Fundamentos de Redes de Computadoras
Grado en Ingeniería Informática
Universidad de Valladolid

Jesús M. Vegas Hernández

Curso 2020-2021

Resumen

Material docente de la asignatura *Fundamentos de Redes de Computadoras* que se imparte en el primer curso del Grado en Ingeniería Informática de la Universidad de Valladolid. El contenido se basa en el libro de texto de James F. Kurose y Keith W. Ross *Computer Networking: a Top-Down Approach* (Pearson, 6/ed, 2012). La mayoría de las figuras están tomadas de este libro de texto o son de creación propia. Cuando no sea así se indicará explícitamente la fuente o se tratará de imágenes de dominio público sin restricciones conocidas.

Índice

1. Redes de Computadoras e Internet

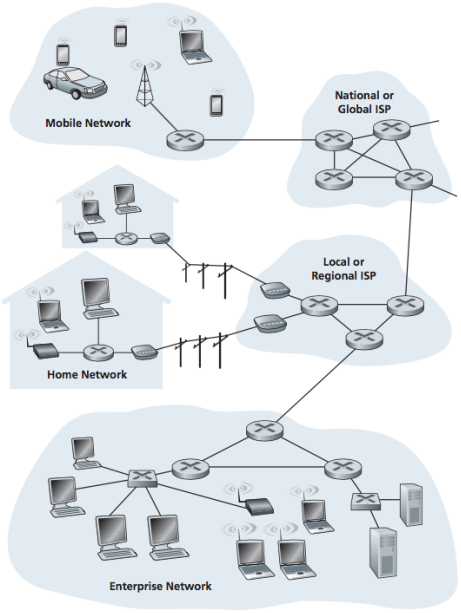
1.1. Qué es Internet?

Qué es Internet?

- Internet como caso de estudio de una red de ordenadores y sus protocolos
- Qué es Internet?
  - Dispositivos
  - Servicios

1.1.1. Descripción basada en Dispositivos

Descripción basada en Dispositivos



Descripción basada en Dispositivos

- Miles de millones de dispositivos conectados, *hosts* o sistemas finales
- Enlaces de comunicaciones y conmutadores de paquetes: *routers* y *switches*
- Acceso a Internet a través de proveedores de servicio de Internet, *ISP*, *Internet Service Provider*
- Protocolos *TCP/IP*

- *Internet Engineering Task Force*, IETF
- *Requests for comments*, RFCs
- Comité de estándares *IEEE 802 LAN/MAN*

1.1.2. Descripción basada en Servicios

Descripción basada en Servicios

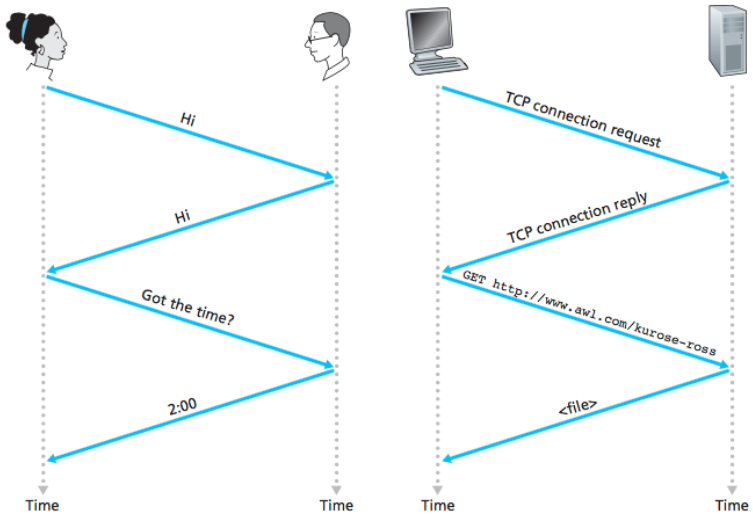
- Internet como infraestructura que proporciona servicios a las aplicaciones
- Aplicaciones ejecutándose en hosts, no en conmutadores o routers
- *Aplicaciones distribuidas*, involucran varios sistemas finales
- *Interfaz de Programación de Aplicaciones*, API (*Application Programming Interface*)

1.1.3. Qué es un Protocolo?

Qué es un Protocolo?

Un *protocolo* define el *formato* y el *orden* de los mensajes intercambiados entre dos o más entidades comunicándose, así como también las *acciones* a tomar en la recepción y/o envío de un mensaje u otro evento

Qué es un Protocolo?

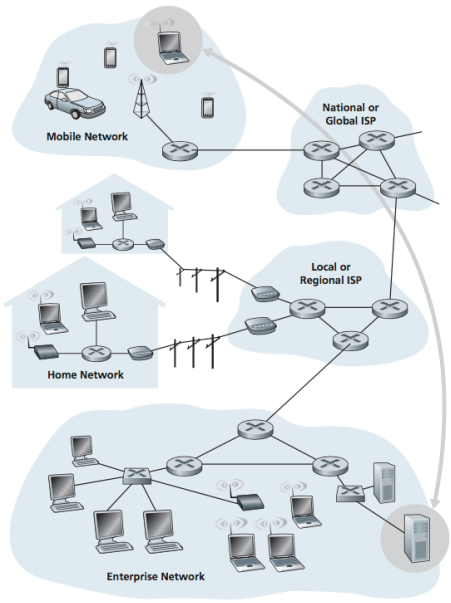


1.2. El Borde de la Red

El Borde de la Red

- Sistemas finales: ordenadores, teléfonos, tabletas, sensores, etc.
- Hosts *clientes* y *servidores*
  - Clientes realizan solicitudes (navegadores web, lectores de correo)
  - que son atendidas por servidores (servidores web, servidores de correo)

Interacción entre Hosts

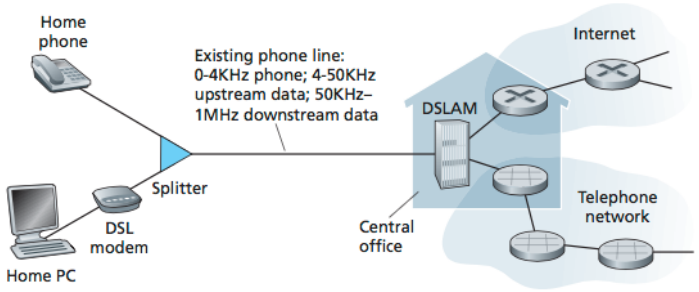


1.2.1. Redes de Acceso

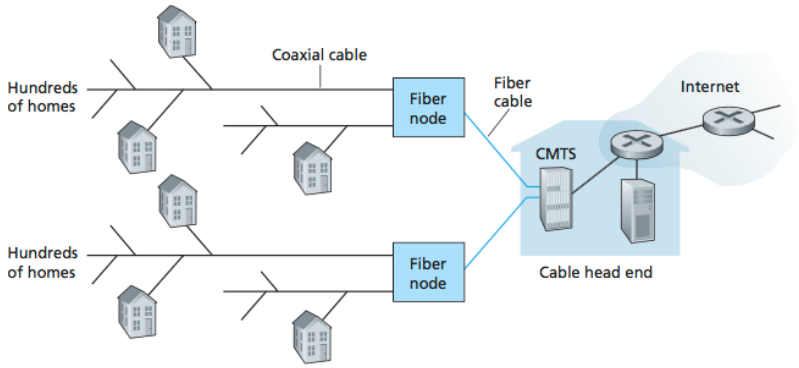
Redes de Acceso

- Redes que conectan físicamente un sistema final al primer router en el camino hacia otro sistema remoto
- Distintos tipos
  - DSL
  - Cable
  - FTTH
  - Modem
  - Satélite
  - 3G y LTE

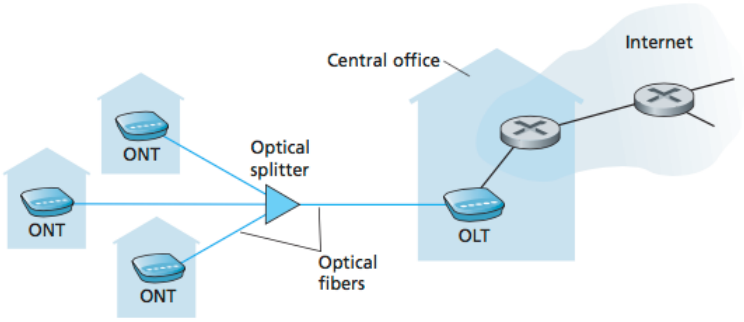
Línea de Subscriptor Digital, xDSL



Híbrido Coaxial-Fibra, HFC



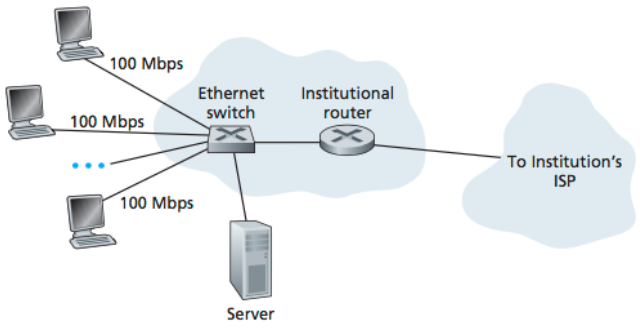
Fibra Hasta el Hogar, FTTH



Redes de Acceso en Organizaciones (y el hogar): Ethernet y WiFi

- Redes de área local, LAN (*Local Area Network*)
- Cable cobre par trenzado: Ethernet 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps
- Inalámbrico: IEEE 802.11 (WiFi)

Redes de Acceso en Organizaciones (y el hogar): Ethernet y WiFi



1.2.2. Medio Físico

Medio Físico

- Emisor transmite un bit al receptor a través del medio físico
- Distintas tecnologías para la transmisión de bits
  - Ondas electromagnéticas (cobre, radio), pulsos ópticos (fibra óptica)
  - Medio guiado o no guiado (inalámbrico)

Par trenzado de Cobre

- El más barato, y más utilizado
- Un par de hilos de cobre, enrollados sobre si mismos para reducir interferencias
- *UTP*, *Unshielded Twisted Pair*
  - De 10 Mbps a 10 Gbps
  - $\leq$ 100 m
  - Actualmente categoría 6a

Cable Coaxial

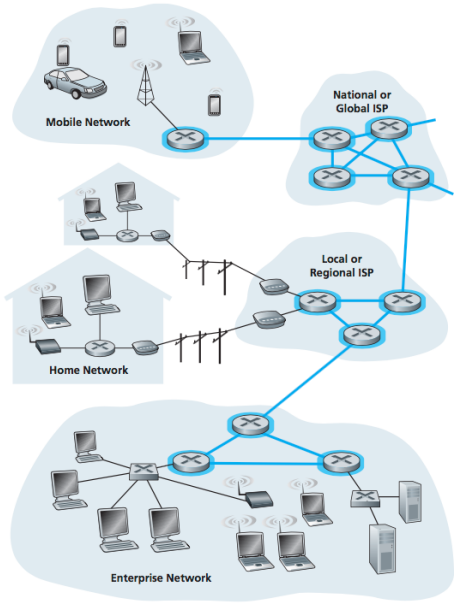
- Dos conectores concéntricos
- Común en transmisión de video y televisión
- Medio guiado compartido

Fibra Óptica

- Hilo fino flexible de material plástico que transmite la luz
- Inmune a interferencias electromagnéticas
- Baja atenuación de la señal, largo alcance ($\leq$ 100 Km)
- Enlaces de alta capacidad y larga distancia

1.3. El Núcleo de la Red

El Núcleo de la Red



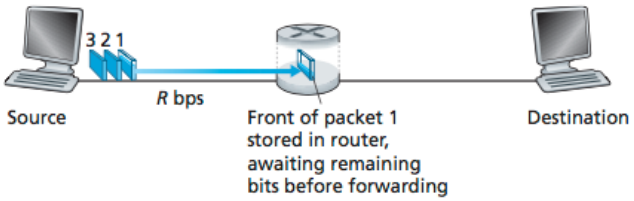
1.3.1. Conmutación de Paquetes

Conmutación de Paquetes

- Mensajes divididos en porciones más pequeñas, *paquetes*
- Cada paquete viaja por los enlaces atravesando *conmutadores de paquetes*
  - Routers
  - Switches
- Los paquetes son transmitidos a la máxima velocidad que permite el enlace
  - Paquete de L bits
  - Enlace de R bits/s (bps)
  - L/R s para transmitir el paquete por el enlace

Almacenamiento y Reenvío

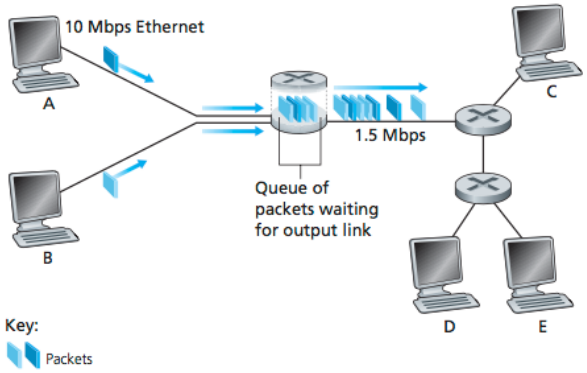
- La mayoría de conmutadores operan en modo *almacenamiento y reenvío*
- Deben recibir el paquete completamente antes de empezar a reenviarlo



- Retardo entre origen y destino? $2L/R$
- Si N enlaces ($N-1$ routers), retardo entre extremos $N\frac{L}{R}$

Retardo de Cola y Pérdida de Paquetes

- Conmutador paquetes con múltiples enlaces de entrada y salida
- Si un enlace salida ocupado, el paquete debe esperar en una *cola*
- Si cola llena, el *paquete se pierde*



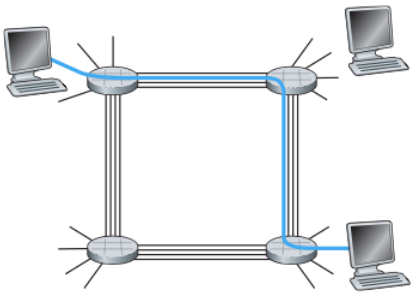
Tablas de Reenvío y Protocolos de Encaminamiento

- Cómo sabe un router el enlace de salida para un paquete?
- Cada host identificado con *dirección IP*
- Cada paquete incluye dirección IP de host *origen* y *destino*
- En routers, *tabla de reenvío* asocia dirección IP destino con enlaces de salida
- Cómo se puebla la tabla de reenvío en cada router?
- *Protocolos de encaminamiento* encuentran el camino más corto

1.3.2. Conmutación de Circuitos

Conmutación de Circuitos

- Reserva de recursos en la ruta para mensajes entre origen y destino, un *circuito*
- Alternativa a la conmutación de paquetes, derivada de compañías telefónicas
- División por tiempo o por frecuencias

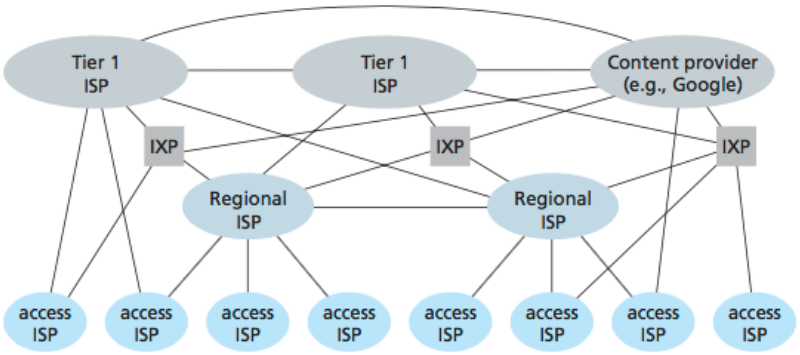


Paquetes vs Circuitos

- Cuál es más adecuado? Conmutación de paquetes o de circuitos?
- Aplicaciones de tiempo real
- Aplicaciones que no envían paquetes constantemente, periodos de inactividad

1.3.3. Red de Redes

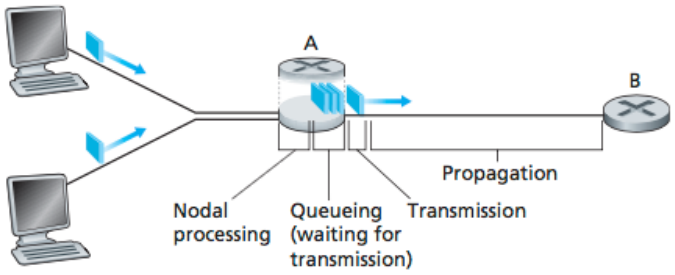
Red de Redes



1.4. Retardos, Pérdidas y Rendimiento en Redes de Conmutación de Paquetes

1.4.1. Visión General del Retardo en Redes de Conmutación de Paquetes

Visión general del Retardo en Redes de Conmutación de Paquetes

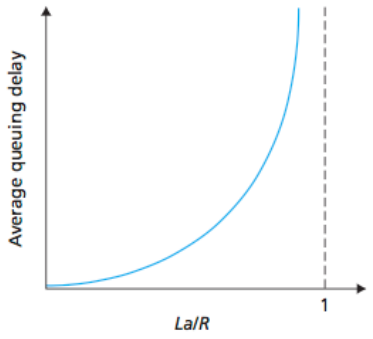


- $d_{nodal} = d_{proc} + d_{cola} + d_{trans} + d_{prop}$

1.4.2. Retardo de Encolado y Pérdida de Paquetes

Retardo en Encolado y Pérdida de Paquetes

- d_{cola} es variable entre paquetes
- a, tasa media de llegada de paquetes/s
- R, tasa de transmisión, bps
- La/R, tasa media de llegada de bits, bps, *intensidad del tráfico*



1.4.3. Retardo entre Extremos

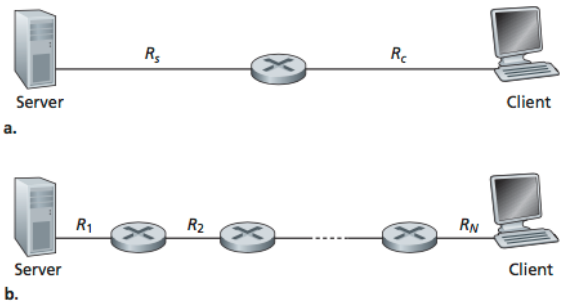
Retardo entre Extremos

```
jvegas — -bash — 89×13
dhcp214:~ jvegas$ traceroute www.google.es
traceroute to www.google.es (216.58.201.131), 64 hops max, 52 byte packets
 1  157.88.124.252 (157.88.124.252)  0.777 ms  0.653 ms  0.687 ms
 2  157.88.29.181 (157.88.29.181)  0.954 ms  1.486 ms  1.002 ms
 3  ge3-0-2.uva.rt1.cyl.red.rediris.es (130.206.201.1)  0.425 ms  0.428 ms  0.416 ms
 4  uva.ae2.ciemat.rt1.mad.red.rediris.es (130.206.245.9)  3.316 ms  35.325 ms  3.439 ms
 5  unizar.ae6.telmad.rt4.mad.red.rediris.es (130.206.245.94)  10.740 ms
    ciemat.ae2.telmad.rt4.mad.red.rediris.es (130.206.245.2)  3.974 ms  3.886 ms
 6  google-router.red.rediris.es (130.206.255.2)  17.695 ms  17.371 ms  10.816 ms
 7  72.14.235.18 (72.14.235.18)  4.470 ms  4.460 ms  11.089 ms
 8  216.239.40.217 (216.239.40.217)  4.278 ms  11.291 ms  11.034 ms
 9  mad06s25-in-f3.1e100.net (216.58.201.131)  10.941 ms  10.956 ms  10.949 ms
dhcp214:~ jvegas$
```

1.4.4. Rendimiento en Redes de Computadoras

Rendimiento en Redes de Computadoras

- Rendimiento *instantáneo* y *medio* (bps)



- Rendimiento es $min(R_1, \ldots, R_N)$, la tasa de transmisión del *cuello de botella*

