receptor.

- Encriptado/desencriptado de ser requerido.
- Encaminamiento de los datos a través de conexiones o nodos seguros de estar disponibles.

3) MECANISMOS DEL PROTOCOLO DE TRANSPORTE:

Los protocolos de transporte, según el caso pueden necesitar ser muy complejos.
Hay dos variantes básicas de servicios de red: **servicio seguro y no seguro**, y en función de ellos se tendrá menor o mayor complejidad en los servicios de transporte.

3.1) SERVICIO DE RED SEGURO CON SECUENCIAMIENTO: el servicio de red aceptará mensajes con un tamaño arbitrario y los enviará en secuencia al destino, con una seguridad teórica del 100%.
Son ejemplos de éstas redes:

- Una red de conmutación de paquetes altamente segura con una interfaz X.25
- Una red Frame Relay usando el protocolo de control LAPF.
- Una LAN IEEE 802.3 usando el servicio LLC orientado a conexión.

La suposición de servicios de red seguros con secuenciamiento permiten el uso de un protocolo de transporte bastante sencillo, con cuatro cuestiones básicas a considerar: direccionamiento, multiplexación, control de flujo y establecimiento/cierre de la conexión.

3.1.a) Direccionamiento: el usuario de una entidad de transporte desea establecer una conexión o efectuar una transferencia de datos sin establecer esa conexión, con un usuario de otra entidad de transporte. El usuario destino se debe especificar mediante:

- Identificación de usuario.
- Identificación de la entidad de transporte.
- Dirección de la estación.
- Número de red.

El protocolo de transporte debe extraer esa información de la cabecera de la PDU de red, siendo lo que en el entorno OSI se llama TSAP (punto de acceso al servicio de transporte), y que en TCP se conoce como “socket” (combinación del puerto y la estación).
Para la determinación de la dirección de transporte destino, se usan estrategias estáticas y dinámicas.

3.1.b) Multiplexación: con respecto a la interfaz entre el protocolo de transporte y el protocolo de la capa superior, el protocolo de transporte implementa una función de multiplexación/demultiplexación, mediante la cual múltiples usuarios se distinguen unos de otros por números de puertos o puntos de acceso al servicio.
¿Por qué se necesitaría multiplexar en capa 4, teniendo por ejemplo los 4095 circuitos virtuales en capa 3 de una red X.25? La respuesta es que el costo depende en parte del tiempo de conexión (un circuito virtual consume algunos recursos de memoria temporal del nodo), por lo que ese tiempo de un único circuito virtual proporcionando servicio a múltiples usuarios TS debe ser compartido hacia arriba.

3.1.c) Control de flujo: mientras que el control de flujo es un mecanismo relativamente sencillo en la capa de enlace, en la capa de transporte es bastante complejo, por dos razones principales:

- El control de flujo en la capa de transporte supone la interacción de usuarios TS, entidades de transporte y el servicio de red. (intervienen colas para los envíos de datos entre una capa y la contigua en el mismo usuario y también entre capas paritarias entre distintos usuarios TS) (Ilustración 2)
- El retardo de transmisión entre entidades de transporte es generalmente grande comparado con el tiempo de transmisión real y, lo que es peor aún, variable. (Ilustración 3)

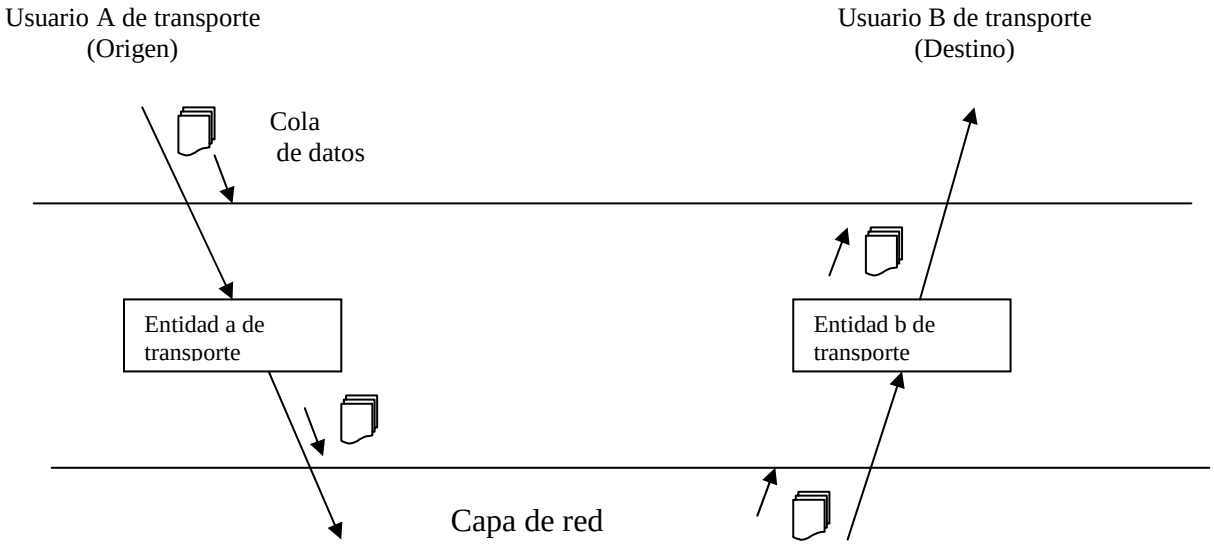
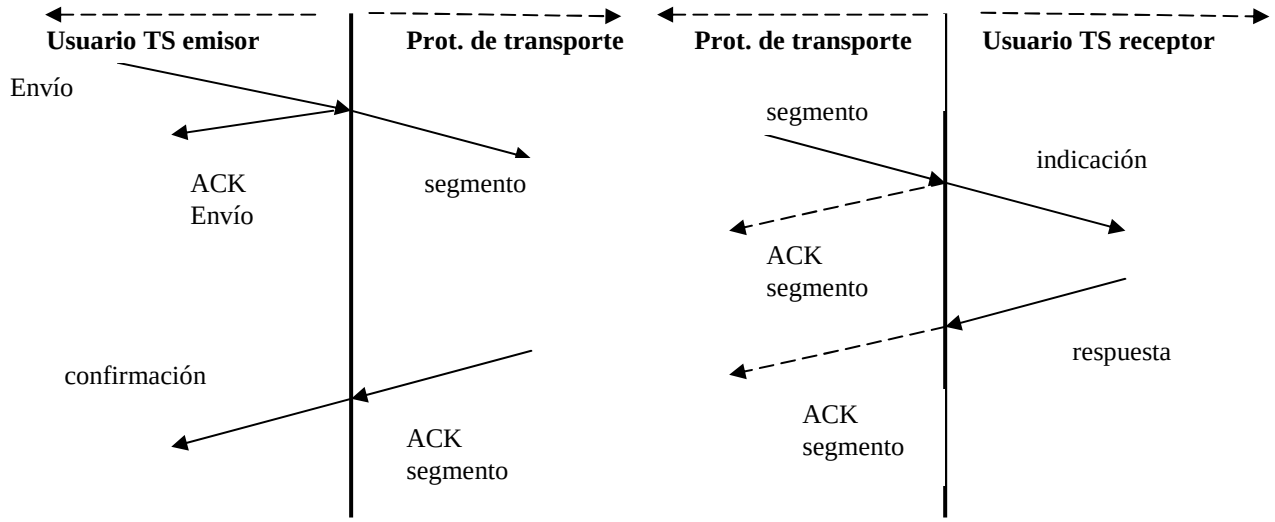


