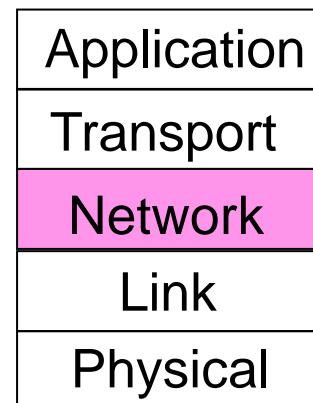


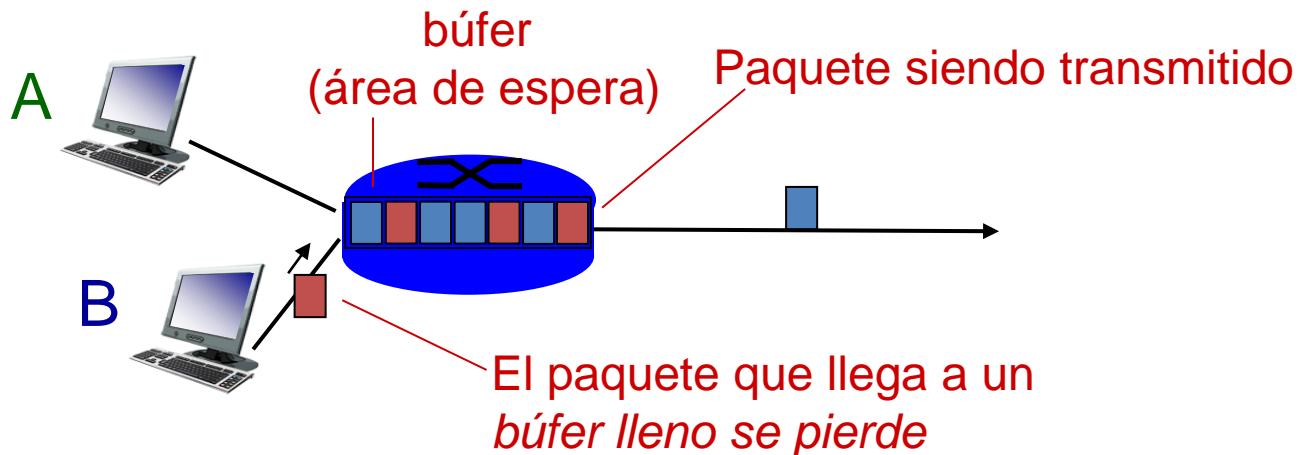
# Capítulo 4

## La Capa de Red Control de congestión



# Congestión

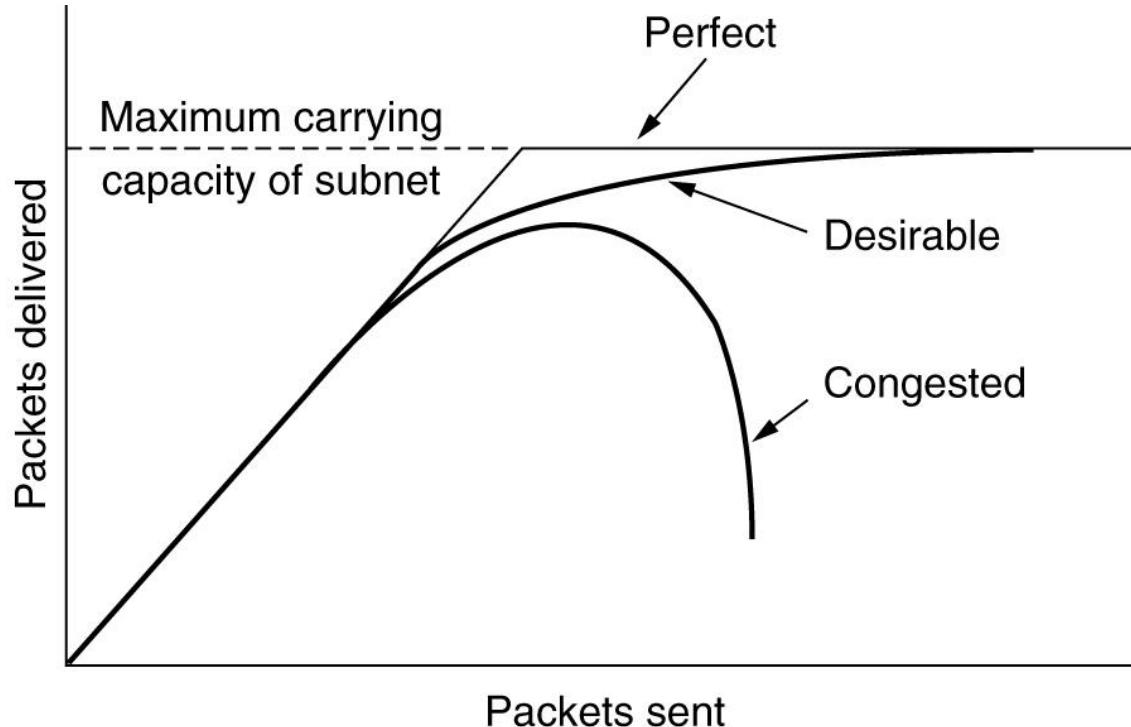
- La cola en un búfer que precede a un enlace tiene capacidad finita.
- ¿Qué pasa con un paquete cuando llega a una línea de salida con buffer lleno?
  - El paquete que llega a un **búfer lleno** se pierde.
  - Los paquetes perdidos deben ser retransmitidos por el enrutador previo o el host emisor.