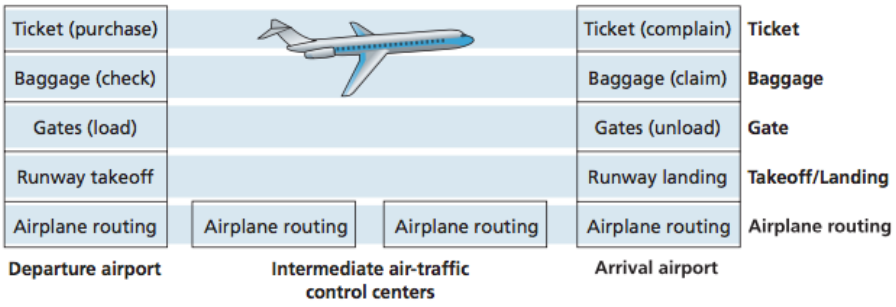
1.5. Protocolos y Modelos de Servicio

Protocolos y Modelos de Servicio

- Internet es un sistema extremadamente complejo: aplicaciones, protocolos, hosts, conmutadores, enlaces
- *Arquitectura de Red*

1.5.1. Arquitectura en Capas

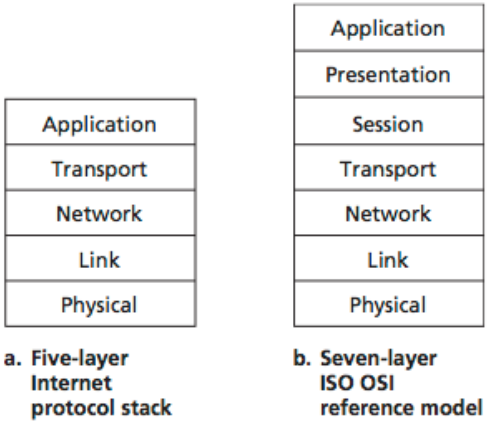
Arquitectura en Capas



Pila de Protocolos

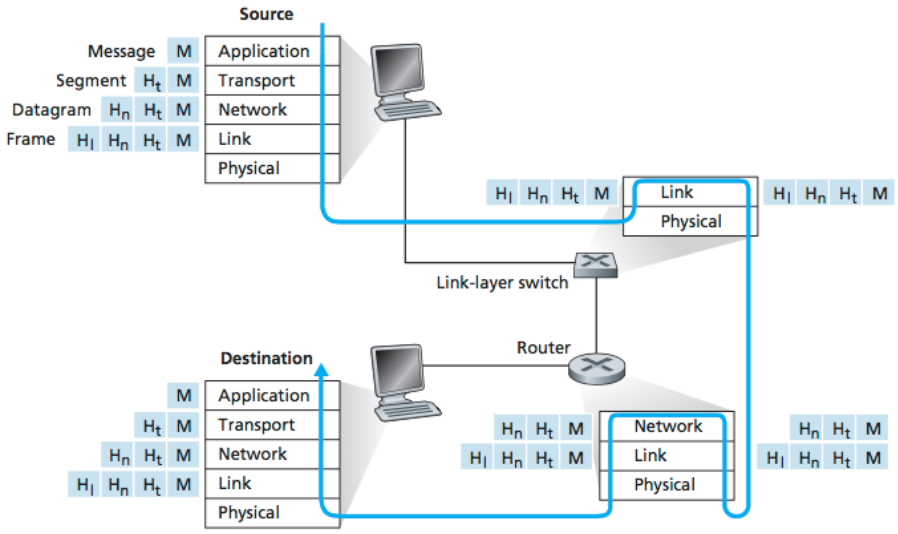
- Arquitectura de red basada en pila de protocolos
- No importa cómo está implementada cada capa, sino el *modelo de servicio* de cada capa a la capa superior
- Modularidad
- Riesgo de duplicar funcionalidad

Pilas TCP/IP e ISO/OSI



1.5.2. Encapsulación

Encapsulación

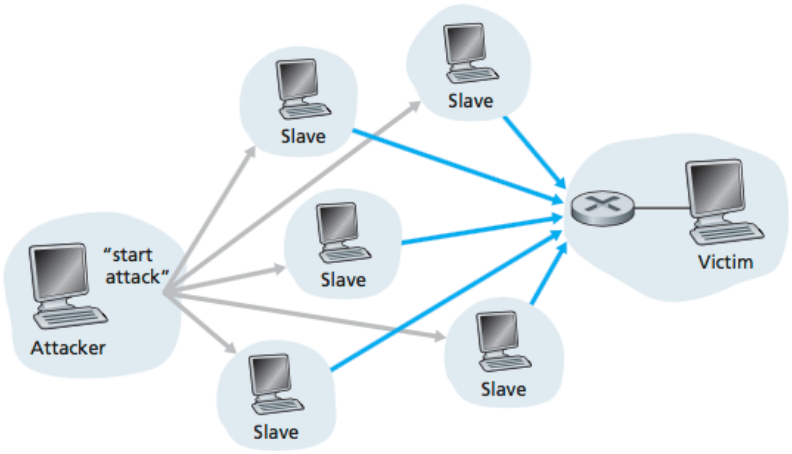


1.6. Ataques a Redes

Ataques a Redes

- Instalar *malware* en hosts accesibles en la red
  - Autoreplicante, *virus* o *gusanos*
  - *Botnet*
- Ataques a la red e infraestructura
  - Denegación de servicio distribuida, *DDoS*, normalmente para consumir recursos
  - *Sniffer*, husmear, fisgar
  - *IP Spoofing*, suplantación

DDoS



1.7. Resumen

Resumen

- Internet como ejemplo de red
- Borde de la red, con sistemas y aplicaciones
- Tecnologías de enlace y físicas en las redes de acceso
- Redes de conmutación de paquetes y de circuitos en el núcleo de la red
- Causas del retardo, la pérdida de paquetes y el rendimiento de la red
- Modelo cuantitativo para el retardo de cola, transmisión y propagación
- Arquitectura de redes multicapa y modelo de servicio
- Ataques a redes y hosts

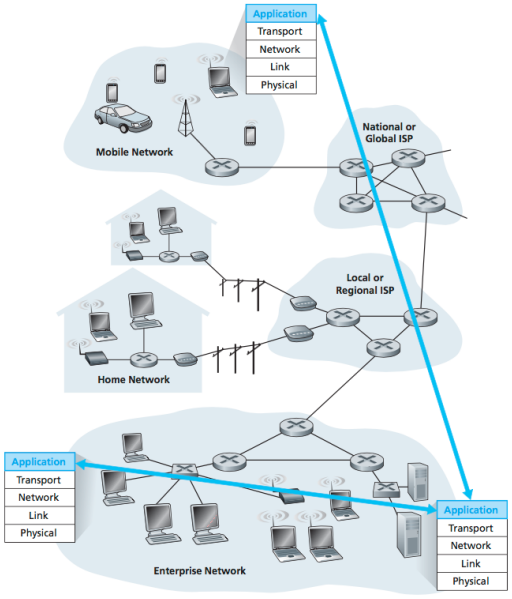
2. Capa de Aplicación

2.1. Principios de Aplicaciones en Red

Principios de Aplicaciones en Red

- Escribir programas que se ejecuten en distintos sistemas finales y se comuniquen a través de la red
- No se necesita escribir ni ejecutar programas en dispositivos de red, switches y routers

Principios de Aplicaciones en Red



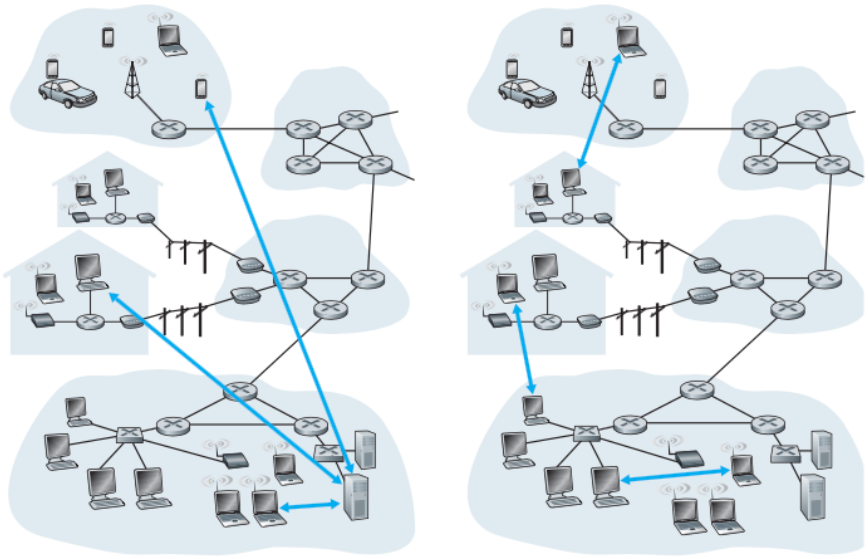
2.1.1. Arquitecturas de Aplicaciones en Red

Arquitecturas de Aplicaciones en Red

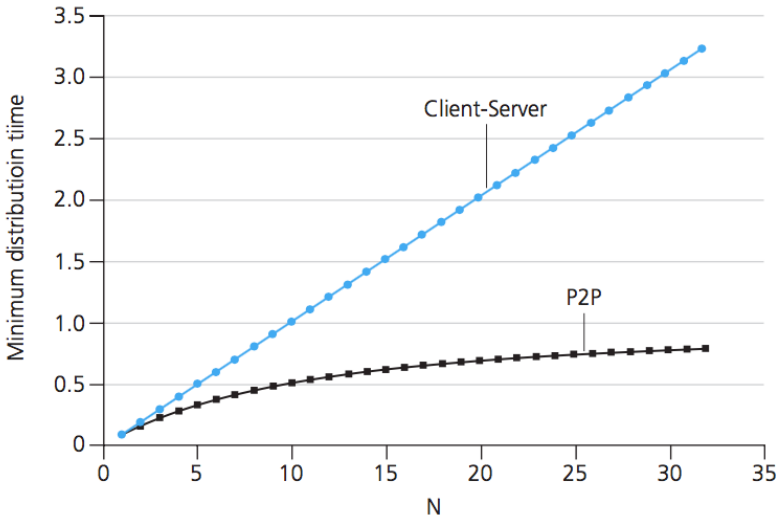
- *Arquitectura de aplicación*, es elegida por el desarrollador y determina cómo se estructura la aplicación sobre los sistemas finales
- Arquitectura *Cliente-Servidor*
  - Un host (*servidor*), acepta y responde peticiones de otros hosts (*cliente*)
  - Clientes no se comunican directamente entre sí
  - *Centros de datos* agrupando servidores

- Arquitectura *Peer-to-Peer*, P2P
  - Comunicación directa entre pares de hosts conectados intermitentemente (*Pares*, *peers*)
  - *Autoescalabilidad*
- Soluciones híbridas

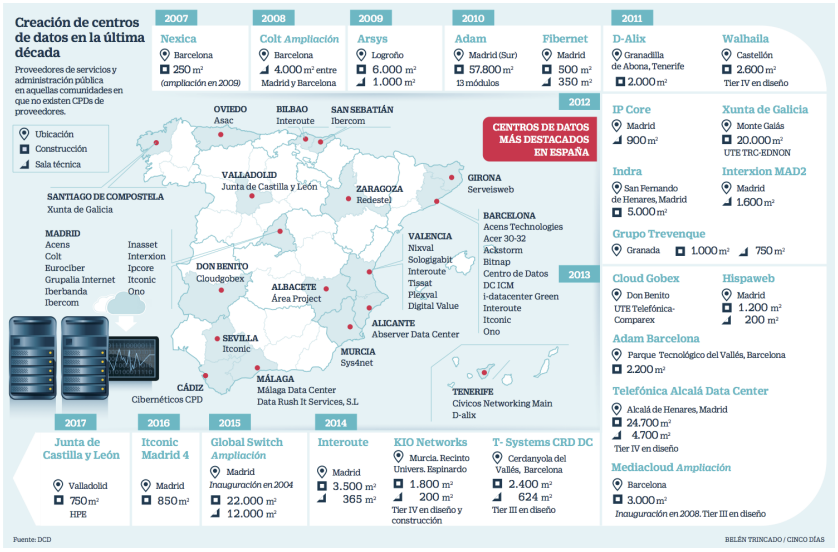
Arquitecturas de Aplicaciones en Red



Distribución de Archivos P2P vs Cliente-Servidor



Centros de Datos



[Cinco Días]

2.1.2. Procesos en Comunicación

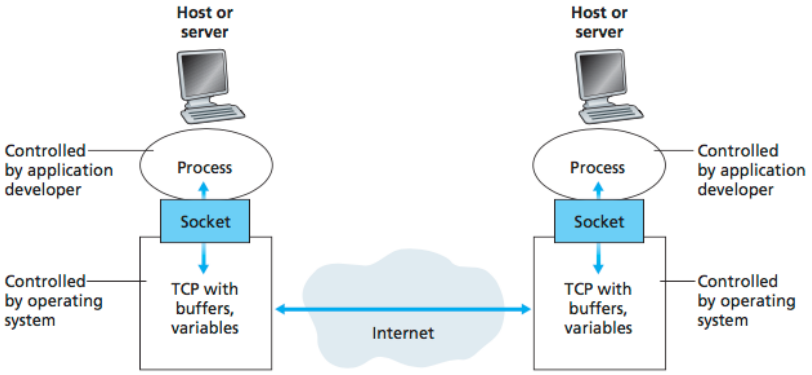
Procesos en Comunicación

- *Proceso*, programa en ejecución en host
- Procesos se comunican entre sí intercambiando *mensajes*

- Varios mensajes pueden formar parte de una misma comunicación, *sesión*

El proceso que inicia la comunicación (contacta a otro proceso para abrir una *sesión*) es etiquetado como *cliente*, el proceso que espera a ser contactado para iniciar una *sesión* es el *servidor*

Socket: Interfaz entre Proceso y Red



Identificar Procesos

- Dirección del host, *Dirección IP* (*Internet Protocol*)
  - IPv4, 32-bit en notación "punto decimal" (`192.0.2.53`)
  - IPv6, 128-bit en hexadecimal (`2001:0db8:582:ae33::29`)
- Dirección del proceso en el host, *puerto* (*port*)
  - Puertos bien conocidos, o de sistema (0-1023)
  - Puertos de usuario y dinámicos (1024-65535)

[iana.org]

2.1.3. Servicios de Transporte Disponibles para las Aplicaciones

Servicios de Transporte Disponibles para las Aplicaciones

- Protocolos de transporte tienen la responsabilidad de entregar los mensajes en el socket destino
- Opciones de transporte
  - Transferencia fiable
  - Rendimiento
  - Temporización
  - Seguridad

2.1.4. Servicios de Transporte Ofrecidos por Internet

Servicios de Transporte Ofrecidos por Internet

Aplicación	Pérdida Datos	Rendimiento	Sensible Tiempo
Transferencia de archivos	Sin pérdida	Elástica	No
Correo electrónico	Sin pérdida	Elástica	No
Web	Sin pérdida	Elástica	No
VoIP	Tolerante	Kbps–Mbps	Si $O(10^2)$ ms
Video *streaming*	Tolerante	Kbps–Mbps	Sí $O(10^3)$ ms
Juegos interactivos	Tolerante	Kbps–10 Kbps	Si $O(10^2)$ ms
Mensajería instantánea	Sin pérdida	Elástica	Si/No

Servicios de Transporte Ofrecidos por Internet

- *TCP*, *Transport Control Protocol*
  - Orientado a conexión
  - Transferencia fiable
  - Control de congestión
- *UDP*, *User Datagram Protocol*
  - No orientado a conexión
  - Transferencia no fiable
  - Sin control de congestión
- Por qué usar UDP si existe TCP?
- Servicios no aportados por Internet
  - Rendimiento
  - Temporización
  - Seguridad

Protocolos de Transporte Utilizados por Aplicaciones

Aplicación	Protocolo Aplicación	Protocolo Transporte
Correo electrónico	SMTP [RFC 5321]	TCP
Acceso a terminal remoto	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transferencia de archivos	FTP [RFC 959]	TCP
Streaming multimedia	HTTP [RFC 2616]	TCP
Telefonía sobre Internet	SIP [RFC 3261], RTSP [RFC 3550] o Skype	UDP/TCP

2.1.5. Protocolos de la Capa de Aplicación

Protocolos de la Capa de Aplicación

- *Protocolos de la capa de aplicación* definen el modo en que intercambian mensajes procesos de aplicación ejecutándose en hosts distintos (o no)
  - Tipos de mensajes
  - Sintaxis
  - Semántica
  - Reglas para enviar/responder mensajes
- Protocolos *abiertos*/*propietarios*
- Aplicación en red $\supset$ Protocolos de la capa de aplicación

Aplicaciones Estudiadas

- Web
- Transferencia de archivos
- Correo electrónico
- Servicio de directorio
- ~~P2P~~

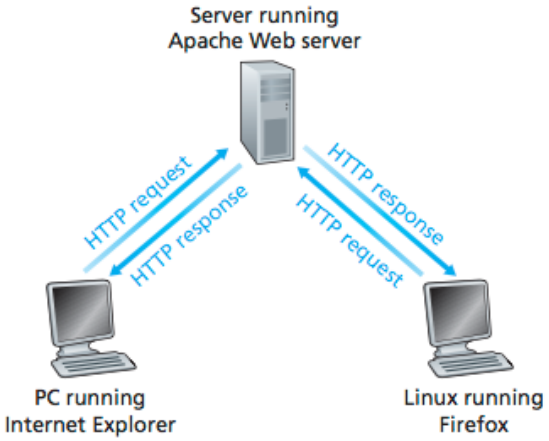
2.2. La Web y HTTP

2.2.1. Visión General de HTTP

Visión General de HTTP

- *HyperText Transfer Protocol*, [RFC 1945] [RFC 2616]
- TCP
- Cliente (navegador, *browser*) - Servidor (servidor web, puerto 80)
- *Sin estado*, servidor HTTP no guarda información sobre clientes
- *Página Web* colección objetos (archivos html, jpeg, etc.) accesible por *URL* (*Uniform Resource Locator*)
  - `http://www.sitio.com/directorio/archivo.gif`

Solicitud-Respuesta HTTP



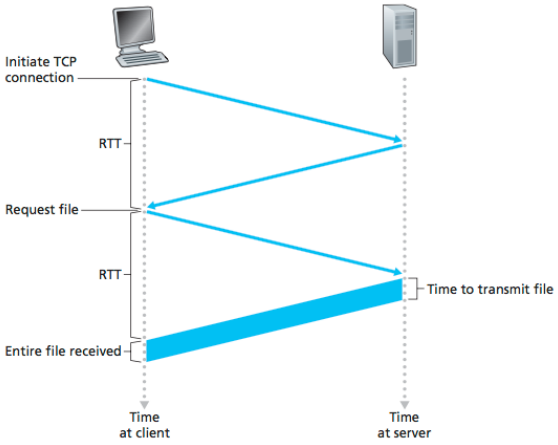
2.2.2. Persistencia de las Conexiones

Persistencia de las Conexiones

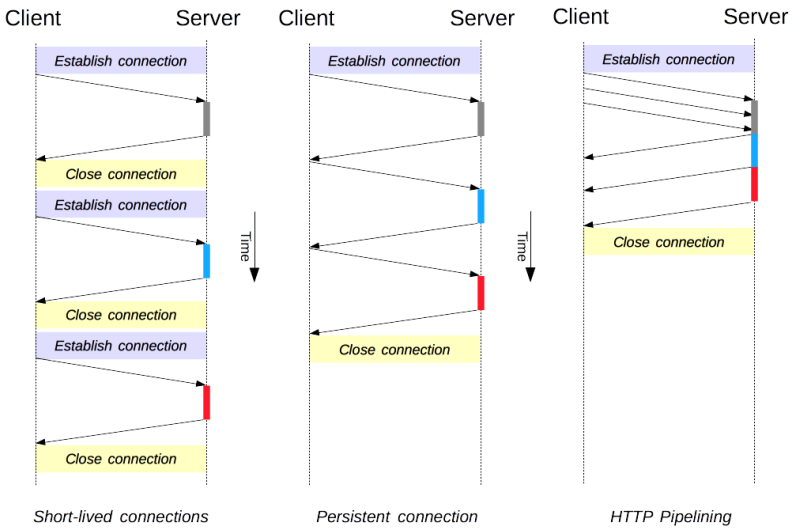
- Solicitudes y respuestas HTTP transportadas sobre TCP
- Decisión: enviar cada solicitud-respuesta en una conexión TCP separada o mantener la conexión TCP abierta para varias?
- *Persistencia*