Ilustración 2: Representación mediante colas de la transferencia orientada a conexión.



(a) Extremo emisor (b) Extremo receptor

Ilustración 3: Interacción entre el usuario del servicio de transporte y el protocolo de transporte.

Una entidad de transporte tiene una cierta capacidad de memoria temporal, en la que se van almacenando los segmentos que llegan. Cada segmento es procesado (por ejemplo, examinando su cabecera de transporte) y los datos se envían al usuario .En caso de saturación de las memorias temporales, la entidad de transporte necesitará tomar medidas para detener o disminuir el flujo de segmentos para evitar la saturación.

Las alternativas para el control de flujo en la entidad de transporte receptora son:

- *No hacer nada*: significaría que los segmentos que llegan, una vez agotadas las memorias temporales, se descartan. La entidad de transporte emisora, viendo que no recibe una confirmación, los retransmitirá.

No es una solución viable, ya que ésta técnica acrecentaría el problema al tener que retransmitirse los segmentos descartados. Además anularía la ventaja fundamental de una red segura (no tener nunca que retransmitir datos).

- *Rechazar la aceptación de más segmentos del servicio de red:* mediante un mecanismo de presión hacia atrás, basándose en el servicio de red para realizar el rechazo. Cuando las memorias temporales de una entidad de transporte están llenas, ésta entidad rechaza datos adicionales del servicio de red. Esto dispara un mecanismo de control de flujo que estrangula el servicio de red en el extremo emisor.
  Es un mecanismo poco riguroso y bastante tosco.
- *Usar un protocolo fijo de ventana deslizante:* similar a los protocolos de la capa de enlace de datos. Los ingredientes clave en ésta técnica eran:
  - Uso de números de secuencia en las unidades de datos.
  - Uso de una ventana de tamaño fijo.
  - Uso de confirmaciones para desplazar la ventana.

  Con un servicio de red segura, la técnica de ventana deslizante funcionaría realmente bien. Sea por ejemplo un protocolo con tamaño de ventana de 7. Siempre que el emisor recibe una confirmación de un segmento determinado, es autorizado automáticamente a enviar los siete segmentos siguientes. Sin embargo, a pesar que el receptor puede admitir otros 7 nuevos segmentos, puede retener las confirmaciones de los segmentos que le van llegando para evitar la saturación. La entidad de transporte emisora puede enviar a lo sumo 7 segmentos adicionales y luego detenerse. Como el servicio de red subyacente es seguro, los temporizadores del emisor no expiran y no retransmitirán ninguno.
  En éstas circunstancias, una entidad de transporte emisora podría haber transmitido cierto número de segmentos sin haber recibido las confirmaciones, pero como se trabaja sobre una red segura, podría suponer dicha entidad que los segmentos han llegado y que la falta de confirmaciones es debida a una táctica de control de flujo.
  Ésta técnica no será efectiva en una red no segura, ya que la entidad de transporte emisora no sabría si la ausencia de confirmaciones es debida al control de flujo o a la pérdida de un segmento.
- *Usar un esquema de créditos:* proporciona al receptor un mayor grado de control sobre el flujo de datos. No es estrictamente necesario para un servicio de red seguro, aunque le otorga un flujo más regular. Además, es un esquema más efectivo con un servicio de red no seguro.
  El esquema de créditos separa las confirmaciones del control de flujo, a diferencia de los protocolos de ventana deslizante fija, como X.25, en que los dos son sinónimos. En un esquema de créditos, un segmento puede ser confirmado sin obtener un nuevo crédito, y viceversa.
  En la Ilustración 4 se muestra el funcionamiento del protocolo, mostrando por simplicidad el flujo de datos en un solo sentido. En el ejemplo, los segmentos de datos son numerados secuencialmente módulo 8. Inicialmente, a través del proceso de establecimiento de la conexión, los números de secuencia de emisión y recepción están sincronizados y A obtiene 7 créditos. A avanza el borde final de su ventana cada vez que transmite, y avanza el borde de la cabecera de la ventana cada vez que obtiene un crédito.
  La Ilustración 5 muestra una perspectiva desde el lado del emisor y desde el del receptor (por supuesto, ambos lados pueden presentar ambas situaciones ya que los datos se pueden intercambiar en ambas direcciones.)
  Desde el punto de vista del emisor, los números de secuencia pueden estar en una de las cuatro regiones siguientes:
  - *Datos enviados y confirmados*: comenzando con el número de secuencia inicial usado en ésta conexión hasta el último número confirmado.

- *Datos enviados pero todavía no confirmados*: representan datos que ya han sido enviados y el emisor está esperando su confirmación.
- *Transmisión de datos permitidos*: la ventana de las transmisiones permitidas, basada en los créditos no usados y que han sido asignados desde el otro extremo.
- *Números no usados e inservibles*: números fuera de la ventana.

