El Nivel de Transporte

Fundamentos de Redes de Ordenadores

GSyC
Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación

Universidad Rey Juan Carlos

Noviembre 2022

Contenidos

Contenidos

Introducción

- El Nivel de Transporte se encarga de gobernar el acceso múltiple a la red de los diversos procesos de la misma máquina que quieran usarla: En TCP/IP se hace a través de puertos.
- Hay dos protocolos que ofrecen un servicio de nivel de transporte:
  - UDP: no orientado a conexión y no fiable
  - TCP: orientado a conexión y fiable

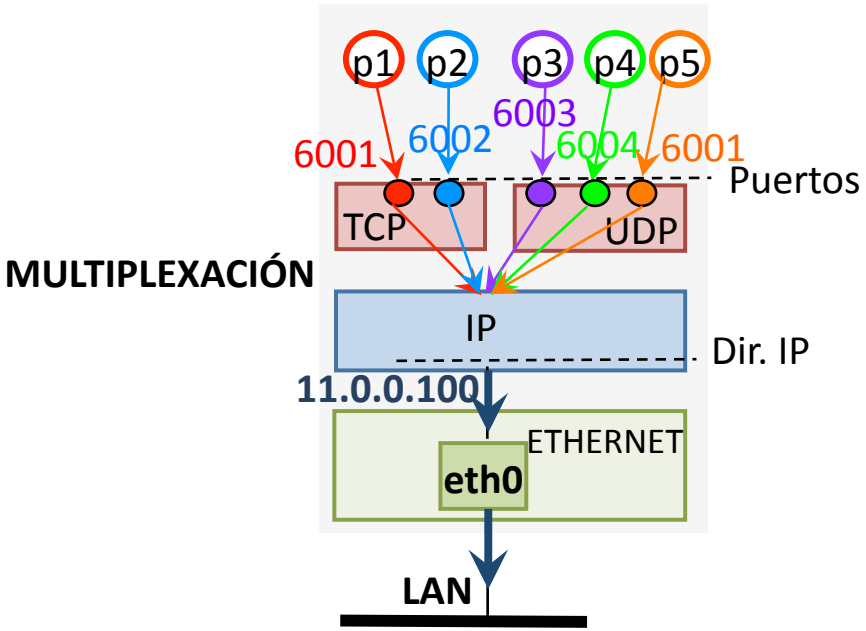
Puertos

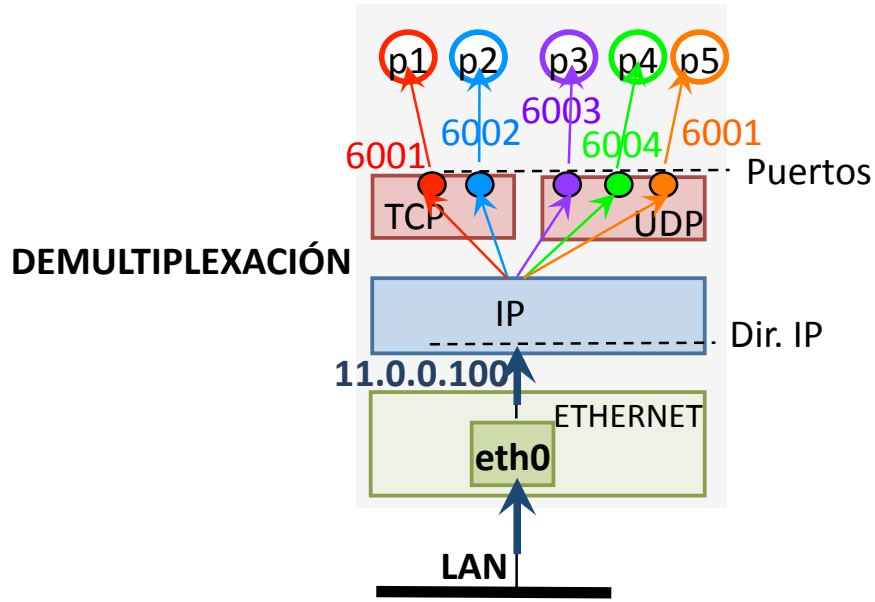
- En una máquina multiproceso, el nivel de transporte debe preocuparse de saber a qué proceso va destinado una unidad de datos de UDP o TCP que acaba de llegar por la red. La dirección IP no es suficiente. Por ello, los procesos utilizan "direcciones de nivel de transporte", denominadas puertos.
- Cada puerto del Nivel de Transporte proporciona a una aplicación un punto de acceso a la red de comunicaciones, con lo que ésta puede dialogar con otra aplicación situada en un puerto de una máquina remota.

