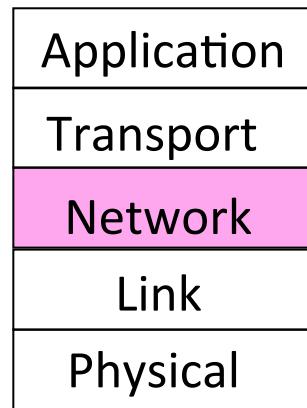


Nivel de Red de Internet

Nivel de Red (1)

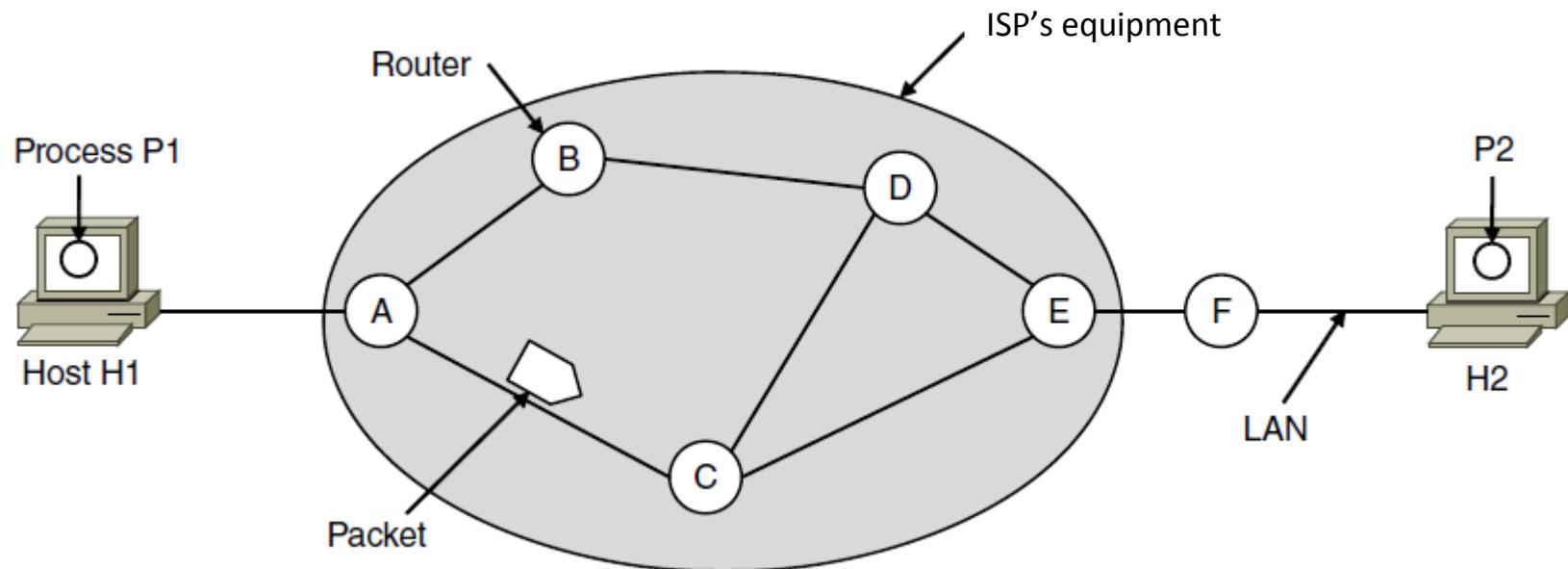
— Nivel de Red

- Responsable de entregar paquetes entre puntos finales a través de múltiples enlaces



Nivel de Red (2)

- Nivel de Red
 - Subred compuesta de routers

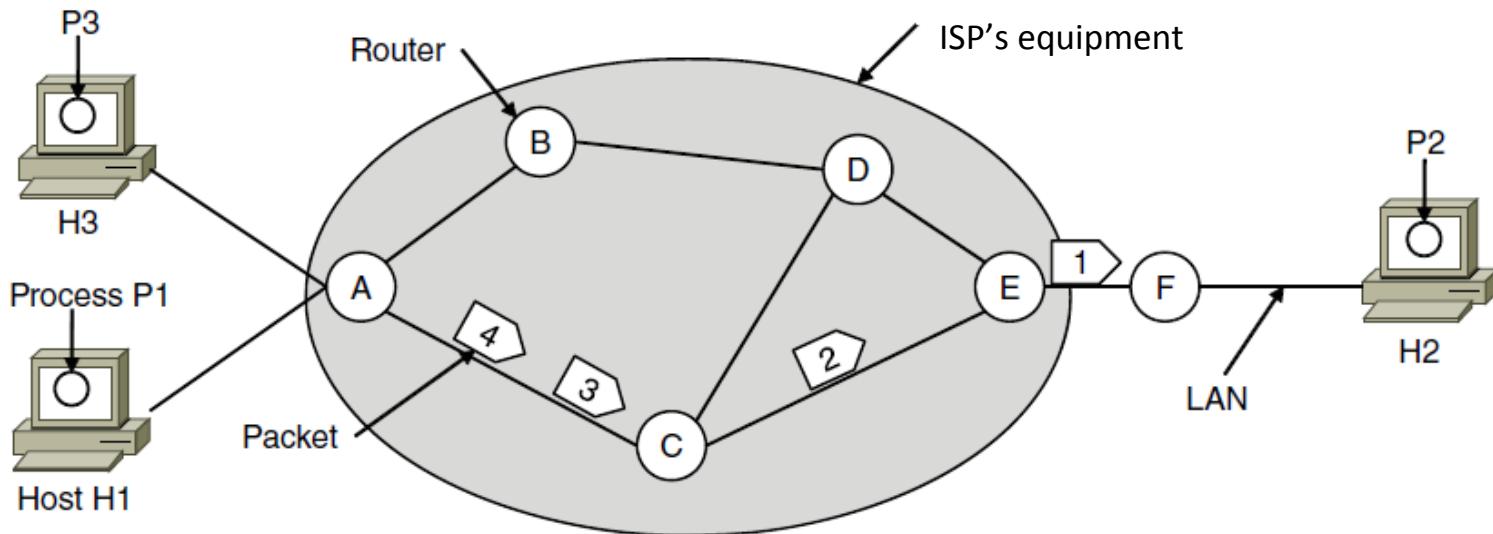


Nivel de Red (3)

- Principales aspectos del Nivel de Red
 - Interfaz Host-Subred
 - Direccionamiento
 - Enrutamiento
 - Adaptación de tamaños de paquetes
 - Prevención de situaciones de Congestión
 - Control de tráfico
 - Trata de evitar que la red se “caiga”

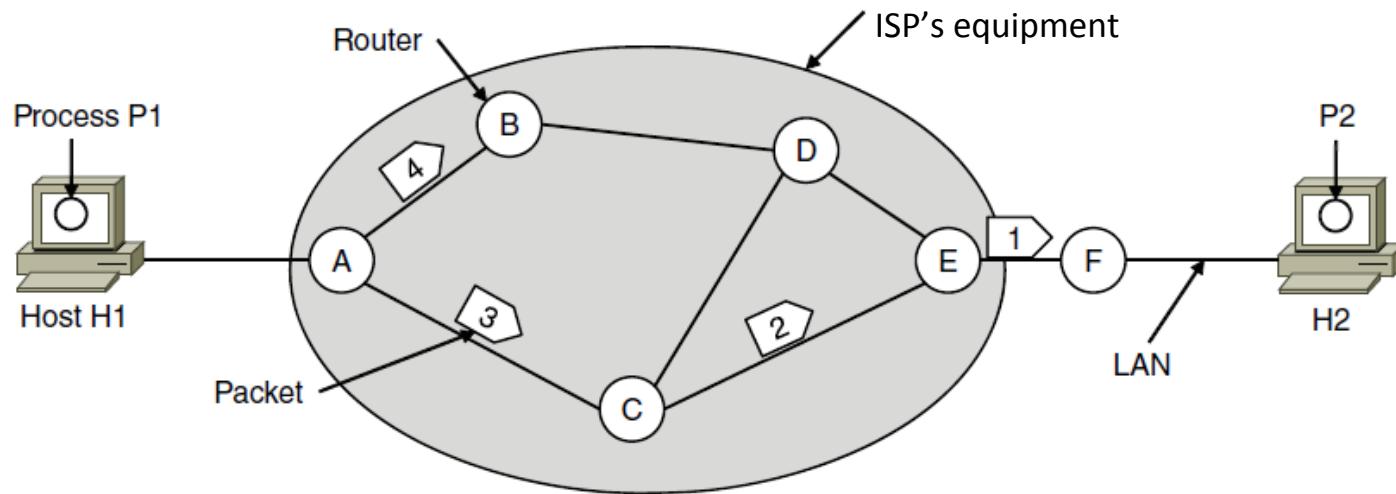
Nivel de Red (4)

- Servicios de Nivel de Red
 - Servicio con conexión



Nivel de Red (5)

- Servicios de Nivel de Red
 - Servicio sin conexión



Nivel de Red (6)

- Principio de Funcionamiento de Subredes
 - Store & Forward
 - Algoritmos de enrutamiento
 - Circuito Virtual
 - » Cada paquete usa la misma ruta
 - Datagrama
 - » Cada paquete descubre una ruta

Circuito Virtual (1)

- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”
 - Se utiliza un número de Circuito Virtual (CV) y una tabla de CV
 - La numeración CV es local
 - El número de CV se escoge como el menor disponible (**localmente**) hacia ese destino
 - La tabla de CV se construye durante la fase de inicialización (**primera fase de la conexión**)
 - Cuando se finaliza la conexión (**tercera fase**) se debe de indicar para poder reutilizar los números de secuencia de CV

Circuito Virtual (2)

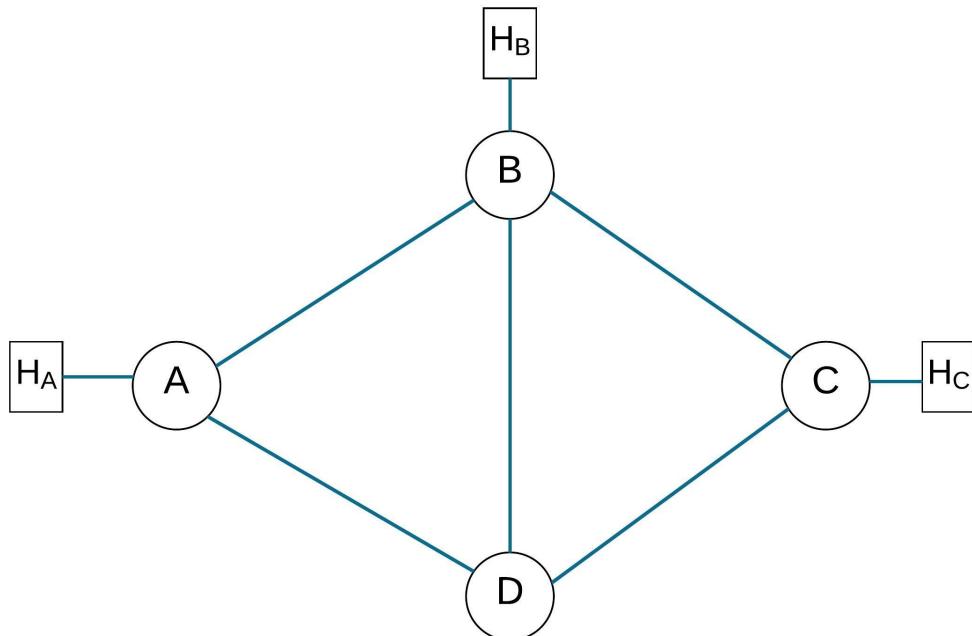
- Estructura Interna Subred Circuito Virtual
 - Entonces, el ruteador construirá una tabla de CV durante la fase de inicialización
 - » Esta tabla relaciona la línea de entrada por donde entra el paquete y tendrá el número de CV del paquete
 - » Esta tabla indicará por cual línea de salida saldrá el paquete y que número de CV tendrá el paquete de salida
 - » La tabla se actualiza conforme se vayan finalizando conexiones

Circuito Virtual (3)

- Estructura Interna Subred Circuito Virtual..
 - Como los paquetes “recuerdan la ruta”....
 - Cuando llega un paquete a un ruteador durante la fase de datos (**segunda fase**):
 - » Este sabe por cual línea de entrada llegó la trama
 - » Ve el número de circuito virtual y revisa su tabla de circuito virtual
 - » La tabla de CV le indicará por cual línea de salida se debe reenviar el paquete
 - » La tabla de CV indicará cual es el número de CV en el paquete de salida

Circuito Virtual (4)

- Estructura Interna Subred Circuito Virtual..
 - Como los paquetes “recuerdan la ruta”....
 - Veamos el siguiente ejemplo:



Circuito Virtual (5)

- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”....
 - Supongamos que se originan los siguientes circuitos virutales:
 - » Primero Origen en Host A
 - 0) ABC
 - 1) ADBC
 - 2) ADCB
 - » Luego Origen en Host B
 - 0) BDA
 - 1) BCDA

Circuito Virtual (6)

- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”.... Ruta 0) ABC

Tabla B

Entrada	Salida
A	0
D	0
C	0
H _B	0
H _B	0
H _B	1

Tabla A

Entrada	Salida
H _A	0
H _A	1
H _A	2
D	2
D	1

Tabla C

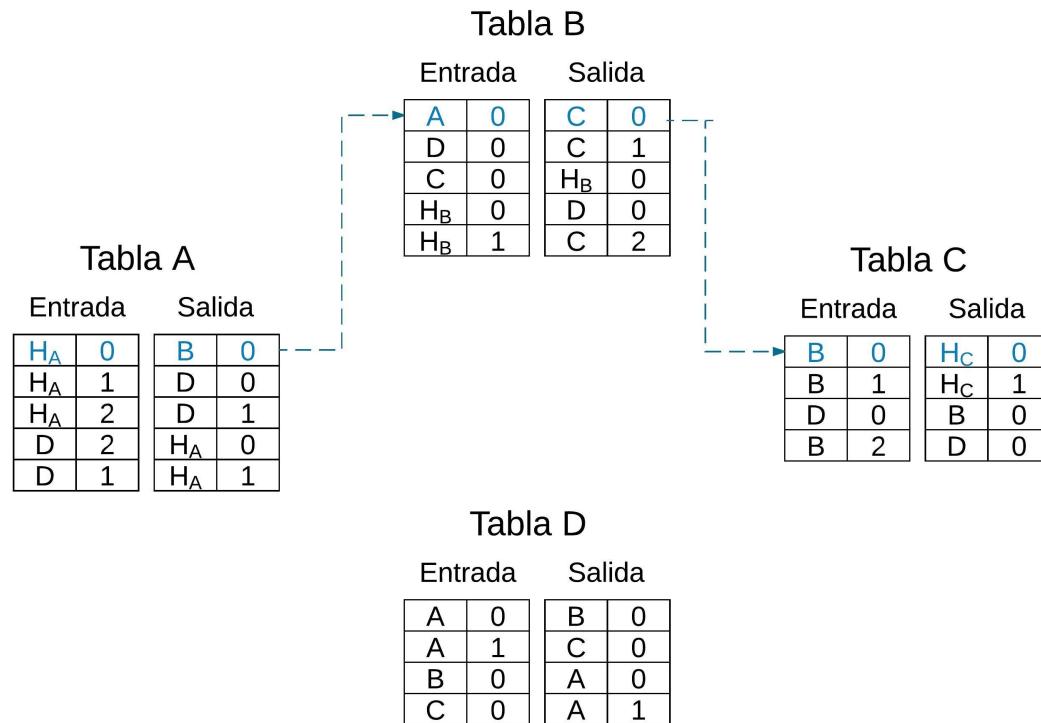
Entrada	Salida
B	0
B	1
D	0
B	2
D	0

Tabla D

Entrada	Salida
A	0
A	1
B	0
C	0
B	0
A	0
C	0
A	1

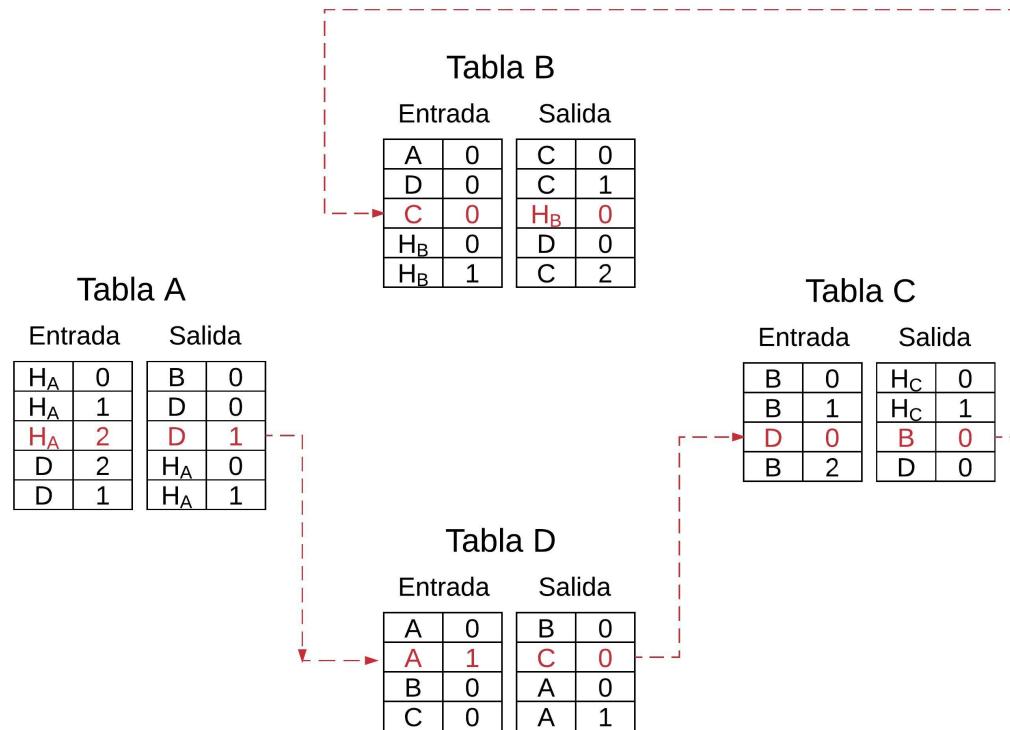
Circuito Virtual (7)

- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”....
 - Origen en A: Ruta 0) ABC



Circuito Virtual (8)

- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”....
 - Origen en A: Ruta 2) ADCB

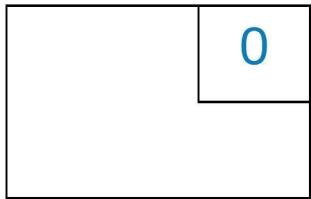


Circuito Virtual (9)

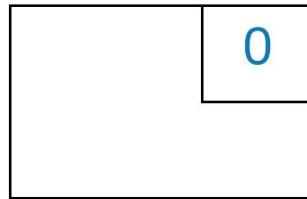
- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”....

Circuito Virtual: 0) ABC

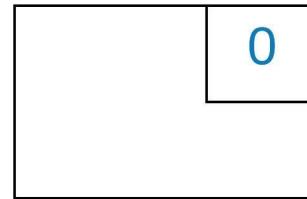
$H_A \rightarrow A$



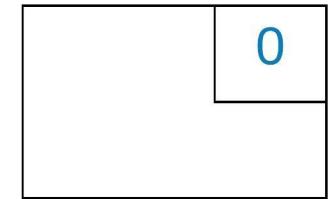
$A \rightarrow B$



$B \rightarrow C$



$C \rightarrow H_C$



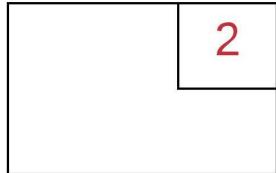
- Siempre fue el mismo número de CV

Circuito Virtual (10)

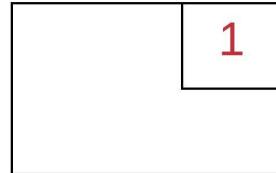
- Estructura Interna Subred Circuito Virtual
 - Como los paquetes “recuerdan la ruta”....

Circuito Virtual: 2) ADCB

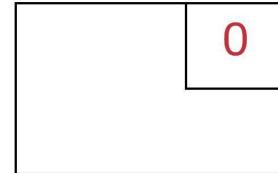
$H_A \rightarrow A$



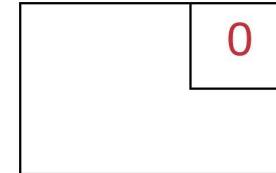
$A \rightarrow D$



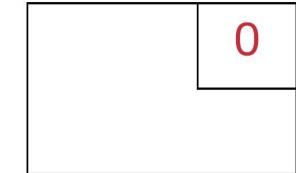
$D \rightarrow C$



$C \rightarrow B$



$B \rightarrow H_B$



- El número de CV cambió

Algoritmos de Enrutamiento (1)

— Carácterísticas:

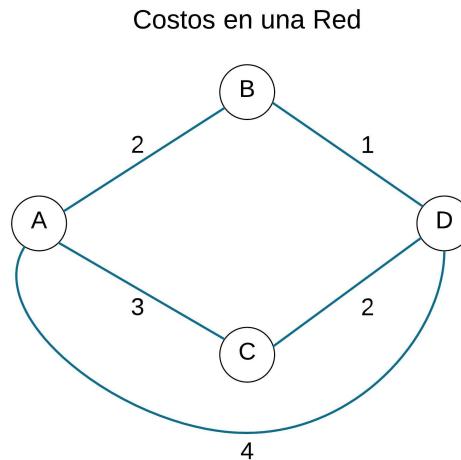
- Simple
 - Menos ciclos de CPU en el router (router más sencillo o de menor costo)
- Robusto
 - Habilidad de la subred para poder entregar paquetes por rutas alternas
- Estable
 - Si hay una falla o problema, no queremos que todo el tráfico se vaya por una sola ruta alterna
 - » Ya que se puede saturar y la falla no sea total
 - » Se quiere que se reaccione rápido

Algoritmos de Enrutamiento (2)

- Técnicas de Enrutamiento: (como se hace)
 - Criterio de Eficiencia:
 - Número de enlaces
 - Capacidad de enlaces
 - Retardo de enlaces
 - Tamaño de la cola
 - Precio de uso de enlaces
 - » Todo esto lo podemos relacionar como un costo para poder calcular la mejor ruta

Algoritmos de Enrutamiento (3)

- Técnicas de Enrutamiento....
 - Criterio de Eficiencia....



- Mejor ruta de red anterior:
 - » ABD: costo = 3
 - » AD: más directa, pero costo = 4
 - » Ver algoritmo de Dijkstra en Libro

Algoritmos de Enrutamiento (4)

- Técnicas de Enrutamiento....
 - Tiempo de Decisión
 - Paquete (datagrama)
 - Sesión (circuito virtual)
 - Lugar de Decisión
 - Cada nodo (distribuido)
 - Nodo central
 - Nodo origen

Algoritmos de Enrutamiento (5)

- Técnicas de Enrutamiento....
 - Fuente de Información
 - Ninguna
 - Local
 - Nodos adyacentes
 - Nodos en ruta
 - Todos los nodos
 - Estrategia de Enrutamiento
 - No adaptiva
 - Fija
 - Adaptiva

Algoritmos de Enrutamiento (6)

- Algoritmos No Adaptivos
 - Flooding (Inundación)
 - Es uno de los algoritmos más simples
 - Cada paquete que entra es transmitido por todas las líneas de salida excepto por donde entró
 - Genera muchos duplicados
 - » Se pueden tomar ciertas medidas
 - » Contador de Hops
 - Cada vez que pasa por un nodo se decrementa un contador
 - Cuando el contador llega a 0 se descarta paquete

Algoritmos de Enrutamiento (7)

— Algoritmos No Adaptivos

- Flooding (Inundación)....
 - » Flooding Selectivo
 - Por ejemplo: Este a Oeste
 - Muy robusto pero es de uso limitado

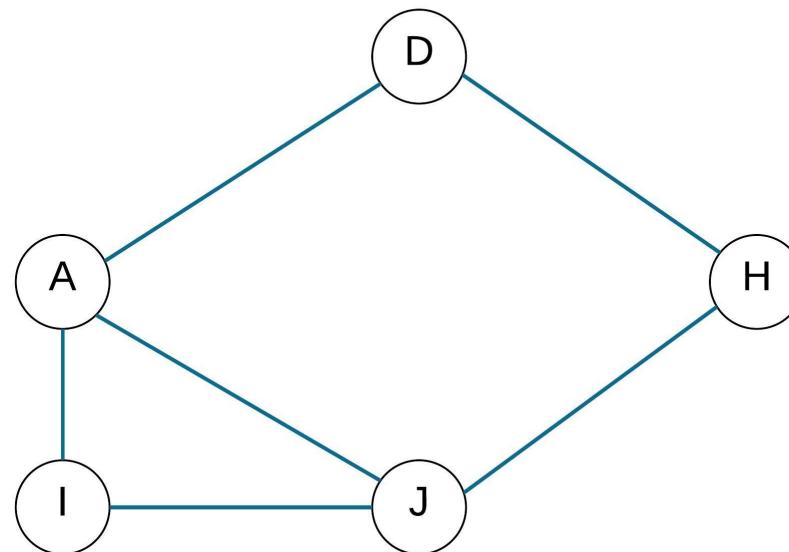
— Algoritmos Fijos

- Enrutamiento Estático
 - Muy usado en redes simples
 - Cada nodo mantiene una tabla con una fila para cada destino posible
 - Cada fila puede tener 1, 2 o 3 opciones

Algoritmos de Enrutamiento (8)

- Algoritmos Fijos..
 - Enrutamiento Estático....
 - Las tablas se hacen manualmente

Enrutamiento Estático



Algoritmos de Enrutamiento (9)

- Algoritmos Fijos..
 - Enrutamiento Estático....

Tabla J

D Opc. 1 Opc. 2

	S	S
A	A 0.7	I 0.3
I	I 0.8	A 0.2
J	- -	- -
D	H 0.5	A 0.5
H	H 0.8	A 0.2

D: Destino deseado

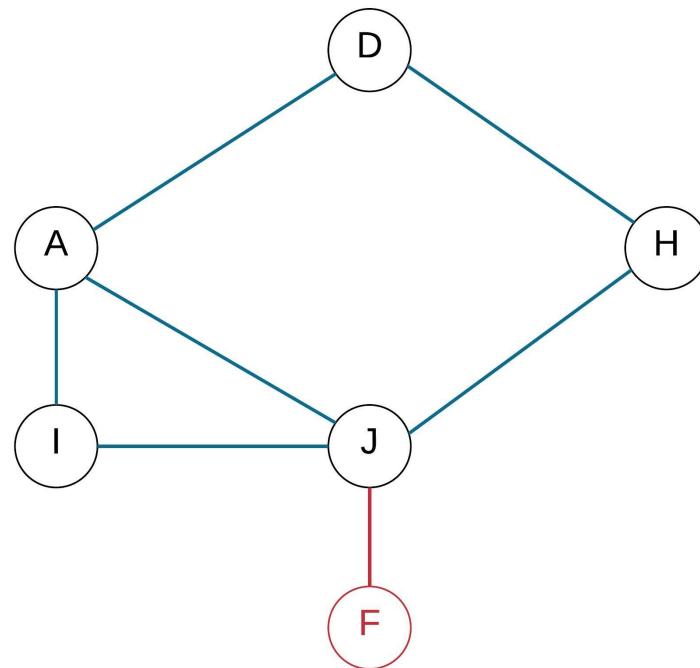
S: Línea de salida

0.7: % de tráfico que toma esta ruta

Algoritmos de Enrutamiento (10)

- Algoritmos Fijos..
 - Enrutamiento Estático....

Enrutamiento Estático
Agregando F



Algoritmos de Enrutamiento (11)

- Algoritmos Adaptivos
 - Distribuidos
 - Se adaptan a cambios
 - » Topológicos
 - » Tráfico
 - » Automáticamente
 - Enrutamiento por Vector de Distancia ([distr.](#))
 - En este algoritmo, cada nodo intercambia periódicamente información de rutas con cada uno de sus vecinos

Algoritmos de Enrutamiento (12)

- Enrutamiento por Vector de Distancia...
 - Cada nodo mantiene una tabla de enrutamiento que contiene:
 - » Línea preferida de salida para un destino
 - » Un estimado de tiempo o distancia a ese destino
 - Posibles métricas:
 - » Cantidad de saltos
 - » Retardo de tiempo (se usa “Hello Packet”)
 - » Tamaño de Cola (dato interno del router)

Algoritmos de Enrutamiento (13)

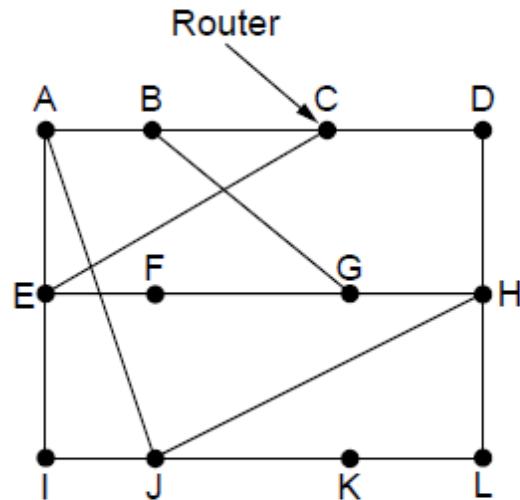
- Enrutamiento por Vector de Distancia...
 - Estrategia para Retardo
 - » Cada T ms cada router envia a sus vecinos una lista de sus retardos a cada destino
 - » Cada router recibe esta lista de sus vecinos
 - » Cada una de estas tablas que viene del vecino “x” con X_i que representa el estimado para llegar al destino “i”
 - » Si el router, “j”, sabe que el retardo a “x” es m y de la tabla consigue el retardo a “i”
 - Significa entonces, que el retardo a “i” = $m + X_i$

Algoritmos de Enrutamiento (14)

- Enrutamiento por Vector de Distancia...
 - Estrategia para Retardo....
 - » Hace esto para cada vecino
 - » De aquí, entonces puede encontrar la mejor ruta a “i”
 - » Actualiza así su propia tabla
- Veamos el siguiente ejemplo de vector de distancia
 - J calcula su retardo a sus vecinos por medio de un “Hello Packet”
 - » JA = 8 ms, JI = 10 ms, JH = 12 ms, JK = 6 ms

Algoritmos de Enrutamiento (15)

- Ejemplo Vector de Distancia



New estimated delay from J

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

Vectors received from J's four neighbors

New routing table for J

Algoritmos de Enrutamiento (16)

- RIP (Routing Information Protocol)
 - Es un protocolo por vector de distancia
 - No usa retardos
 - Usa número de enlaces (hops) como métrica
 - Desarrollado en los 60's
 - Usado en Arpanet
 - Hoy en día se usa para redes muy pequeñas
 - » Problema de tiempo de conversión
 - » Problema de cuenta a infinito

Algoritmos de Enrutamiento (17)

- Problema Cuenta a Infinito
 - Converge lentamente

A	B	C	D	E
•	•	•	•	Initially
1	•	•	•	After 1 exchange
1	2	•	•	After 2 exchanges
1	2	3	•	After 3 exchanges
1	2	3	4	After 4 exchanges