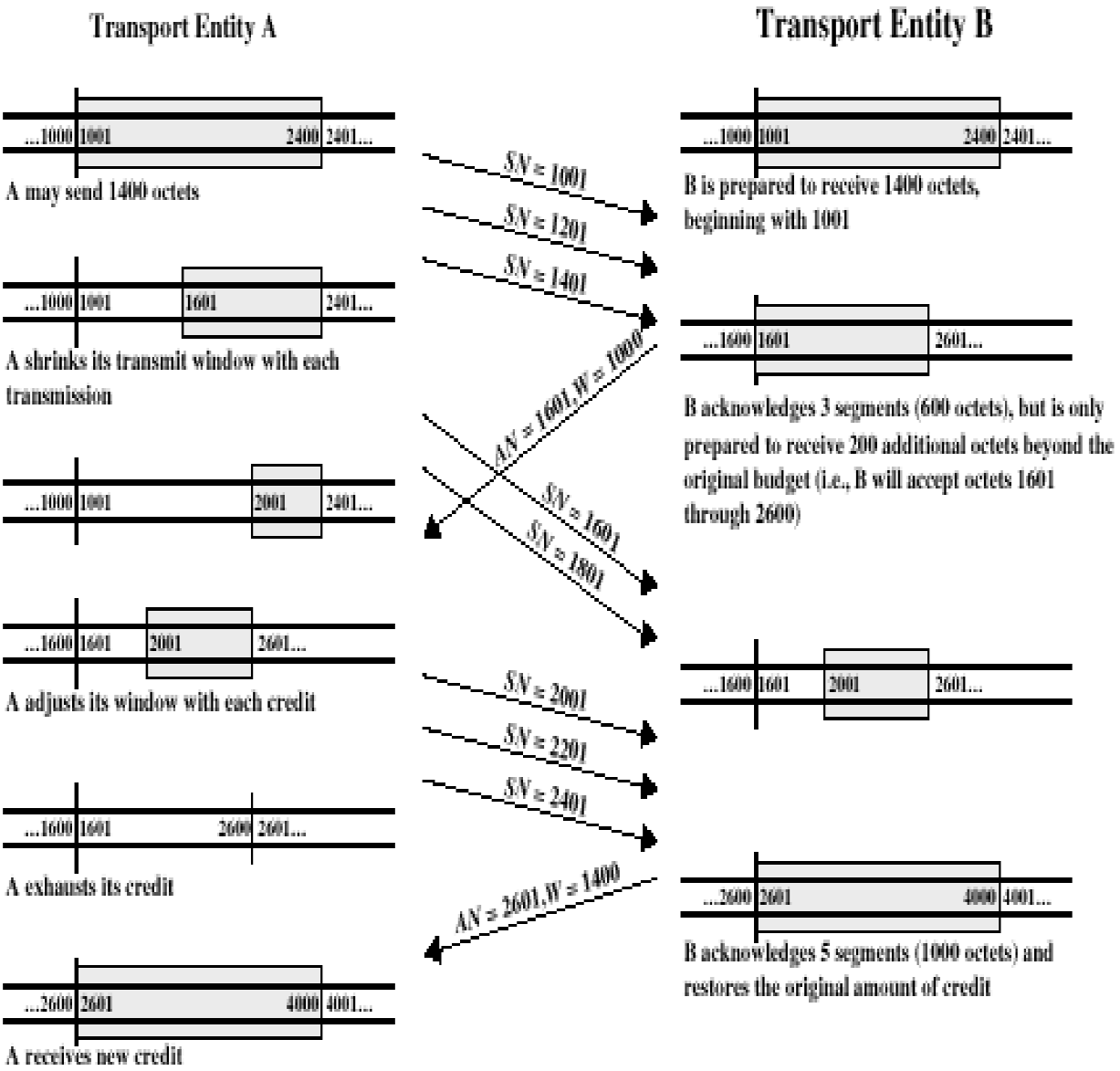


Ilustración 4: Ejemplo de mecanismo de asignación de créditos.

Desde el punto de vista del receptor, el problema se centra en la recepción de datos y en la ventana de créditos que ha sido asignada. Hay que notar que el receptor no está obligado a confirmar inmediatamente los segmentos que llegan, sino que puede esperar y emitir una confirmación conjunta para un número determinado de segmentos