Contenidos

Puertos

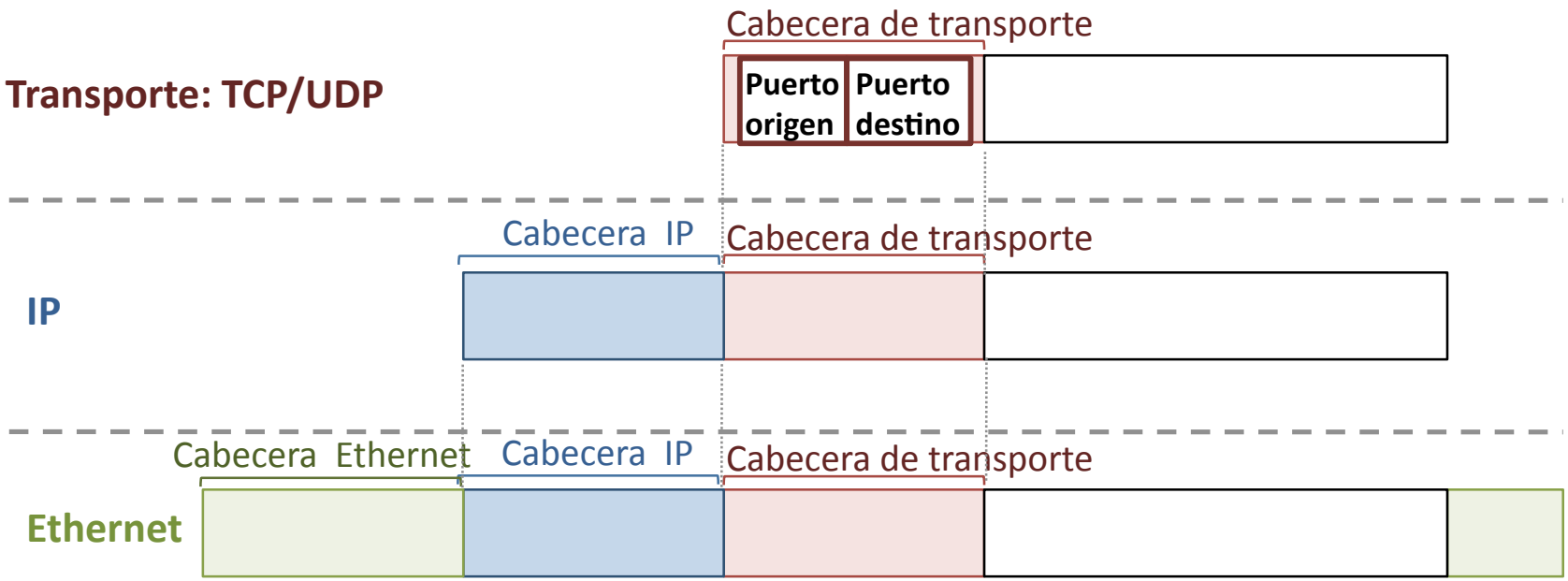
- El Nivel de Transporte TCP/IP:
  - multiplexa las unidades de datos que envían las aplicaciones a través de los puertos, encapsulándolas en unidades de datos de UDP o TCP
  - demultiplexa las unidades de datos de UDP y TCP, pasando los datos a las aplicaciones.
- Los puertos se identifican por un número de 16 bits. Los puertos UDP y TCP se manejan por separado:
  - el puerto 6001 UDP y el puerto 6001 TCP son puertos distintos.



