

Capa de Transporte

Redes de Computadores

FIEC04705

Sesión 19

Agenda

- Terminología
- TCP
- Control de congestión

Terminología

Terminología

- **Throughput** en una red es el número de paquetes pasando a través de la red en una unidad de tiempo.
- **El desempeño de la red** viene dado por dos factores que miden el desempeño de la red: **delay and throughput**.

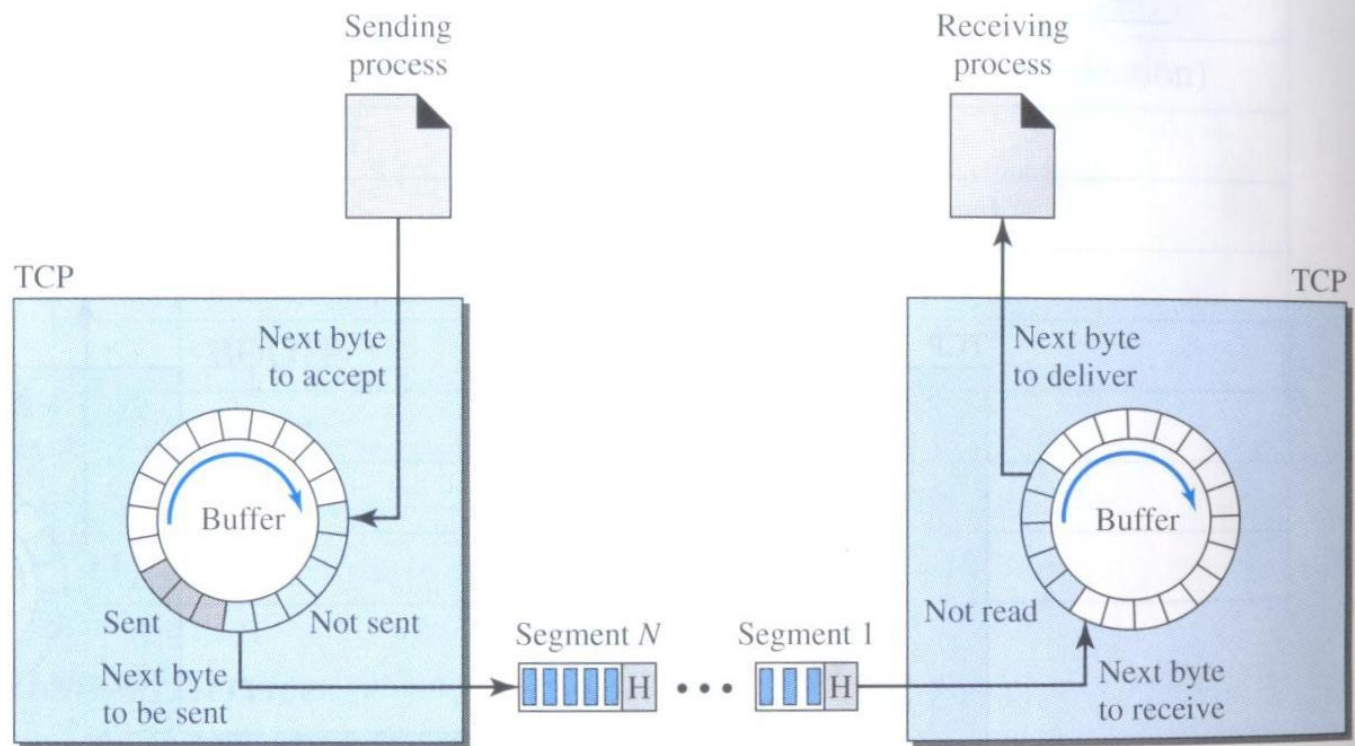
TCP

TCP

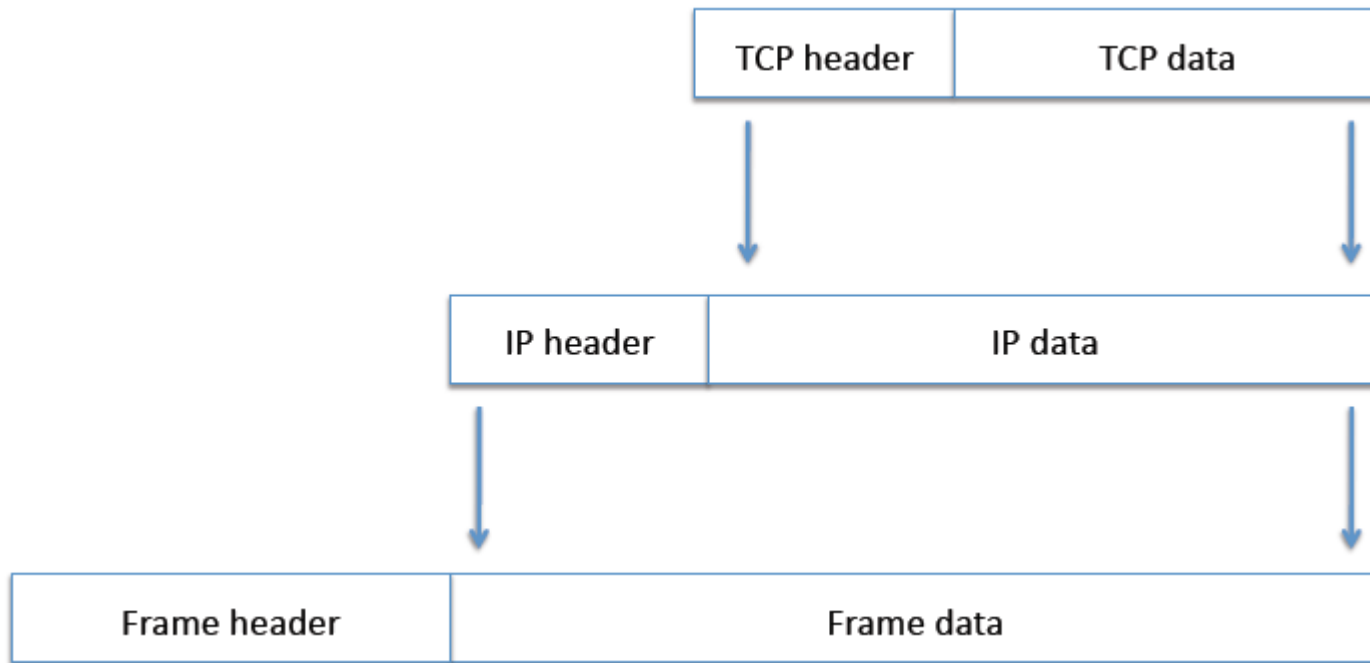
- **Transmission Control Protocol (TCP)** es otro protocolo de capa de transporte.
- Es un protocolo confiable y orientado a conexión.
- Crea una conexión virtual entre dos TCPs para enviar datos.
- TCP utiliza mecanismos de control de flujo y errores a nivel de la capa de transporte.

Segmento TCP

Figure 23.15 TCP segments



TCP encapsulation



Características de TCP

- **Sequence number:** define el número del primer byte de datos contenido en el segmento.
- **Acknowledgment number:** define el número del próximo byte esperado por el host.
- **Control de flujo:** el receptor controla la cantidad de datos que van a ser enviados por el transmisor.
- **Control de errores:** considerando el segmento como la unidad de datos para detección de errores, TCP implementa un mecanismo de control de errores.
- **Control de congestión:** la cantidad de datos enviados por el transmisor no es solamente controlada por el receptor (control de flujo), sino que además es determinado por el nivel de congestión en la red.

Segmento TCP

0	4	8	12	16	20	24	28	31
Source port				Destination port				
Sequence number								
Acknowledgment number								
Hdr len	Reserved	Flags		Window				
Checksum				Urgent pointer				
Options (if <code>hdr_len > 5</code>)							Padding	
Data								