# Congestión

- Si comienzan a llegar *muchos* paquetes por algunas líneas de entrada y todas necesitan la misma línea de salida,
  - Se irán acumulando los paquetes en una cola.
  - Si no hay suficiente memoria para almacenar todos los paquetes, muchos de ellos se perderán.

# Congestión



- Cuando hay demasiado tráfico, surge la congestión y el desempeño se degrada rápidamente
- La meta del control de congestión es
  - ❑ asegurar que la subred sea capaz de transportar el tráfico ofrecido

# Congestión

- **Problemas de los algoritmos de control de congestión de TCP estudiados**
  - El host de destino demora demasiado en enterarse de congestión (solo por expiración de temporizador de retransmisiones o 3 acks duplicados).
  - Los hosts solo se enteran de pérdidas de paquetes, no pueden controlar qué paquetes perder y cuáles no.

# Congestión

- **Razones para estudiar control de congestión en la capa de red.**
  - Para resolver los problemas de los protocolos de control de congestión de TCP mencionados.
  - Para entender las medidas que pueden tomar los enrutadores para detectar la congestión y colaborar con la capa de transporte para ayudar a controlar mejor la congestión.

# Congestión

- **Tipos de soluciones a la congestión que vienen a la mente:**
  - aumentar los recursos
  - disminuir la carga en la subred (nos concentraremos en esto)
- **Formas de disminuir la carga en la subred:**
  - Regulación del tráfico**
    - hacer que hosts responsables de la congestión se enteren más rápido (que con protocolos de TCP) de la congestión y reduzcan su tasa de transferencia
  - Desprendimiento de carga**
    - Enrutadores descartan paquetes inteligentemente antes que se saturen búferes.

# Aprenderemos

- **Agenda:**

1. **Cómo identificar la congestión**

- **Para que los enrutadores puedan tomar algunas medidas y colaborar.**

2. Cómo regular el tráfico

- Método de paquetes reguladores
- Método de paquetes reguladores salto por salto
- Método de bit de advertencia y su implementación ECN

3. Cómo desprenderse de carga

- Algoritmo de detección temprana aleatoria

# Identificación de la Congestión

- **Problema:** ¿Cómo puede hacer un enrutador para darse cuenta si tiene algún puerto de salida congestionado?
- **Solución:** Cada enrutador monitorea la **demora de la cola de línea de salida**.
  - Asociar a cada línea:  $d = \text{demora reciente de cola de esta línea}.$
  - Tomar periódicamente una muestra de la **longitud de cola instantánea de la línea**,  $s$
  - Actualizar  $d$  periódicamente usando:

$$d_{nvo} = a \cdot d_{ant} + (1 - a) * s$$

donde  $a$  determina la rapidez con que el enrutador olvida la historia reciente.

# Identificación de la Congestión

- Siempre que  $d$  rebasa un **umbral**, la línea de salida entra un **estado de advertencia**.
  - Cada paquete nuevo que llega se revisa para ver si su línea de salida está en estado de advertencia.
  - Si es así, se realiza alguna acción.

# Aprenderemos

- **Agenda:**

1. Cómo identificar la congestión

- Para que los enrutadores puedan tomar algunas medidas y colaborar.

2. Cómo regular el tráfico

- Método de paquetes reguladores
- Método de paquetes reguladores salto por salto
- Método de bit de advertencia y su implementación ECN

3. Cómo desprenderse de carga

- Algoritmo de detección temprana aleatoria

# Regulación de Tráfico

- **Regulación de tráfico** es cuando los emisores ajustan sus transmisiones para enviar un tráfico que la red pueda soportar.
- La congestión se da en los enrutadores (y no en los hosts).
- **¿Cómo se puede enterar un host de que hay congestión?**
  - Se le avisa de la congestión.
- **Una vez que un enrutador tiene una línea de salida en estado de advertencia puede**
  - avisar a los hosts responsables de los paquetes que llegan a esa línea congestionada.