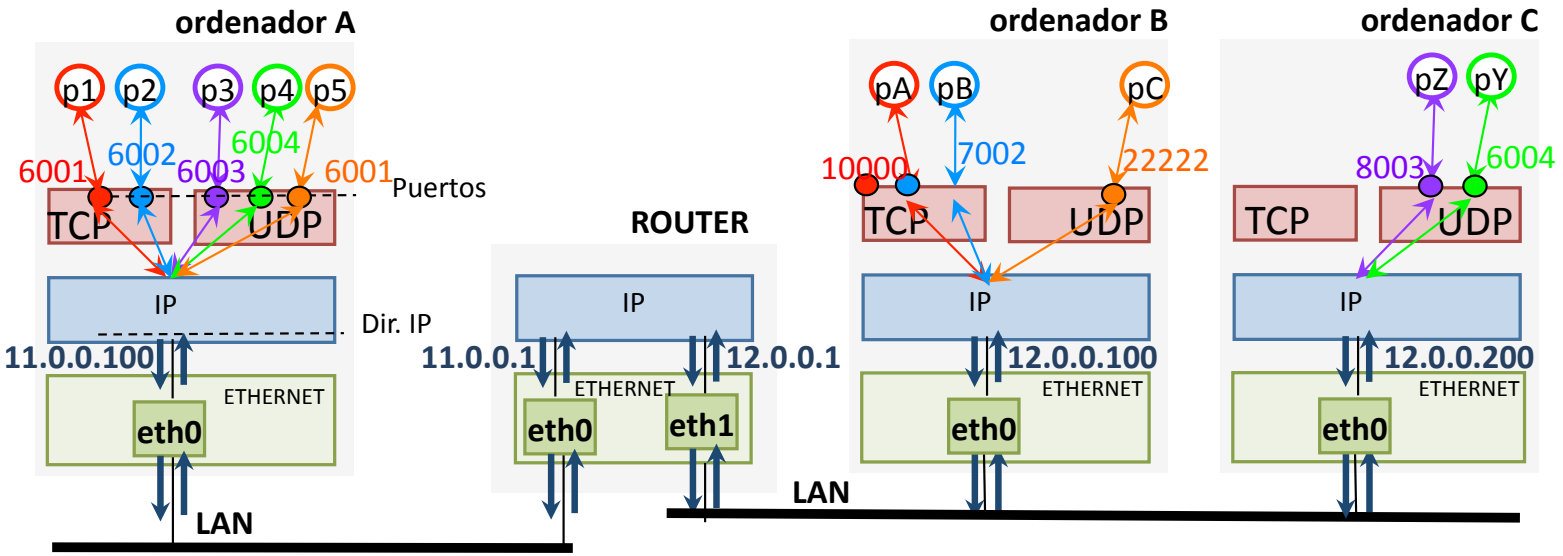


Puertos en la cabecera del nivel de transporte

- Cuando se envía una unidad de datos del nivel de transporte se especifica el puerto de destino y el puerto desde el que se envía.



Ejemplo de comunicaciones en el nivel de transporte



Comunicaciones desde el ordenador A	Protocolo	Dirección IP origen	Puerto origen	Dirección IP destino	Puerto destino
⟷	TCP	11.0.0.100	6001	12.0.0.100	10000
⟷	TCP	11.0.0.100	6002	12.0.0.100	7002
⟷	UDP	11.0.0.100	6003	12.0.0.200	8003
⟷	UDP	11.0.0.100	6004	12.0.0.200	6004
⟷	UDP	11.0.0.100	6001	12.0.0.100	22222

- Una comunicación de nivel de transporte queda completamente identificada con los siguientes parámetros: protocolo, dirección IP origen, puerto origen, dirección IP destino y puerto destino.

Puertos reservados

- Los puertos menores que 1024 (puertos privilegiados) están reservados y asignados universalmente a aplicaciones de red conocidas.
  - En una máquina Unix esta asignación está en el fichero `/etc/services`:

```
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
ftp-data        20/tcp          # default ftp data port
ftp             21/tcp
ssh             22/tcp          # SSH Remote Login Protocol
ssh             22/udp
telnet          23/tcp
smtp            25/tcp          mail
www             80/tcp          http    # WorldWideWeb HTTP
www             80/udp                  # HyperText Transfer Protocol
```