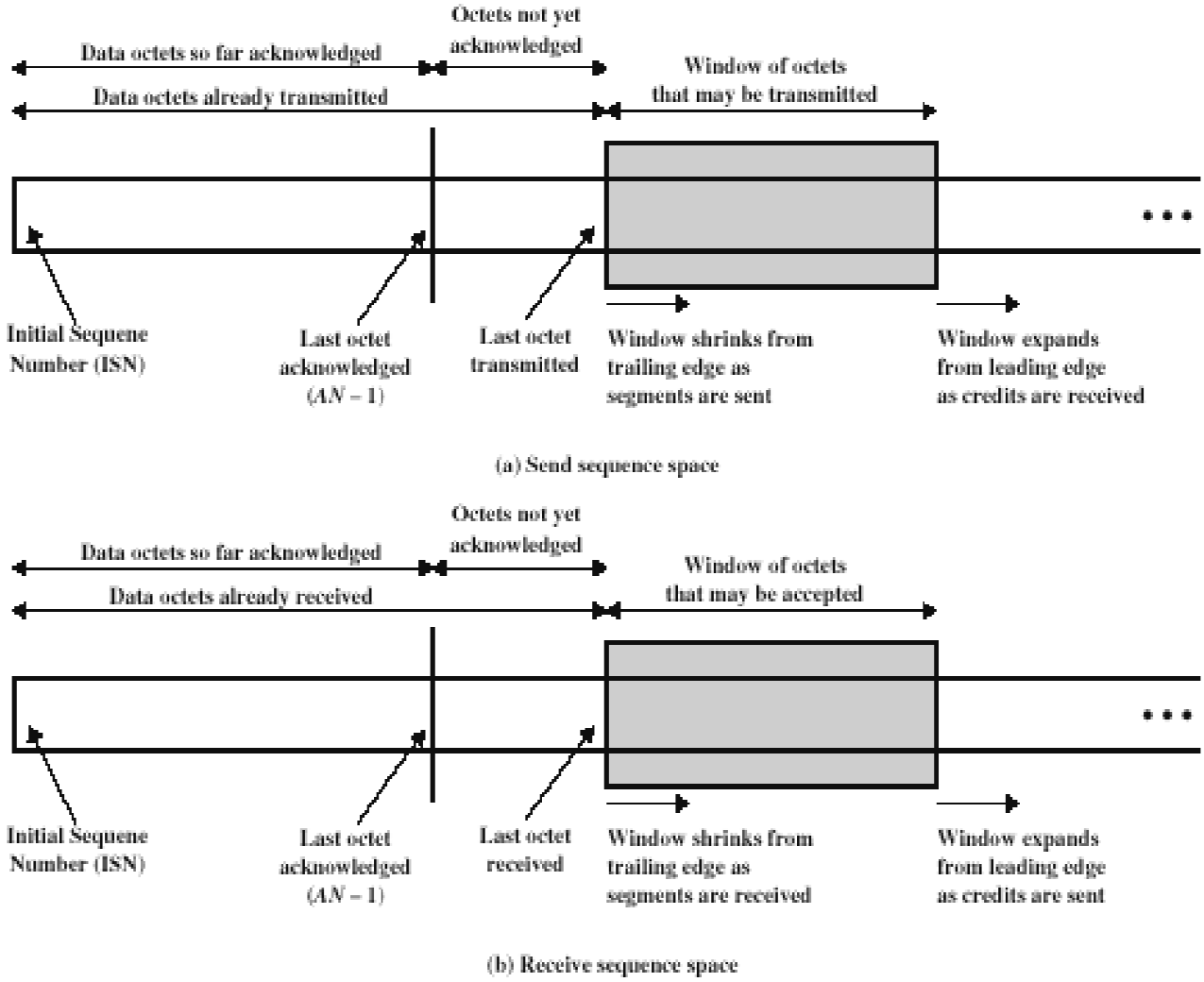


Ilustración 5: Perspectiva del control de flujo emisor y receptor.

3.1.d) Establecimiento y cierre de la conexión: incluso con servicios de red seguros, se requiere de procedimientos de establecimiento y cierre de conexión para ofrecer un servicio orientado a conexión.
Características del *establecimiento* de la conexión:

- Es por mutuo acuerdo entre ambos extremos, y se puede llevar a cabo mediante un conjunto sencillo de órdenes de usuario y segmentos de control.
- Cualquiera de los dos extremos puede iniciar la conexión.
- Permite a cada extremo asegurarse de que el otro existe.
- Permite la negociación de parámetros opcionales (por ejemplo, el tamaño del segmento, tamaño máximo de la ventana, calidad de servicio).
- Permite poner en marcha la reserva de recursos de la entidad de transporte (por ejemplo, espacio de memoria, entradas en la tabla de conexiones).

Características del *cierre* de la conexión:

- Es por mutuo acuerdo entre ambos extremos.
- Cualquiera de los dos extremos puede iniciar la desconexión.
- Se puede hacer un cierre abrupto o un cierre ordenado.
- El cierre ordenado asegura que ambos extremos han recibido todos los datos pendientes y que ambos están de acuerdo en terminar la conexión antes del cierre real.

3.2) SERVICIOS DE RED NO SEGUROS:

El caso más difícil para un protocolo de transporte es aquel en que se ofrece un servicio de red no seguro. Ejemplos de estas redes son:

- Una interconexión de redes utilizando IP.
- Una red de Frame Relay, usando solamente el núcleo del protocolo LAPF.
- Una LAN IEEE 802.3 usando un servicio no orientado a conexión LLC sin confirmaciones.

El problema ahora no es sólo que los segmentos pueden perderse ocasionalmente o ser duplicados, sino que los segmentos pueden llegar fuera de secuencia debido al retardo variable de tránsito.
Esta combinación de inseguridad y no secuenciamiento, crea problemas a resolver en todos los mecanismos vistos hasta ahora.
Básicamente, se deben tratar siete cuestiones:

- Transporte en orden.
- Estrategia de retransmisión.
- Detección de duplicados.
- Control de flujo.
- Establecimiento de la conexión.
- Cierre de la conexión.
- Recuperación de las cancelaciones no deseadas (“caídas”)

a) ***Transporte en orden****:* con un servicio de red no seguro, es posible que los segmentos, incluso si todos llegan, lo hagan en forma desordenada. La solución es numerar los segmentos secuencialmente en forma similar a los protocolos de control de enlace de datos como HDLC ó X.25, en los que cada unidad de datos (trama, paquete) se numera secuencialmente siendo cada de secuencia sucesivo, uno más que el número de secuencia anterior. Este esquema es utilizado en algunos protocolos de transporte, como los ISO. Sin embargo TCP usa un esquema algo diferente, ya que cada octeto de datos que se transmite está implícitamente numerado. Así, el primer segmento podría tener el número de secuencia 0. Si ese segmento contiene 1000 octetos de datos, el segundo segmento tendría el número de secuencia 1000, y así sucesivamente.

b) ***Estrategia de retransmisión***: existen dos eventos que requieren la retransmisión de un segmento:

- El segmento es dañado en el camino, pero llega a destino. Si se incluye en el segmento una secuencia de comprobación de trama, la entidad de transporte receptora puede detectar el error y descartar el segmento.
- El segmento no llega a destino.

En ambos casos, la entidad de transporte *emisora* no se entera que la transmisión del segmento fracasó.
La solución al primer problema es un esquema de confirmaciones positivas (ACK) de segmentos, con reconocimientos inclusivos, lo que permite la retransmisión de segmentos dañados.
La solución al segundo problema obliga a incluir un temporizador de retransmisión, ya que nunca se emitirá el ACK de un segmento perdido, que sin embargo debe ser retransmitido.
La duración del temporizador es un valor de diseño crítico:

- Si es muy pequeño: habrán muchas retransmisiones innecesarias (desperdicio de la capacidad de la red y/o potencial problema de congestión).
- Si es muy grande: el protocolo es muy lento en dar respuesta a la pérdida de un segmento.