Tiempo de ida y vuelta, RTT

RTT, *Round-Trip Time*
Intervalo de tiempo que tarda un paquete pequeño en ir del cliente al servidor y vuelta



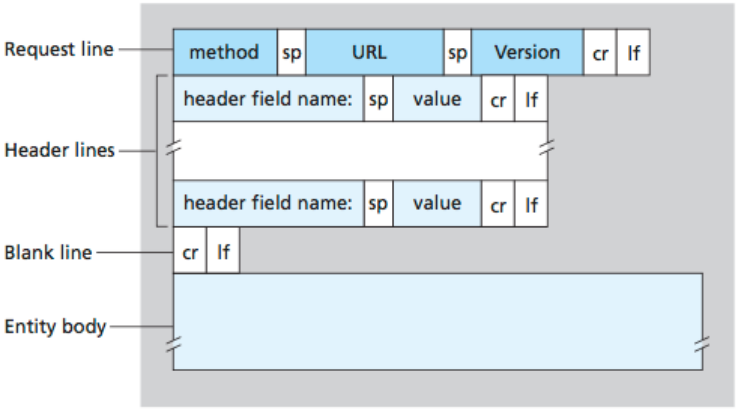
Conexiones HTTP persistentes *pipelining*



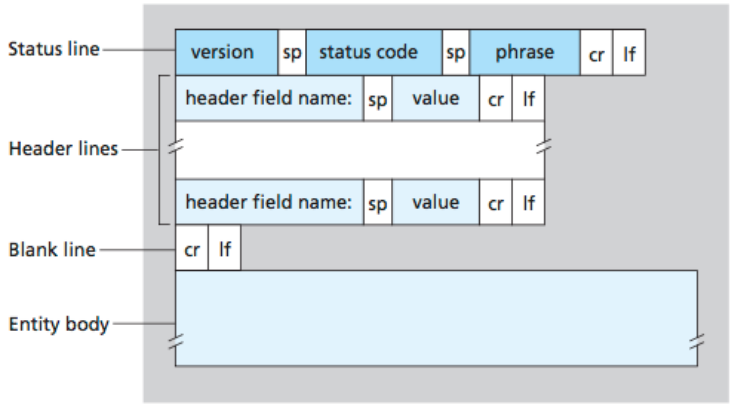
[Mozilla Contributors]

2.2.3. Formato Mensajes HTTP

Formato Solicitud HTTP

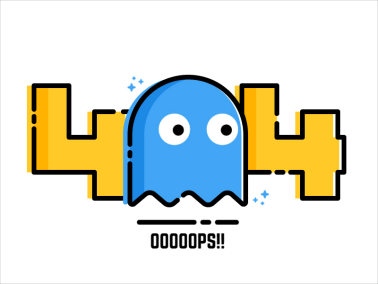


Formato Respuesta HTTP



Códigos Estado

- *1xx* Informativo, solicitud recibida y entendida
- *2xx* Solicitud recibida, entendida y aceptada
- *3xx* Redirección, el cliente debe realizar acciones adicionales
- *4xx* Error de cliente
- *5xx* Error de servidor



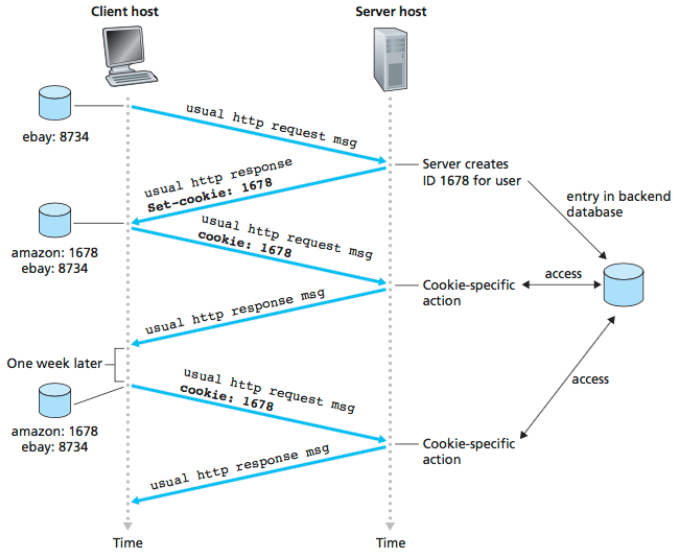
[Arturo Muñoz @ Dribbble]

Ejemplo Solicitud-Respuesta HTTP

```
jvegas — telnet frc.lab.inf.uva.es 80 — 100×37
dhcp216:~ jvegas$ telnet frc.lab.inf.uva.es 80
Trying 157.88.125.187...
Connected to frc.lab.inf.uva.es.
Escape character is '^]'.
GET / HTTP/1.1
Host: frc.lab.inf.uva.es

HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Tue, 13 Feb 2018 10:21:34 GMT
Content-Type: text/html
Content-Length: 412
Last-Modified: Thu, 08 Feb 2018 11:18:12 GMT
Connection: keep-alive
ETag: "5a7c31f4-19c"
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<head>
<title>Bienvenidos a Fundamentos de Redes!</title>
<style>
    body {
        width: 45em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Bienvenidos a Fundamentos de Redes!</h1>
<p>Si ves esta p&aacute;gina es que el servidor web est&aacute; instalado y funcionando.</p>

<p><em>Saludos.</em></p>
</body>
</html>
```

2.2.4. Interacción Usuario-Servidor: Cookies

Interacción Usuario-Servidor: *Cookies*



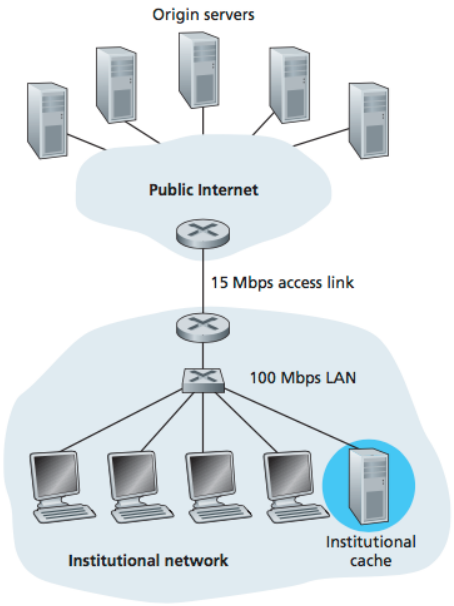
[chrome://settings/siteData]

2.2.5. Caché Web

Caché Web

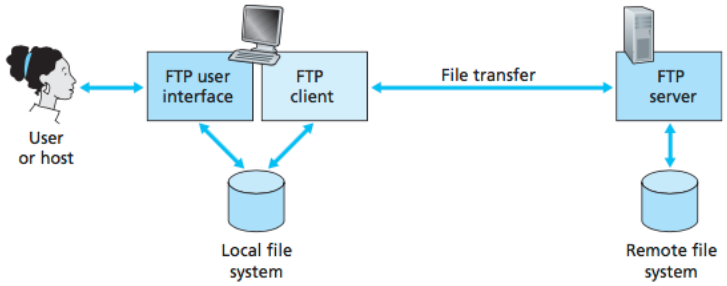
- *Caché Web* (o también *proxy web*) es una entidad de red que satisface solicitudes HTTP en lugar del servidor web original
  - Reducción tiempo respuesta
  - Reducción tráfico en segmento salida a Internet
  - Control sobre contenidos accedidos
- *CDNs*, *Content Distribution Networks*
- Get condicional

Caché Web



2.3. FTP, Transferencia de Archivos

FTP, Transferencia de Archivos

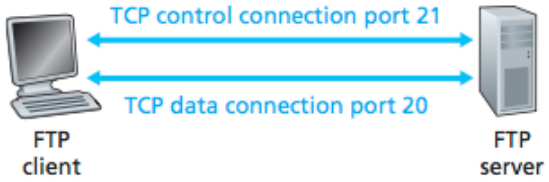


FTP, Transferencia de Archivos

- FTP, *File Transfer Protocol* [RFC 959]
- Conexión de datos (TCP puerto 20)
- Conexión de control *fuera de banda* (TCP puerto 21)
  - Diferencia con HTTP, información de control enviada *en banda*
- FTP mantiene información de estado del usuario
- Comandos ASCII 7 bits
  - `USER nombreUsuario`
  - `PASS contrasenia`
  - `LIST`
  - `RETR archivo`
  - `STOR archivo`

[Tabla ASCII]

FTP, Conexiones de Control y Datos



Comandos y Respuestas FTP

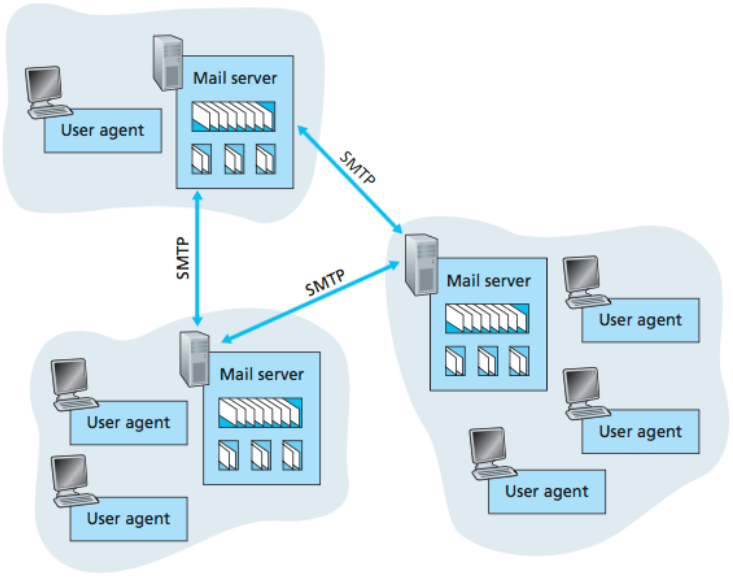
```
jvegas — -bash — 100×30
dhcp216:~ jvegas$ ftp
ftp> open frc.lab.inf.uva.es
Connected to frc.lab.inf.uva.es.
220 (vsFTPd 3.0.3)
Name (frc.lab.inf.uva.es:jvegas): hackme0
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||22210|)
150 Here comes the directory listing.
-rw-------    1 1001     1001        36422 Feb 27  2017 CP1.pdf
dr-xr-xr-x    3 65534    65534        4096 Feb 27  2017 ftp
-rw-------    1 1001     1001        24599 Feb 27  2017 rfc1350.txt
-rw-------    1 1001     1001       147316 Feb 27  2017 rfc959.txt
-rw-------    1 1001     1001         9892 Feb 27  2017 ventana.png
226 Directory send OK.
ftp> get rfc959.txt
local: rfc959.txt remote: rfc959.txt
229 Entering Extended Passive Mode (|||14194|)
150 Opening BINARY mode data connection for rfc959.txt (147316 bytes).
100% |**********************************************************|   143 KiB    5.62 MiB/s    00:00 ETA
226 Transfer complete.
147316 bytes received in 00:00 (5.45 MiB/s)
ftp> close
221 Goodbye.
ftp> quit
dhcp216:~ jvegas$
```

2.4. Correo Electrónico

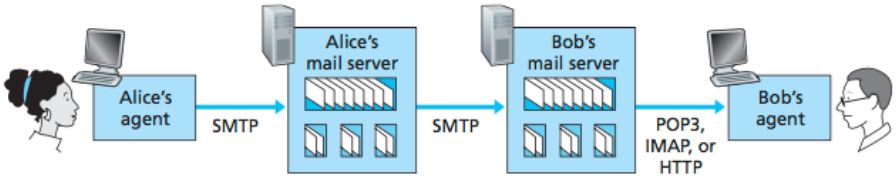
Correo Electrónico

- Aún una de las aplicaciones de Internet más utilizadas e importantes
- Agentes de usuario, Servidores de correo
- *SMTP*, *Simple Mail Transfer Protocol* [RFC 5321]
  - Cliente, servidor enviando
  - Servidor, servidor recibiendo
  - TCP, puerto 25
  - Protocolo Push, (HTTP es *pull*)
  - ASCII 7 bits $\rightarrow$ *MIME Multipart Internet Mail Extension* [RFC 2045] [RFC 2046]
- *Buzón*, alojado en un servidor donde se reciben los mensajes de un usuario
- Protocolos de acceso al correo
  - *POP3*, *Post Office Protocol* versión 3 [RFC 1939]
  - *IMAP*, *Internet Mail Access Protocol* [RFC 3501]
  - *HTTP*

SMTP



Envío y Recepción de Correo Electrónico



2.5. DNS, Servicio de Directorio de Internet

DNS, Servicio de Directorio de Internet

- Servicio de directorio: hostname ↔ dirección IP
- *DNS*, *Domain Name System* [RFC 1034] [RFC 1035]
  - Base de datos distribuida sobre nombres, direcciones (y más) de hosts
  - Protocolo capa aplicación para consultas
  - Puerto 53 UDP
- Utilizado por otras apps como paso previo

Nombre de Dominio
cadena de identificación que define un ámbito de autonomía administrativa, autoridad o control dentro de Internet

[https://www.iana.org/domains]

2.5.1. Servicios DNS

Servicios DNS

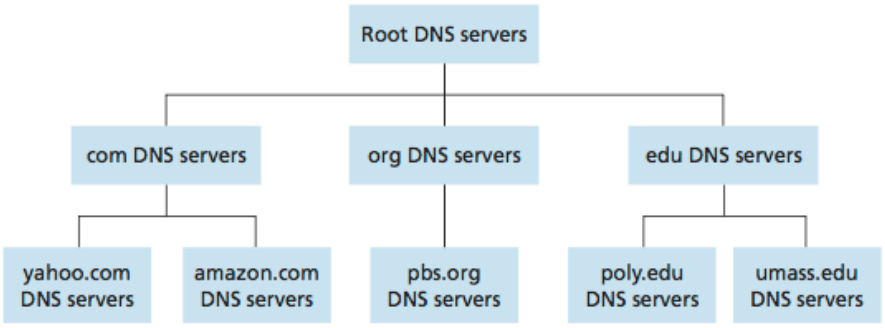
- Traducción directa o *inversa* entre nombres y direcciones
- Permitir que un host tenga uno o más *alias* además de su *nombre canónico*
- Alias para *servidor de correo* (mnemotécnico)
- Distribución de carga, varias direcciones IP para un nombre canónico

2.5.2. Visión General del Funcionamiento de DNS

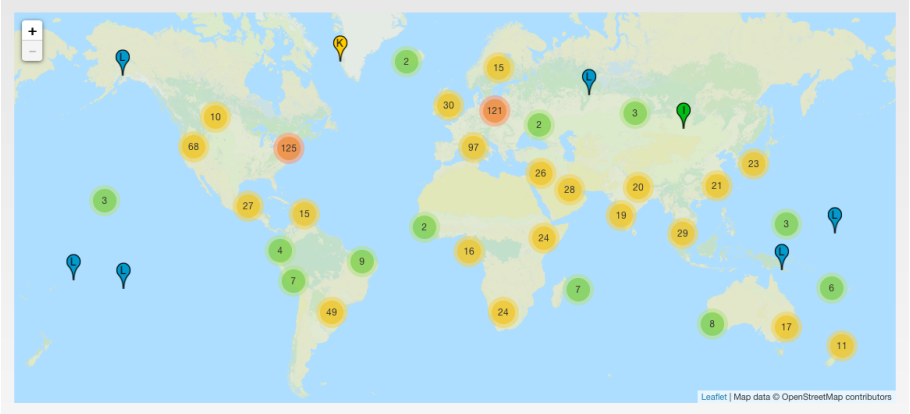
Visión General del Funcionamiento de DNS

- BD jerárquica y distribuida
  - Alternativa centralizada: punto único de fallo, problemas de tráfico, distancia y mantenimiento
- 3 niveles de servidores
  - *Raíz*, *Root*
    - 13, replicados en cientos de instancias
  - *Dominio de primer nivel*, *Top Level Domain*, TLD
    - `.arpa`, `.com`, `.edu`, `.gov`, `.mil`, `.net`, `.org`
  - *Autorizados*, *Authoritative*
    - Registros DNS de los hosts públicamente accesibles en Internet de cada organización
- Aún hay otro servidor más
  - *Local*, o *por defecto*
    - Punto de entrada a DNS para los hosts

BD *Jerárquica* y Distribuida

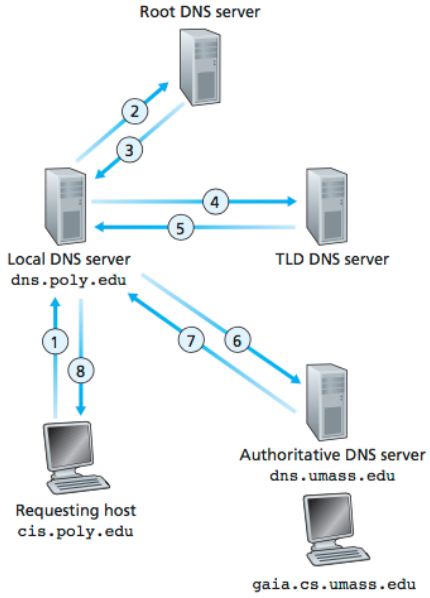


BD Jerárquica y *Distribuida*



[http://www.root-servers.org]

Consultas DNS y Caché

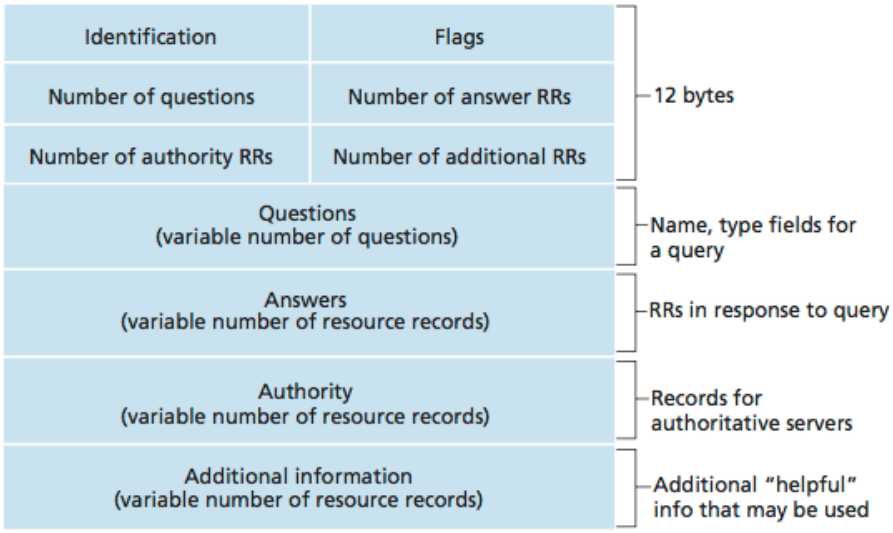


2.5.3. Registros y mensajes DNS

Registros DNS

- (Nombre, Valor, Tipo, TTL)
  - *TTL* tiempo de validez en caché (s)
  - *Nombre* y *valor* dependen del *tipo* de registro
- `(inf.uva.es, 157.88.109.243, A, 3600)`
- `(inf.uva.es, ns1.inf.uva.es, NS, 3600)`
- `(eiiva.uva.es, inf.uva.es, CNAME, 3600)`
- `(inf.uva.es, mx01.puc.rediris.es, MX, 3600)`