# Regulación de Tráfico

- **Método de paquetes reguladores.**
  1. Usar **paquetes reguladores** si línea de salida en estado de advertencia.
    - El enrutador regresa un **paquete regulador (PR)** al host de origen, proporcionándole el destino encontrado en el paquete.
  2. **Para que el paquete original no genere más PR más adelante en la ruta**
    - en el paquete original se activa un bit del encabezado y después se reenvía.
  3. **El PR** le pide al host de origen que reduzca en un porcentaje X el tráfico enviado al destino especificado.

# Regulación de Tráfico

- **Método de paquetes reguladores (cont).**
  4. El host ignora los PR que se refieran a ese destino por un intervalo fijo.
  5. Una vez que haya expirado ese tiempo, el host escucha más PR durante un intervalo  $I$ .
    - Si llega alguno el host reduce el flujo aún más y comienza a ignorar nuevamente los PR.
    - **Si no llega ningún PR durante  $I$**  el host incrementa el flujo.
- Un ejemplo de paquetes reguladores se usó en la internet temprana (se llamaba Quench) por los 80s.

# Regulación de Tráfico

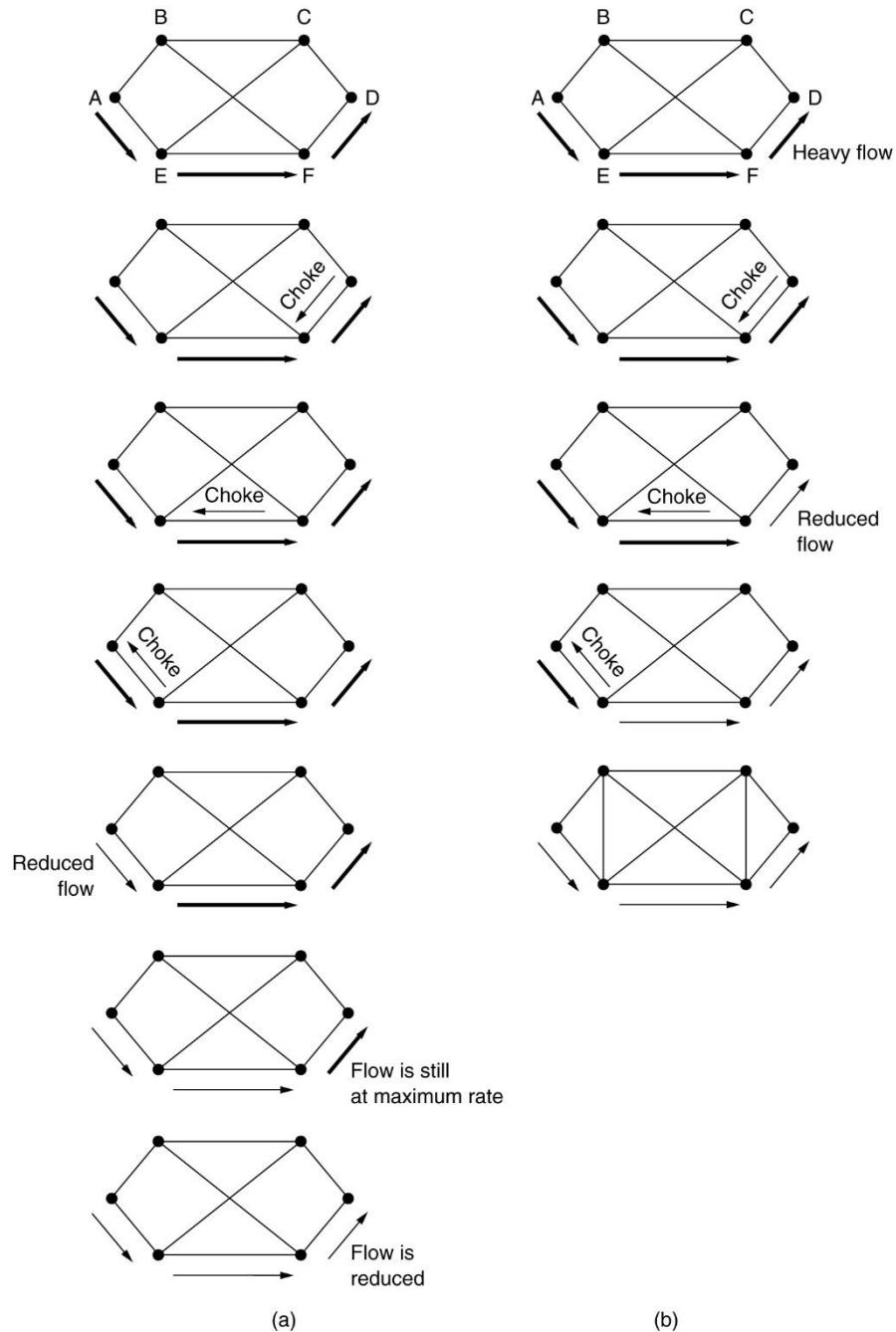
- **Problema del método de paquetes reguladores:**
  - A altas velocidades o distancias grandes, el envío de un paquete regulador a los hosts de origen no funciona bien porque la reacción es muy lenta.