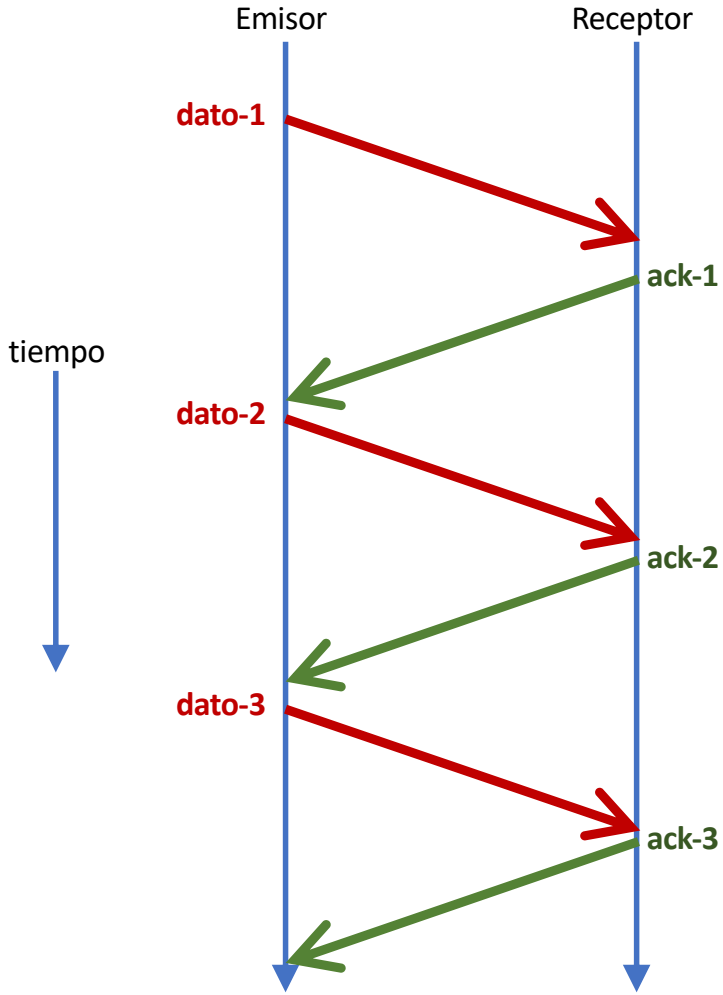
Contenidos

Protocolos de retransmisión

- Fiabilidad en el **nivel de enlace**: retransmisión de tramas perdidas o descartadas por errores de transmisión entre dos vecinos. Opcional en TCP/IP.
- Fiabilidad en el **nivel de transporte**: retransmisión de paquetes que faltan en el receptor (puede deberse a tramas perdidas/descartadas en el nivel de enlace o a paquetes perdidos por congestión de algún *router*).
- En TCP/IP el primer protocolo fiable está en el nivel de transporte, ya que Ethernet e IP son no fiables.
  - Excepción: Los protocolos de enlace Wifi con frecuencia son fiables
- Los protocolos que proporcionan fiabilidad se basan en asentimientos (ACKs) y plazos para detectar los paquetes que no tiene el receptor y por tanto hay que retransitir.

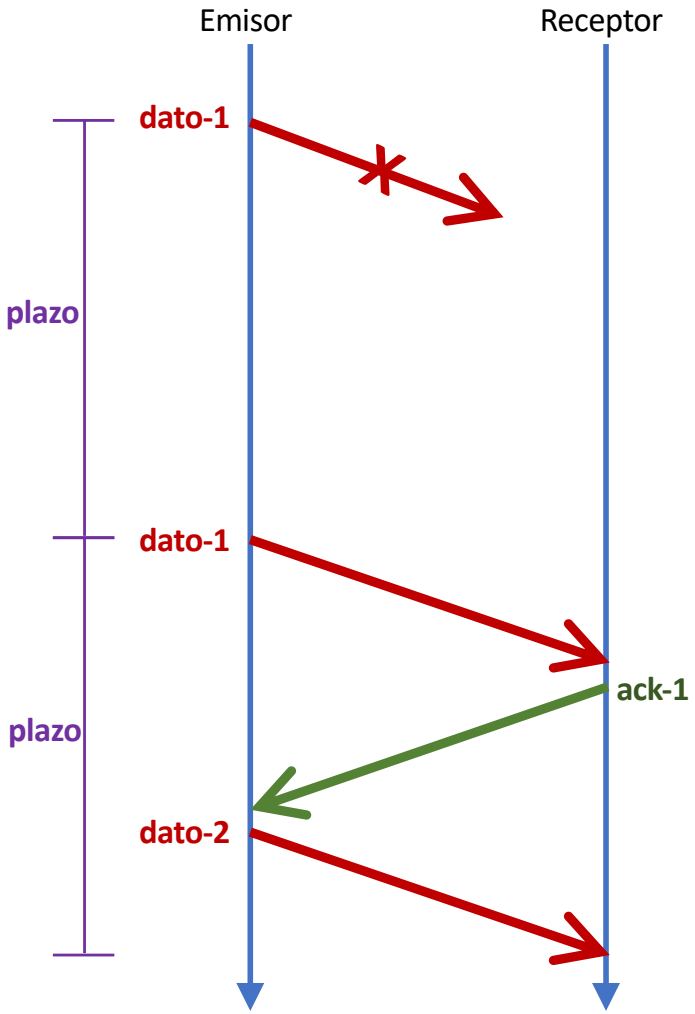
Protocolo de Parada y Espera (I)