Mensajes DNS




```
$ dig inf.uva.es any
```

```
jvegas — -bash — 122×38
vegax:~ jvegas$ dig inf.uva.es any

; <<>> DiG 9.8.3-P1 <<>> inf.uva.es any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62896
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 9

;; QUESTION SECTION:
;inf.uva.es.                    IN      ANY

;; ANSWER SECTION:
inf.uva.es.             3600    IN      A       157.88.109.243
inf.uva.es.             3600    IN      NS      ns1.uva.es.
inf.uva.es.             3600    IN      NS      ns2.uva.es.
inf.uva.es.             3600    IN      NS      ns1.inf.uva.es.
inf.uva.es.             3600    IN      NS      ns2.inf.uva.es.
inf.uva.es.             3600    IN      MX      10 mx01.puc.rediris.es.
inf.uva.es.             3600    IN      MX      10 mx02.puc.rediris.es.
inf.uva.es.             3600    IN      TXT     "v=spf1 ip4:157.88.0.0/16 include:spf.puc.rediris.es -all"
inf.uva.es.             3600    IN      SPF     "v=spf1 ip4:157.88.0.0/16 include:spf.puc.rediris.es -all"
inf.uva.es.             3600    IN      SOA     ns1.inf.uva.es. informatica.inf.uva.es. 2018020900 28800 900 2419200 7200

;; ADDITIONAL SECTION:
ns1.inf.uva.es.         3600    IN      A       157.88.109.249
ns1.uva.es.             86400   IN      A       157.88.18.190
ns2.inf.uva.es.         3600    IN      A       157.88.109.248
ns2.uva.es.             86400   IN      A       157.88.18.189
mx01.puc.rediris.es.    139     IN      A       130.206.19.17
mx01.puc.rediris.es.    139     IN      A       130.206.19.25
mx01.puc.rediris.es.    139     IN      A       130.206.19.33
mx02.puc.rediris.es.    298     IN      A       130.206.19.81
mx02.puc.rediris.es.    298     IN      A       130.206.19.89

;; Query time: 28 msec
;; SERVER: 157.88.18.190#53(157.88.18.190)
;; WHEN: Thu Feb 15 17:08:23 2018
;; MSG SIZE  rcvd: 500
```

2.5.4. Insertar Registros en DNS

Insertar Registros en DNS

1. Registrar nombre de dominio (p.e. `frc.es`) en algún *registrador* ([http://www.dominios.es/dominios/])

2. Proporcionar al registrador las IP (obtenida del *ISP*) del servidor de nombres primario (y secundario), que la insertará en el servidor TLD (`.es`)

3. Insertar registros en servidores DNS autorizados (local) para servidor Web y de correo (p.e. `www.frc.es` y `mx.frc.es`)

2.5.5. Vulnerabilidades DNS

Vulnerabilidades DNS

- DNS servicio esencial para funcionamiento de Internet
- Cómo se puede atacar?
  - DDoS, denegación de servicio distribuida, *Distributed Denial-of-Service*
  - Hombre interpuesto, *man-in-the-middle*
  - Falsificación origen de consulta, *spoofing*

2.6. Resumen

Resumen

- Aspectos arquitectónicos de las aplicaciones de red
- Cliente-Servidor, cuasi omnipresente
- Procesos en comunicación, sockets como API de acceso a la red
- Distintos servicios de transporte, TCP o UDP, dependiendo de las necesidades de aplicaciones
- Web y HTTP
- Transferencia de archivos con FTP
- Correo electrónico con SMTP, POP3, IMAP
- Servicio de Directorio DNS

3. Capa de Transporte

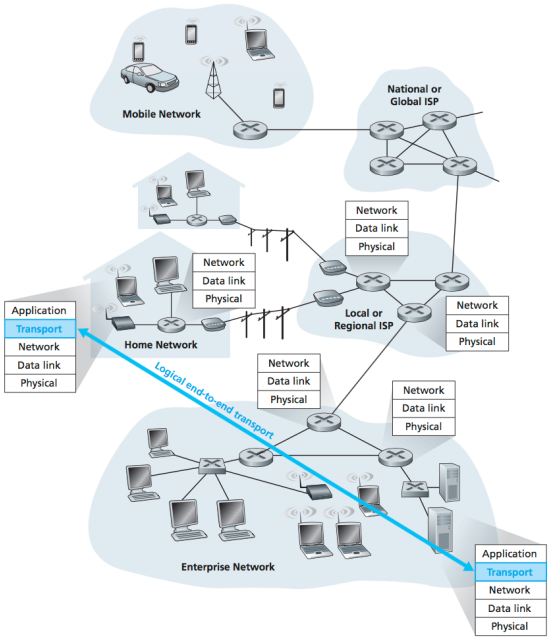
3.1. Introducción a los Servicios de la Capa de Transporte

Servicios de la Capa de Transporte

- Capa transporte proporciona *comunicación lógica* entre procesos ejecutándose en hosts distintos
- *Segmentos*
- Internet proporciona dos protocolos: *TCP* y *UDP*

3.1.1. Relación entre Capas de Transporte y Aplicación

Relación entre Capas de Transporte y Aplicación



3.1.2. Visión General de la Capa de Transporte

Visión General de la Capa de Transporte

- UDP, *User Datagram Protocol* [RFC 768]
  - Servicio no orientado a conexión
  - Entrega no fiable
- TCP, *Transport Control Protocol* [RFC 793]

    - Servicio orientado a conexión
    - Entrega fiable
- Cómo puede la capa de transporte ofrecer un servicio no ofrecido por la capa de red subyacente?
- IP, *Internet Protocol* modelo de servicio *best-effort*
    - Sin garantía de entrega de paquetes, *no fiable*, y posiblemente en desorden

Visión General de la Capa de Transporte

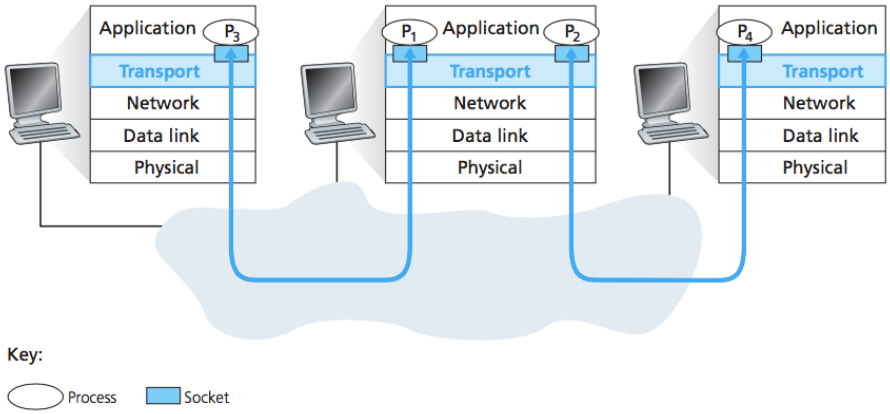
- Servicio básicos UDP y TCP
    - Extender el servicio IP de entrega entre hosts a entrega entre procesos: *multiplexación-demultiplexación*
    - Comprobación de la integridad de los segmentos (*checksum*)
- TCP, además
    - Transferencia fiable de datos (control de flujo, números de secuencia, reconocimiento, temporizadores)
    - Control de congestión

3.2. Multiplexación y Demultiplexación

Multiplexación y Demultiplexación

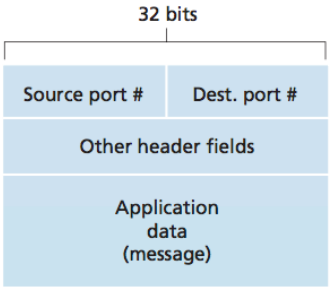
- *Multiplexación*
    - Recopilar en el host origen los datos enviados por cada proceso a través de su socket, crear segmentos añadiendo cabeceras a esos datos y entregárselos a la capa de red para su envío
- *Demultiplexación*
    - Tomar los segmentos que entrega la capa de red en el host destino, extraer de ellos los datos que contienen y entregarlos al proceso destinatario mediante el socket correspondiente

Multiplexación y Demultiplexación

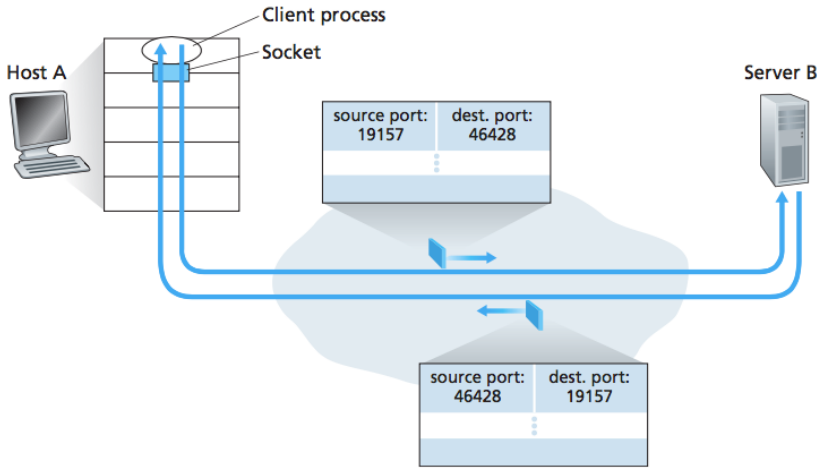


Número de Puerto
Multiplexación y Demultiplexación

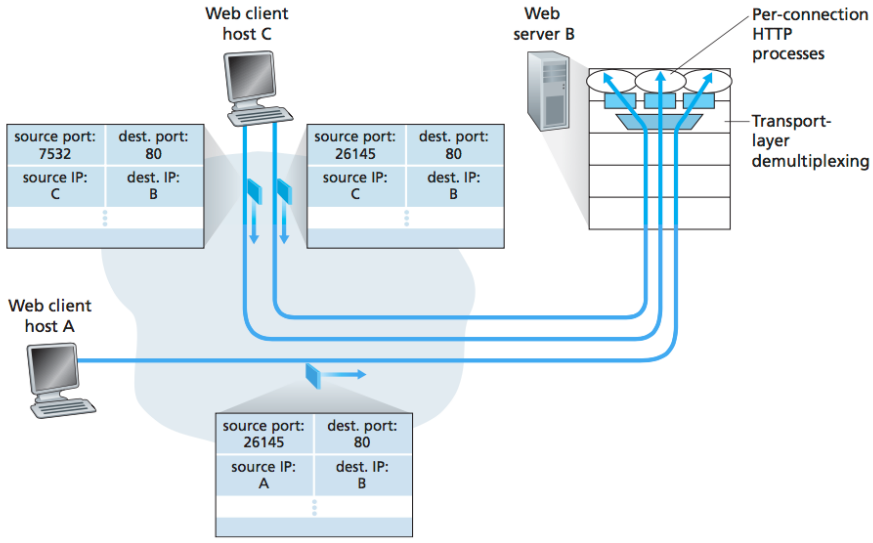
- 16 bits ($2^{16} \approx 65 \cdot 10^3$ puertos)
- Puerto *origen*
- Puerto *destino*
- Puertos *bien conocidos* [IANA Service Name and Transport Protocol Port Number Registry]



Inversión Números de Puerto Origen y Destino
Multiplexación y Demultiplexación



Dos Clientes y Un Mismo Servidor
Multiplexación y Demultiplexación



3.3. Transporte Sin Conexión: UDP

Transporte Sin Conexión: UDP

- UDP, multiplexación/demultiplexación y comprobación de errores

    - UDP toma los datos de aplicación, les añade num. puerto origen y destino, y otros campos información (*checksum*) y pasa el segmento resultante a la capa de red

- Sin necesidad de acuerdo previo para enviar un segmento: *no orientado a conexión*
    - Mayor control sobre cuándo y qué datos se envían
    - Sin establecimiento y cierre de conexión
    - No guarda estado de conexión
    - Poca sobrecarga por cabeceras (8 bytes)

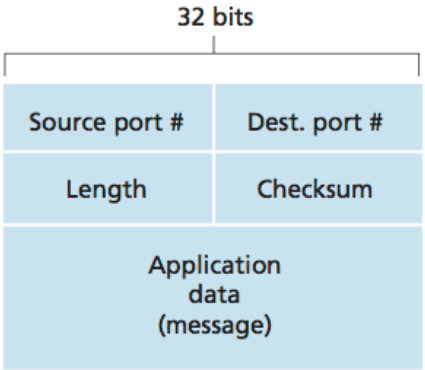
Aplicaciones y sus Protocolos de Transporte

Aplicación	Protocolo de Aplicación	Protocolo de Transporte
Correo electrónico	SMTP	TCP
Acceso remoto a terminal	Telnet	TCP
Web	HTTP	TCP
Transferencia de archivos	FTP	TCP
Servidor archivos remoto	NFS	UDP *
Streaming multimedia	Propietario *	UDP o TCP
Telefonía sobre Internet	Propietario *	UDP o TCP
Administración de red	SNMP	UDP *
Encaminamiento dinámico	RIP	UDP *
Servicio de directorio	DNS	UDP *

*: Normalmente

3.3.1. Estructura del Segmento UDP

Estructura del Segmento UDP



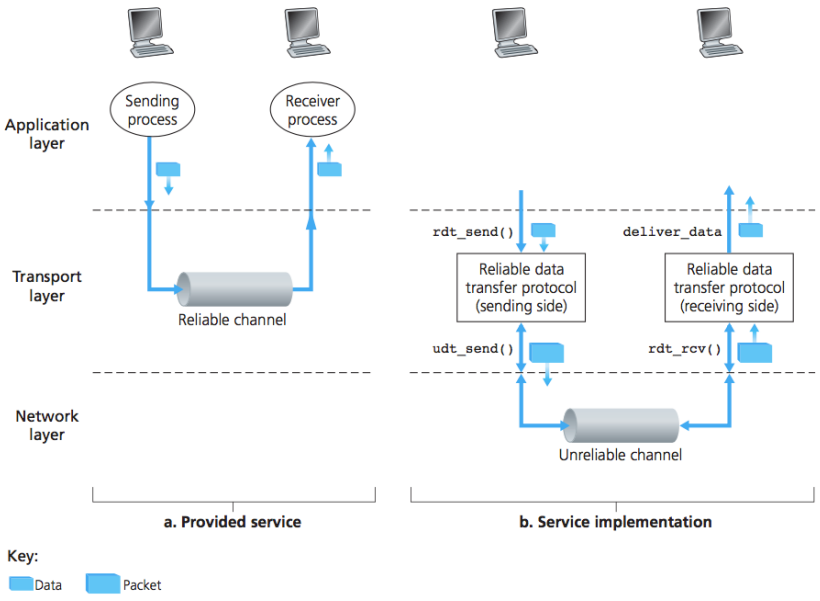
3.3.2. Checksum

Checksum

- Suma de comprobación, *cheksum* [RFC 1071]
- Complemento a 1 de la suma complemento a 1 de todas las palabras de 16 bits en el segmento
- La suma de la cabecera, incluyendo el checksum, debería ser -0 (todos a 1 en complemento a 1) si no ha habido corrupción
- Sobre el checksum
  - Posibles errores de transmisión, escritura/lectura en memoria
  - Capas inferiores pueden (o no) realizar comprobaciones integridad
  - Falsos positivos si varios bits cambian
  - Rápido

3.4. Principios de Transferencia Fiable

Principios de Transferencia Fiable



3.4.1. Construcción de un Protocolo Transferencia de Datos Fiable

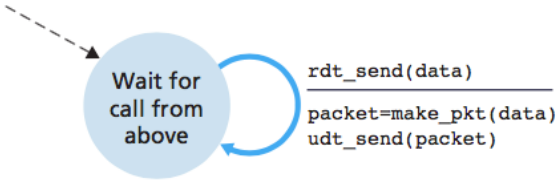
Construcción de un Protocolo Transferencia de Datos Fiable

- Consideraremos comunicación unidireccional, en bidireccional es igual pero más engorroso
- Enfoque incremental
  - Canal fiable
  - Canal con errores de bit
  - Canal con errores de bit y pérdida de paquetes

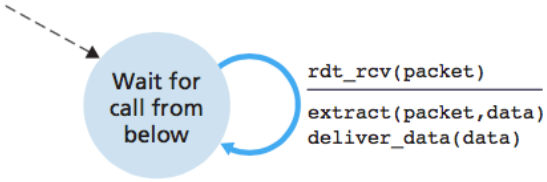
Transferencia en Canal Fiable: `rdt1.0`

- Si canal fiable, protocolo trivial
- Máquina estados finitos para emisor y receptor

Transferencia en canal fiable: `rdt1.0`



a. rdt1.0: sending side



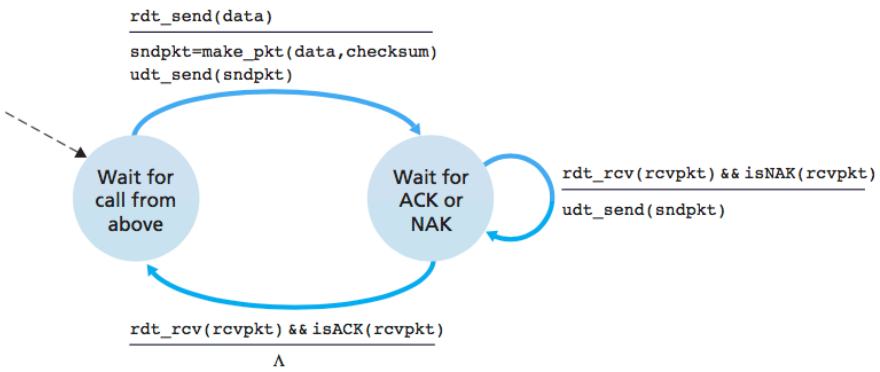
b. rdt1.0: receiving side

Transferencia en Canal con Errores de Bit: `rdt2.0`

- Hipótesis
  - Los bits en un paquete pueden ser corrompidos
  - Todos los paquetes llegan a su destino
- Cómo trataríamos esto en una conversación? "*Puede repetir?*"

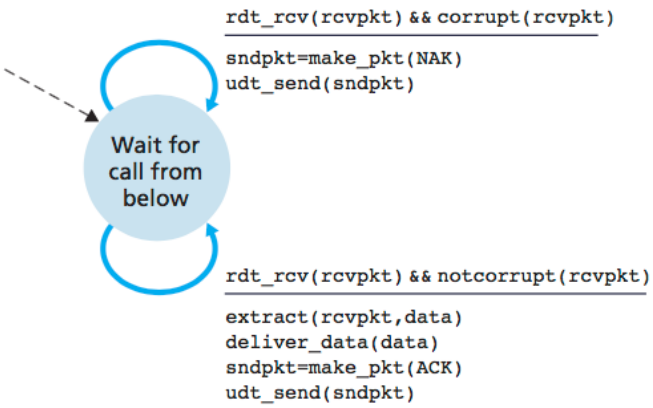
- *Protocolos de Repetición Automática*, *Automatic Repeat reQuest protocols*, ARQ
  - Detección de errores en recepción (*checksum*)
  - Realimentación del receptor (*ACK*, *NACK*)
  - Reenvío si necesario
- Protocolos de parada y espera (*stop-and-wait*)