TCP flags

Figure 23.17 *Control field*

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push

RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

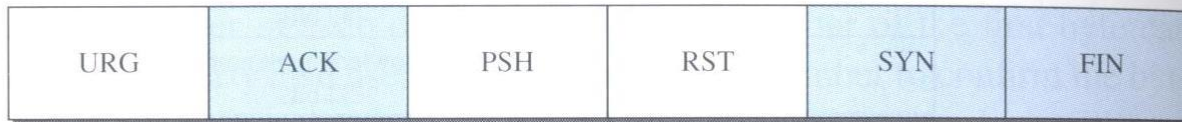


Table 23.3 *Description of flags in the control field*

Flag	Description
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

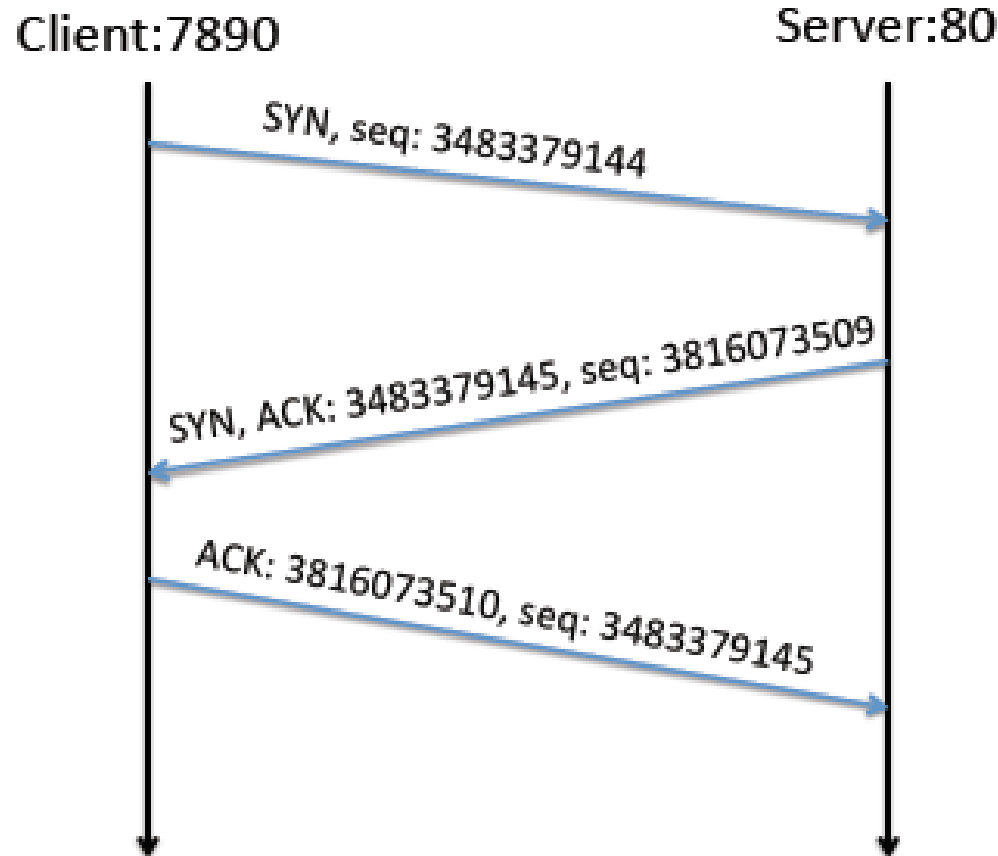
Conexión TCP

- Estableciendo de la conexión
 - Three-Way handshaking
- Transferencia de datos
 - Pushing Data (los datos deben ser enviados al programa tan pronto como sea posible)
 - Urgent Data
- Terminación de la conexión
 - Three-Way handshaking
 - Half close

Estableciendo la conexión

- La configuración es hecha a través del three-way handshake
- El cliente envía un SYN al servidor; sequence number es I_A
- El servidor contesta con SYN-ACK; el ack es definido como $I_A + 1$; el sequence number es I_B
- El cliente envía ACK; el ack es establecido a $I_B + 1$; el sequence number es $I_A + 1$