Mecanismo básico: El emisor, después de enviar cada paquete de datos, para y espera a que le llegue su asentimiento por parte de receptor:



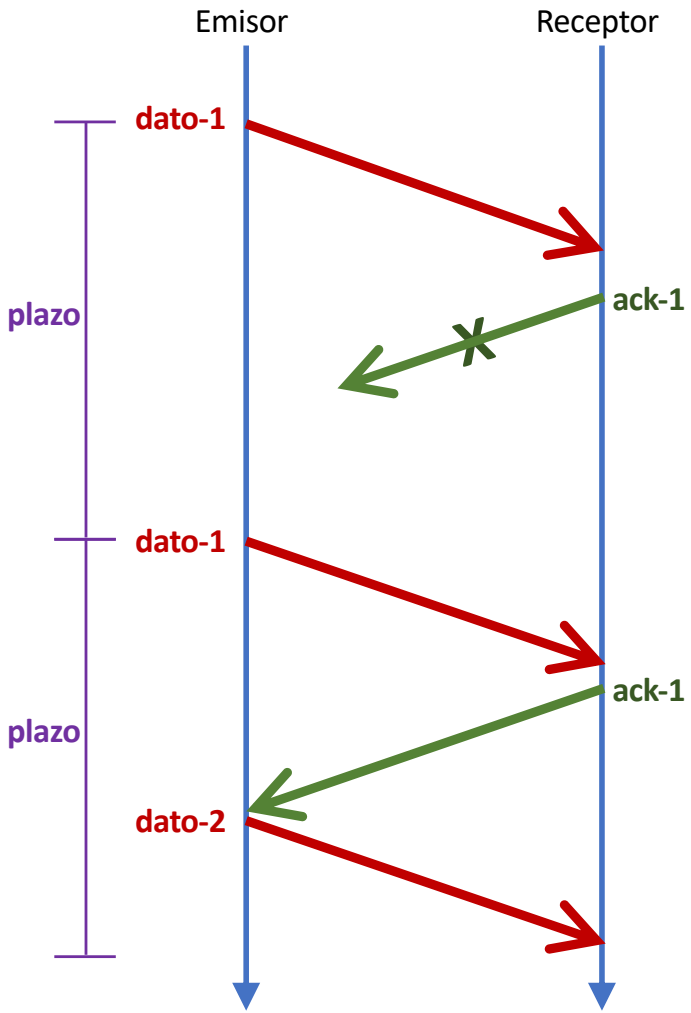
Protocolo de Parada y Espera (II)

- Cada vez que se envía un paquete de datos se arranca un temporizador (plazo).
- Si no se recibe el ACK pasado el plazo, se retransmite el paquete de datos.
- Es importante elegir un tamaño de plazo adecuado



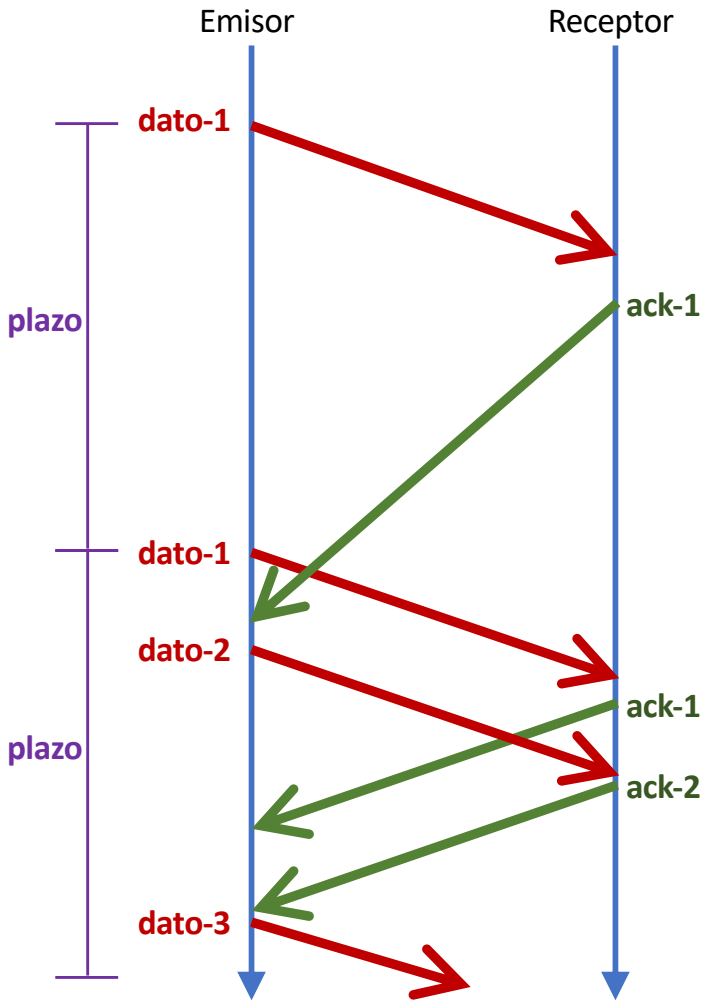
Protocolo de Parada y Espera (III)