Se puede fijar a un valor ligeramente superior al retardo de ida y vuelta (RTT-“Round Trip Time” = tiempo transcurrido entre el envío de un segmento y la recepción del ACK), aunque

éste retardo es variable aún para el caso de una carga constante de la red. Peor aún, la estadística del retardo variará con condiciones de red variables.
La fijación del RTT admite dos estrategias:

- Esquema estático: el más simple, establece un tiempo fijo en base a un comportamiento típico de la red. Al no ser adaptable a la carga instantánea de la red, un valor alto de RTT produce lentitud en las retransmisiones, mientras que un valor bajo de RTT puede contribuir a la congestión.
- Esquema dinámico: es más complejo, ya que va variando los temporizadores de retransmisión en base al valor medio de los retardos que observa en la red.
  Este tiempo no es fiable por tres razones:
  - ✓ La entidad de transporte paritaria puede que no confirme inmediatamente un segmento, ya que maneja la opción de ACK inclusivo.
  - ✓ Si un segmento debió retransmitirse, el emisor no puede saber si el ACK que recibe es la confirmación del segmento original o del retransmitido.
  - ✓ Las condiciones de la red pueden cambiar muy rápidamente.

Cada uno de estos problemas es la causa de una complicación adicional en el algoritmo de transporte, que no admite una solución completa. Siempre habrá un grado de incertidumbre sobre el mejor valor para el temporizador de retransmisión.

c) ***Detección de duplicados*****:** Si un segmento se pierde y después se retransmite, no se producirá confusión. Sin embargo, si se pierde un ACK, uno o más segmentos serán retransmitidos y, si llegan correctamente, se tendrán duplicados del segmento recibido previamente. El receptor debe ser capaz de reconocer duplicados, lo que se facilita por el hecho de que cada segmento lleve un número de secuencia, aunque la tarea no es fácil.

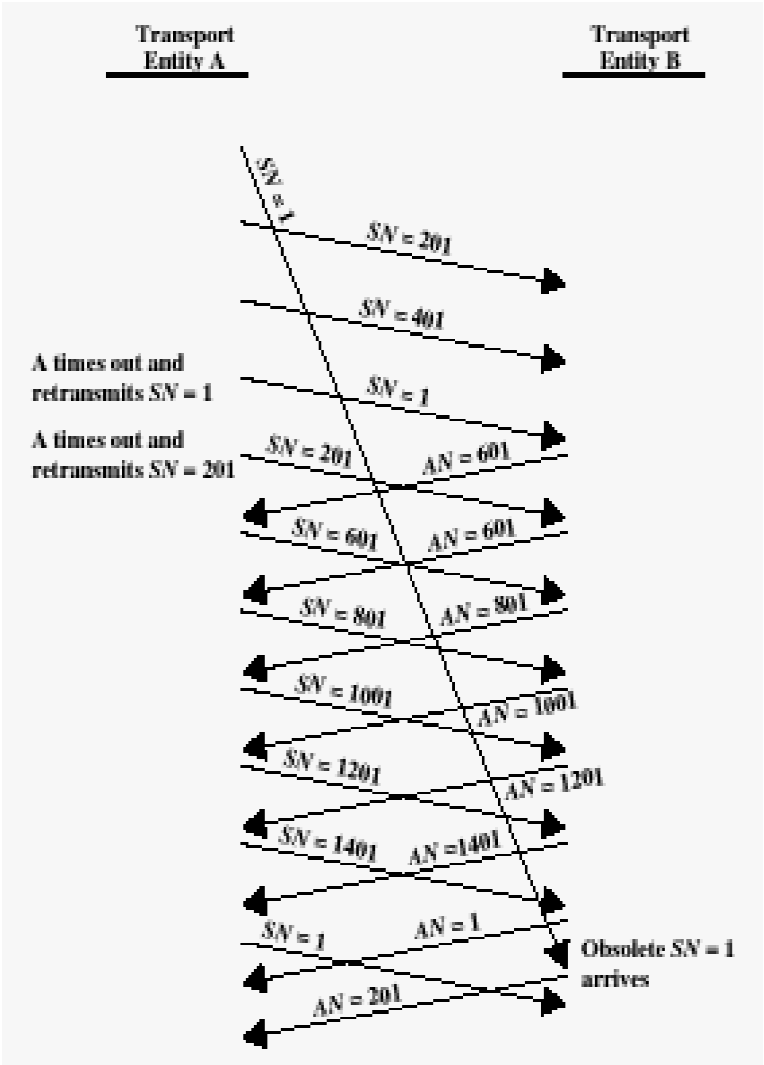


Ilustración 6: Ejemplo de una detección incorrecta de duplicados

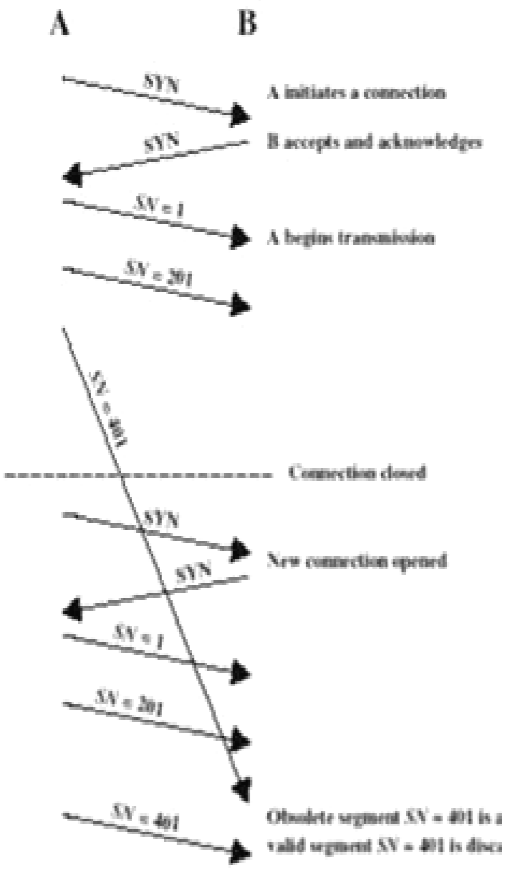
Hay dos casos posibles para los segmentos duplicados:

*1ro) Se recibe un duplicado antes del cierre de la conexión****:***
El receptor debe asumir que su confirmación se perdió, y por lo tanto debe confirmar el duplicado (el emisor no debe confundirse al recibir múltiples ACK de un mismo segmento).
El espacio de números de secuencia debe ser lo suficientemente grande para no agotarse en menos tiempo que la vida máxima posible de un segmento.
Vemos la Ilustración 6, con un ejemplo en el que el espacio de secuencia es de longitud 1600, con un protocolo de ventana deslizante con tamaño de

ventana 3. Suponemos que A ha transmitido los segmentos de datos 1, 201 y 401 y no recibe confirmaciones. Al cabo del tiempo, se produce una expiración y retransmite el segmento 1. B ha recibido el 201 y el 401, pero el 1 se ha retrasado en el camino. Por lo tanto, B no envía ninguna confirmación. Cuando llega el duplicado del segmento 1, B confirma al 1, 201 y 401. Mientras tanto, en A se produce otra expiración y se retransmite el 201, que es confirmado por B con otro AN=601. Las cosas parecen haberse areglado solas y la transferencia de datos continúa. Cuando el espacio de secuencia se agota A vuelve a comenzar con el número de secuencia 1 y continúa. Desafortunadamente, el viejo segmento 1 hace una aparición tardía y es aceptado por B antes de que el nuevo segmento 1 llegue. Queda así claro que la aparición fuera de tiempo de un segmento antiguo no habría causado dificultades si los números de secuencia no hubieran dado la vuelta. Se evita éste problema, adoptando un espacio de secuencia suficientemente grande en función del tiempo de vida del paquete y de su tasa de transmisión.

2do) Se recibe un duplicado después de cerrada la conexión:
Si se abre otra conexión entre las dos mismas entidades de transporte, un segmento de la conexión antigua podría llegar y ser aceptado en la nueva conexión. De igual forma, un ACK retrasado puede entrar en una nueva conexión y causar problemas.
Alternativas para tener una solución:

Ilustración 7: Dialogo en dos sentidos; problema con segmento de datos viejo



☑ En forma simétrica, cada unidad de transporte debe "recordar" el último número de secuencia que usó en la transmisión para cada conexión terminada, y así no tener problemas en una nueva conexión entre ambas.
☑ En forma simétrica, cada nueva conexión debe tener un único identificador específico.
☑ Si el sistema completo se cae, las alternativas anteriores no funcionarían ya que no habría registro del último número utilizado. La alternativa entonces sería simplemente esperar el tiempo suficiente como para que envejezcan todos los segmentos de una conexión anterior, antes de establecer una nueva con el mismo extremo. Esto podría introducir retardos innecesarios.

d) ***Control de flujo:*** el mecanismo de control de flujo por medio de asignación de créditos descrito antes es bastante robusto en presencia de un servicio de red no seguro y requiere pocas mejoras. En general se tiene que el esquema de asignación de créditos está ligado a las confirmaciones de la siguiente forma: para confirmar segmentos y conceder créditos, una entidad de transporte envía un segmento de control de la forma (ACK N, CREDIT M) donde ACK N confirma todos los segmentos de datos hasta el número N y CREDIT M permite que los segmentos de números N + 1 hasta N + M sean transmitidos. Este mecanismo es bastante potente, ya que la pérdida de un segmento ACK/CREDIT tiene escaso impacto en el funcionamiento del esquema, ya que confirmaciones posteriores re-sincronizarán el protocolo. Además, si no hay nuevas confirmaciones en camino,

en el emisor expirará un temporizador y se retransmitirá un segmento de datos, lo cual disparará una nueva confirmación.

e) ***Establecimiento de la conexión:*** al igual que con la desconexión, el establecimiento debe tener en cuenta la no seguridad del servicio de red en el intercambio de SYNs.

☑ Dialogo en dos sentidos (two-way handshake): A emite un SYN a B y espera un SYN de vuelta como confirmación de la conexión. Pueden haber dos problemas: el SYN de A se puede perder o la respuesta de B se puede perder. Para ambos, la solución sería el uso de temporizadores de retransmisión, pero el nuevo problema potencial sería la aparición de datos duplicados debido a segmentos de datos obsoletos, como ejemplifica la gráfica de la hoja anterior. La solución podría ser empezar cada nueva conexión con un número de secuencia diferente, elegido lejos del último número utilizado en la conexión más reciente. Para nuevas conexiones, se podrían crear también problemas por SYNs duplicados como se plantea en la Ilustración 8:

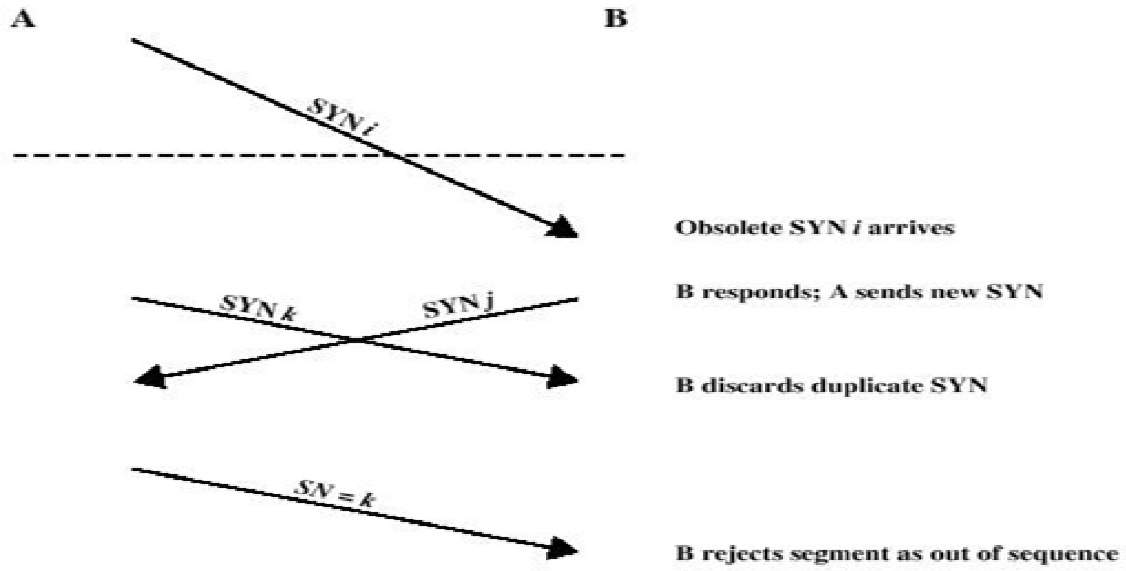
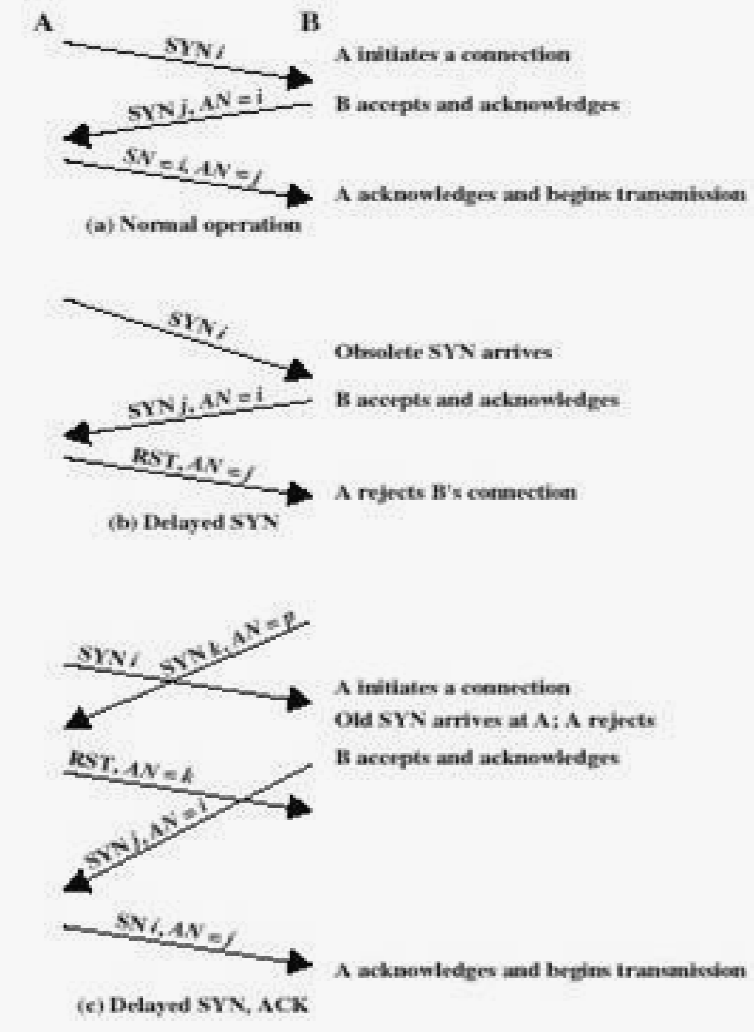