La buena noticia
camino a A se propaga
rápidamente

A	B	C	D	E	
1	2	3	4	Initially	
3	2	3	4	After 1 exchange	
3	4	3	4	After 2 exchanges	
5	4	5	4	After 3 exchanges	
5	6	5	6	After 4 exchanges	
7	6	7	6	After 5 exchanges	
7	8	7	8	After 6 exchanges	
⋮	⋮	⋮	⋮	⋮	⋮
•	•	•	•	•	⋮

Mala noticia de que se cayó camino a A
se propaga lentamente

Algoritmos de Enrutamiento (18)

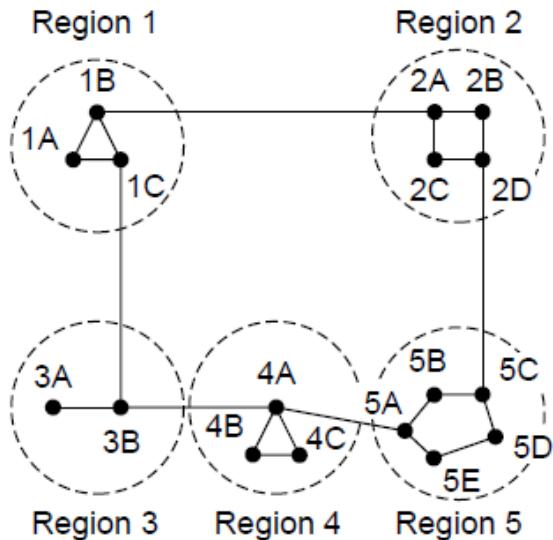
- Problema Cuenta a Infinito....
 - Entonces, mala noticia viaja lento
 - En algún momento hay que definir que 10: es Infinito
 - » Para poder decir que A se cayó
 - En retardo es más difícil
 - Por eso, RIP solo se usa en redes pequeñas o sencillas

Algoritmos de Enrutamiento (19)

- Enrutamiento Jerárquico
 - A medida que crece la red, crece la tabla de enrutamiento
 - Crece la memoria en el router
 - Crece tiempo de procesamiento
 - Entonces, se propone una red jerárquica
 - Los routers se agrupan en zonas
 - Cada router conoce todos los detalles de las rutas de su zona
 - No conoce los detalles de las rutas de otras zonas
 - Se pueden tener varios niveles de jerarquía
 - Número de Niveles = $\ln N$ (donde N número de routers)

Algoritmos de Enrutamiento (20)

- Enrutamiento Jerárquico...



Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

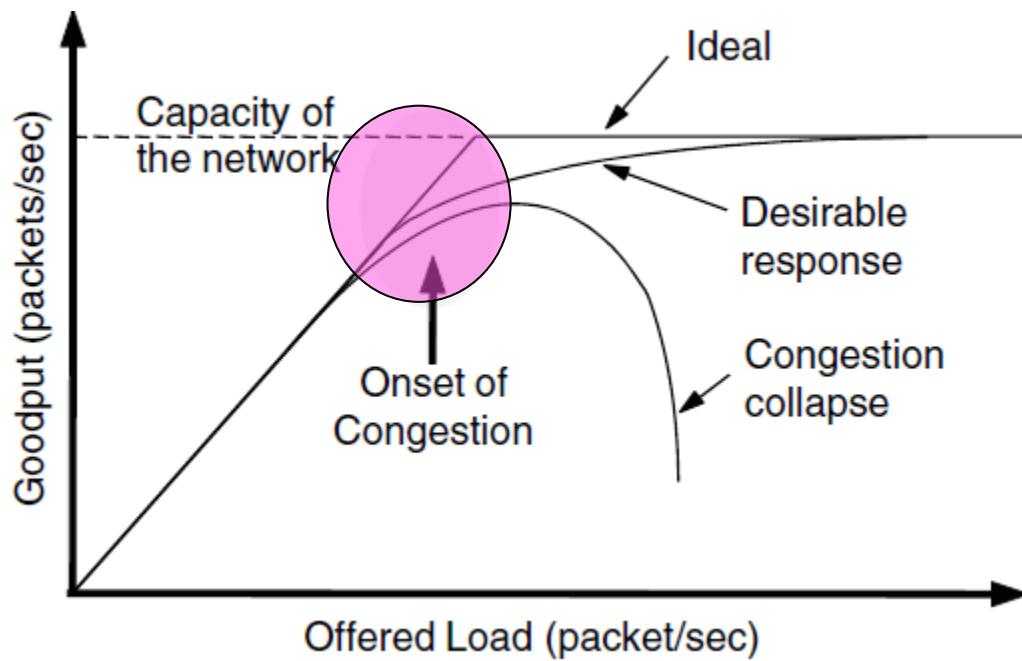
Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Mejor escogencia
para llegar a routers
en zona 5, excepto
para 5C

Congestión de Red (1)

- Control de Congestión
 - Cuando hay demasiados paquetes en una parte de la red
 - Su desempeño se degrada
 - » Congestión
 - Cuando hay una cantidad normal de paquetes, se pierde por BER ([curva deseada](#))
 - Si se aumenta más de la cuenta (los paquetes), los routers ya no pueden manejar el tráfico y terminará perdiéndose paquetes también ([hasta un posible colapso](#))
 - Veamos el siguiente gráfico

Congestión de Red (2)



Congestión de Red (3)

- Factores que hacen que ocurra Congestión
 - Exceso de paquetes que quieren la misma línea de salida
 - » Aumenta la cola de salida por esa línea
 - Falta de memoria
 - » Se pueden perder paquetes
 - » Más memoria: ayuda hasta cierto punto
 - Porque si aumenta mucho la cola, por time out se tendrá que retransmitir
 - CPU lenta
 - » Para llevar funciones administrativas
 - Aumenta colas, aun cuando haya ancho de banda de sobra

Congestión de Red (4)

- Capacidad de líneas de salida
 - » También aumenta las colas
- Diferencia de Control de Congestión vrs Control de Flujo
 - Control de Congestión
 - » Se ocupa de asegurar que la subred sea capaz de transportar el tráfico ofrecido
 - » Asunto global
 - Control de Flujo
 - » Se relaciona a tráfico punto a punto
 - » Se preocupa de que un TX rápido no sature a un RX lento

Congestión de Red (5)

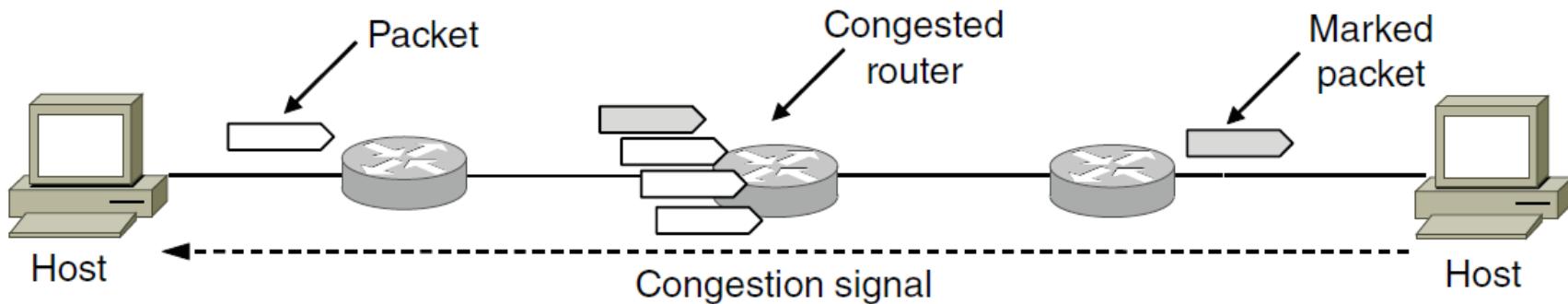
- » Necesita retroalimentación directa entre TX y RX
- Algunas veces ambos controles se confunden porque algunos algoritmos de control de congestión reducen la cantidad de paquetes que transmite un TX
- Métodos de Control de Congestión
 - Responsabilidad de Nivel 3 y Nivel 4
 - Nivel 3
 - » Enrutamiento (**mejores algoritmos**)
 - » Control de Admisión
 - » Regulación de Tráfico
 - » Desprendimiento de Carga

Congestión de Red (6)

- Control de Admisión
 - Para redes con circuitos virtuales
 - » No se establecen nuevos circuitos virtuales cuando hay mucho tráfico y la red no lo podrá manejar
 - » Si baja el tráfico, se vuelven a generar más circuitos virtuales
- Regulación de tráfico
 - Método 1: Routers congestionados le avisan a los hosts que bajen tráfico
 - » ECN (Explicit Congestion Notification) es un ejemplo

Congestión de Red (7)

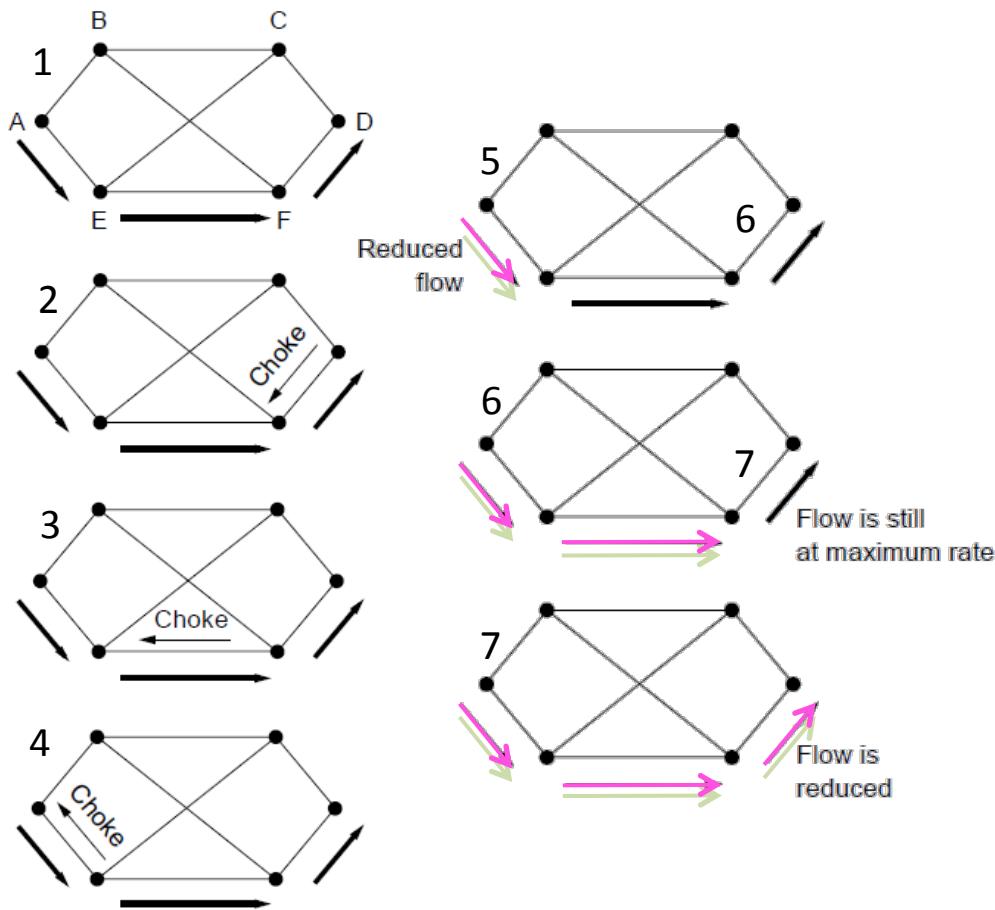
- Regulación de Tráfico....
 - » Un router marca paquetes y cuando llega este paquete al Host RX, se manda señal a Host TX para que reduzca tráfico



- » Se manda señal de congestión usando protocolo de transporte (más lento)

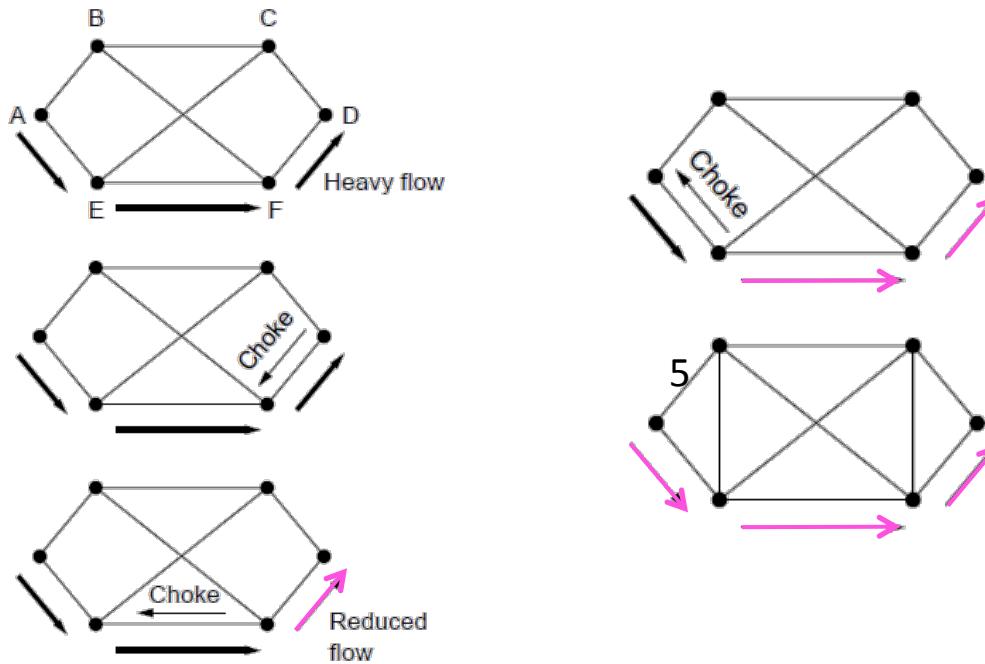
Congestión de Red (8)

- Cuando hay largas distancias o velocidades muy altas podría ser más lento



Congestión de Red (9)

- Otros métodos de Regulación de paquetes....
 - » Método 2: Cuando hay largas distancias o velocidades muy altas se propagan paquetes reguladores (choke) (más rápido)



Congestión de Red (10)

- Regulación Desprendimiento de Carga
 - Artillería pesada (medida extrema)
 - Se descartan paquetes por reglas
 - Por ejemplo:
 - » Desecha paquete más nuevo en una transferencia de archivo, ya que el viejo puede reconocer varios:
 - 6, 7, 8, 9, 10
 - Ya que el 6, permite pasar 6, 7, 8, 9 al nivel superior

Congestión de Red (10)

» Video comprimido

- Hay dos tipos de tramas
- Trama entera
- Trama diferencia
 - Mejor desecha trama diferencia
- Otros protocolos de nivel superior deben de arreglar esta pérdida de paquetes
- Esto tiene implicaciones muy importantes en el rendimiento de la red
- Diferentes métodos de hacerlo
 - » No siempre se implemente algo como ECN
 - » Nivel de Transporte (TCP) tiene varias alternativas que no se verán en este curso

Direcciones IP (1)

— IPv4

- Se definió en 1981
- 2 niveles de jerarquía
 - Nivel 1: Direcciones de Red ([Network Address](#))
 - Nivel 2: Direcciones de Host ([Host Address](#))
- Las direcciones de red son las que se intercambian para encontrar las rutas
 - Mejora rendimiento
 - Se reduce el tamaño de la tabla en routers
- Direcciones
 - IPv4: 32 bits para direcciones

Direcciones IP (2)

- Direcciones....
 - Se organiza en bytes separados por puntos
 - » X.X.X.X
 - Se escogen números en base 10
 - » Más fácil de recordar
 - 8 bits: de 0 a 255
 - 0.0.0.0: dirección mínima
 - 255.255.255.255: dirección máxima
 - Con 32 bits: 4,294,967,296 direcciones diferentes
 - Sin embargo, no se pueden aprovechar todas tan fácilmente
 - » Hay desperdicio de direcciones