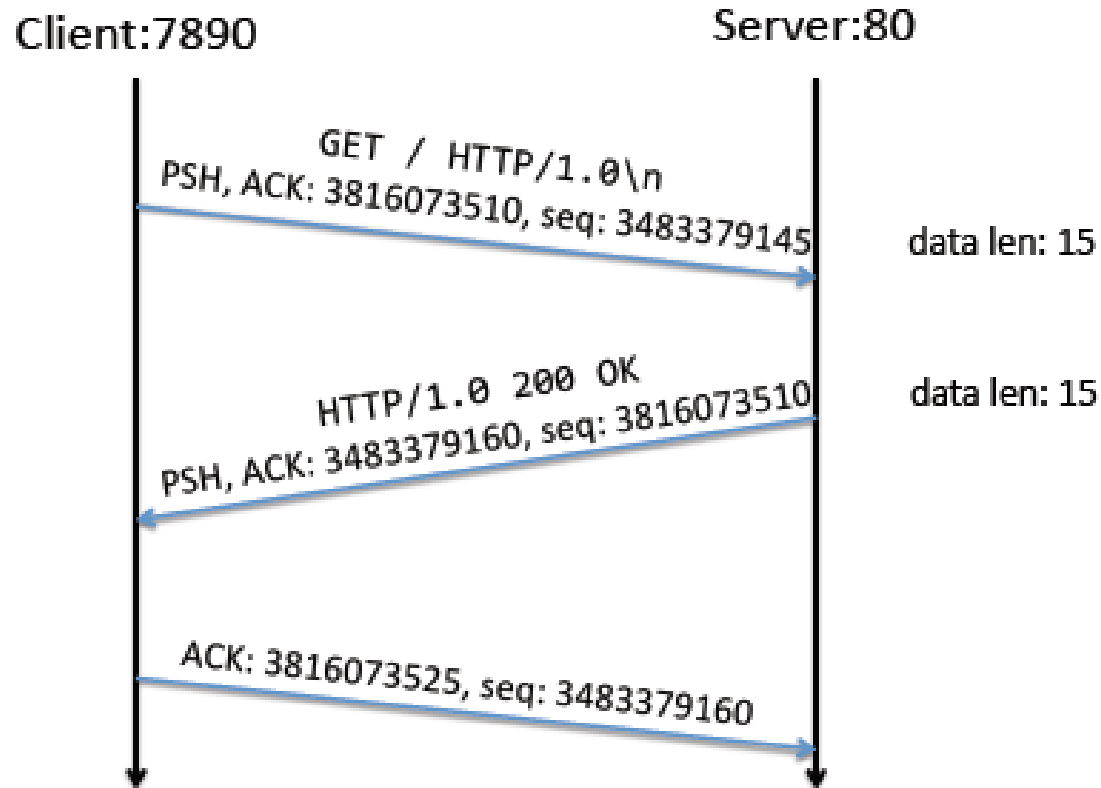
Initial sequence number



Initial sequence numbers

- Qué utilizar como el número de secuencia inicial?
- El standard original especifica que el sequence number debe ser incrementado cada 4 microsegundos
- BSD UNIXes inicialmente usaron un número que es incrementado por 64000 cada medio segundo y por 64000 cada vez que una conexión es establecida
- *Veremos al final de curso si estas son buenas opciones.*