Ilustración 8: Diálogo en dos sentidos; problema con segmentos SYN viejos

☑ Dialogo en tres sentidos (three-way shake): para solucionar el problema anterior, cada lado confirma explícitamente el SYN y el número de secuencia del otro.(a): La entidad de transporte A inicia la conexión. El SYN de A incluye el número de secuencia de envío, i.El SYN de repuesta confirma el número e incluye el número de secuencia para el otro extremo. A confirma el SYN/ACK en su primer segmento de datos. (b): Un SYN X viejo llega a B después de cerrar la conexión relevante. B supone que es una petición nueva y responde con SYN j, ACK X. Cuando A recibe éste mensaje, se da cuenta que él no ha solicitado una conexión y por tanto envía un RST, ACK j. Hay que notar que la porción ACK j del mensaje RST es esencial para que un RST duplicado viejo no cancele un establecimiento de conexión legítimo. (c): un SYN/ACK viejo llega en medio del establecimiento de una nueva conexión. Debido al uso de números de secuencia en las confirmaciones, no hay perjuicio.

Ilustración 9: Ejemplo de diálogo en tres sentidos.



f) ***Cierre de la conexión:*** se adopta un mecanismo similar al de la conexión, mediante un diálogo en tres sentidos.

g) ***Recuperación de caídas:*** Cuando el sistema en el que una entidad de transporte se está ejecutando falla y consecuentemente re-arranca, la información de estado de todas las conexiones se pierden. Las conexiones afectadas llegan a estar "medio abiertas", ya que el lado que no se vio afectado por la caída no se ha dado cuenta todavía del problema.

- El lado todavía activo de la conexión medio abierta puede cerrar la conexión usando un temporizador de renuncia (mide el tiempo máximo después de las sucesivas retransmisiones de un mismo segmento, sin recibir respuesta). Superado el temporizador, cierra la conexión e indica un cierre no normal al usuario TS.
- Si se cancela una entidad de transporte y re-arranca rápidamente, esto se detecta y también se da un aviso de cierre no normal al usuario.

El usuario TS en el lado que no se cayó sabe cuantos datos ha recibido, pero el otro usuario no, si es que la información de estado se ha perdido. De ese modo, existe el peligro de que algunos datos de usuario se pierdan o dupliquen.

3.3) PROTOCOLOS DE TRANSPORTE:

El conjunto de protocolos TCP/IP incluye dos variantes de protocolos en la capa de transporte:

- **TCP** (Transmisión Control Protocol- Protocolo de control de la transmisión), que es un protocolo orientado a conexión.
- **UDP** (User Datagram Protocol-Protocolo datagrama de usuario), que es un un protocolo no orientado a conexión.

3.3.1) PROTOCOLO DE CONTROL DE TRANSMISIÓN (TCP)

Principales características:

- Especificado en la RFC 793, está diseñado para proporcionar una comunicación segura entre procesos (usuarios de TCP) paritarios a través de una gran variedad de redes seguras e inseguras, así como a través de un conjunto de redes interconectadas.
- TCP está orientado a conexión, es decir que los usuarios establecen una conexión extremo a extremo, intercambian luego un flujo de datos transmitido por TCP en segmentos y finalmente, se libera la conexión.
- Al igual que en IP, los servicios suministrados por TCP incluyen el concepto de primitivas y parámetros de servicio, brindando QoS ("Calidad de servicio"), aunque en el protocolo TCP dichos servicios son mucho mas ricos y complejos que los de capa de red.
- El protocolo utiliza para el **control de secuenciamiento** un mecanismo de la forma de "ventana deslizante" tal como HDLC, pero a diferencia de éste, separa la confirmación de datos recibidos del permiso para enviar más

  Este mecanismo se conoce como Esquema de Otorgamiento de Créditos

  Cada octeto de datos se considera que tiene un número de secuencia.