- Si se pierde un ACK, también hay que retransmitir
- El receptor detecta que el paquete es un duplicado del anterior y lo descarta, pero debe enviar el ACK



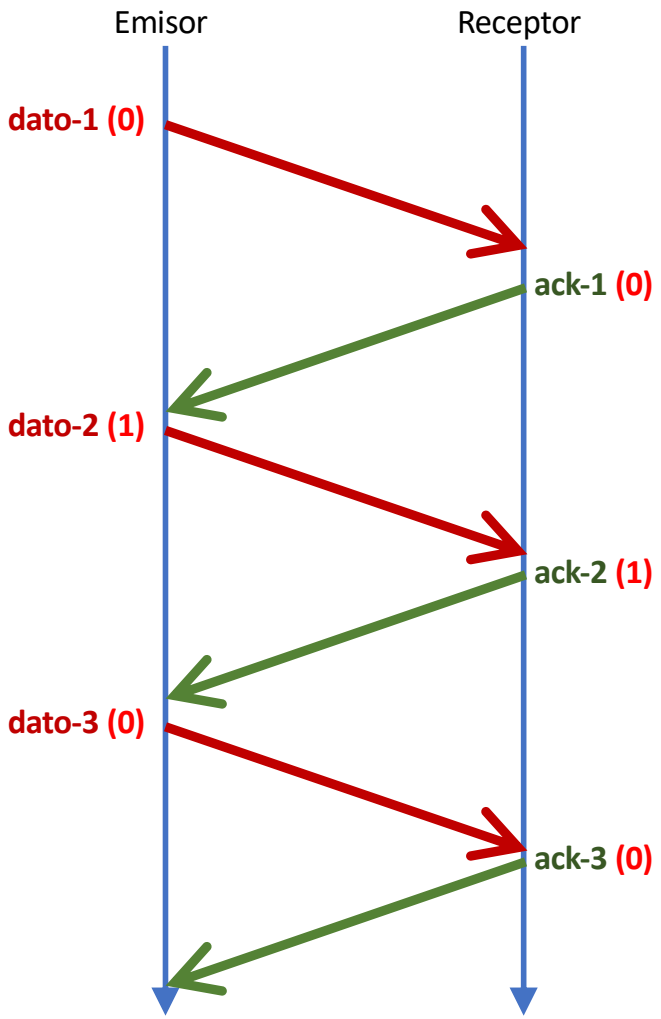
Protocolo de Parada y Espera (IV)

- Si un ACK se retrasa, habrá también retransmisión.
- En cuanto llega el ACK retrasado ya puede transmitirse el paquete siguiente
- El ACK duplicado no puede confundirse con el ACK del nuevo paquete, y debe ignorarse.



Protocolo de Parada y Espera (V)

- Los paquetes de datos y de acks se numeran con un campo de cabecera
- En realidad basta con usar un bit (0/1) para numerar datos y acks, por esto al protocolo también se le conoce como "protocolo del bit alternante".