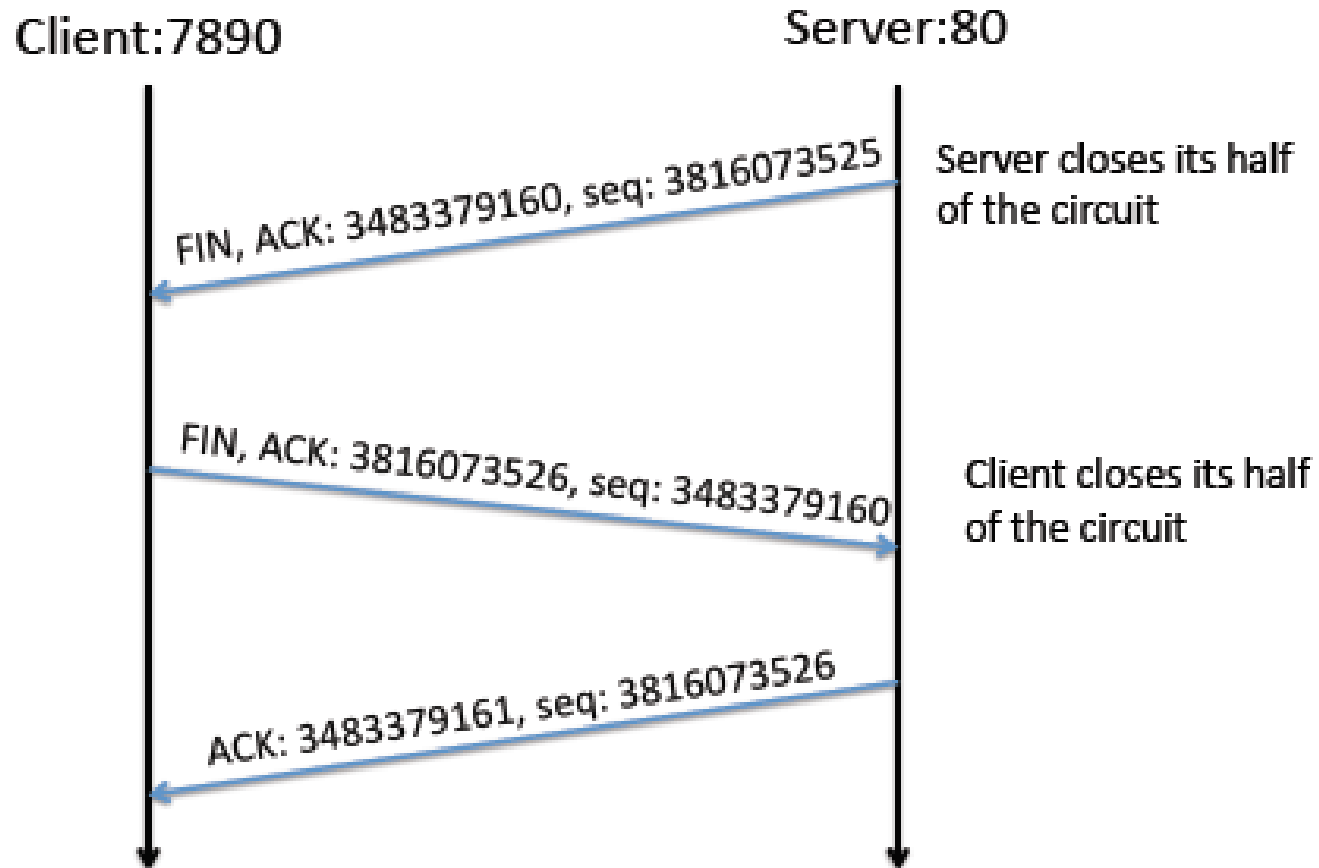
Transferencia de datos



Terminando la conexión

- Uno de los hosts, digamos el servidor, termina su flujo por medio del envío de un segmento con el flag establecido a FIN.
- El otro host, el cliente, confirma (acknowledges) la recepción
- En este punto, el servidor no enviará más datos
 - Solo envía ACKs para el dato que recibe
- Cuando el cliente termina su flujo, el circuito virtual es cerrado.