  TCP confirma la recepción de datos con un mensaje de la forma (A=i; W=j) donde:
  - se confirma la recepción de todos los octetos hasta i-1, se espera recibir i
  - se permite enviar una nueva ventana de datos (W= j octetos). Esto es: desde i hasta i+j+1

  El receptor confirmará los segmentos y otorgará más crédito, solo si tiene espacio disponible en buffer

  La tasa a la cual se envían segmentos está determinada por la tasa a la cual se reciben las confirmaciones de los segmentos enviados
- Como en los protocolos de capa 2, TCP incluye un mecanismo de **control de errores**

  No existe en TCP una confirmación de rechazo, tal como REJ o SREJ en HDLC

  TCP se basa en la confirmación positiva de la recepción y retransmite cuando la confirmación no llega dentro de un período determinado (RTO)
- El mecanismo de ventana deslizante provee una herramienta para apaciguar al transmisor, o sea efectuando un **control de la congestión**.
- TCP brinda dos servicios útiles para etiquetar datos:
  - *Cargar flujo de datos*: si un usuario detecta una interrupción lógica, puede requerir mediante una etiqueta, (PUSH), que TCP transmita todos los datos pendientes, aunque no se halla completado un segmento.
  - *Indicación de datos urgentes*: es una posibilidad que sirve para notificar al usuario TCP destino que el flujo de datos entrante contiene datos significativos o "urgentes", siendo responsabilidad de éste último realizar la acción adecuada.
- TCP está diseñado específicamente para trabajar con IP, por lo que algunos parámetros de usuario se pasan a través de TCP a IP para su inclusión en la cabecera IP y no en el segmento TCP. Entre los más importantes de éstos parámetros se tiene:
  - Prioridad: un campo de 3 bits.
  - Retardo normal/ bajo retardo.
  - Rendimiento normal/ rendimiento alto.
  - Seguridad normal/seguridad alta.
  - Protección: un campo de 11 bits.

Formato de la cabecera TCP:

TCP utiliza un único tipo de unidad de datos de protocolo, llamado segmento TCP. Ya que la cabecera debe servir para implementar todos los mecanismos del protocolo, ésta es mas bien grande, con una longitud mínima de 20 octetos

- Puerto origen (16 bits): punto de acceso del servicio origen.

- Puerto destino (16 bits): punto de acceso del servicio destino.

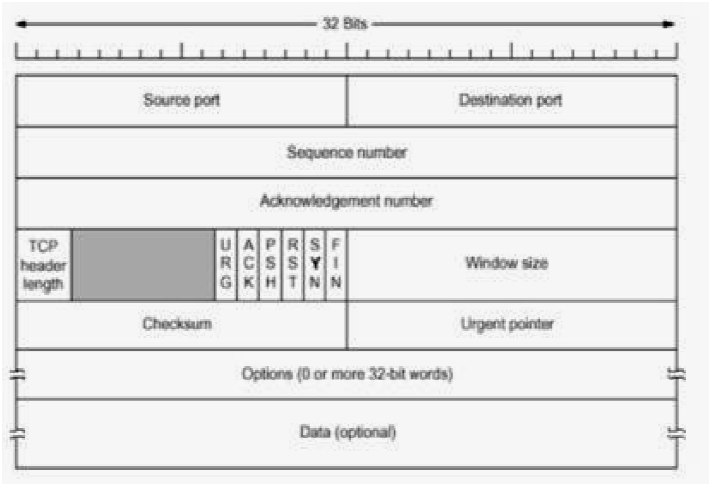


Ilustración 10: Cabecera TCP

- Numero de secuencia (32 bits): número de secuencia del primer octeto en éste segmento, excepto si el indicador SYN está presente. Si SYN está presente, es el número de secuencia inicial (ISN, "inicial sequence number") y en éste caso el primer octeto de datos es el ISN+1.
- Número de confirmación (32 bits): una confirmación incorporada ("piggybacking"). Contiene el número de secuencia del siguiente octeto que la entidad TCP espera recibir.
- Longitud de la cabecera (4 bits): número de palabras de 32 bits en la cabecera.
- Reservados (6 bits): bits reservados para un uso futuro.
- Indicadores (6 bits):
  - URG: el campo puntero urgente es válido.
  - ACK: el campo de confirmación es válido.
  - PSH: función de carga.(Push)
  - RST: puesta a cero de la conexión.(Reset)
  - SYN: sincronizar los números de secuencia.
  - FIN: el emisor no tiene más datos.
- Ventana (16 bits): asignación de créditos de control de flujo, en octetos. Contiene el número de octetos de datos comenzando con el que se indica en el campo de confirmación y que el que envía está dispuesto a aceptar.
- Suma de verificación (16 bits): el complemento a uno de la suma módulo 2^{16} -1 de todas las palabras de 16 bits en el segmento más una pseudo-cabecera.
- Puntero urgente (16 bits): señala el octeto que sigue a los datos urgentes. Esto permite al receptor conocer cuantos datos urgentes llegan.
- Opciones (variable): si está presente, solamente se define una opción, que especifica el tamaño máximo del segmento que será aceptado.

3.3.2) PROTOCOLO DATAGRAMA DE USUARIO (UDP)

Principales características:

- Especificado en la RFC 768, está diseñado para brindar un servicio no orientado a conexión ("connectionless") para los procedimientos de la capa de aplicación.
- Por ser un servicio no seguro, la entrega y protección contra duplicados no está garantizada.
- Se reduce la información suplementaria del protocolo, lo que puede ser adecuado en muchos casos en el contexto de gestión de red.
- UDP está situado encima de IP, con reducidas funciones para efectuar, esencialmente incorporar un direccionamiento a puerto a las capacidades de IP.

Formato de la cabecera:

Ilustración 11: Cabecera UDP

- La cabecera incluye un puerto origen y un puerto destino.
- El campo longitud contiene la longitud del segmento UDP entero, incluyendo la cabecera y los datos.
- La suma de verificación es el mismo algoritmo usado para TCP e IP. En UDP, la suma de verificación se aplica al segmento UDP entero, más una pseudo-cabecera incorporada a la cabecera UDP cuando se calcula la suma y es la misma que la usada para TCP. Si se detecta un error, el segmento se descarta sin tomar ninguna medida adicional.
  El campo de suma de verificación en UDP es opcional. Si no se utiliza, éste se pone todo a cero. Sin embargo, hay que indicar que la suma de verificación de IP se aplica solo a la cabecera IP y no al campo de datos, que está compuesto, en éste caso de la cabecera UDP y los datos de usuario. De ese modo, si UDP no implementa ningún cálculo de suma de verificación, los datos de usuario no se comprueban.