Direcciones IP (3)

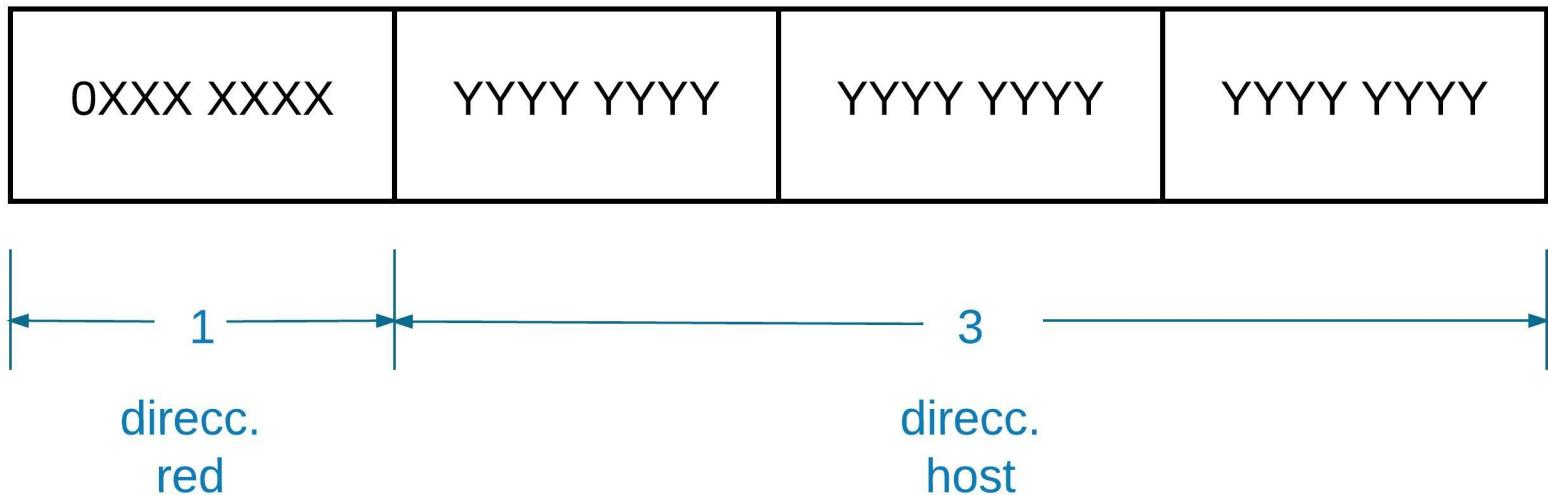
- Direcciones....

- 1 1 1 1 1 1 1 1
 - 128 64 32 16 8 4 2 1

- Las direcciones se dividieron en 5 clases
 - » Clase A: redes grandes
 - » Clase B: redes medianas
 - » Clase C: redes pequeñas
 - » Clase D: direcciones de multicast
 - » Clase E: direcciones reservadas IETF

Direcciones IP (4)

CLASE A



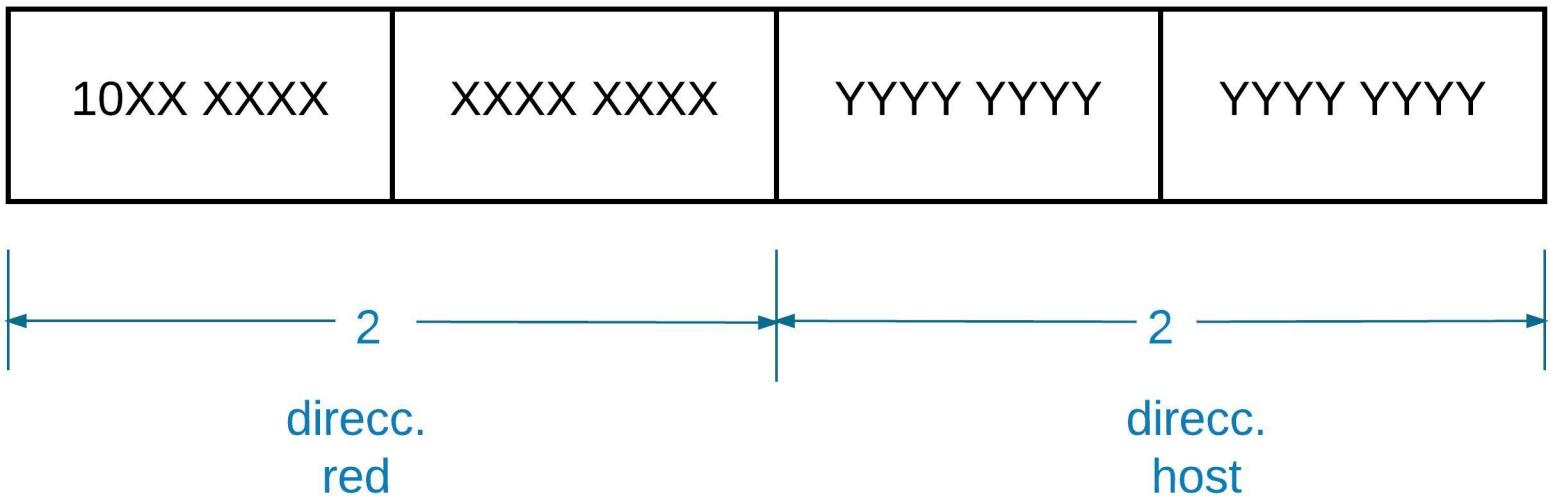
- Primer bit = 0: define que es dirección Clase A
- Quedan 7 bits: $2^7 = 128$ redes
 - pero no todas se usan

Direcciones IP (5)

- 0.0.0.0 se utiliza para “esta red”
- 127.0.0.0 dirección de loopback
 - Dirección especial
- Entonces, solo 126 redes clase A
- Rango: 1.0.0.0 a 126.0.0.0
- $2^{24} = 16,777,216$ posibles hosts
 - Pero no todos se usan
- X.0.0.0 → reservado a red x
- X.255.255.255 → dirección broadcast red x
- Entonces, solo 16,777,214 hosts (se pierden 2)

Direcciones IP (6)

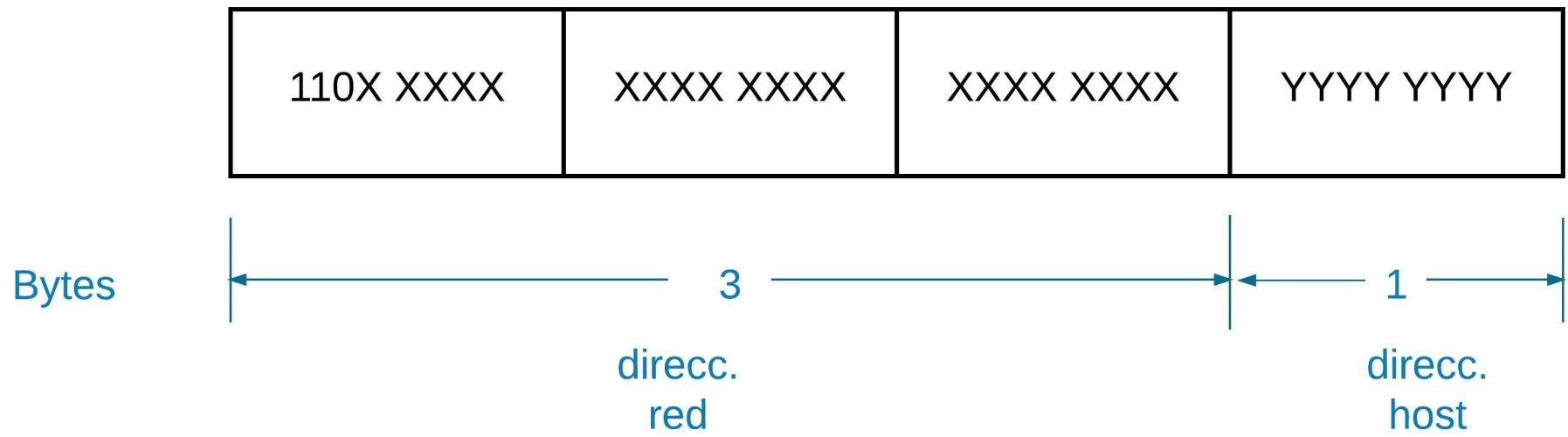
CLASE B



- Número de redes: $2^{14} - 2 = 16,384$
- Número de hosts: $2^{16} - 2 = 65,534$
- Rango: 128.0.0.0 a 191.255.0.0

Direcciones IP (7)

CLASE C



- Número de redes: $2^{21} - 2 = 2,097,150$
- Número de hosts: $2^8 - 2 = 254$
- Rango: 192.0.0.0 a 223.255.255.0

Direcciones IP (8)

CLASE D

1110 YYYY	YYYY YYYY	YYYY YYYY	YYYY YYYY
-----------	-----------	-----------	-----------

- Para direcciones multicast
- 1 dirección múltiples destinos
- Reduce tráfico en una red
- Rango: 224.0.0.0 a 239.255.255.255

Direcciones IP (9)

CLASE E

1111 YYYY	YYYY YYYY	YYYY YYYY	YYYY YYYY
-----------	-----------	-----------	-----------

- Para uso reservado de la IETF
- Rango: 240.0.0.0 a 255.255.255.255

Direcciones IP (10)

- Además, hay direcciones destinadas para uso privado
 - No se usan en redes públicas
 - No hay que solicitarlas a nadie
 - Clase A: 10.0.0.0
 - Clase B: 172.16.0.0
 - Clase C: 192.168.0.0

Direcciones IP (11)

- Limitaciones
 - Se desperdician muchas direcciones
 - No son como las direcciones MAC (físicas)
 - No hay redes tan grandes como para una clase A
 - Para redes pequeñas:
 - » 300 hosts: 2 clases C
 - Se desperdician 208 direcciones
- Hoy hay pocas direcciones IPv4 disponibles
 - IPv6: 128 bits para direcciones
 - NAT / PAT
 - Subnet Mask (subredes): fija y variable

Subredes (1)

- Crea un nivel más en la jerarquía
 - Crea una red lógica (subred)
 - Se necesita ruteador para interconectar estas subredes
 - Niveles:
 - Dirección de Red
 - Dirección Subred
 - Dirección Host
 - Los ruteadores del backbone de Internet solo ven la dirección de red
 - No entienden subredes

Subredes (2)

- El número de subredes depende de la clase de la red
- Las subredes se identifican usando una “seudo dirección” IP de 32 bits
 - Máscara
- Esta máscara indica cuantos bits de dirección de hosts se usarán para definir subredes
 - Los bits de la máscara que identifican la red y subred: 1
 - Los bits de la máscara que identifican los hosts: 0

Subredes (3)

– Ejemplo:

- Máscara de Clase C

255 . 255 . 255 . 192

1111 1111 . 1111 1111 . 1111 1111 . 11 00 0000



bits para subredes bits para hosts

- $2^2 - 2 = 2$ subredes
- $2^6 = 62$ hosts

Subredes (4)

— Posibilidades de Subredes Clase C

# de Bits	Máscara	# de Subredes	# de Hosts
2	255.255.255.192	2	62
3	255.255.255.224	6	30
4	255.255.255.240	14	14
5	255.255.255.248	30	6
6	255.255.255.252	62	2

Subredes (5)

- Posibilidades de Subredes Clase C.....

Pesos

1000 0000 = 128
1100 0000 = 192
1110 0000 = 224
1111 0000 = 240
1111 1000 = 248
1111 1100 = 252
1111 1110 = 254
1111 1111 = 255

Subredes (5)

— Ejemplo

- Se quiere “subnetear” la dirección de clase C:
 - 193.168.125.0
 - Además se requieren 6 subredes
- Solución:
 - Se escoge la máscara de la subred
 - » 255.255.255.**224**

Subredes (6)

— Ejemplo...

255 . 255 . 255 . 224

1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000



3 bits para
subredes

- $2^3 - 2 = 8 - 2 = 6$ subredes

Subredes (7)

— Ejemplo...

Subred	Dirección Binaria	Dirección Decimal
1	1100 0001 . 1010 1000 . 0111 1101 . 001 0 0000	193.168.125. 32

Diagram illustrating the breakdown of a subnetted IP address:

- The binary address is split into four octets separated by dots.
- Arrows point from the first three octets to their corresponding decimal values: 193, 168, and 125.
- A red arrow points to the fourth octet, labeled "bits de subred" (subnet bits).
- A blue arrow points to the last three bits of the fourth octet, labeled "bits de host" (host bits).
- The fourth octet is shown as 001 0 0000, where 001 represents the subnet bits and 0 0000 represents the host bits.
- The final value 32 is highlighted in red.

Subredes (8)

— Ejemplo...

Subred	Dirección Binaria	Dirección Decimal
0	No Válida	193.168.125.0
1	1100 0001 . 1010 1000 . 0111 1101 . 001 0 0000	193.168.125.32
2	1100 0001 . 1010 1000 . 0111 1101 . 010 0 0000	193.168.125.64
3	1100 0001 . 1010 1000 . 0111 1101 . 011 0 0000	193.168.125.96
4	1100 0001 . 1010 1000 . 0111 1101 . 100 0 0000	193.168.125.128
5	1100 0001 . 1010 1000 . 0111 1101 . 101 0 0000	193.168.125.160
6	1100 0001 . 1010 1000 . 0111 1101 . 110 0 0000	193.168.125.192
7	No Válida	193.168.125.255

Subredes (9)

— Ejemplo...

- Entonces, si tenemos 193.168.125.129

- Subred 4, host 1

- » . 100 0 0001

- » 128 + 1 = 129

- Se pueden tener hasta 30 hosts

Subredes (10)

- Máscaras Default
 - Es como no usar subredes
 - Clase C: 255.255.255.0
 - Clase B: 255.255.0.0
 - Clase A: 255.0.0.0
- Recordemos, fuera de la red, la subred no es visible

Introducción a TCP/IP (1)

- La Arquitectura de TCP/IP es más que estos 2 protocolos
 - De hecho, está compuesta de protocolos:
 - Operaciones Internas
 - Aplicaciones
- Nuevamente, revisitando la Arquitectura con más detalle en la siguiente diapositiva
 - Veremos que se tienen 2 opciones en nivel de transporte
 - En nivel de Internet, no solo se tiene IP y enrutamiento. Se requieren otros protocolos de operaciones internas (e.g. ARP, RARP, etc.)