Terminando la conexión



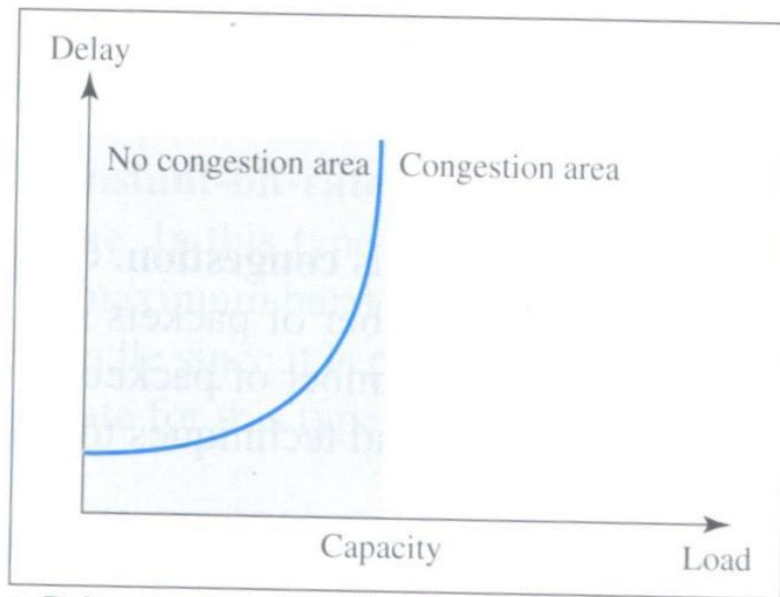
Control de congestión

Congestión

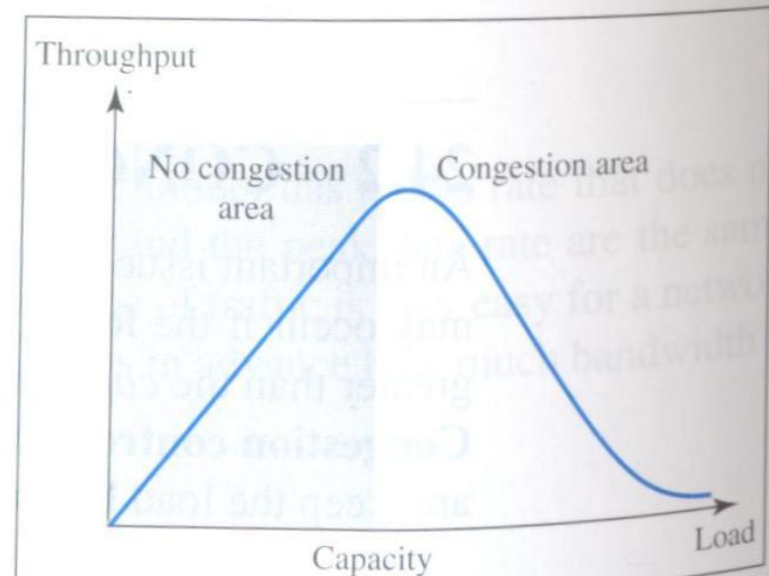
- Un aspecto importante en redes conmutadas por paquete es la **congestión**.
- La congestión en una red puede ocurrir cuando la **carga** de la red - el número de paquetes enviados a la red - es mayor que la capacidad de la misma - el número de paquetes que la red puede manejar
- **El control de congestión** se refiere a las técnicas y mecanismos para controlar la congestión y mantener la carga por debajo de la capacidad

Congestión: desempeño de la red

Figure 24.4 *Packet delay and throughput as functions of load*



a. Delay as a function of load



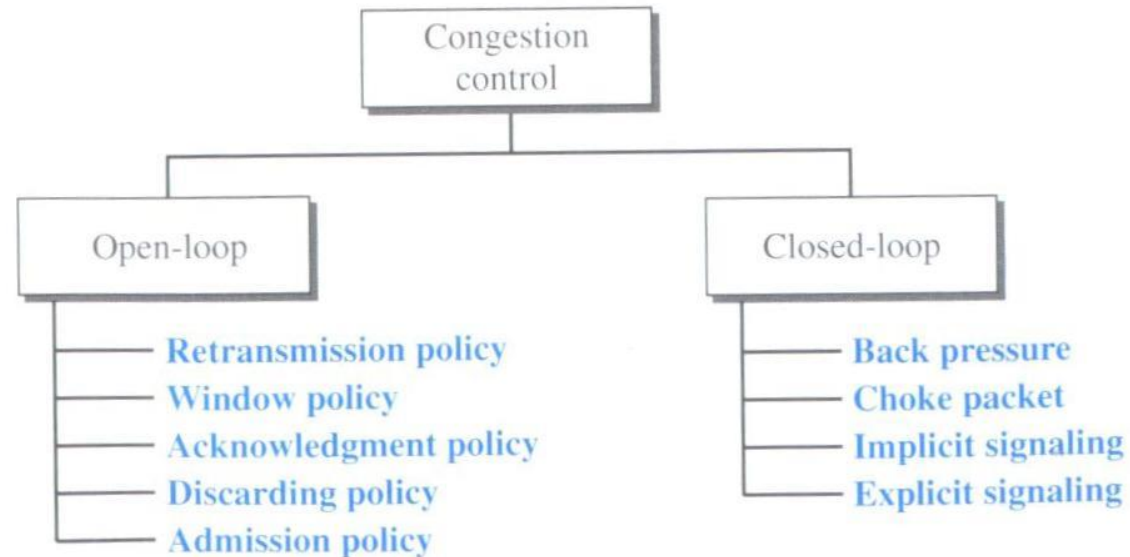
b. Throughput as a function of load

Control de congestión

- Control de congestión se refiere a las técnicas y mecanismos para prevenir o terminar con la congestión. Se dividen en dos categorías:
 - Open-loop congestion control (prevención)
 - Closed-loop congestion control (terminación)