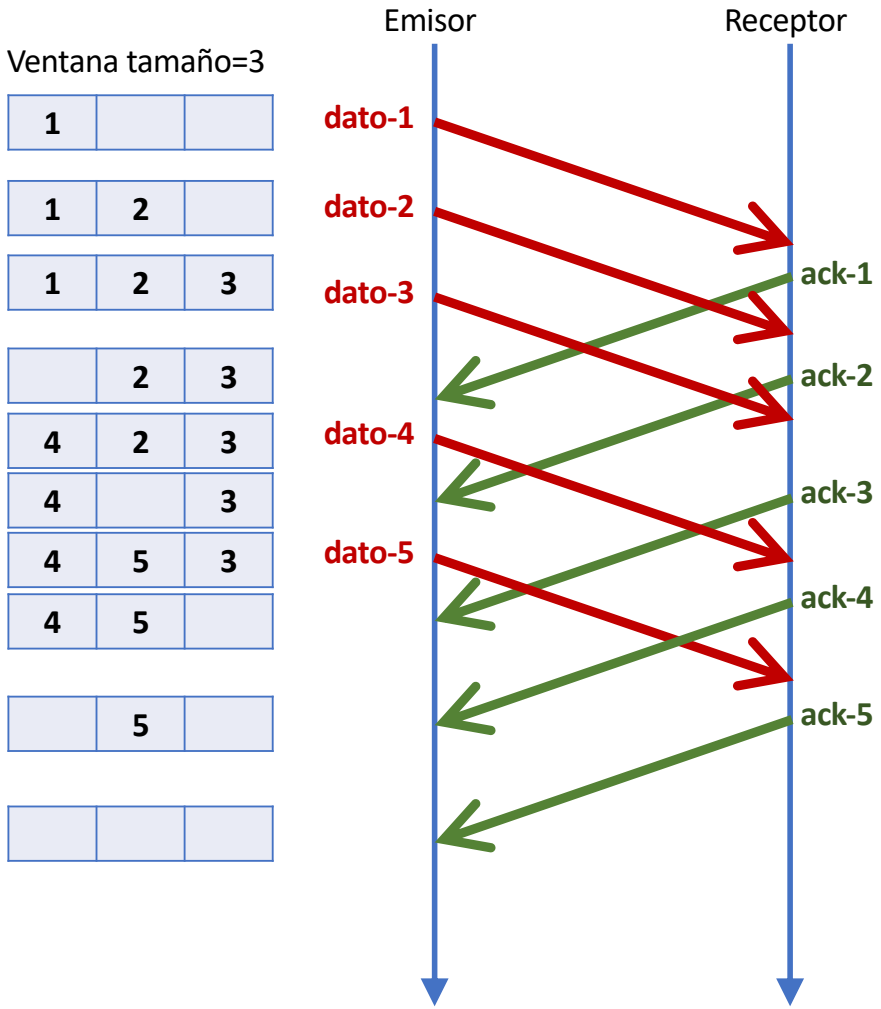


Protocolo de Ventana (I)

- El protocolo de Parada y Espera es muy lento: aunque no se pierda ningún paquete, es un freno la espera sin progresar hasta recibir cada ACK.
- Podría aprovecharse el tiempo de espera hasta que llegue el ACK para ir enviando nuevos paquetes de datos, todos numerados para distinguir unos de otros (el campo de cabecera ya tendrá que tener varios bits).
- Protocolo de Ventana de tamaño N: El emisor puede enviar un máximo de N paquetes antes de parar a la espera de que lleguen los ACK.
- Normalmente es el receptor el que fija el tamaño de ventana que debe usar el emisor.
- Un protocolo de Parada y Espera puede verse como un protocolo de Ventana de tamaño 1.