Introducción a TCP/IP (2)

Arquitectura TCP/IP

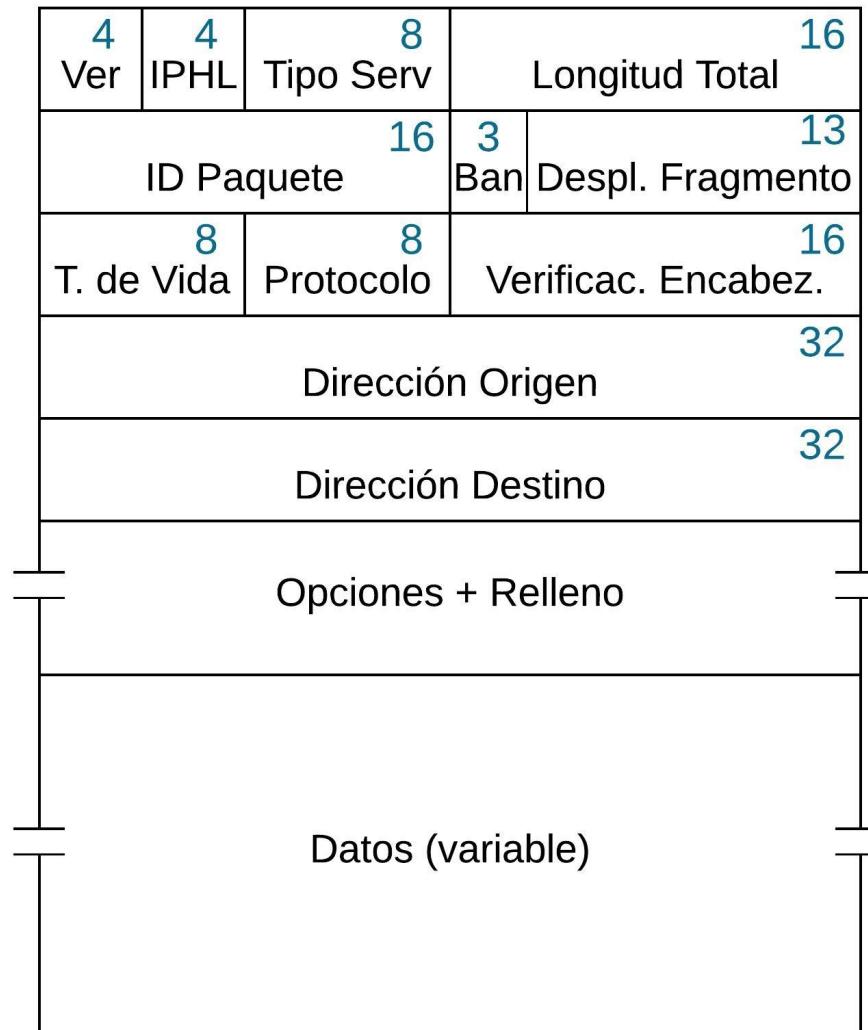
7	Aplicación FTP Telnet SMTP	Aplicación DHCP BootP DNS TFTP
6		
5		
4	Transporte TCP	Transporte UDP
3	Internet IP ICMP ARP / RARP PROT. ENRUT (rip, ospf, bgp)	
2	Host-to-Network	
1	No especificado	

Protocolo IP (1)

- Protocolo IP
 - Funciones principales:
 - Proveer servicios sin conexión (**mejor esfuerzo en la subred**)
 - Proveer fragmentación de paquetes para soportar niveles de enlace de datos con tramas de diferentes tamaños
 - En la siguiente diapositiva se muestra el formato de un paquete IP

Protocolo IP (2)

Paquete IP



Protocolo IP (3)

- Análisis Paquete IP
 - Versión: Versión del protocolo IP (hoy IPv4)
 - IPhL: IP Header Length: longitud de encabezado IP
 - Tipo de Servicio: Varios Niveles de importancia del paquete
 - Retardo
 - Throughput
 - Confiabilidad
 - Precedencia

Protocolo IP (4)

- Análisis Paquete IP....
 - Longitud Total: tamaño del paquete en bytes
 - Longitud máxima del paquete: 65,535 bytes
 - Incluye Encabezado
 - Entonces, Datos = Long. Total – IPHL
 - ID de Paquete: Identificador de paquete
 - A cada paquete se le asigna un identificador (ID) único
 - Esto permite identificar los fragmentos de un paquete

Protocolo IP (5)

- Análisis Paquete IP....
 - **Banderas**: 3 bits para banderas
 - Reservado
 - DF: Don´t Fragment: No fragmentar
 - » DF=1
 - MF: More Fragments: Permite fragmentar
 - » MF=1 : siguen más fragmentos
 - » MF=0 : último fragmento
 - **Desplazamiento de Fragmento** (fragment offset)
 - Mide el desplazamiento de un fragmento con respecto al paquete

Protocolo IP (6)

- Análisis Paquete IP....
 - Desplazamiento de Fragmento (fragment offset)....
 - El desplazamiento se da en incrementos de 64 bits o 8 bytes
 - Se tienen 13 bits:
 - » $2^{13} = 8192$ posibles fragmentos
 - » $8192 * 8 \text{ bytes} = 65,536 \text{ bytes}$
 - Un byte más del tamaño máximo del paquete

Protocolo IP (7)

- Análisis Paquete IP....
 - Tiempo de Vida (Time to Live)
 - No se permite que un paquete deambule por siempre en una red
 - » Este valor va de 0 a 255 posibles números de enlaces que puede viajar un paquete
 - » Se inicia con valor máximo
 - » Cuando llega a 0, el paquete se bota
 - Protocolo: Identifica al protocolo de nivel superior que se usará
 - Por ejemplo: TCP, UDP, OSPF, ICMP

Protocolo IP (8)

— Análisis Paquete IP....

- Verificación de Suma Encabezado: Control de errores en el encabezado
 - Solo verifica el encabezado
 - No revisa los datos
 - » Es mejor esfuerzo
- Dirección IP Origen y Destino: Direcciones
- Opciones - Relleno (Padding): opciones
 - Se rellena con 0's para tener siempre múltiplos de 32 bits
- Datos: datos del paquete IP

Operaciones Internas (1)

- Se mencionó en la Arquitectura TCP/IP que se requieren ciertos protocolos de “operaciones internas”
- Estos protocolos son necesarios para que el protocolo IP funcione
 - Dan información de control de la red
 - Dan información de direccionamiento
 - Etc.

ICMP (1)

- ICMP (Internet Control Message Protocol)
 - Brinda reportes de errores
 - Viaja en un paquete IP
 - Usado por hosts y routers
 - Reportes de errores, hosts o redes que no se les puede llegar
 - Se desarrollan herramientas como:
 - Ping (echo request/reply)
 - traceroute

ICMP (2)

— ICMP (Internet Control Message Protocol)

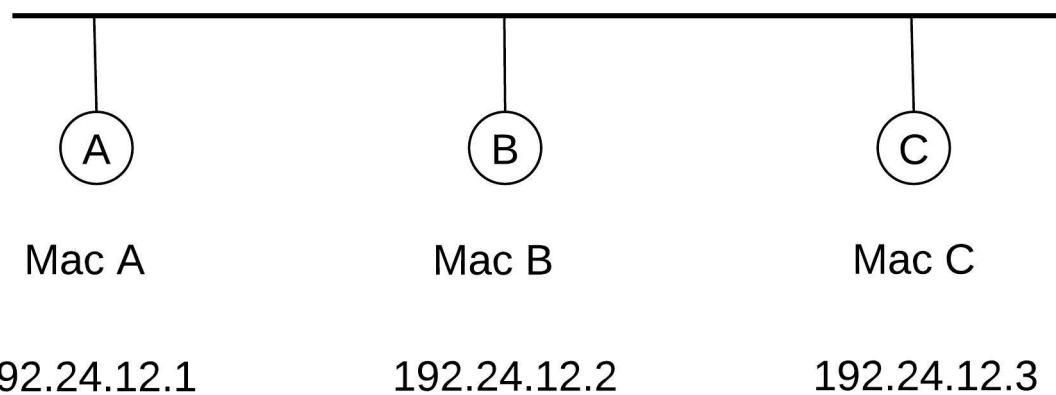
Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and Echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

ICMP (3)

- ICMP (Internet Control Message Protocol)
 - Traceroute:
 - Fuente envia series de un segmento UDP a un destino
 - » Primero se hace con TTL = 1, luego con TTL = 2, etc.
 - » Se escoge puerto poco usado
 - » Se envian 3 paquetes de cada uno para promedio
 - Cuando el paquete con TTL = 1 llega a router:
 - » Router lo descarta (TTL = 0)
 - Router envia mensaje fuente ICMP que TTL expiró y incluye nombre del router y su dirección IP
 - Cuando llega este mensaje a host origen, graba el tiempo de ida y vuelta y así sucesivamente

ARP (1)

- ARP (Address Resolution Protocol)
 - Para que 2 hosts se comuniquen en una LAN deben de conocer la dirección MAC del host destino



- C quiere comunicarse con A?

ARP (2)

- ARP (Address Resolution Protocol)....
 - Se supone que C debe de conocer la dirección IP de A
 - Si me quiero comunicar con María por teléfono, debería de tener el número de María
 - Pero, no conoce la dirección MAC de A
 - Probablemente, no conocemos a que puerto de la Central telefónica está conectada María
 - Se utiliza ARP en la arquitectura de TCP/IP para averiguar la dirección MAC de A
 - Veamos.....

ARP (3)

- ARP (Address Resolution Protocol)....
 - Primer Paso:
 - C manda trama con dirección destino broadcast y origen MAC de C
- | | | |
|-----------------|-------|------|
| 1111111111...11 | Mac C | etc. |
|-----------------|-------|------|

↓

Destino
Direcc. Mac

 - No hay problema ya que Ethernet es una red de propagación (broadcast)

ARP (4)

- ARP (Address Resolution Protocol)....
 - Segundo Paso:
 - Todas las estaciones reciben trama broadcast
 - Esta trama es recibida y pasada a nivel superior
 - » En este caso: Protocolo ARP
 - En la trama ethernet, tipo = ARP
 - ARP:
 - » Tiene MAC Origen C
 - » Tiene IP Origen C (192.24.12.3)
 - » Tiene IP Destino A (192.24.12.1)

ARP (5)

— ARP (Address Resolution Protocol)....

- Tercer Paso:
 - Solo host A reconoce su IP
 - » En el campo IP Destino de ARP
- Cuarto Paso:
 - Estación A graba información de dirección MAC C y dirección IP C (192.24.12.3)
 - Enviará trama a C con la siguiente información en ARP:
 - » MAC Origen A
 - » Dirección IP Origen A (192.24.12.1)
 - » MAC Destino C
 - » Dirección IP Destino C (192.24.12.3)

ARP (6)

- Dado que no es deseable manejar tanto tráfico de control dentro de una red
 - Se implementa un cache ARP dentro del host
 - Permite almacenar pares de IP-MAC resueltas
 - Pueden utilizarse en otro momento para no tener que enviar trama broadcast
 - Se usa un temporizador
 - Es un cache pequeño

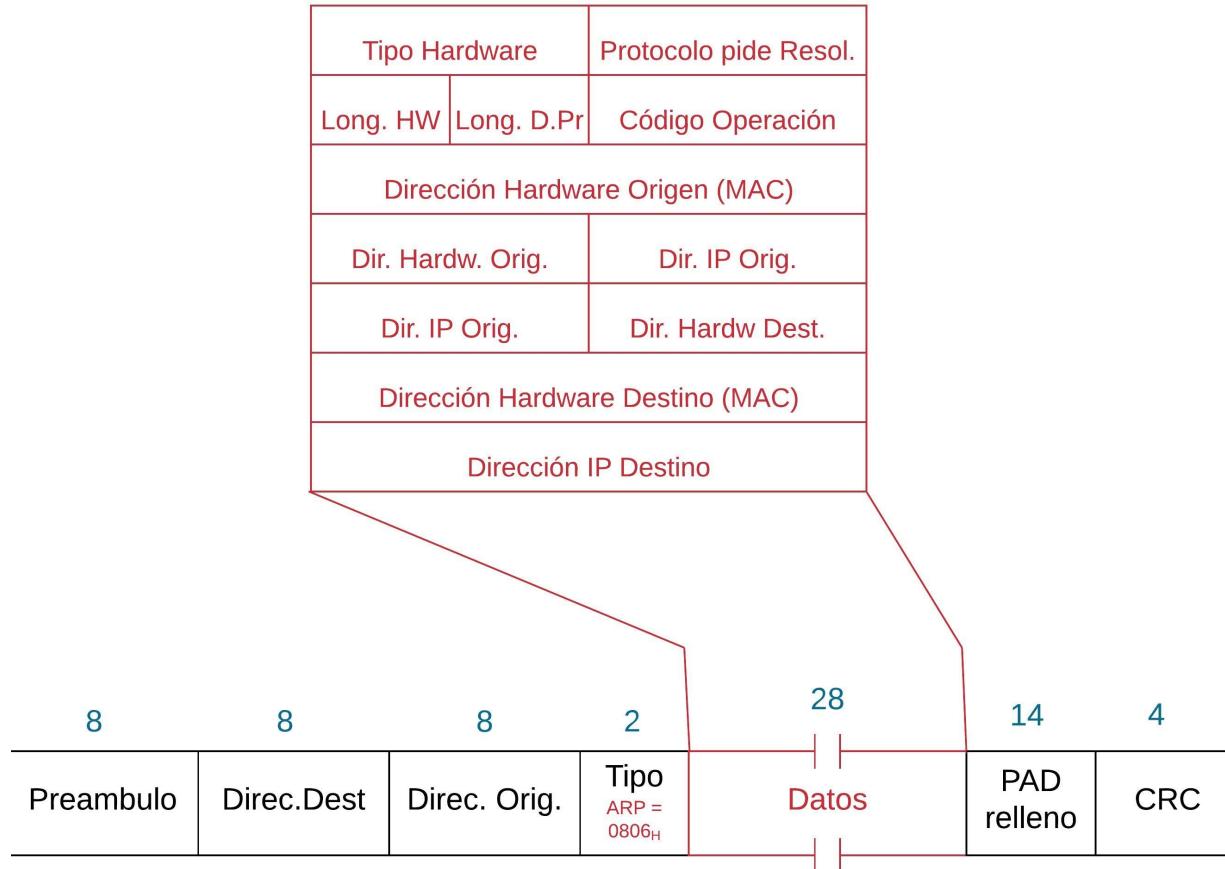
ARP (7)

Protocolo ARP

Tipo Hardware 16		Protocolo pide Resol. 16		
Long. HW 8	Long. D.Pr 8	Código Operación 16		
32 Dirección Hardware Origen (MAC)				
Dir. Hardw. Orig. 16		Dir. IP Orig. 16		
Dir. IP Orig. 16	Dir. Hardw Dest. 16			
32 Dirección Hardware Destino (MAC)				
32 Dirección IP Destino				

ARP (8)

Protocolo ARP



Trama Ethernet

Otros Protocolos (1)

- RARP (Reverse Address Resolution Protocol)
 - Mapea Direcciones MAC a Direcciones IP
 - Para estaciones sin disco duro que no saben su IP cuando arrancan
 - Hay un servidor RARP en cada red
 - Se usa dirección broadcast para llegar al servidor
 - El servidor RARP le indica a la estación cual es su IP

Otros Protocolos (2)

— BOOTP

- A diferencia de RARP, un paquete UDP si es retransmitido por los routers
- Se hace como una aplicación ya que usa UDP
- Da dirección IP de servidor que tiene información de direcciones IP para hosts sin disco duro
- Requiere configuracion manual
- Problema para redes grandes

Otros Protocolos (3)

- DHCP (Dynamic Host Configuration Protocol)
 - Asignación de IP manual y automática
 - Es lo que se usa hoy en día
 - Pero también es una aplicación y usa UDP
- DNS (Domain Name Server)
 - Para hacer relación de nombres de dominios y direcciones IPs
 - Es como un servicio de Directorio de Internet

Servicios y Protocolos (1)

- Antes de proseguir, es importante devolvernos a conceptos que vimos muy rápidamente en el Capítulo 1
 - Servicios
 - Primitivas
 - Interfaces
 - Protocolos
- Empecemos con terminología OSI
 - Servicios
 - La función de cada nivel es la de brindar un servicio al nivel Superior

Servicios y Protocolos (2)

- Empecemos con terminología OSI....
 - Entidades
 - Elementos activos a cada nivel
 - » Software: hablamos del proceso
 - » Hardware: hablamos de chips
 - Hardware hará hacer CRC
 - Entidades equivalentes
 - Entidades del mismo nivel pero de hosts (máquinas) diferentes
 - Interfaces
 - Conjunto de reglas para que 2 niveles intercambien información

Servicios y Protocolos (4)

- Empecemos con terminología OSI....
 - Interfaces....
 - Ejemplos en OSI
 - » SAP (Service Access Point)
 - Los servicios se encuentran en los SAPs
 - Cada SAP tiene una dirección única
 - Analogía en Sistema Telefónico
 - SAP: caja RJ-11
 - Dirección SAP: Número Telefono
 - » En TCP/IP es solo la interfaz