Categorías del control de congestión

Figure 24.5 *Congestion control categories*



Control de congestión Open-Loop

- Políticas son aplicadas para prevenir la congestión. El control viene dado por el origen o el destino. Entre las políticas tenemos:
 - **Retransmission Policy:** Retransmission timers para optimizar la eficiencia.
 - **Window Policy:** Selective Repeat Window
 - **Acknowledgment Policy:** Enviar menos acknowledgments. Por ejemplo, no confirmando cada paquete recibido sino cada N
 - **Discarding Policy:** los routers pueden descartar paquetes menos sensibles
 - **Admission Policy:** Chequear el requerimiento del recurso antes de admitirlo en la red (redes de circuitos virtuales)

Control de congestión Closed-Loop

- Mecanismo que tratan de **aliviar** la congestión una vez que esta sucede. Algunos mecanismos utilizados por los diferentes protocolos son:
 - **Backpressure**: nodo congestionado deja de recibir datos desde los nodos inmediatos en el flujo
 - **Choke Packet**: el router envía un aviso de congestión directamente a la estación origen.
 - **Implicit Signaling**: no hay comunicación entre los nodos congestionados y el nodo origen. El origen supone que hay congestión
 - **Explicit Signaling**: el nodo que detecta la congestión envía un mensaje al origen o destino.
 - **Backward signaling**: en sentido opuesto a la congestión
 - **Forward signaling**: en la dirección de la congestión.

Control de congestión Closed-Loop

Figure 24.6 *Backpressure method for alleviating congestion*

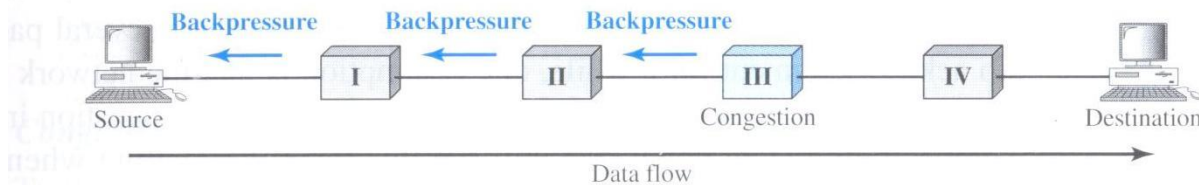
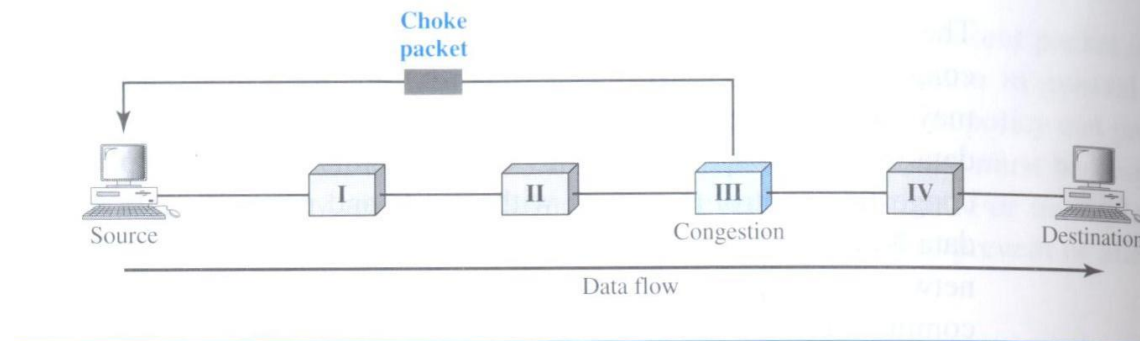


Figure 24.7 *Choke packet*



Puntos para recordar

- TCP connection
- Three-Way handshing
- Control de congestión

Próxima Sesión

- Programación con sockets