# Regulación de Tráfico

- **Solución: Método de Paquetes reguladores de salto por salto.** Hacer que el paquete regulador ejerza su efecto en cada salto que da.
  - Cuando el paquete regulador llega a un enrutador F, se le obliga a F a reducir el flujo al siguiente enrutador D (F deberá destinar más búferes al flujo).
  - Luego el paquete regulador llega al enrutador E anterior a F e indica a E que reduzca el flujo a F. Esto impone una mayor carga a los búferes de E, pero da un alivio inmediato a F. Y se sigue así sucesivamente.

# Paquetes Reguladores Salto por Salto

(a) Un paquete regulador que afecta solo al origen.



# Regulación de Tráfico

- **Método de bit de advertencia.** Señalar el estado de advertencia activando un bit especial en el encabezado del paquete.
  - Cuando el paquete llega a su destino, la entidad transportadora copia el bit en la siguiente confirmación de recepción que se regresa al origen.
  - A continuación el origen reduce el tráfico.
  - Mientras el enrutador está en estado de advertencia, continua activando el bit de advertencia, lo que significa que el origen continua obteniendo confirmaciones de recepción con dicho bit activado.

# Regulación de Tráfico

- ❑ El origen monitorea la fracción de confirmaciones de recepción con el bit activado y ajusta su tasa de transmisión de manera acorde.
  - En tanto los bits de advertencia continúan fluyendo, el origen continua disminuyendo su tasa de transmisión.
- ❑ Cuando la tasa de transmisión disminuye lo suficiente, el origen incrementa su tasa de transmisión.
  - Debido a que cada enrutador a lo largo de la ruta puede activar el bit de advertencia, el tráfico se incrementa solo cuando no había enrutadores con problemas.

# Regulación de Tráfico

- Una implementación de bit de advertencia usada por TCP es **ECN (Explicit Congestion Notification)**:
  - Se usa en TCP/IP.
  - Se marcan 2 bits en el encabezado IP con distintos fines:
    - 00: transporte no capaz de ECN
    - 10: transporte capaz de ECN, **ECT(0)**
    - 01: transporte capaz de ECN, **ECT(1)**
    - 11: congestión encontrada, **CE**
  - Si ambos extremos soportan ECN mandan sus paquetes con ECT(0) y ECT(1) respectivamente.
  - Si paquete atraviesa cola congestionada y el enrutador soporta ECN, se cambia código en el paquete a CE para avisar al receptor de la congestión.

# Regulación de Tráfico

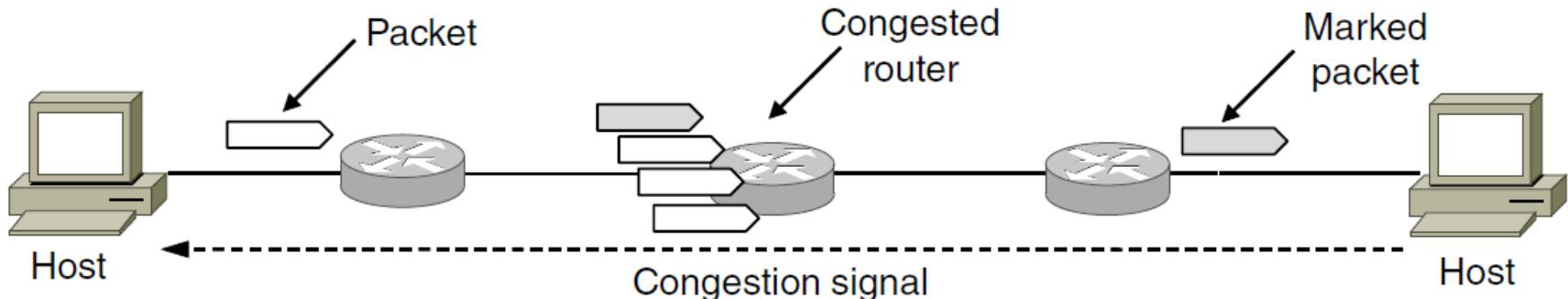
- **ECN continuación:**
  - El uso de ECN en conexión TCP es opcional.
  - Para usar ECN, debe ser negociado al establecer conexión TCP incluyendo opciones adecuadas en segmentos SYN y SYN-ACK.
  - Se usan dos banderas en encabezado TCP para soportar ECN:
    - **ECE (ECN echo):** se usa para mandar indicación de congestión al emisor.
    - **CWR (ventana de congestión reducida):** es usada para confirmar que la indicación ECE fue recibida.