Transferencia en Canal con Errores de Bit: `rdt2.0`



a. rdt2.0: sending side

Transferencia en Canal con Errores de Bit: `rdt2.0`

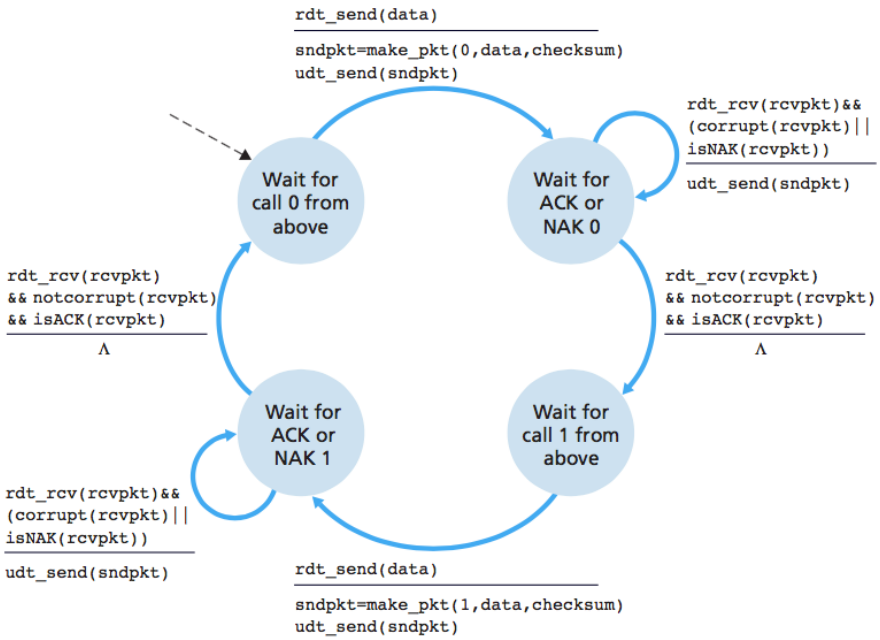


b. rdt2.0: receiving side

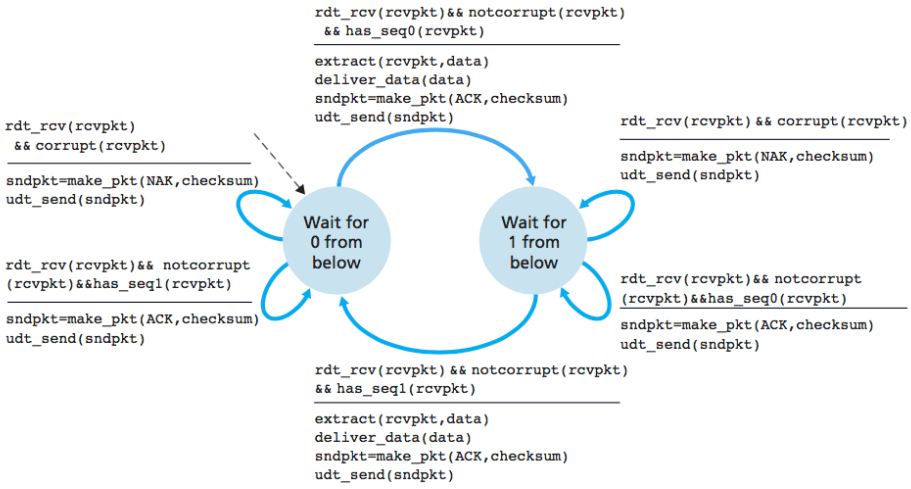
Transferencia en Canal con Errores de Bit: `rdt2.1`

- Cómo saber si un ACK o NACK ha sido corrompido?
- Reenviar el actual paquete si recibe un NACK o un ACK corrompido
- Posible paquete repetido: receptor no puede distinguir si el paquete recibido es nuevo o una retransmisión
- *Número de secuencia*
  - En un protocolo de parada y espera suficiente con 1 bit y aritmética módulo 2 (`1 + 1 = 0`)

Transferencia en Canal con Errores de Bit: `rdt2.1`



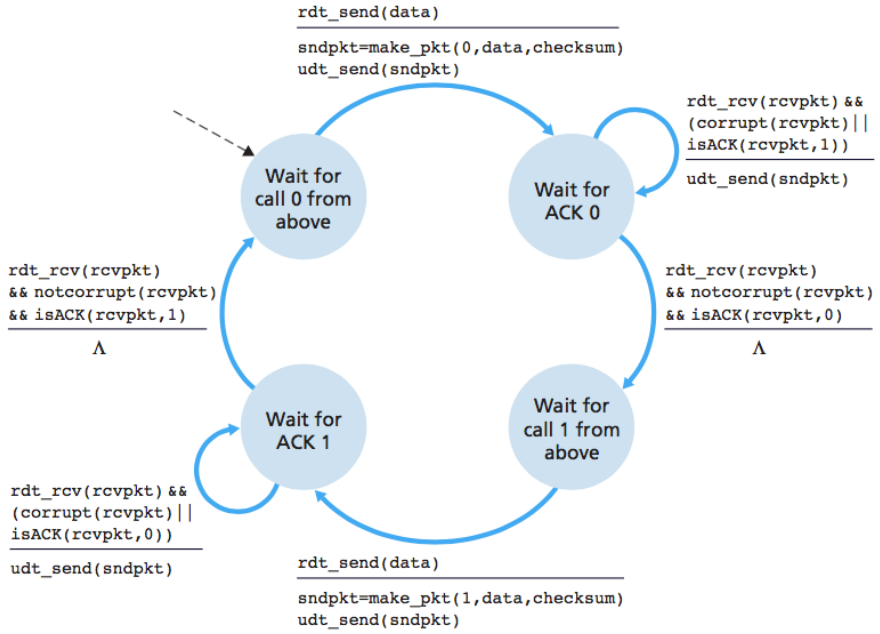
Transferencia en Canal con Errores de Bit: `rdt2.1`



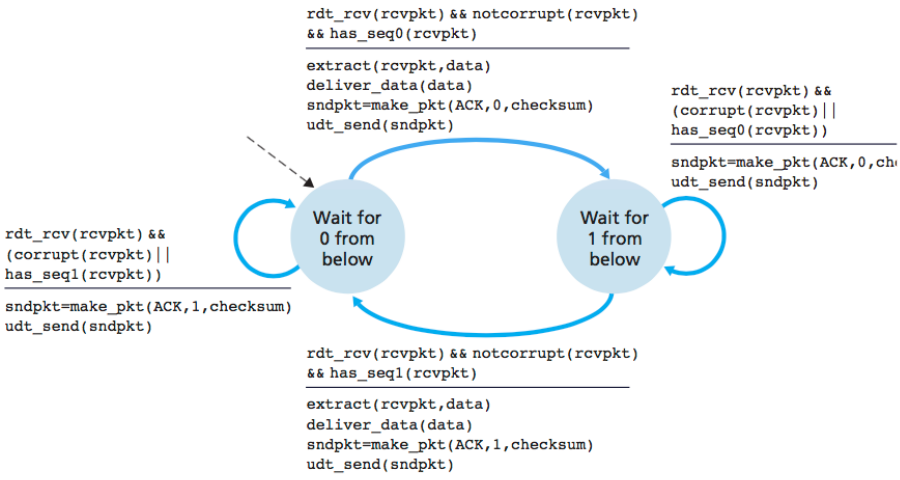
Transferencia en Canal con Errores de Bit: `rdt2.2`

- `rdt2.1` usa reconocimientos positivos y negativos
- Simplifiquemos
- En lugar de enviar un NACK, podemos enviar un ACK del último paquete correctamente recibido
- Si emisor recibe un *ACK duplicado* del paquete i, deduce que el receptor no recibió correctamente el paquete $i + 1$
- El receptor debe indicar el número de secuencia del paquete reconocido

Transferencia en Canal con Errores de Bit: `rdt2.2`



Transferencia en Canal con Errores de Bit: `rdt2.2`



Transferencia en Canal con Errores de Bit y Pérdida de paquetes: `rdt3.0`

- Además de errores de bit, el canal puede perder paquetes
- Cómo detectar paquetes perdidos?

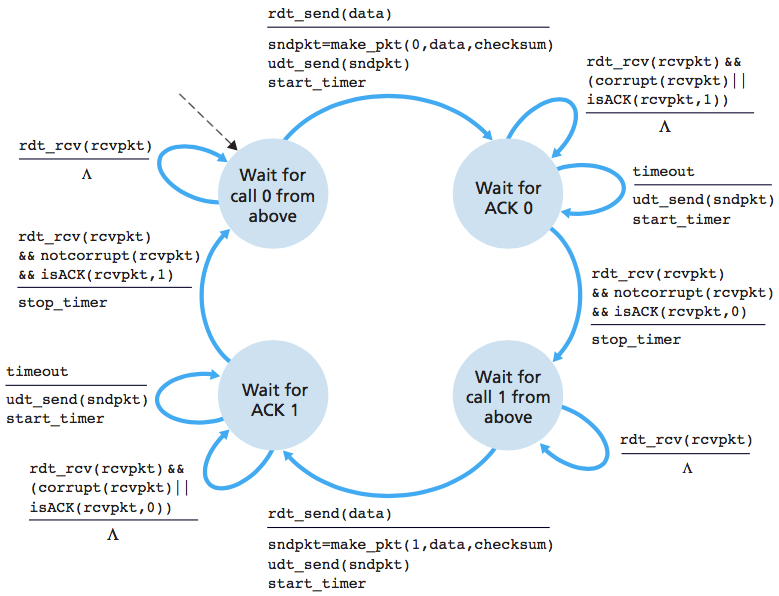
- Número secuencia, ACKs

- Qué hacer en ese caso?
  - Reenvío

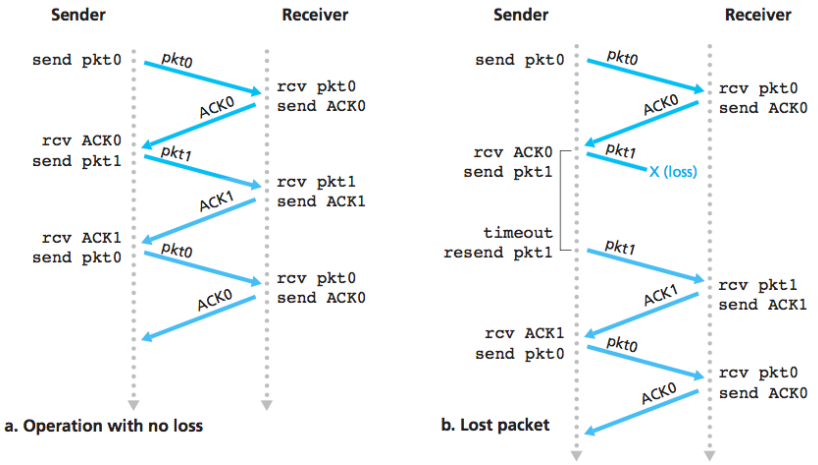
Transferencia en Canal con Errores de Bit y Pérdida de paquetes: `rdt3.0`

- Emisor vuelve a transmitir un paquete si:
  - El paquete no llegó
  - No llegó el ACK
  - El paquete se retrasó
    - Si el paquete acaba llegando, ha sido vuelto a transmitir $\rightarrow$ paquete duplicado
- Cuánto esperar? $>$ (RTT + procesamiento)
- *Temporizador de retransmisión* interrumpe emisor cuando expira el tiempo (timeout)
- Emisor podrá
  - Iniciar un temporizador cada vez que envía un paquete
  - Responder a un evento de interrupción de temporizador
  - Parar el temporizador
- *Protocolo de bit alternante*, los números de secuencia alternan entre `0` y `1`

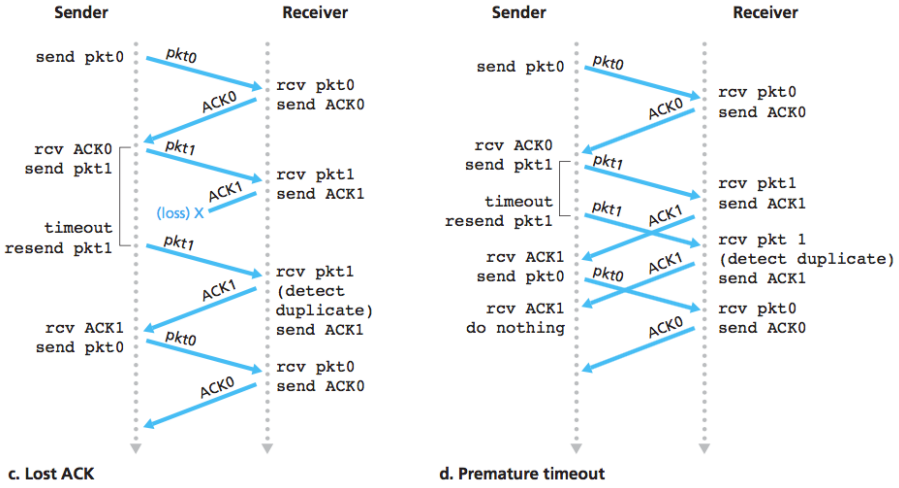
Transferencia en Canal con Errores de Bit y Pérdida de paquetes: `rdt3.0`



Funcionamiento de `rdt3.0`: protocolo de bit alternante



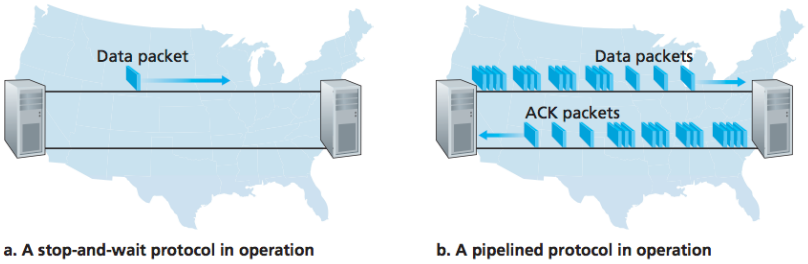
Funcionamiento de `rdt3.0`: protocolo de bit alternante



3.4.2. Protocolos de Transferencia Fiable Canalizados

Protocolos de Transferencia Fiable Canalizados

- Bit alternante con problemas de rendimiento

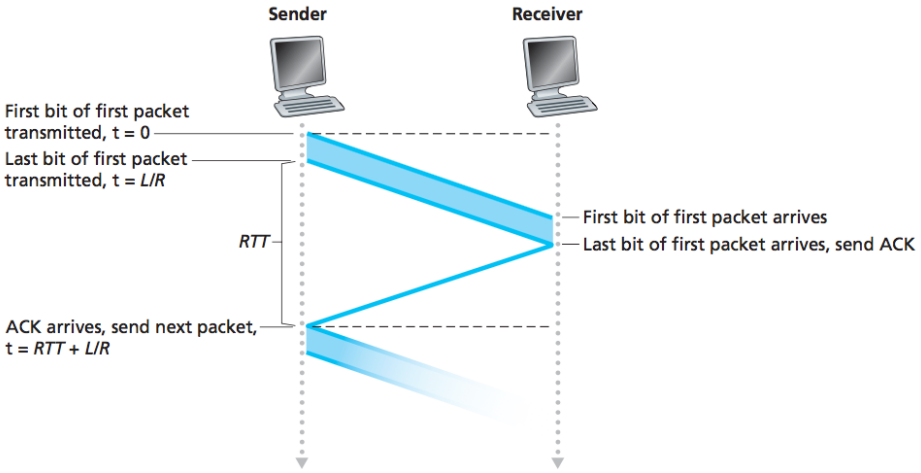


a. A stop-and-wait protocol in operation

b. A pipelined protocol in operation

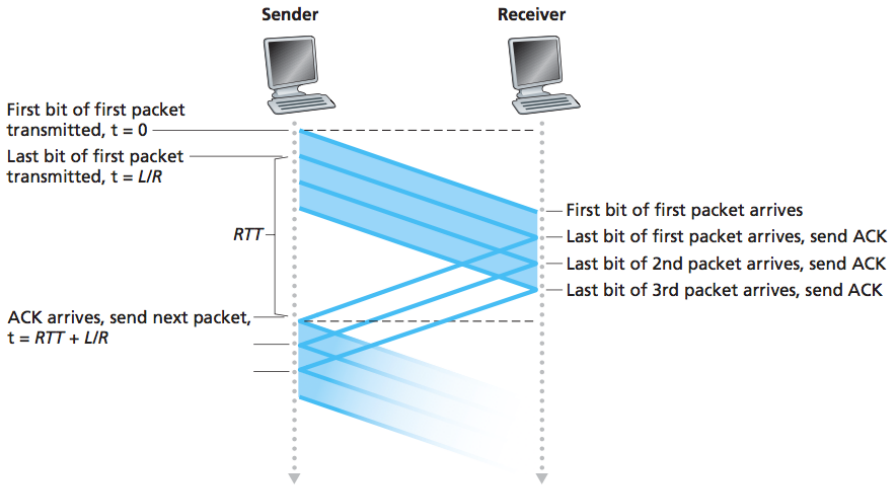
- *Canalizar*, permitir que el emisor envíe varios paquetes sin esperar ACKs (pipelining)

Protocolos de Transferencia Fiable Parada y Espera vs Canalizados



a. Stop-and-wait operation

Protocolos de Transferencia Fiable Parada y Espera vs Canalizados



b. Pipelined operation

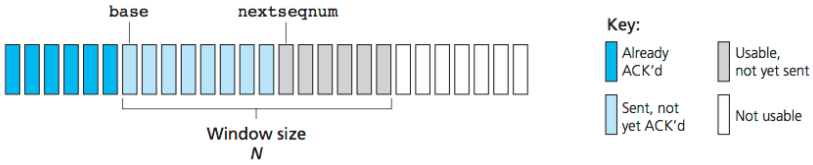
Protocolos de Transferencia Fiable Canalizados

- Aumentar el rango de números de secuencia disponibles
- Tanto emisor como receptor deben disponer de espacio almacenamiento suficiente (*buffer*)
- Determinar el modo de recuperar un error:
  - *Retroceder N* (*Go-Back-N*, GBN)
  - *Repetición selectiva* (*Selective Repeat*, SR)

3.4.3. Go-Back-N, GBN

Go-Back-N, GBN

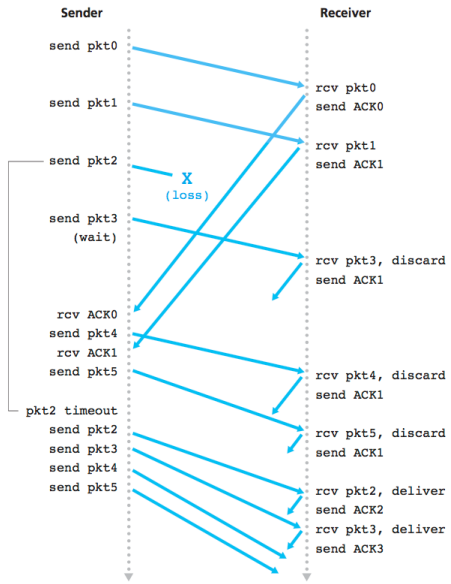
- Emisor hasta N paquetes pendientes de reconocer
- Protocolo de *ventana deslizante*



Eventos GBN

- Emisor
  - Invocación desde arriba (aplicación), si ventana disponible, enviar
  - Recepción de ACK$_i$, *reconocimiento acumulado*
  - Evento temporizador, reenviar todos los paquetes aún no reconocidos
- Receptor
  - Si recibe paquete n en orden, enviar ACK$_n$
  - En otro caso, desechar paquete y enviar ACK del último paquete recibido en orden