Servicios y Protocolos (5)

- Empecemos con terminología OSI....
 - Primitivas de un Servicio
 - Un servicio se caracteriza por medio de un conjunto de primitivas (operaciones)
 - En OSI, se definen 4 tipos de primitivas
 - Request: una entidad quiere que el servicio realice un trabajo
 - Indication: una entidad va a ser informada de un evento
 - Response: una entidad quiere responder ante un evento
 - Confirm: una entidad a a ser informada sobre el trabajo que solicitó

Servicios y Protocolos (6)

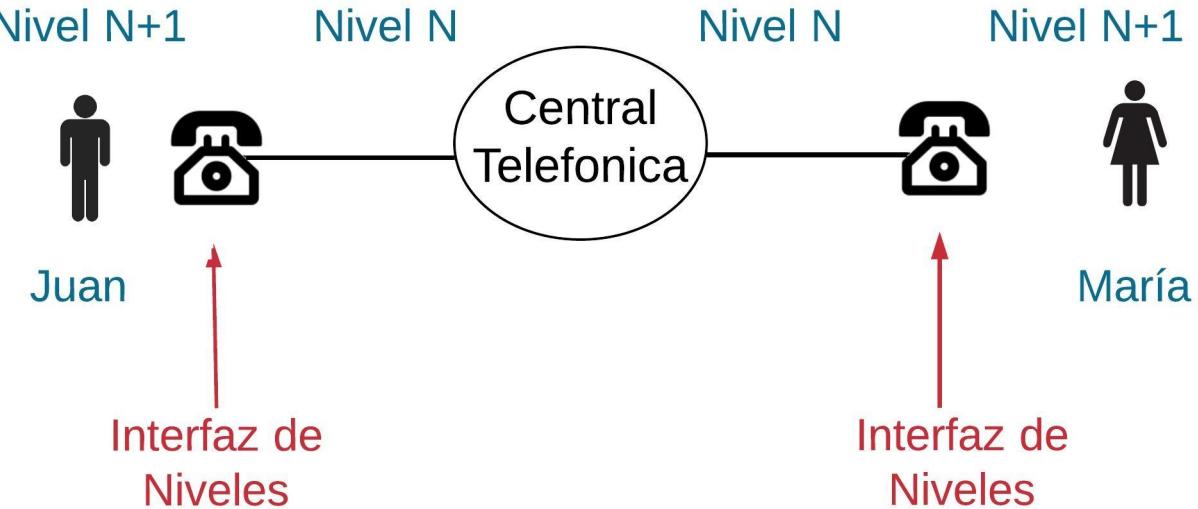
- Empecemos con terminología OSI....
 - Las primitivas tienen parámetros
 - Dirección
 - Tipo de servicio
 - Tamaño de unidad de datos (segmento, paquete, trama)
 - etc
 - Negociación
 - Ponerse de acuerdo en los parámetros
 - Servicios con Conexión: Ej. 4 primitivas
 - Servicios sin Conexión: Ej. 2 primitivas

Servicios y Protocolos (7)

- Empecemos con terminología OSI....
 - Redefiniendo Servicios
 - Conjunto de primitivas que el Nivel N brinda al Nivel N+1
 - Define cuales operaciones deben de hacerse
 - Pero no indica como hacerlo
 - Veamos un ejemplo
 - Sistema Telefónico se simplifica a un solo nivel
 - Los teléfonos representan donde se accede el servicio que presta el Nivel N del Sistema Telefónico
 - Teléfono: interfaz
 - Persona: Nivel N+1
 - “Juan invita a María a una fiesta”

Servicios y Protocolos (8)

— Uso de Primitivas



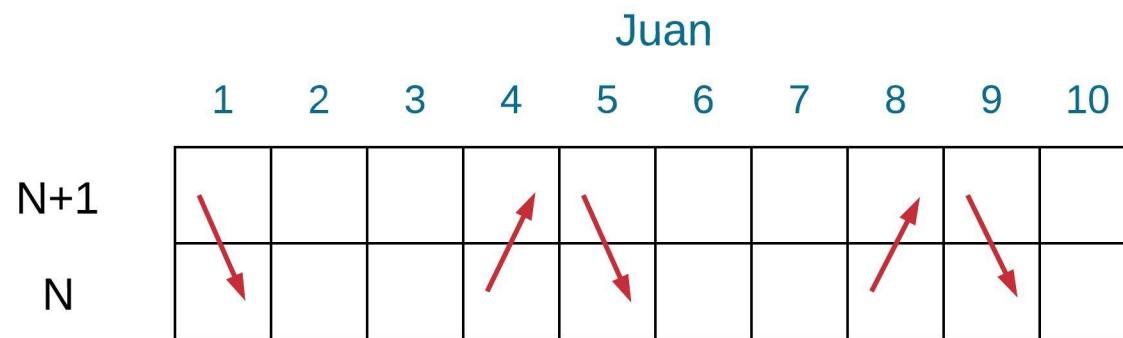
Servicios y Protocolos (9)

— Uso de Primitivas....

- 1) Connect.Request: Juan marca número de María
- 2) Connect.Indication: Suena teléfono de María
- 3) Connect.Response: María levanta auricular
- 4) Connect.Confirm: Juan deja de oír tono de espera
- 5) Data.Request: Juan invita a María a fiesta
- 6) Data.Indication: María oye invitación de Juan
- 7) Data.Request: María dice que sí acepta
- 8) Data.Indication: Juan oye respuesta de María
- 9) Disconnect.Request: Juan cuelga auricular
- 10) Disconnect.Indication: María oye que Juan cuelga y cuelga

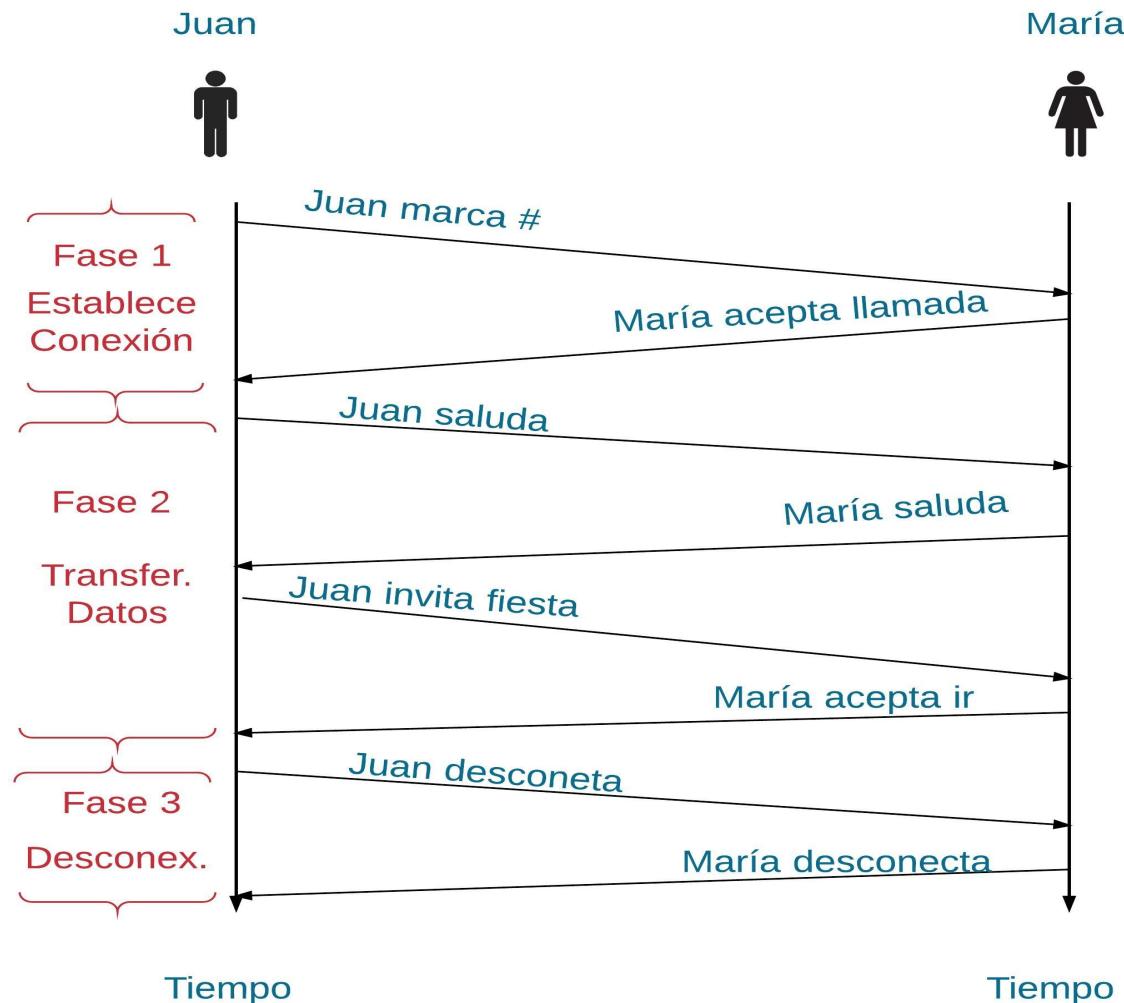
Servicios y Protocolos (10)

— Uso de Primitivas



Servicios y Protocolos (11)

- Protocolo: veamos ejemplo anterior



Servicios y Protocolos (12)

- Protocolo....
 - En ejemplo anterior “con conexión”
 - Fase 1
 - » Se envian mensajes de control
 - Fase 2
 - » Se intercambian datos
 - Saludos
 - Preguntas
 - Fase 3
 - » Se envian mensajes de control

Servicios y Protocolos (13)

— Protocolo....

- Cada mensaje genera una respuesta
- Se toman acciones de acuerdo a respuesta
- Se espera que personas tengan costumbres similares, hable mismo idioma
 - De otra forma no funciona el protocolo
- En una red, serían entidades:
 - Intercambiando mensajes (primitivas)
 - » En terminología TCP/IP: procesos remotos
 - Y se toman acciones (hardware o software)
 - » Estos procesos remotos son definidos por el protocolo

Servicios y Protocolos (14)

- Protocolo....
 - Entonces el Protocolo:
 - Define formato y orden de mensajes que se intercambian entre 2 o más entidades
 - Define las acciones a tomar en la transmisión o recepción de un mensaje o evento
 - Define el significado de bits en las tramas, paquetes o segmentos que se intercambian por las entidades (idioma)

Introducción TCP (1)

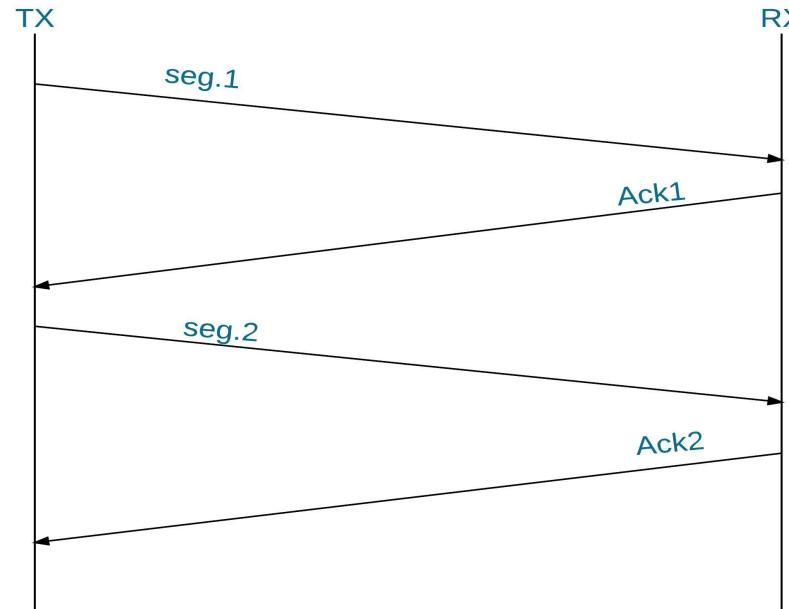
- TCP (Transmission Control Protocol)
 - Provee una transmisión confiable en un ambiente IP
 - Los protocolos superiores no se tienen que preocupar
 - Entrega de flujos de datos sin ninguna estructura
 - Se identifican por su número de secuencia
 - Orientado a conexión: Confiable
 - Sistema de Acknowledgments (Acks)
 - Se maneja segmentos perdidos, retardados, duplicados o con errores

Introducción TCP (2)

- TCP (Transmission Control Protocol)....
 - Ofrece buen control de flujo
 - Mecanismo de Ventanas
 - Operación Full Duplex
 - Se puede transmitir y recibir simultáneamente
 - Multiplexación
 - Se permite tener varias conexiones simultáneas
- Operación
 - Se reconoce por byte y no por segmento
 - PAR (Positive Ack Retransmissions)

Introducción TCP (3)

- Operación....
 - Es como un Stop & Wait
 - Pero no sería muy eficiente si lo hacemos Seg.= 1 byte
 - Además de tiempos involucrados son muy altos

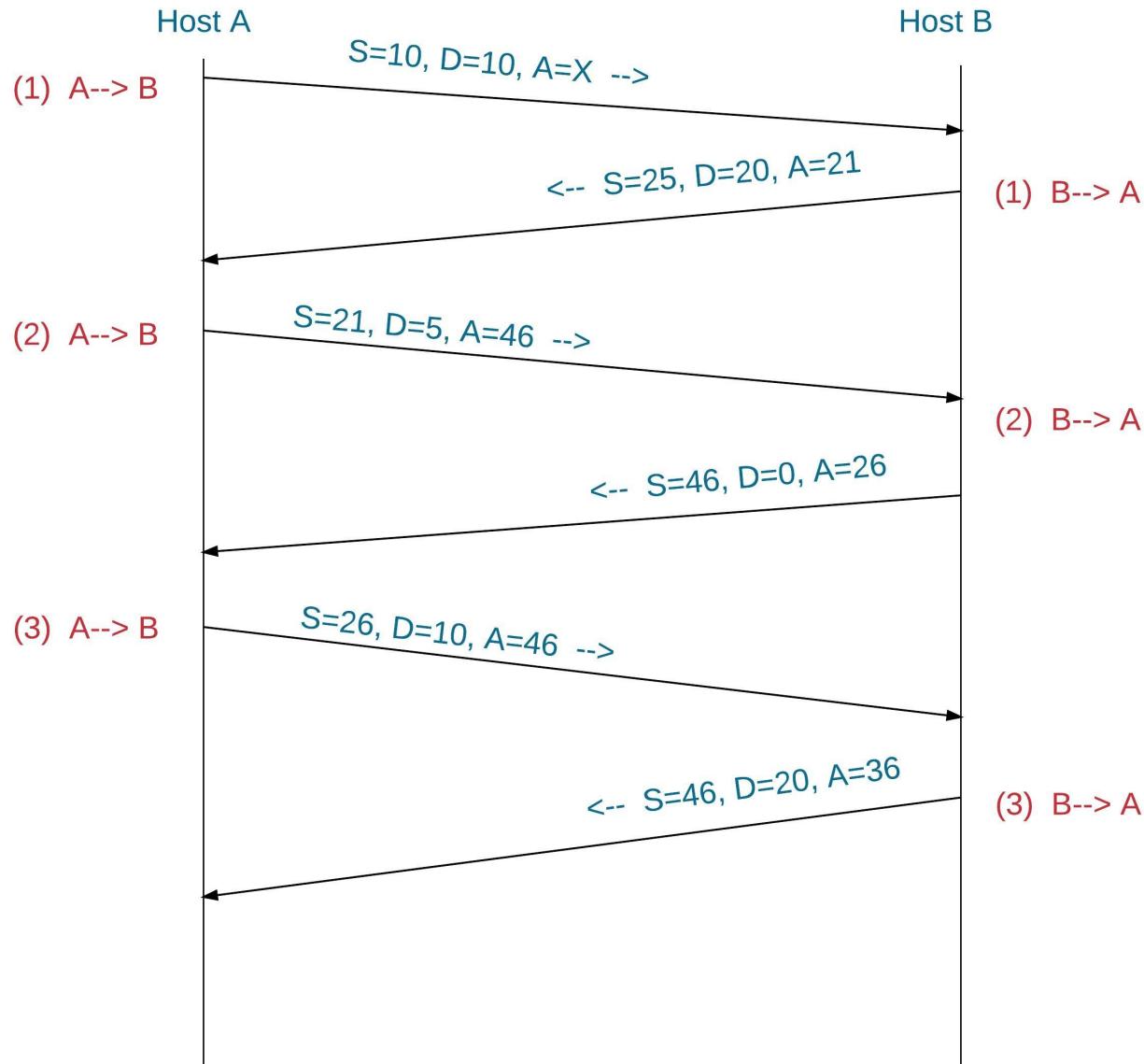


Introducción TCP (4)

— Operación....