# Regulación de Tráfico

- **Secuencia de ejecución de ECN típica:**

1. Se negocia ECN en conexión TCP
2. Emisor manda paquete IP  $P$  con ECT(0)
3.  $P$  llega a enrutador congestionado que soporta ECN y enrutador marca  $P$  con CE.
4. Receptor recibe  $P$  con CE y manda segmento  $Q$  (con ACK de  $P$ ) de vuelta usando bandera ECE prendida.
5. Emisor recibe  $Q$  con ECE prendido, entonces emisor reduce ventana de congestión.
6. Emisor manda siguiente segmento al otro extremo usando bandera CWR prendida para confirmar recepción de aviso de congestión.

**Nota:** Se continúa transmitiendo segmentos con ECE prendido hasta recibirse segmento con CWR prendido.



# Aprenderemos

- **Agenda:**

1. Cómo identificar la congestión

- Para que los enrutadores puedan tomar algunas medidas y colaborar.

2. Cómo regular el tráfico

- Método de paquetes reguladores
- Método de paquetes reguladores salto por salto
- Método de bit de advertencia y su implementación ECN

3. Cómo desprenderse de carga

- Algoritmo de detección temprana aleatoria

# Desprendimiento de Carga

- **Situación:** En el esquema anterior cuando se satura una línea de salida de un enrutador, se pierden paquetes indiscriminadamente.
- **Problema:** evitar la pérdida descontrolada de paquetes.
- **Conviene preocuparse de esto porque:**
  - No todos los paquetes tienen la misma importancia.
  - Por eso es mejor controlar qué paquetes se descartan.
- **Solución 1:** descartar paquetes inteligentemente antes de que se ocupe todo el espacio de búfer
  - cuando hay estado de advertencia en una línea de salida.

# Desprendimiento de Carga

- **Algunos criterios para escoger qué paquetes descartar:**
  - ❑ **Según el tipo de aplicación que se está usando.**
    - **Estrategia Vino:**
      - Descartar primero los paquetes más nuevos.
      - P.ej. en la transferencia de archivos.
    - **Estrategia Leche:**
      - Descartar primero los paquetes más viejos.
      - P.ej. en multimedia.
  - ❑ **Según la importancia de los paquetes.**
    - Marcar los paquetes con **clases de prioridades.**
    - Los enrutadores primero se desprenden de paquetes de la clase más baja, luego los de la siguiente clase, etc.

# Desprendimiento de Carga

- Se suele usar desprendimiento de carga junto con reducción de tráfico.
  - La respuesta a paquetes perdidos por desprendimiento de carga es que el origen disminuya su tasa de transferencia.
  - Si expira el temporizador de retransmisiones, el emisor lo toma como pérdida de paquete.
  - Vemos ahora una implementación de esta solución.

# Desprendimiento de Carga

- **Implementación: Algoritmo de detección temprana aleatoria (RED).**
  - Para detectar cuándo comenzar a descartar paquetes, los enrutadores mantienen un **promedio móvil de sus longitudes de cola**.
  - **Cuando este promedio de una cola  $C$  sobrepasa el umbral**
    - Una **pequeña fracción** de los paquetes son descartados al azar.
  - **Con cada uno de esos paquetes:**
    1. El enrutador **elige un paquete al azar** de  $C$ .
    2. Se descarta el paquete seleccionado.
    3. El origen notará falta de ACK y la **capa de transporte** disminuirá la velocidad de transmisión.
- **Consecuencias de elegir paquetes al azar:**
  - hace más probable que los hosts emisores más rápidos pierdan un paquete, lo noten, y reduzcan su tasa de transferencia.

# Desprendimiento de Carga

- El método RED se usa en internet cuando los hosts no pueden recibir señales explícitas de congestión.
  - Tanembaum dice que la mayoría de los hosts de internet no reciben mensajes explícitos de congestión de los enrutadores.