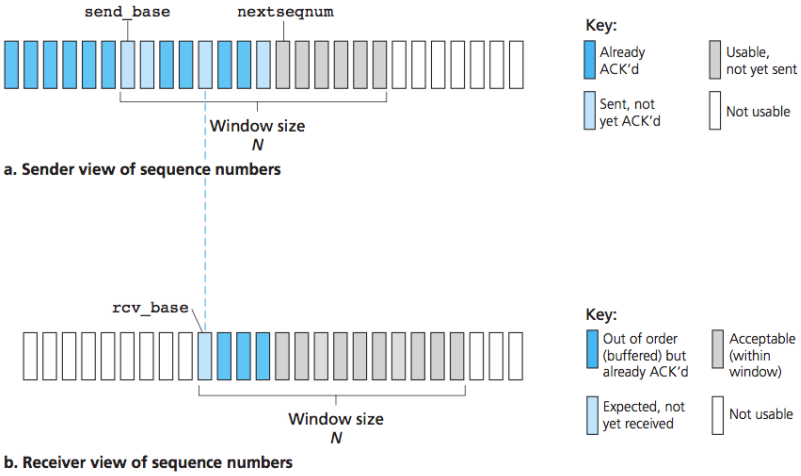
Funcionamiento de GBN



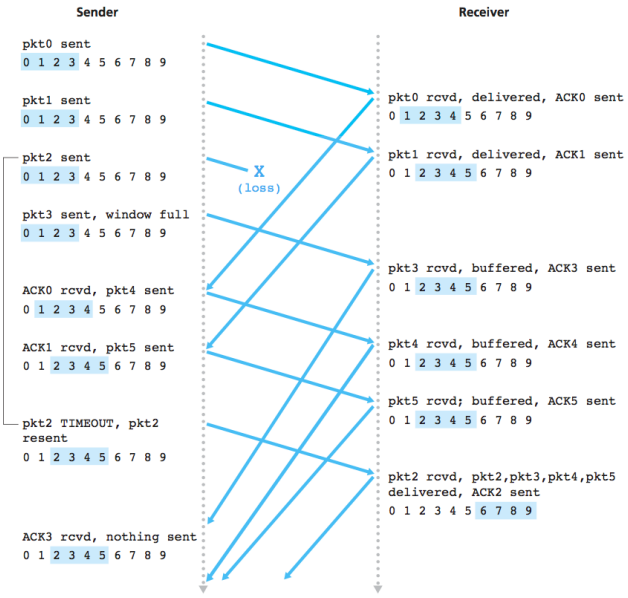
3.4.4. Repetición Selectiva, SR

Repetición Selectiva, SR

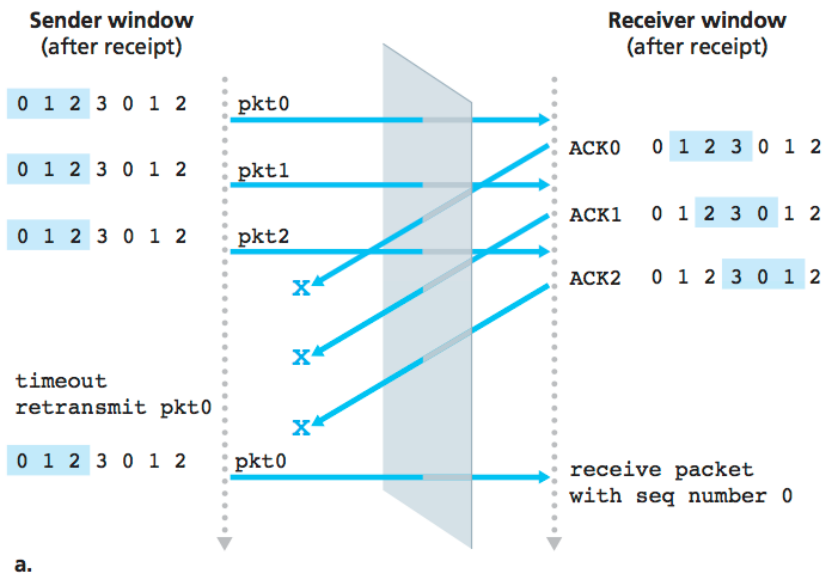
- Problemas rendimiento en GBN con tamaño ventana y retardo grandes
- Evitar reenvíos innecesarios con reconocimientos individuales



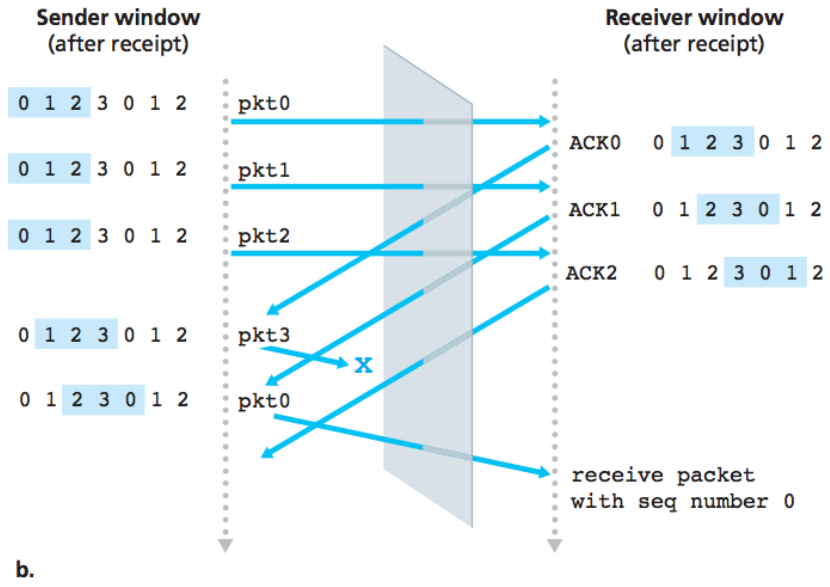
Funcionamiento de SR



SR, paquete nuevo o retansmisión?



SR, paquete nuevo o retansmisión?



Resumen Mecanismos Transferencia Fiable

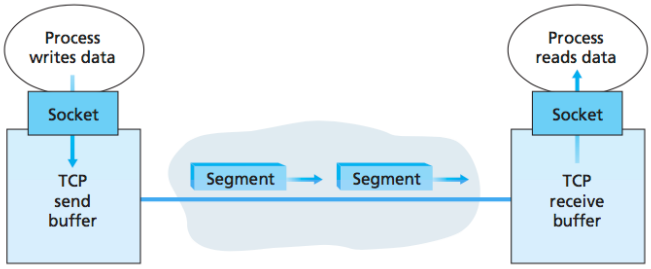
Mecanismo	Uso, Comentarios
Temporizador	Reenviar un paquete, porque el paquete o su ACK se perdieron. Paquetes retrasados (*timeout* prematuro) o ACKs perdidos producen paquetes duplicados
Número de Secuencia	Permite la secuenciación de los paquetes, detección de agujeros (pérdidas) y duplicados
Reconocimiento, ACK	Indica recepción de paquete con num. secuencia, individual o acumulado
Reconocimiento Negativo, NACK	Indica que un paquete no fue recibido correctamente
Ventana, Canalización	Emisor puede enviar paquetes sin esperar ACKs hasta un máximo N (ventana deslizante) lo que incrementa rendimiento. Útil para control flujo y congestión

3.5. Transporte Orientado a Conexión: TCP

3.5.1. La Conexión TCP

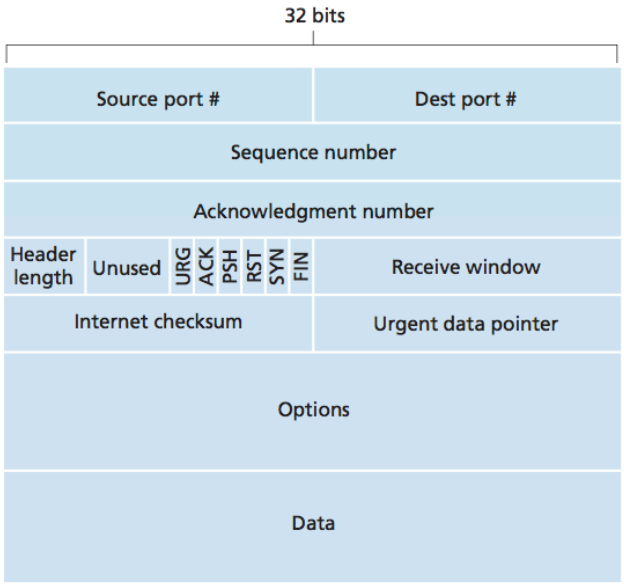
La Conexión TCP

- *Full-duplex*, punto a punto
- TCP es orientado a conexión, debe haber acuerdo entre procesos antes de enviar datos
  - Acuerdo en tres pasos
  - Los dos primeros segmentos intercambiados no portan datos
- *MSS*, *Maximum Segment Size*, dependiente de *MTU*, *Maximum Transmission Unit*

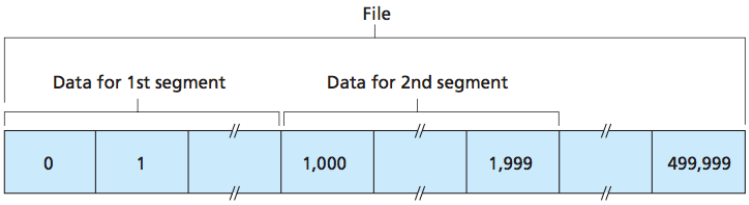


3.5.2. Estructura del Segmento TCP

Estructura del Segmento TCP

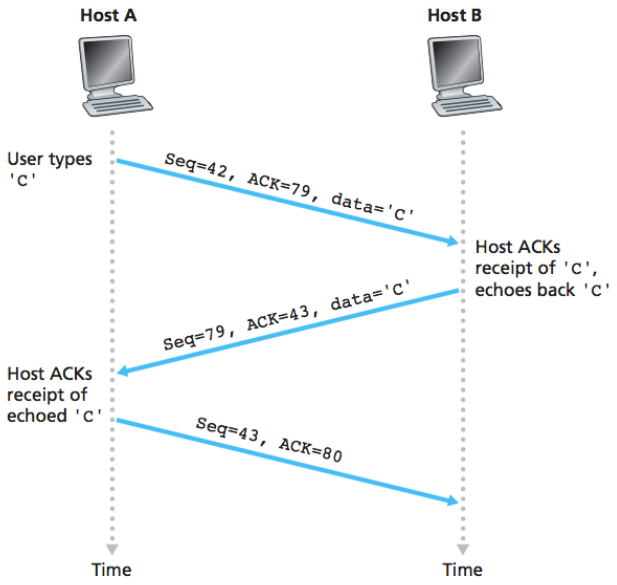


Número de Secuencia y de Reconocimiento



- *Número de secuencia*, número del orden dentro del flujo de datos del primer byte de datos transportado por el segmento
- *Número de reconocimiento*, el número del byte en el flujo de datos que está esperando recibir
  - Reconocimiento acumulativo
  - Qué pasa con los segmentos desordenados?

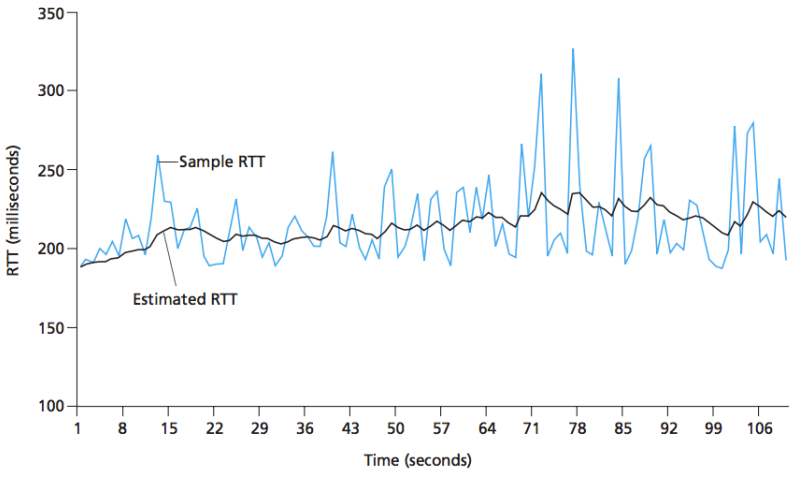
Número de Secuencia y de Reconocimiento



3.5.3. Estimación RTT y Temporizador de Retransmisión

Estimación RTT y Temporizador de Retransmisión

- Temporizador > RTT



Estimación RTT y Temporizador de Retransmisión

- $RTT_{i+1}^{estimado} = (1-\alpha) \cdot RTT_i^{estimado} + \alpha \cdot RTT_i^{muestreado}$

- $\alpha = 0,125$

- $DevRTT_{i+1} = (1 - \beta) \cdot DevRTT_i + \beta \cdot |RTT_i^{muestreado} - RTT_i^{estimado}|$
  - $\beta = 0,25$
- $Temporizador = RTT^{estimado} + 4 \cdot DevRTT$

3.5.4. Transferencia Fiable

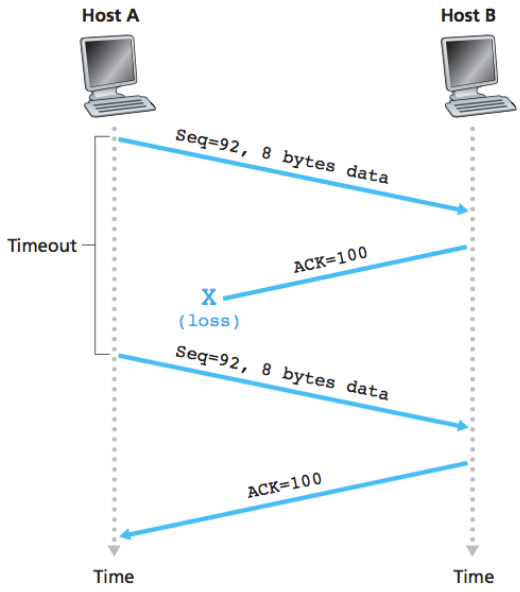
Transferencia Fiable

- TCP crea un servicio de transferencia fiable sobre el no fiable servicio de IP *best-effort*
- Asegura que el proceso destino recibe un flujo de datos continuo no corrompidos sin pérdidas, huecos o duplicados
- Temporizador reenvío único, reconocimientos acumulados
- Eventos envío de datos
  - Recibir datos de la aplicación
  - Fin de temporizador, *timeout*
  - Recepción ACK

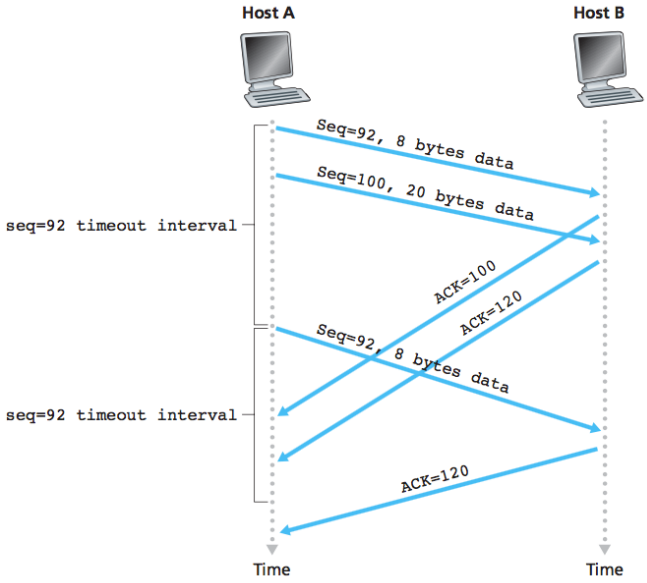
Emisor TCP simplificado

```
NextSeqNum=InitialSeq
Number SendBase=InitialSeqNumber

loop (forever) {
  switch(event)
     event: data received from application above
           create TCP segment with sequence number NextSeqNum
           if (timer currently not running) start timer
           pass segment to IP
           NextSeqNum = NextSeqNum + length(data)
           break;
     event: timer timeout
           retransmit not-yet-acked segment with smallest seqNum
           start timer
           break;
     event: ACK received, with ACK field value of y
           if (y > SendBase) {
                 SendBase = y
                 if (any not-yet-acked segments) start timer
           }
           break;
} // end of loop forever
```

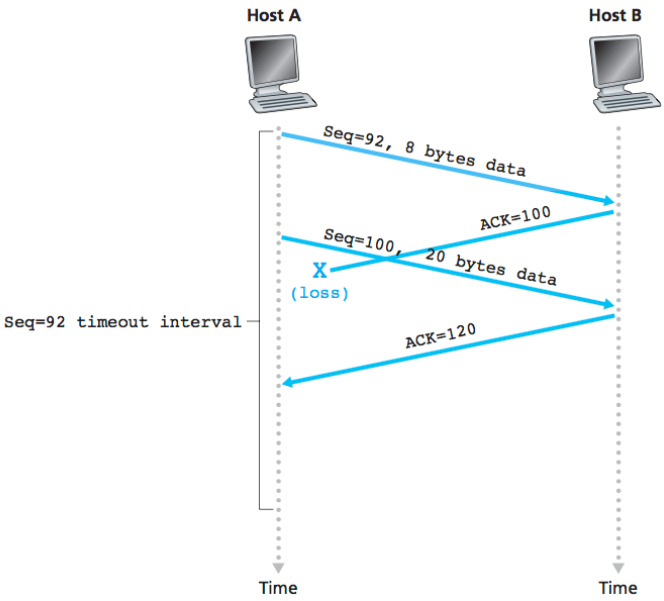
Retransmisición por pérdida de ACK



Segmento no Retransmitido



Renocimiento Acumulativo



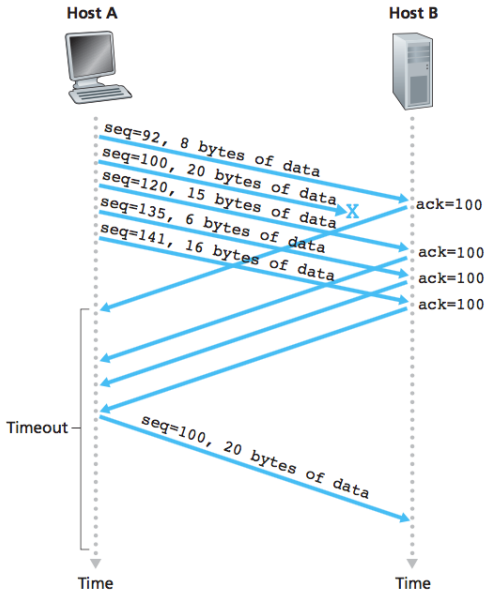
Duplicación del Temporizador

- Si transmisión por evento de fin de temporizador, duplicar el valor del temporizador de reenvío
- Si transmisión por otros 2 eventos (datos de aplicación, ACK), temporizador según la estimación
- Forma de control de congestión, pero produce temporizadores con valores grandes

Retransmisión Rápida

- Temporizador puede llegar a ser relativamente grande
- Detectar pérdida de paquetes al recibir un ACK por duplicado, volviendo a reconocer un segmento ya reconocido anteriormente
- Receptor detecta un hueco, una pérdida, reconoce el último segmento en secuencia
- Como emisor envía una gran número de segmentos al receptor, si uno se pierde, el receptor devolverá muchos ACKs duplicados
- Al recibir tres ACKs repetidos, emisor asume la pérdida del siguiente segmento y lo reenvía antes de fin de temporizador

Retransmisión Rápida



Emisor TCP con Retransmisión Rápida

```
NextSeqNum=InitialSeq
Number SendBase=InitialSeqNumber

loop (forever) {
  switch(event)
      event: data received from application above
             ...
             break;
      event: timer timeout
             ...
             break;
      event: ACK received, with ACK field value of y
             if (y > SendBase) {
                   SendBase = y
                   if (any not-yet-ACKed segments) start timer
             } else { // duplicate ACK for already ACKed segment
                   nDuplicateACKS_y ++
                   if (nDuplicateACKS_y == 3)
                   resend segment with seqNum y // fast retransmit
             }
             break;
} // end of loop forever
```

Generación ACKs [RFC 5681]

Evento	Acción Receptor TCP
Llega segmento en orden con num. secuencia esperado, todos los datos anteriores reconocidos	Retardar ACK, esperando 500 ms la llegada de otro segmento en orden. Si no llega enviar ACK
Llega segmento en orden con num. secuencia esperado y otro segmento en orden espera ser reconocido	Enviar un único ACK acumulado, reconociendo ambos segmentos
Llega un segmento desordenado con num. secuencia mayor del esperado (hueco)	Enviar ACK duplicado, con el num. del siguiente byte esperado (el inicio del hueco)
Llega un segmento que rellena parcialmente un hueco por su principio	Enviar ACK

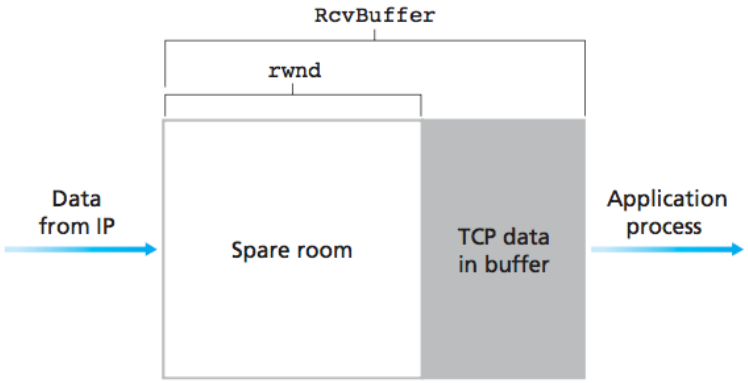
TCP: GBN o SR?