- Obviamente sería un desperdicio de recursos de la red enviar un byte a la vez
- Entonces, se envian varios bytes por segmento
- En la siguiente figura, notar que dado que es una transmisión Full Duplex, no tiene sentido escribir TX y RX, sino más bien A y B
 - Se envian varios bytes por segmento
 - » D: # bytes en segmento
 - » S: # de secuencia
 - » A: # de secuencia del Ack

Introducción TCP (5)



Introducción TCP (7)

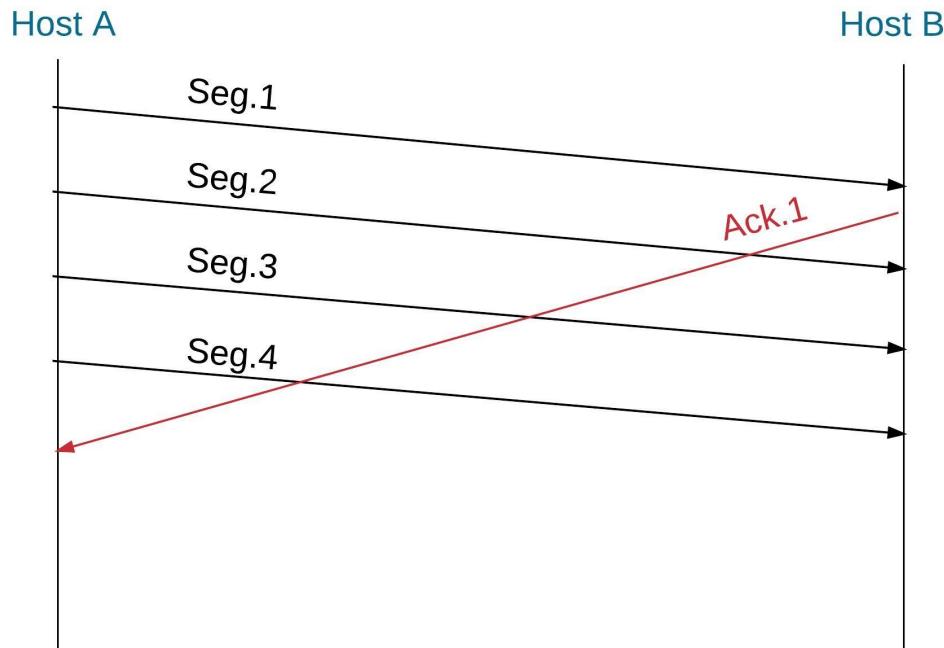
— Operación....

- Como se mencionó, nos da confiabilidad pero solo un segmento a la vez (de varios bytes)
- Sabemos que los tiempos involucrados son altos y no parece muy eficiente
- Se puede agregar concepto general de Ventana deslizante
 - Permite mandar varios segmentos sin un ack
 - No se puede tratar igual que Nivel 2, GBN o SR
 - Es un problema de espacio en memoria y que no corresponde a enlace punto-a-punto como Nivel 2
 - No sería tema de este curso ver como se haría

Introducción TCP (8)

— Operación....

- Se mandan varios segmentos antes de requerir un Ack. **Mejora rendimiento**



Introducción TCP (9)

- Operación....
 - Control de Flujo y Ventana Deslizante
 - Datos en ventana de TX no se pueden remover hasta recibir Ack
 - Datos en ventana de RX no se pueden remover hasta que sean leidos
 - » Esto puede tomar cierto tiempo
 - Por esto, es que en este Nivel 4 es más complejo el manejo de memoria que en Nivel 2
 - Se utiliza un concepto de ventana deslizante diferente a resolver 2 problemas:

Introducción TCP (9)

— Operación....

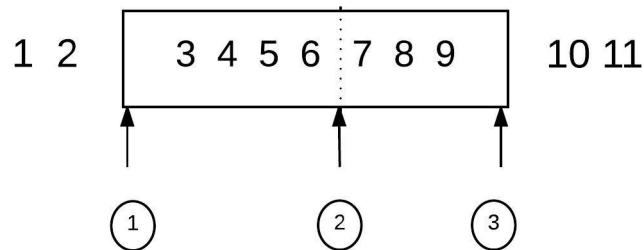
- » Enviar múltiples segmentos antes de requerir un Ack (**transmisión más eficiente**)
- » Control de Flujo
 - Le permite al RX restringir al TX transmitir más de la cuenta para asegurarse que tiene la cantidad de memoria adecuada

— Concepto de Ventana Deslizante

- » Permite enviar múltiples bytes sin necesidad de un Ack
- » Opera a nivel de bytes y no segmentos
- » Cada segmento viaja en un paquete
- » Los bytes son los que se numeran secuencialmente y no los segmentos

Introducción TCP (10)

- Operación....
 - Manejo de Ventana
 - » Se manejan 3 punteros



- » Bytes 1 y 2: ya han sido transmitidos y reconocidos
- » Bytes 3,4,5,6: ya se transmitieron pero no se han reconocidos

Introducción TCP (11)

- Operación....
 - Manejo de Ventana...
 - » Bytes 7,8,9: no han sido transmitidos, pero se enviarán de inmediato
 - No es necesario recibir un Ack para transmitirlos
 - » Bytes 10,11: no se podrán transmitir hasta recibir Acks 3,4....
 - Ya no hay espacio en memoria
 - Tamaño Ventana TX: 7 en este ejemplo
 - RX: requiere otra ventana
 - Como es full duplex: se requieren 4 ventanas

Introducción TCP (12)

— Operación....

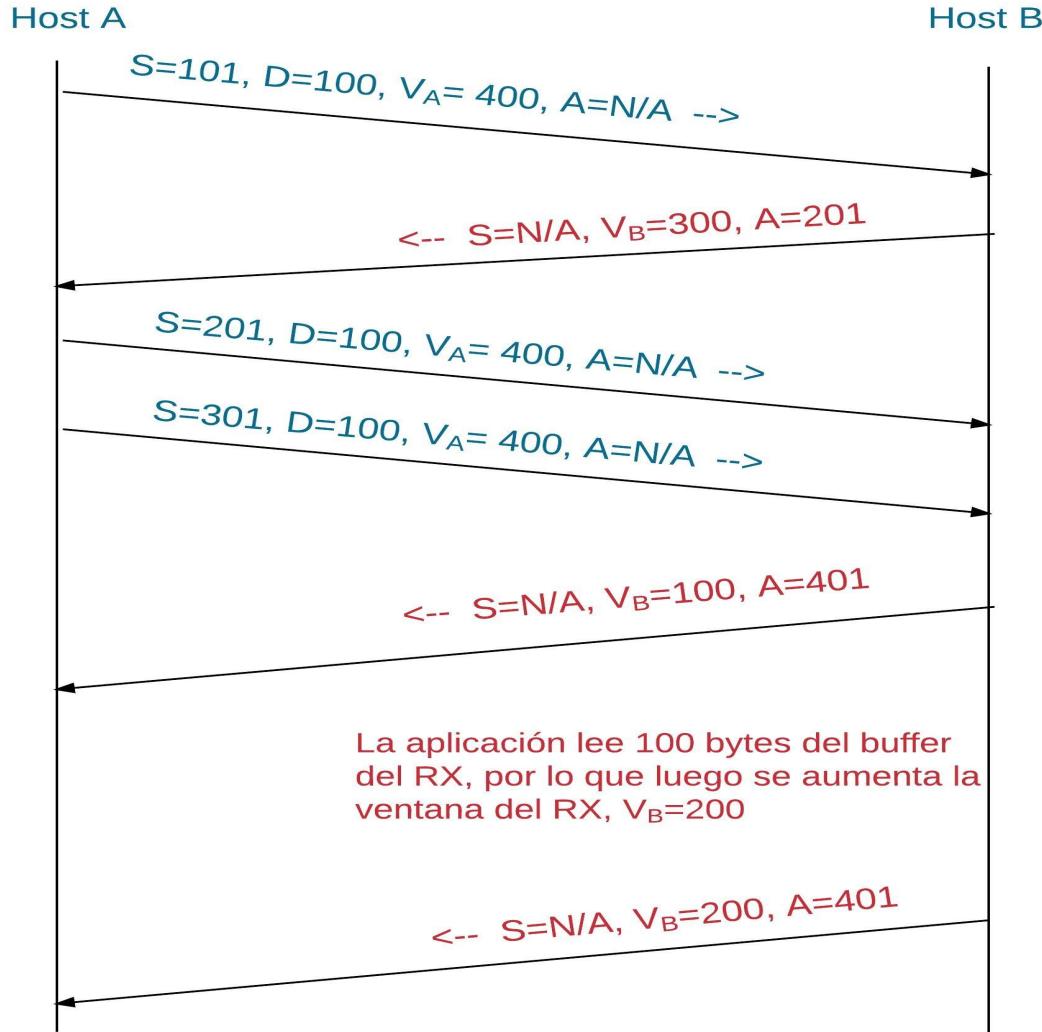
- Si ventana fuera de 1 byte: sería PAR
- Se permite variar el tamaño de la ventana en el tiempo
- El RX es quién define el tamaño
- Esto define el control de flujo
- TCP separa los aspectos de Ack's de los segmentos de la asignación de buffers (tamaño de ventana) en el RX
- Para esto, el RX cuando envia un Ack, indica la cantidad de bytes que está preparado para recibir

Introducción TCP (13)

— Operación....

- De esta forma, el TX sabe cuanto puede enviar
 - Control de Flujo
- Al inicio, durante la inicialización de la conexión, se define una cantidad inicial
- Veamos la siguiente diapositiva donde se ve este concepto

Introducción TCP (14)



Introducción TCP (15)

— Puertos y Puntos Finales

- TCP le permite a múltiples programas de aplicación en una máquina comunicarse concurrentemente
- TCP utiliza los número de puerto para indicar el último destino dentro de un Host
- Sockets: Puntos Finales
 - Es la concatenación del número de:
 - (dirección IP Host + Puerto) = Socket
 - Dirección IP: 32 bits
 - Número Puerto: 16 bits

Introducción TCP (16)

— Puertos y Puntos Finales...

- Existen Puertos reservados
 - “Well Known Ports” (puertos bien conocidos)

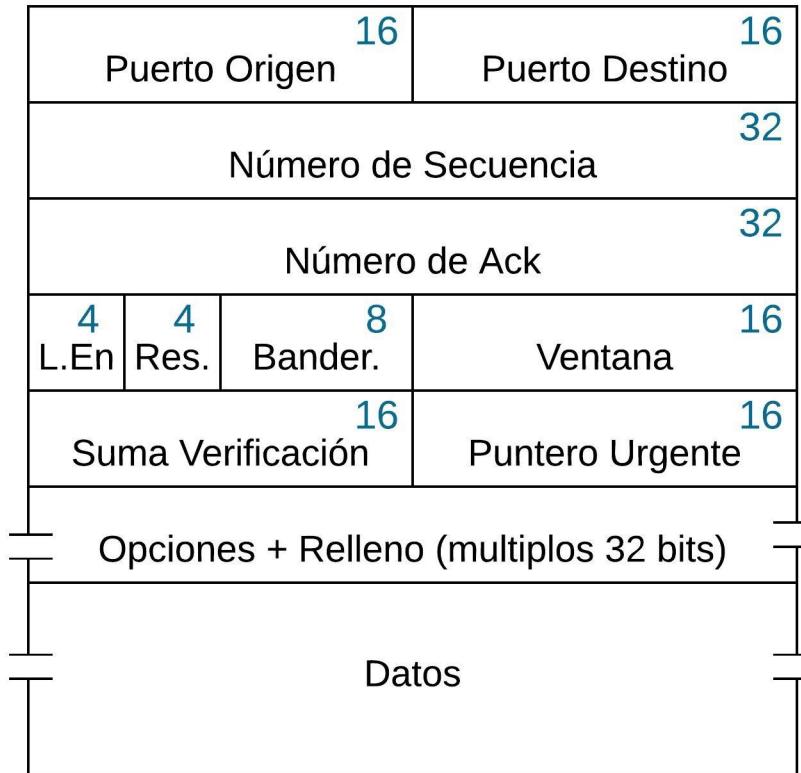
# Puerto Decimal	# Puerto Hexadec.	Aplicación
20	0014	ftp-datos
21	0015	ftp-control
23	0017	telnet
22	0016	ssh

- Originalmente habían 256, pero hoy hay 1024
- Si se quiere hacer una aplicación, se sugiere usar un puerto > 1024

Introducción TCP (17)

– Formato Segmento TCP

Protocolo TCP



Introducción TCP (18)

- Puerto Origen, Destino: identifica la aplicación
- Número de Sec.: representa el número de secuencia del primer byte de datos en este segmento.
 - Excepto si SYN = 1, entonces es el ISN (Initial Sequence Number)
 - Hay $2^{32} = 4,294,967,296$ posibilidades
 - Número aleatorio (se saca con reloj host)
 - El RX lo usa para reconstruir mensaje

Introducción TCP (19)

- Num. Ack: contiene el número del primer byte esperado en el siguiente segmento
 - Solo válido si Bandera Ack = 1
 - Se usan Ack's acumulativos, reconoce hasta Ack-1
- Long. Encabezado: Longitud encabezado en múltiplos de 32 bits
- Reservado: para uso futuro 4 bits (antes 6)
- Banderas de Control: 8 bits

Introducción TCP (20)

- URG: si campo de puntero urgente es valido
- ACK: define si Núm.Ack es válido
- PSH: si se pidió un Push para este segmento
- RST: resetea conexión
- SYN: sincroniza Núm.Sec, si SYN=1 sería el ISN
- FIN: TX terminó de transmitir, indica desconex.
- CWR, ECN: se usan para indicar congestión
(fuera del alcance de este curso)

Introducción TCP (21)

- Ventana: se usa para que RX indique cuantos bytes está preparado para aceptar
- Suma Verific: suma de encabezado, datos y pseudoencabezado
 - Pero si se implementa control de errores (ACKs)
- Puntero Urgente: válido si URG=1
 - Le indica a RX adonde dentro del flujo de datos hay datos urgentes
 - Valor de Punt. Urg.: desplazamiento (offset) desde número de secuencia hasta donde se encuentra datos urgentes