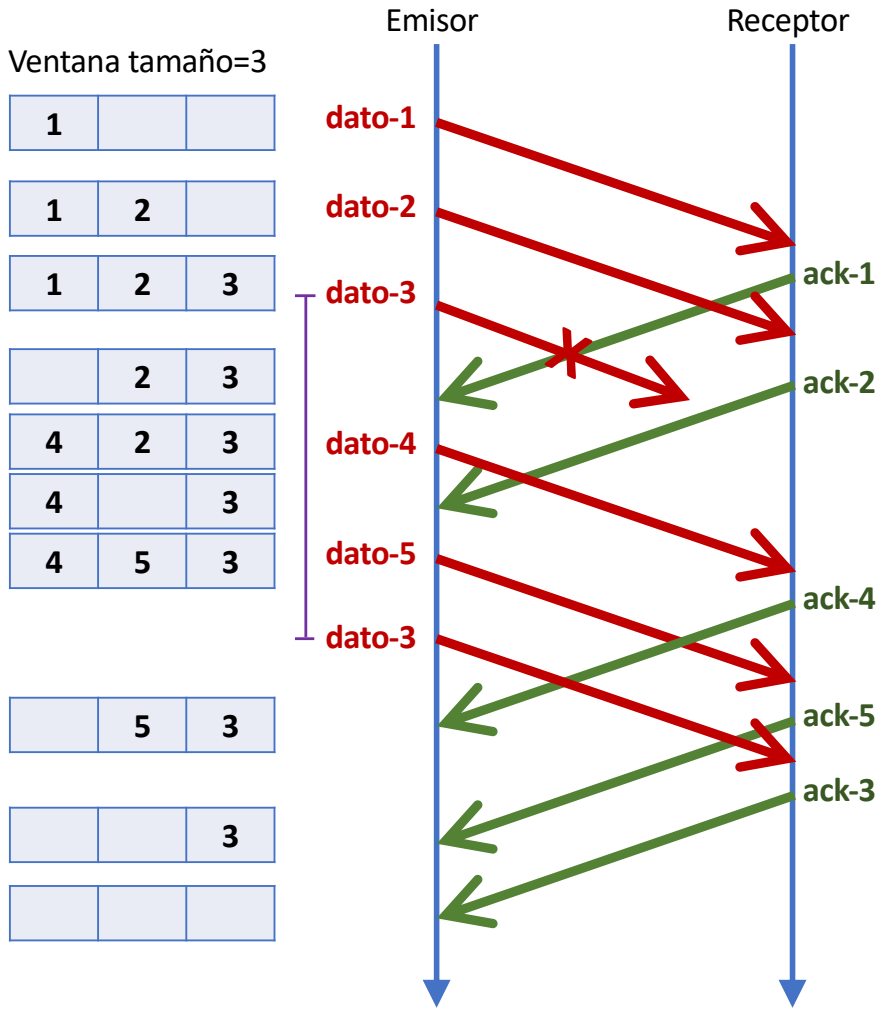
Protocolo de Ventana (II)

- Cada vez que se envía un nuevo paquete de datos, se añade a la ventana
- Cada vez que llega un ACK, se elimina de la ventana el paquete correspondiente
- Si la ventana se llena, hay que parar de transmitir hasta que llegue ACKs.



Protocolo de Ventana (III)

- Cada vez que se envía un paquete de datos se arranca un plazo
- Si vence el plazo asociado a un paquete, se retransmite



Contenidos

Modelos de interacción extremo-a-extremo

- Comunicación extremo-a-extremo: Con este nombre nos referimos al intercambio de mensajes extremo-a-extremo, que puede darse en un protocolo de transporte, en un protocolo de nivel de aplicación, o directamente en una aplicación (por encima de todos los niveles).
- Modelo Cliente/Servidor:
  - Entre los participantes en la comunicación se distinguen dos roles claramente diferenciados:
    - **Clientes**: los que **piden** un servicio
    - **Servidores**: los que **ofrecen** el servicio
  - El código (sw) que ejecuta un cliente es diferente del que ejecuta el servidor.
  - Ejemplo: clientes web (piden páginas web), servidores web (tienen las páginas web). La interacción entre un cliente web y un servidor web sigue el Modelo Cliente/Servidor.
- Modelo Entre Iguales o P2P *Peer-to-Peer*:
  - Todos los participantes en la comunicación, **Peers**, tienen el mismo rol.
  - Todos los Peers ejecutan el mismo código (sw).
  - En ocasiones se dice que cada Peer a ratos "se comporta como cliente" y a ratos "se comporta como servidor".
  - Ejemplo: sistemas P2P de intercambio de ficheros.

Arranque de aplicaciones cliente/servidor

- Al arrancar una aplicación que funciona como **servidor**, ésta se quedará esperando a recibir mensajes de un determinado protocolo de nivel de transporte y en un determinado puerto.
- Al arrancar una aplicación que funciona como **cliente**, ésta tomará la iniciativa de enviar el primer mensaje a la aplicación servidor utilizando el protocolo de nivel de transporte que está usando la aplicación servidor y enviando los mensajes a la dirección IP y puerto en los que está escuchando la aplicación servidor.
- Es necesario arrancar primero la aplicación que funciona como servidor y posteriormente arrancar la aplicación que funciona como cliente.

Contenidos

Referencias

- C. M. Kozierok, **The TCP/IP guide: A Comprehensive Illustrated Internet Protocols Reference**: Cap. 28. Disponible on-line:

  `http://www.tcpipguide.com/free/t_IPNetworkAddressTranslationNATProtocol.htm`

- J. F. Kurose, K. W. Ross, **Computer Networking: A Top-Down Approach (4th ed)**: Cap. 3 (3.1), Cap. 4 (4.4).