- Parece GBN, pero con matices
  - Algunas implementaciones aceptan segmentos en desorden
  - No se retransmite el segmento n cuyo ACK se perdió si llega el ACK del $n+1$ antes de que expire el temporizador para el segmento n
  - *Reconocimiento Selectivo*, [RFC 2018], permite reconocer segmentos en desorden de manera selectiva, en lugar del acumulado
- Recuperación de fallos en TCP es un híbrido GBN - SR

3.5.5. Control de Flujo

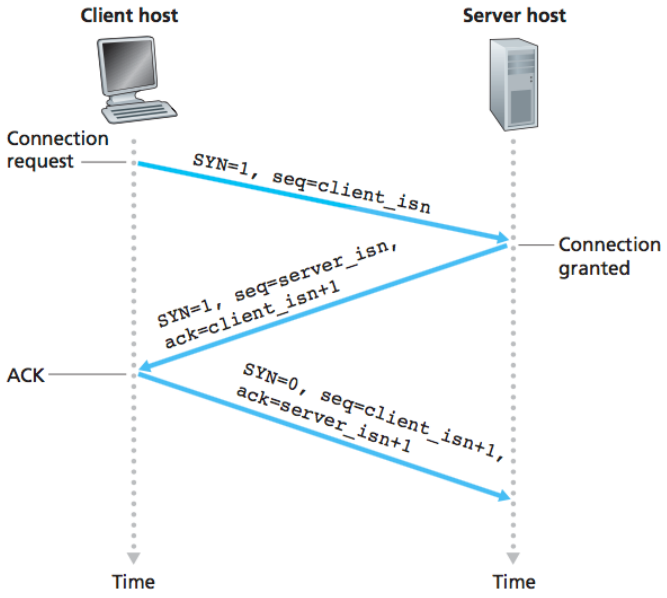
Control de Flujo

- *Ventana de recepción*, `rwnd`
- Receptor: `rwnd = RcvBuffer - [LastByteRcvd - LastByteRead]`
- Emisor: `LastByteSent - LastByteAcked` $\leq$ `rwnd`

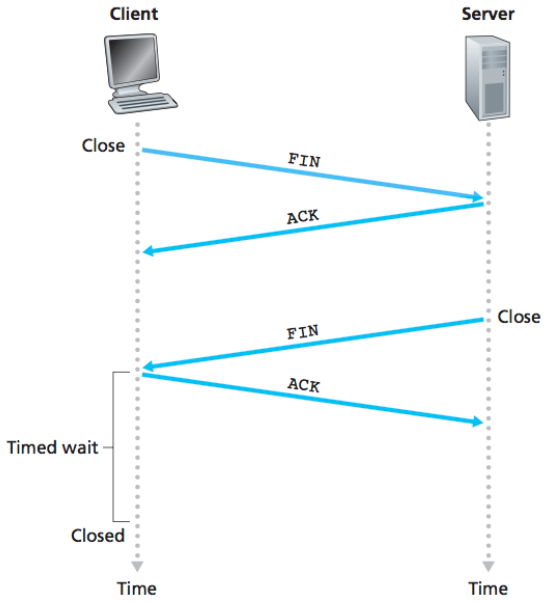


3.5.6. Gestión de Conexión TCP

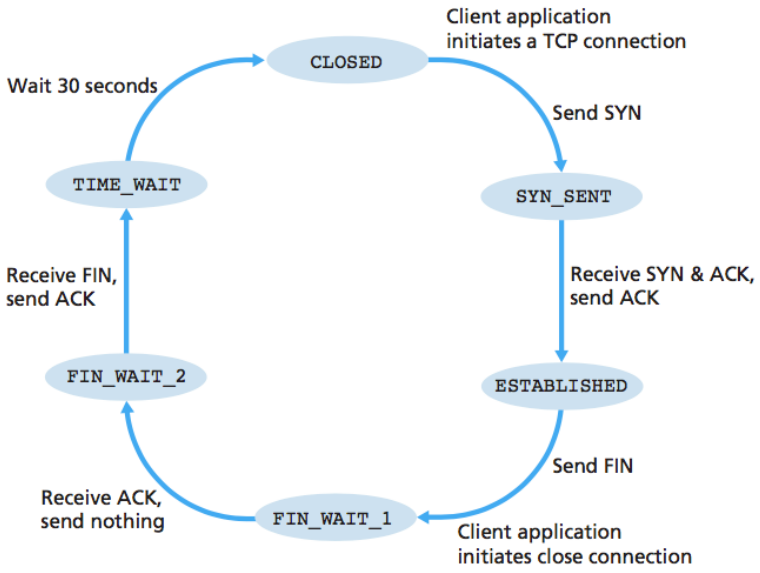
Acuerdo en Tres Pasos



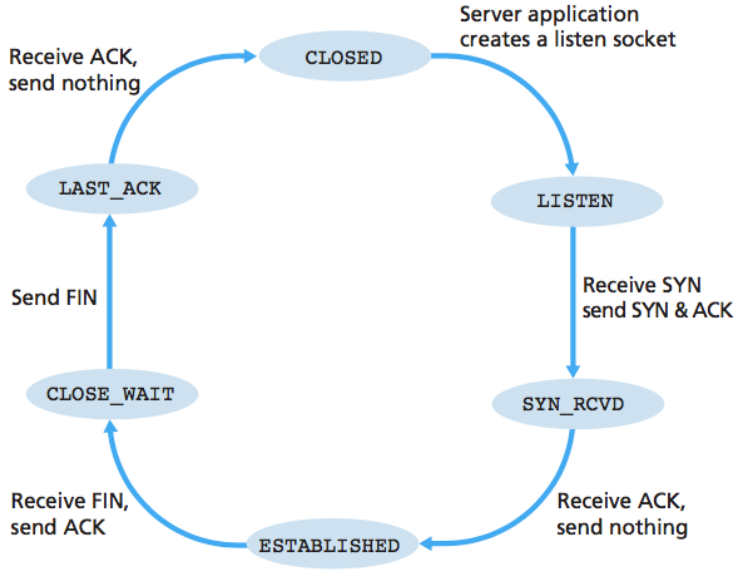
Finalización Conexión



Estados Cliente



Estados Servidor



3.6. Principios del Control de la Congestión

Principios del Control de la Congestión

- Paquetes perdidos por desbordamiento buffer en routers, debido a congestión en la red
- El reenvío de paquetes es un síntoma de congestión
- Solución: estrangular emisores
- Enfoques
  - Control de congestión entre extremos (TCP/IP)
  - Control de congestión asistido por la red (ABR de ATM)

3.7. Control de Congestión en TCP

Control de Congestión en TCP

- Emisor limita la tasa de envío en función de la congestión percibida
  - Si no congestión, incrementa la tasa de envío
  - Si congestión, decrementa la tasa de envío
- *Ventana de congestión*, `cwnd`
- `LastByteSend - LastByteAcked` $\leq$ `min{cwnd, rwnd}`
- Si `rwnd` = $\infty \rightarrow$ tasa envío = `cwnd`/RTT bytes/s

Mecanismo Ajuste Ventana Congestión

- Si evento de pérdida, reducir `cwnd`
  - Timeout
  - Recepción tres ACKs repetidos
- Si recibe ACK, todo está bien, aumentar `cwnd`
- *Sondeo de ancho de banda*, aumentar la tasa de envío respondiendo a los ACKs que llegan hasta que se de un evento de pérdida, que decrementará la tasa de envío

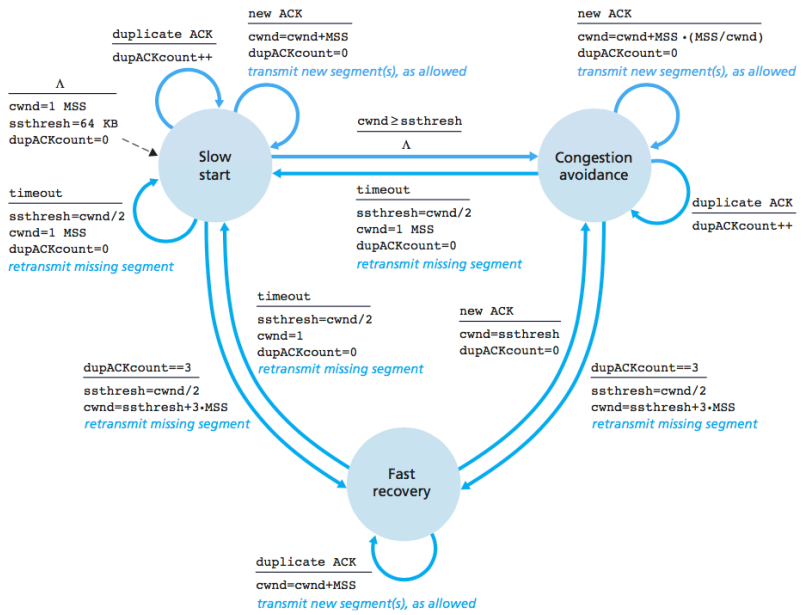
Algoritmo Control Congestión TCP

- Arranque lento
  - Incrementar `cwnd` por cada primer ACK, hasta umbral `ssthresh`, pasar a evitación congestión
  - Si evento de fallo, volver al principio: $\texttt{cwnd}_0$ = 1 MSS
- Evitación de la congestión

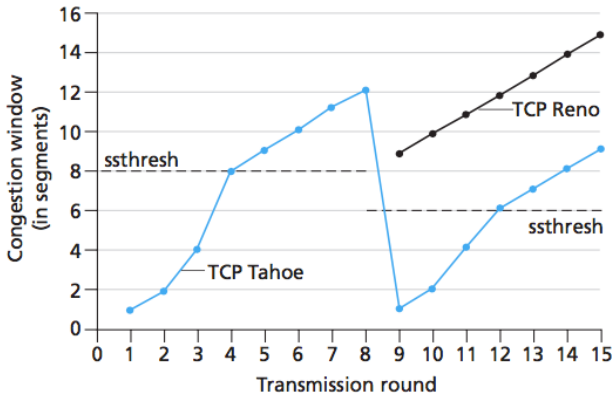
- Incrementar `cwnd` de forma menos agresiva que el arranque lento, 1 MSS por RTT
- En evento fallo volver a arranque lento

- Recuperación rápida (* no en *TCP Tahoe*)
  - Distinguir entre fallo por `timeout` y triple ACK repetido

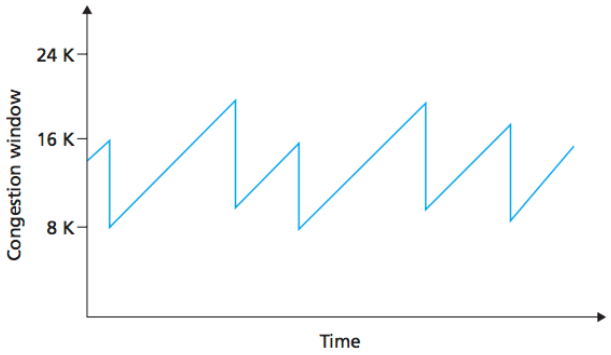
Algoritmo Control Congestión TCP



Evolución de la Ventana de Congestión



Incremento aditivo, decremento multiplicativo
Control de la Congestión



3.7.1. Equidad

Equidad
Control de la Congestión

- El control de congestión es *equitativo* si cada conexión consigue aproximadamente R/K
- TCP converge hacia el reparto equitativo del ancho de banda en los cuellos de botella
- Aplicaciones sobre UDP no colaboran con el resto de competidoras por el ancho de banda

3.8. Resumen

Resumen

- Comenzamos estudiando los servicios que la capa de transporte puede proporcionar a la de aplicación
- UDP es un ejemplo de protocolo de transporte sin adornos
- Los servicios que se pueden ofrecer a menudo están restringidos por la capa de red
- Hemos desarrollado incrementalmente un modo de proporcionar entrega fiable aunque la capa de red no ofrezca ese servicio y además pueda corromper los paquetes
- Hemos estudiado TCP como un protocolo de entrega fiable orientado a la conexión y con gestión de flujo y de congestión

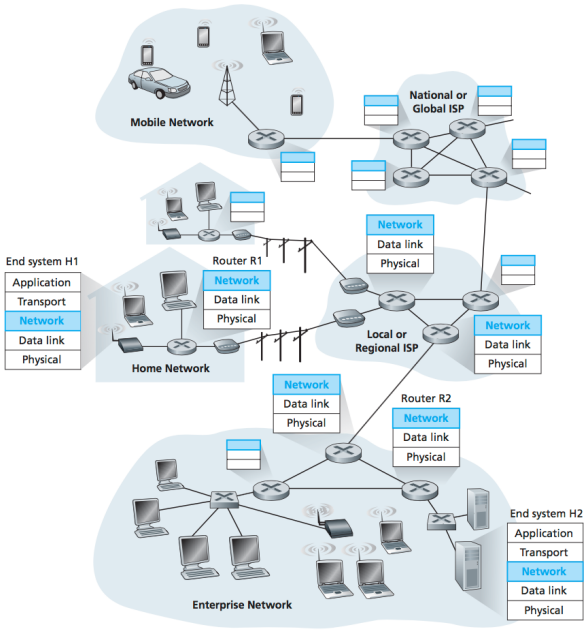
4. Capa de Red

4.1. Introducción

Introducción

- Capa de red, toma segmento de la capa de transporte, lo encapsula en un *datagrama* y lo encamina hacia su destino
- *Router* reenvía datagramas desde puertos de entrada hacia puertos de salida
- Routers implementan hasta la capa 3, no transporte ni aplicación

La Capa de Red

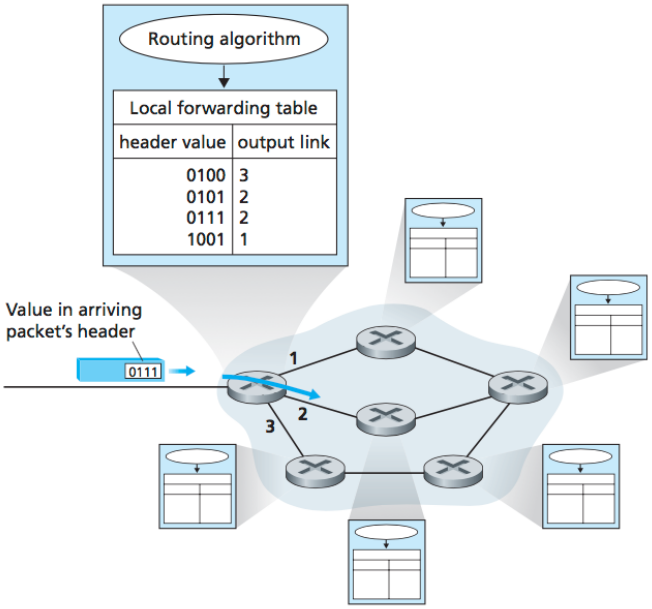


4.1.1. Reenvío y Encaminamiento

Reenvío y Encaminamiento

- Rol capa de red: mover paquetes del host origen al host destino
- Funciones
  - *Reenvío*, de puerto entrada router a puerto salida
  - *Encaminamiento*, determinar ruta de emisor al receptor
- Basado en dirección destino
- Si conmutación circuitos, establecimiento de conexión

Algoritmos Encaminamiento y Tablas de Reenvío



4.1.2. Modelos de Servicio de Red

Modelos de Servicio de Red

- Posibles servicios a la capa de transporte
  - Entrega garantizada
  - Entrega garantizada con retardo limitado
  - Entrega ordenada de paquetes
  - Ancho de banda mínimo garantizado
  - *Jitter* máximo garantizado
  - Seguridad
- Internet proporciona un servicio único: *Best-effort*
  - Sin ningún compromiso

4.2. Redes de Circuitos Virtuales y de Datagramas

Redes de Circuitos Virtuales y de Datagramas

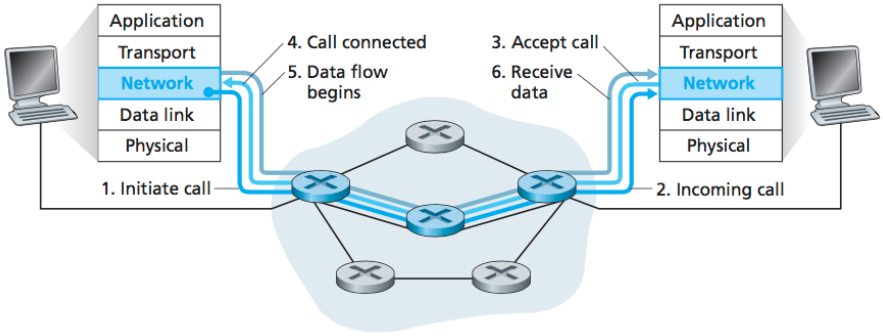
- *Redes de Circuitos Virtuales*, proporcionan un servicio orientado a conexión

- Origen en redes telefónicas: redes inteligentes, terminales tontos
- Frame relay, ATM

- *Redes de Datagramas*, servicio no orientado a conexión
  - Dispositivos finales más sofisticados y capaces, permitan una red más simple, mínimos requisitos para interconectar redes con tecnologías enlace diferentes
  - Internet, IP

4.2.1. Redes de Circuitos Virtuales

Establecimiento de Circuito Virtual



4.2.2. Redes de Datagramas

Red de Datagramas

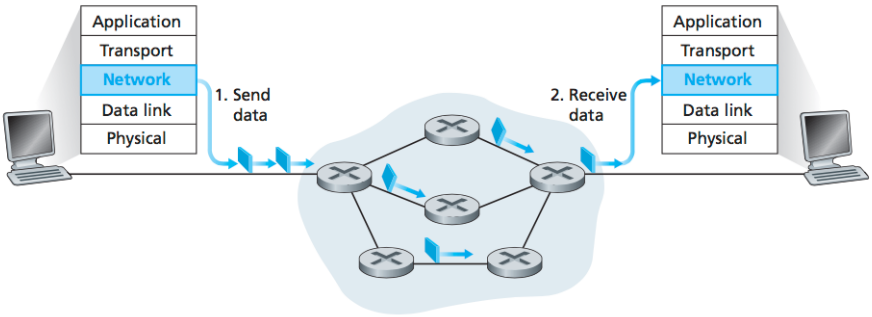


Tabla Reenvío Router con 4 interfaces

- Direcciones 32 bits $\rightarrow 2^{32} \approx 4 \cdot 10^9$ destinos

Rango Direcciones Destino	Interfaz Salida
11001000 00010111 00010000 00000000 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 11001000 00010111 00011111 11111111	2
En cualquier otro caso	3

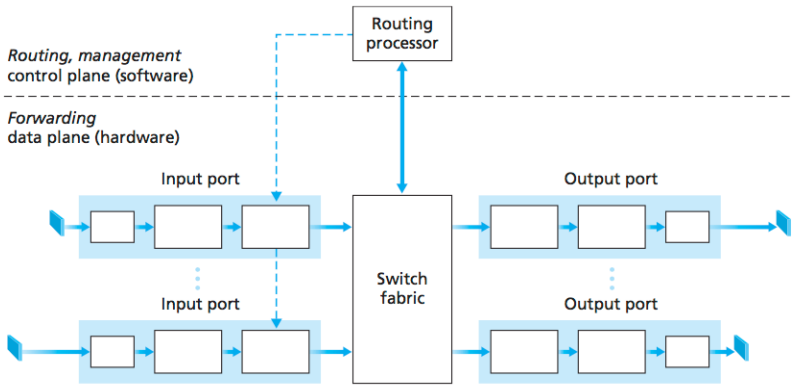
Tabla Reenvío Router con 4 interfaces

- Regla coincidencia del *mayor prefijo* posible

Prefijo Direcciones Destino	Interfaz Salida
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
En cualquier otro caso	3

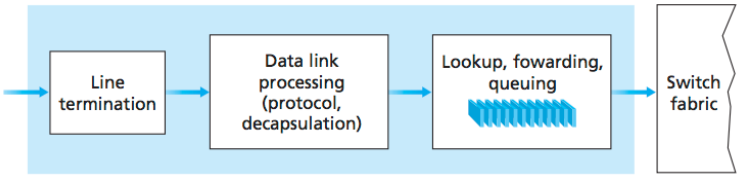
4.3. Qué hay Dentro de un Router?