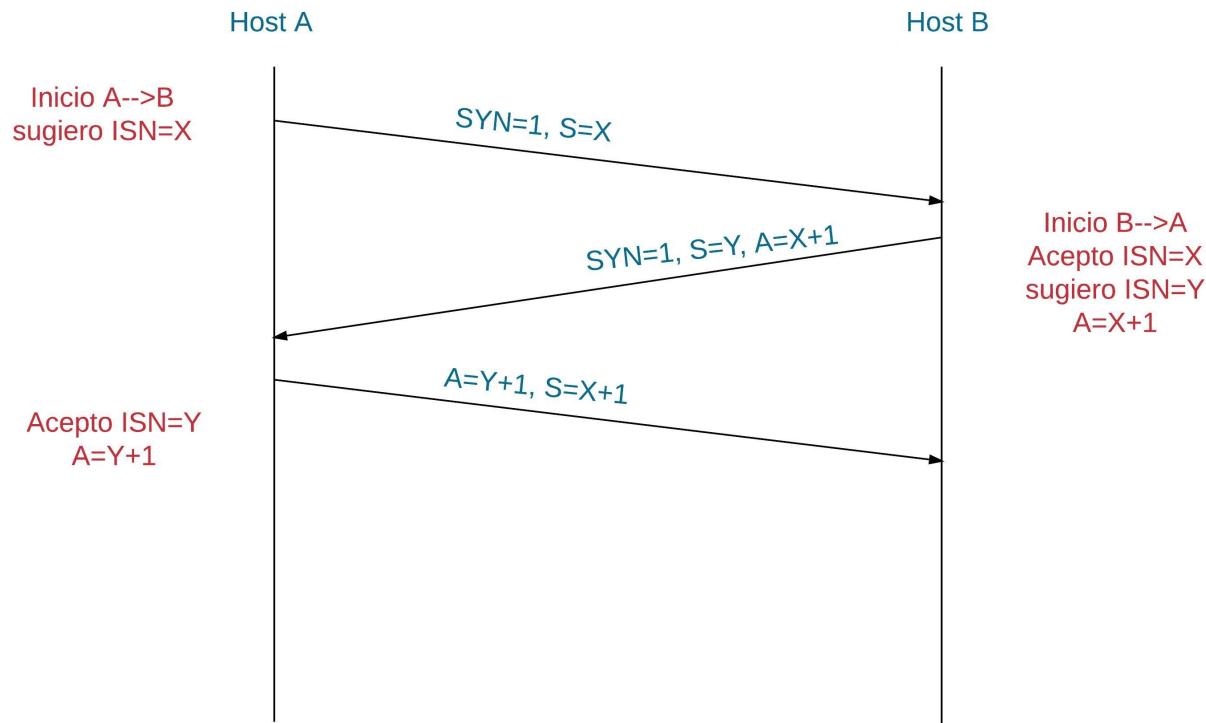
Introducción TCP (22)

- Aclaración adicional Push
 - Permite que pequeñas cantidades de datos sean transmitidas en un segmento
 - A pesar de que no se llene buffer
 - Apura la transmisión (aplicaciones interactivas)
 - Solicitado por la aplicación
 - Si Push=1, se envia los datos que se tengan en ese momento

Introducción TCP (23)

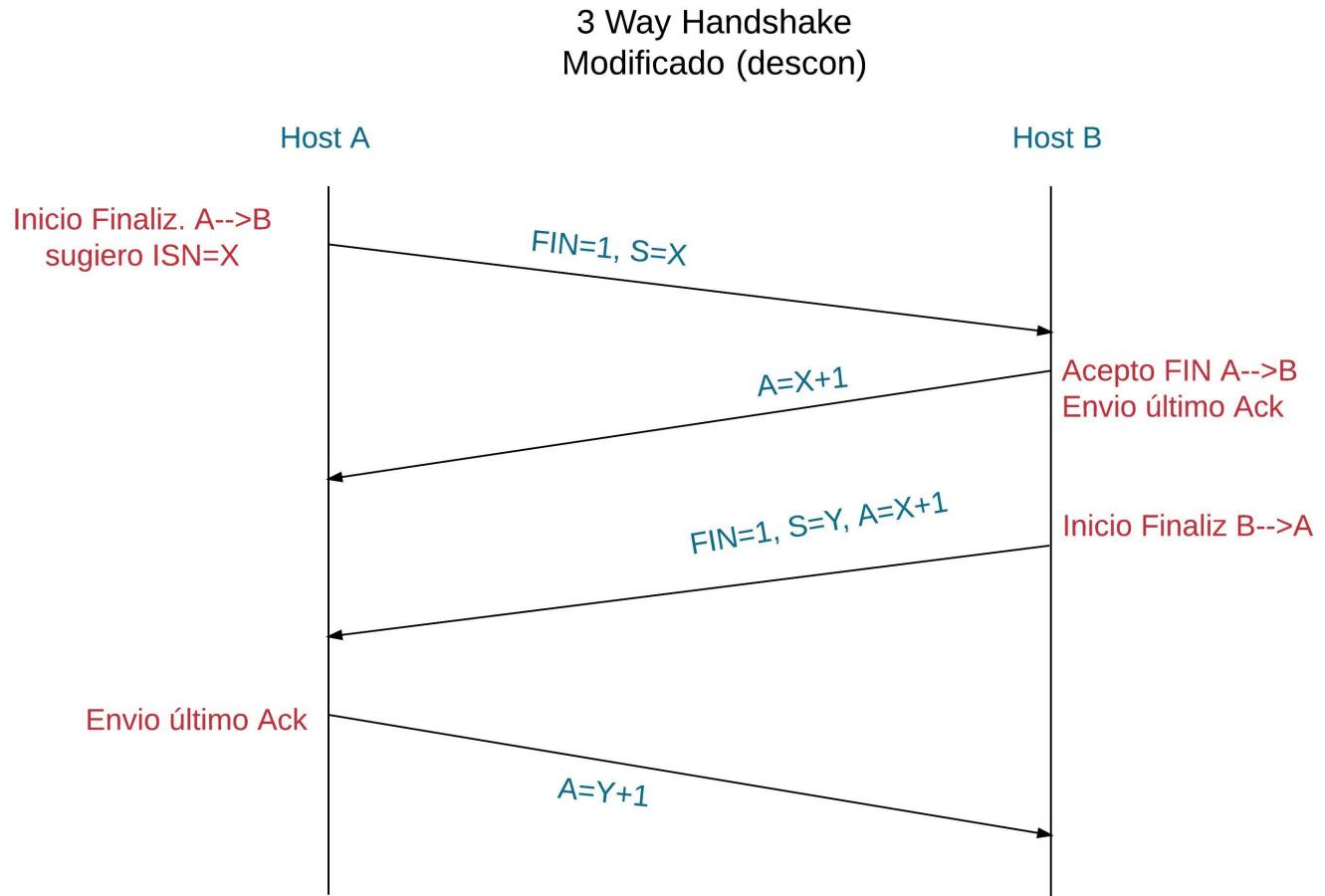
– Establecimiento de una Conexión

3 Way Handshake



Introducción TCP (24)

– Finalización de una Conexión



Introducción UDP (1)

- UDP (User Datagram Protocol)
 - Sin conexión
 - Sin control de flujo
 - Sin confiabilidad
 - No hay recuperación de errores
 - Aplicaciones:
 - TFTP
 - DNS
 - Aplicaciones tiempo real

Introducción UDP (2)

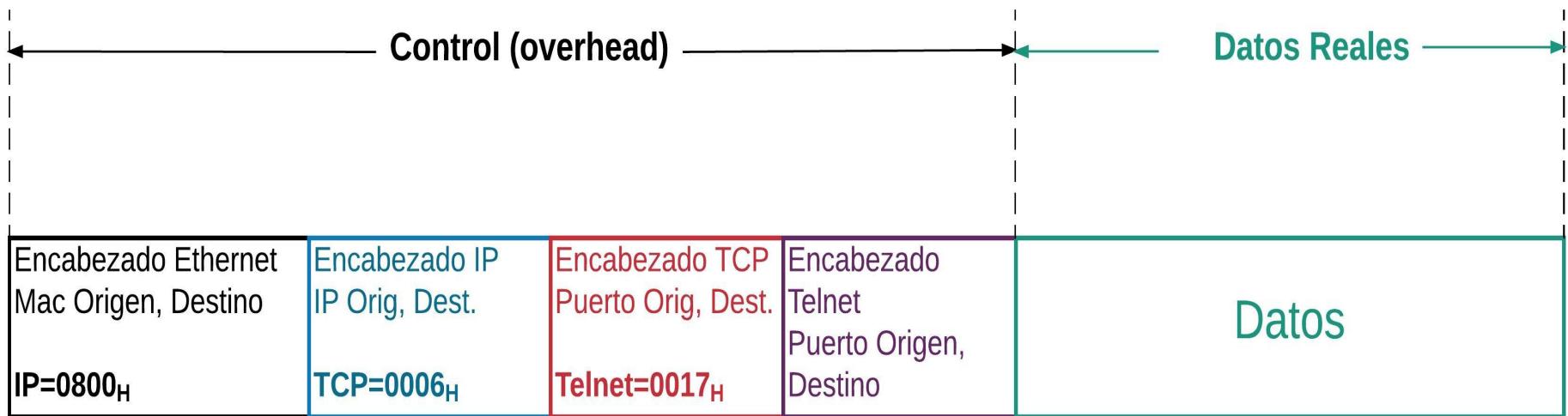
– Formato de Protocolo

Protocolo UDP



- Longitud: datos + encabezado
- Suma Verificación: para encabezado+ datos y seudoencabezado (incluye direcciones IP)

Datos Reales (1)



Introducción a Aplicaciones en Red (1)

- Ahora, queremos dar otra visión a la pregunta: que es Internet?
- Podemos contestar de 2 formas
- Como lo hicimos en este curso: Internet es una colección de componentes
 - Red: hosts + subredes
 - Subredes: routers + switches
 - Protocolos principales de una arquitectura
 - etc.

Introducción a Aplicaciones en Red (2)

- O, podemos usar otro punto de vista:
Desde el punto de vista de Servicio
 - Internet es una infraestructura que provee servicios a Aplicaciones
 - Provee interfaces de programación para que las aplicaciones se conecten en red

Introducción a Aplicaciones en Red (3)

- Arquitectura de de Aplicación
 - Aspectos de implementación de aplicaciones en redes
 - Define estructura de aplicación:
 - Por ejemplo: Cliente – Servidor (lo que se verá en otro curso, aparte de mayor entendimiento protocolos de transporte y routers, direcciones IP)
 - Define modelos de servicio del protocolo de transporte
 - Requerimientos de aplicaciones
 - » Retardo, Rendimiento, Confiabilidad, Seguridad
 - Aplicaciones “reales” y “administrativas

Introducción a Aplicaciones en Red (4)

— Ejemplo de Cliente – Servidor

- Siempre hay 1 host activo (servidor)
- Servidor necesita dirección pública fija para que los clientes lo puedan encontrar
- Los clientes no se comunican directamente
 - Ejemplo:
 - » Navegar en la Web
 - » Servidor: Google
 - » Cliente: Navegadores (firefox, Safari, etc)

Introducción a Aplicaciones en Red (5)

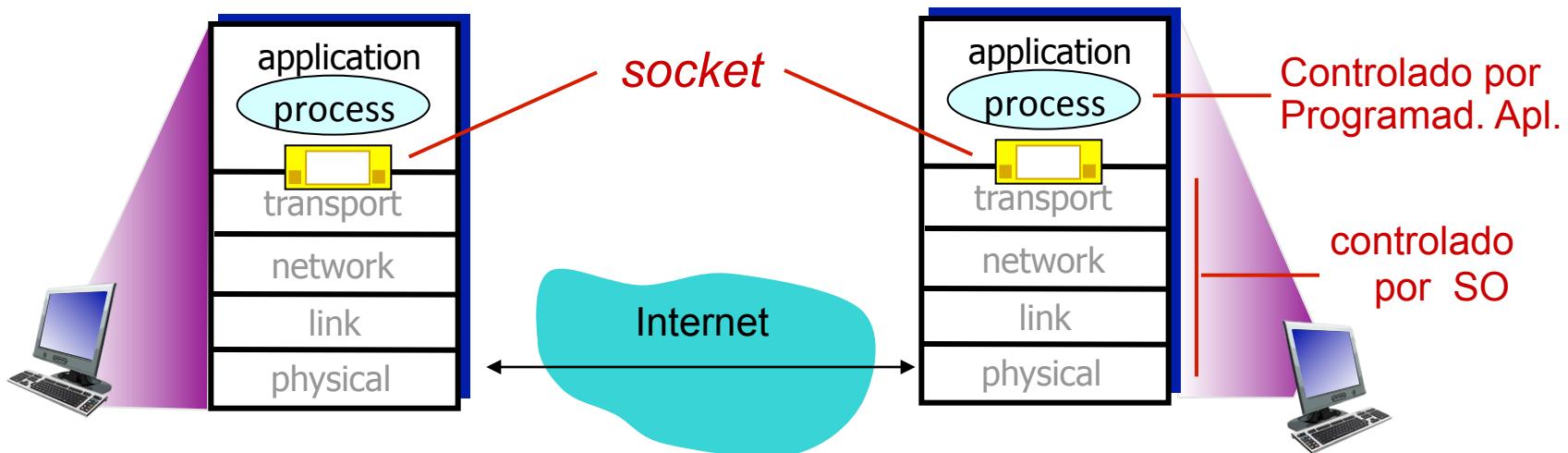
- Concepto de Comunicación de Procesos
 - En sistemas operativos: no son los programas en múltiples hosts los que se comunican, sino más bien, son los procesos dentro de un programa los que se comunican
 - Una API (Application Programming Interface) le especifica a un programa que corre en un host como puede transferir datos a otro programa en otro host mediante procesos
 - Estos procesos intercambian mensajes entre un TX y RX (o un cliente y un servidor)

Introducción a Aplicaciones en Red (6)

- Un proceso envia mensajes a la red y recibe mensajes de la red a través de un **Socket**
- Una analogía:
 - El proceso es análogo a una “casa”
 - El socket es análogo a la puerta de la “casa”
 - Esto permite llegar a la calle (o la red)
- Para enviar un mensaje se hace a través de un socket
 - Este socket es el API entre el nivel de aplicación y el nivel de transporte en un host

Introducción a Aplicaciones en Red (12)

- Este socket es el API entre el nivel de aplicación y el nivel de transporte en un host



Introducción a Aplicaciones en Red (7)

- El programador tiene control absoluto del lado del socket hacia el proceso
- Pero, del lado del socket a Nivel Transporte, es muy poco control, e.g.:
 - Bufer máximo
 - Segmento máximo
- Para identificar un proceso:
 - Dirección IP + Dirección Puerto = Socket
 - Ejemplo: Servidor Web usa puerto 80

Introducción a Aplicaciones en Red (8)

- Que es requerido de los servicios de transporte (por parte de la Aplicación)?
 - Integridad de datos
 - Aplicaciones como FTP requiere 100% de confiabilidad para la transferencia de datos
 - Otras aplicaciones como audio, puede tolerar algunas pérdidas
 - Temporización
 - Algunas aplicaciones, como telefonía en Internet, juegos interactivos, requiere retardos bajos para que la aplicación funcione bien

Introducción a Aplicaciones en Red (9)

- Rendimiento (throughput)
 - Algunas aplicaciones, como multimedia (audio, video), requieren un mínimo de rendimiento para que funcionen bien
 - Si no se logra este rendimiento, las aplicaciones se tienen que codificar con una calidad menor (video, audio). Algunos usan Codificación Adaptiva
 - Otras aplicaciones no tienen problemas para sobrevivir con cualquier ancho de banda
 - Estas aplicaciones se conocen como aplicaciones elásticas: e.g. email, ftp, transferencias de dato por web

Introducción a Aplicaciones en Red (10)

- Seguridad
 - Algunas aplicaciones necesitan encriptación por ejemplo

Introducción a Aplicaciones en Red (11)

- Que pasa si los protocolos de transporte no tienen estas características?
 - Se desarrollan otros protocolos para proveer estas características:
 - Seguridad: SSL, provee seguridad a TCP (encripta)
 - Temporizacion: RTP