Arquitectura de un Router



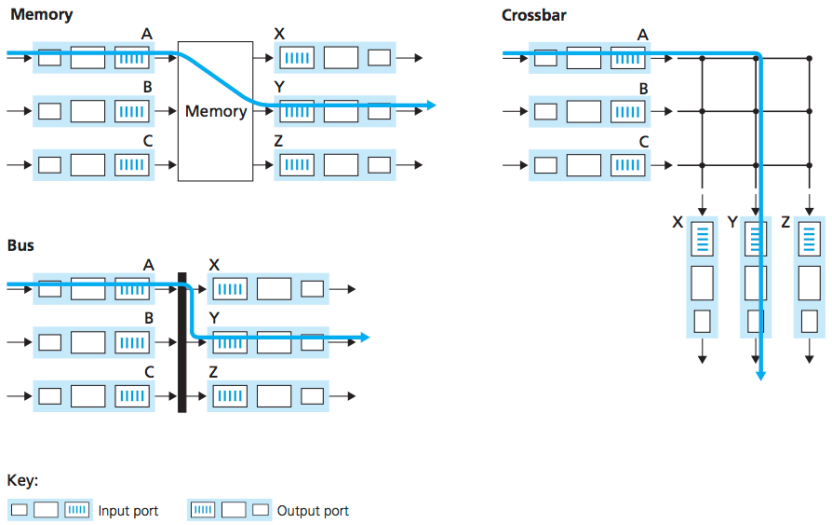
4.3.1. Procesamiento de la Entrada

Procesamiento de Puerto de Entrada



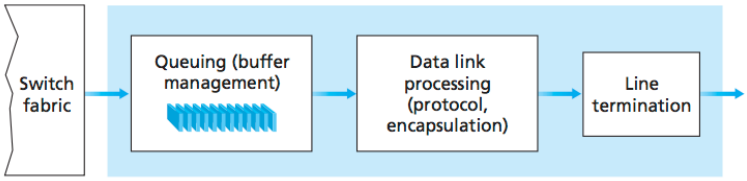
4.3.2. Conmutación

Técnicas de Conmutación



4.3.3. Procesamiento de la Salida

Procesamiento de Puerto de Salida



4.3.4. Dónde se Producen las Colas?

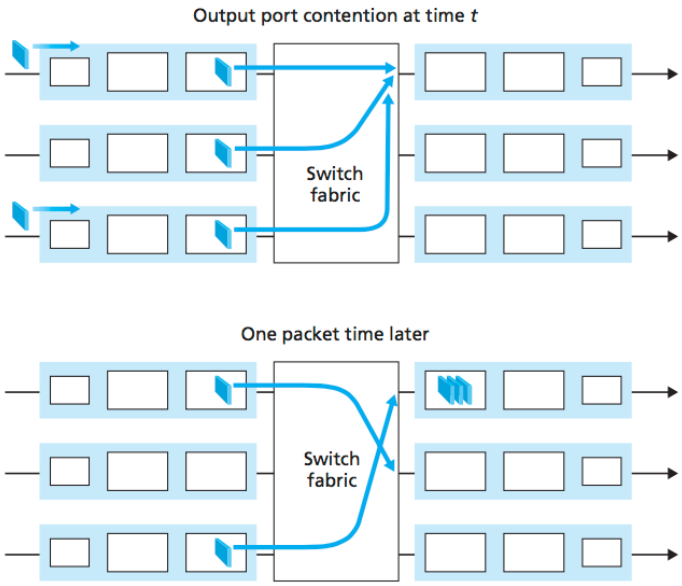
Dónde se Producen las Colas?

- Puertos entrada y salida
- Depende de la carga de tráfico, la velocidad relativa de la malla de interconexión y de la velocidad de línea
- *Pérdida de paquetes* si no espacio disponible
- Buffer requerido: $B = RTT \cdot C$ con C capacidad enlace bps
- En cola salida
    - Planificador de salida de paquetes: FCFS, WFQ (*Quality of Service*, QoS)

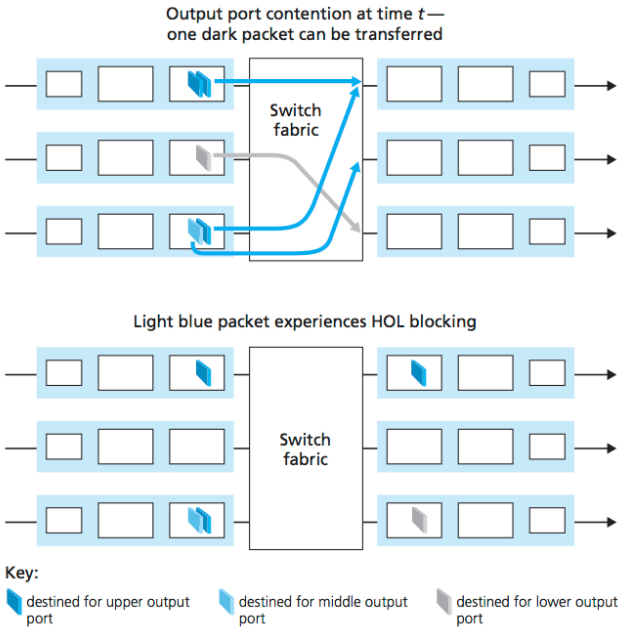
- Gestión de cola activa, *Random Early Detection* RED

- En puertos de entrada
  - Bloqueo de paquetes al frente debido a malla conmutación ocupada, bloqueo *head-of-the-line*, HOL

Colas en Puerto de Salida

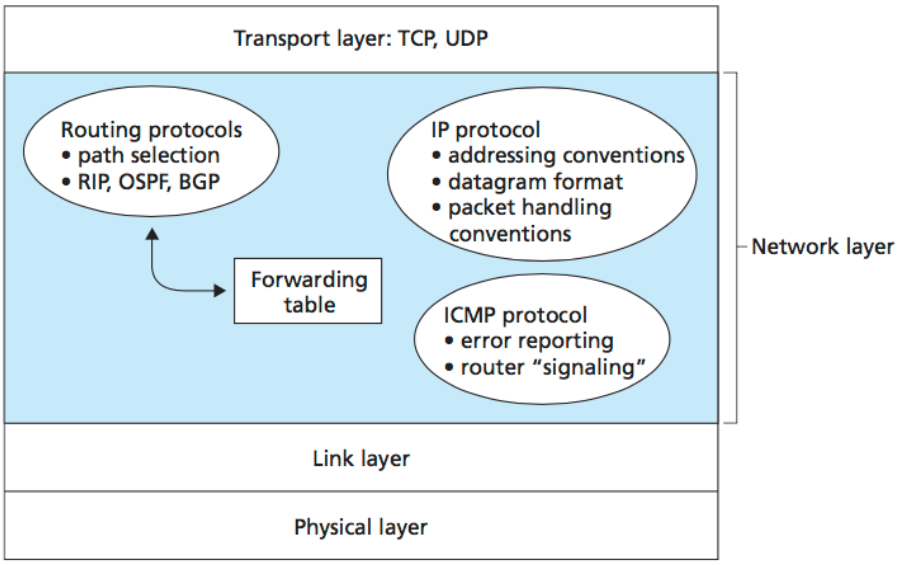


Bloqueo en Puerto de Entrada



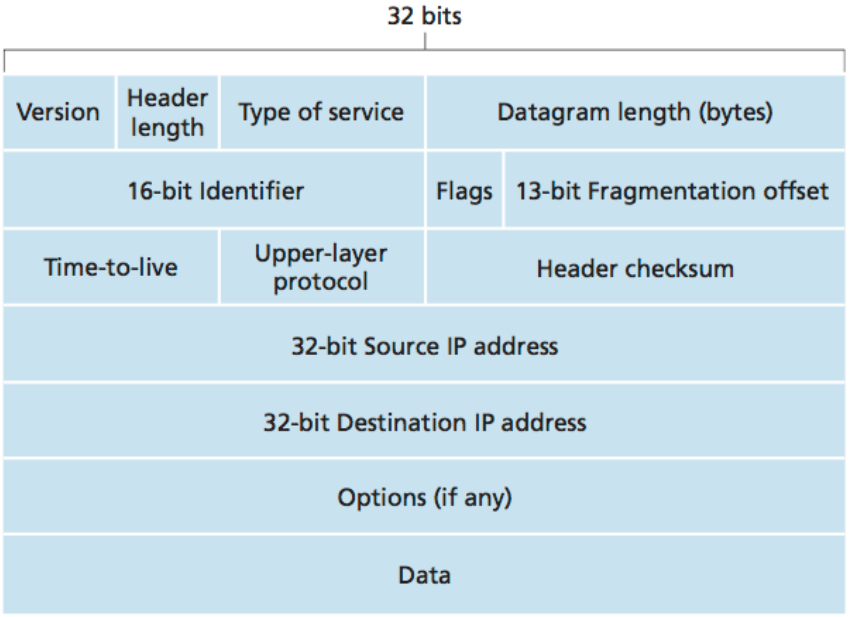
4.4. El Protocolo de Internet, IP: Reenvío y Direccionamiento en Internet

La Capa de Red

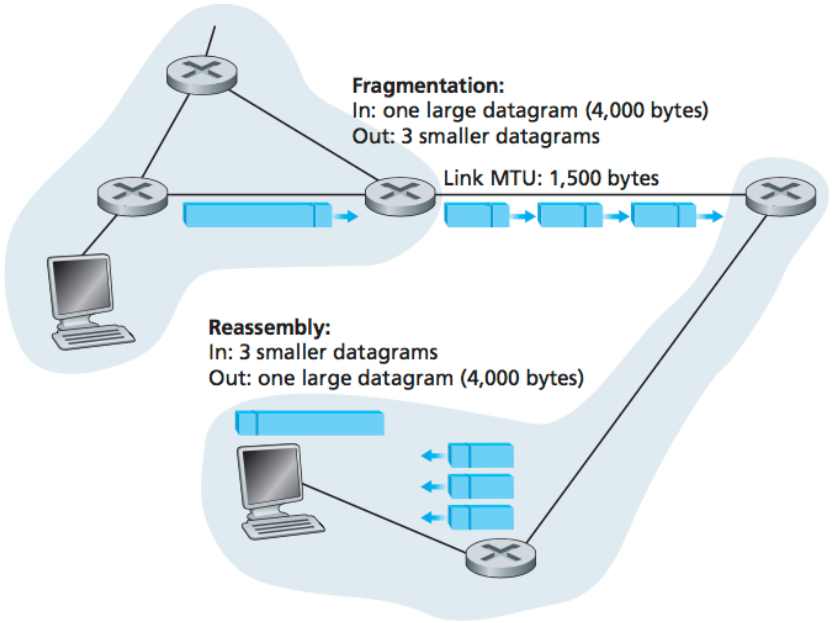


4.4.1. Formato de Datagrama

Formato del Datagrama IPv4



Fragmentación y Reensambado en IP



Fragmentación y Reensambado en IP

- Datagrama 4000 bytes (20 + 3980)
- MTU 1500 bytes

Fragmento	Bytes	ID	Desplazamiento	Flag
1º	1480	777	0	1
2º	1480	777	185 (1480/8)	1
3º	1020 (3980 − 2960)	777	370 (2960/8)	0

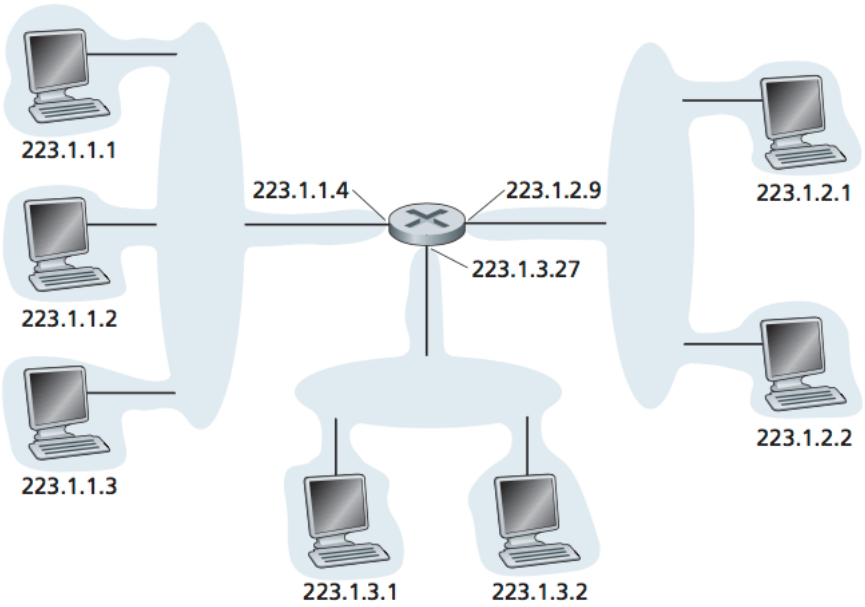
4.4.2. Direcciones IPv4

Direcciones IPv4

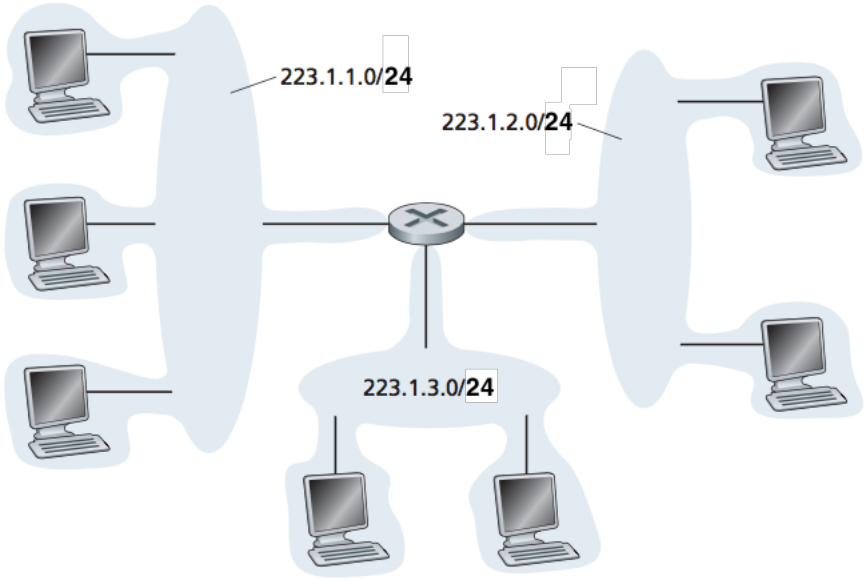
- Host, $\geq$ 1 interfaces de red
- Router, $\geq$ 2 interfaces de red
- Interfaz con *dirección IP única* asignada
- 32 bits, notación punto decimal
  - 193.32.216.9
  - `11000001 00100000 11011000 00001001`
- Tipos

- *Unicast*, uno a uno
- *Multicast*, uno a un grupo
- *Broadcast*, uno a todos

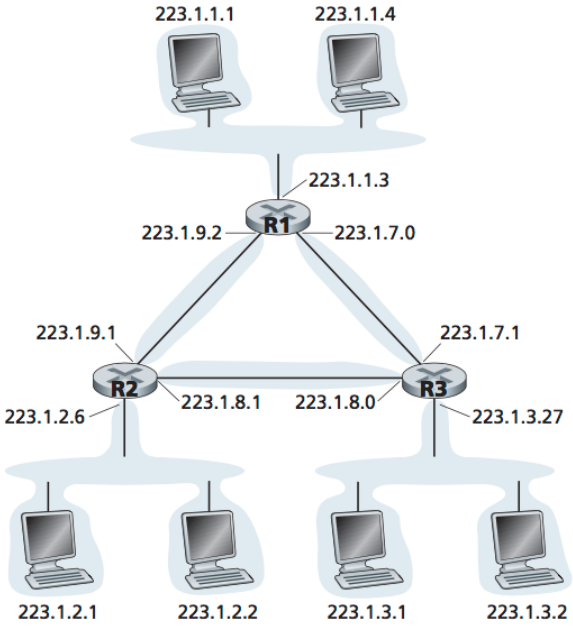
Direcciones de Interfaces y Subredes



Direcciones de Subredes



Tres Routers Conectando Seis Subredes

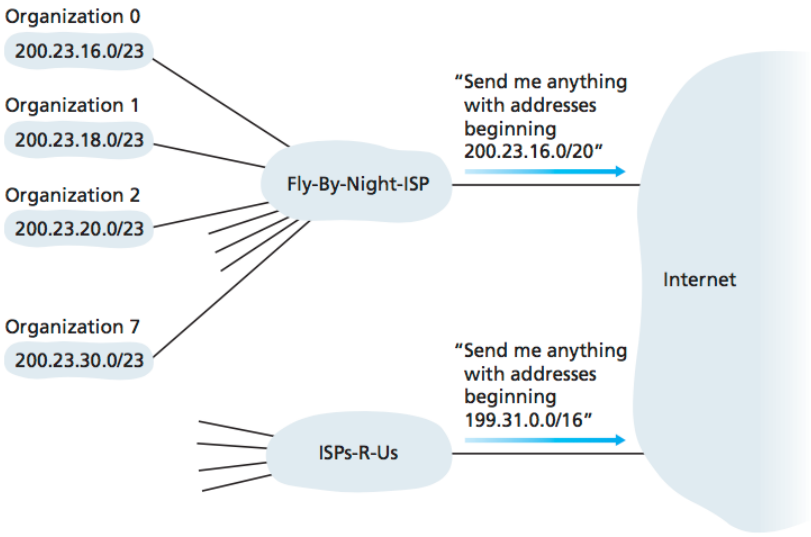


Encaminamiento entre Redes sin Clases, CIDR

- *Classless Interdomain Routing* [RFC 4632]

- `a.b.c.d/x`
- Los x bits más significativos son la dirección de la subred
- Bloques de dirección consecutivas compartiendo prefijo
- *Broadcast*: `255.255.255.255`
- Si dir IP host: `157.88.125.250/24`
  - Máscara: `255.255.255.0`
  - Subred: `157.88.125.0/24`
  - Todos los hosts de la subred: `157.88.125.255/24`
- Anteriormente, clases de direcciones
  - Clase A (/8): $2^{24} - 2$ hosts/subred
  - Clase B (/16): $2^{16} - 2$ hosts/subred
  - Clase C (/24): $2^{8} - 2$ hosts/subred
  - Clase D : *multicast*

Direccionamiento Jerárquico y Agregación de Rutas

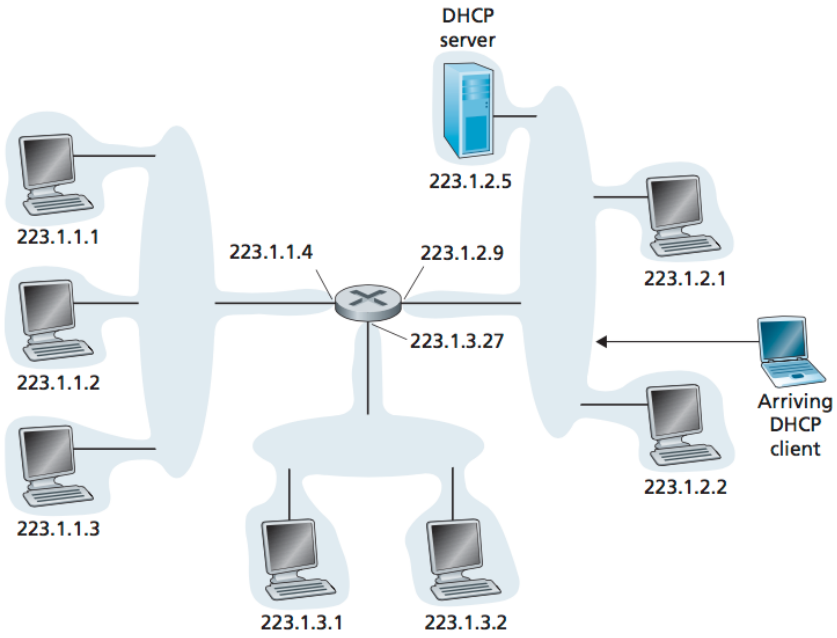


Obteniendo Dirección IP: Protocolo de Configuración Dinámica de Host, DHCP

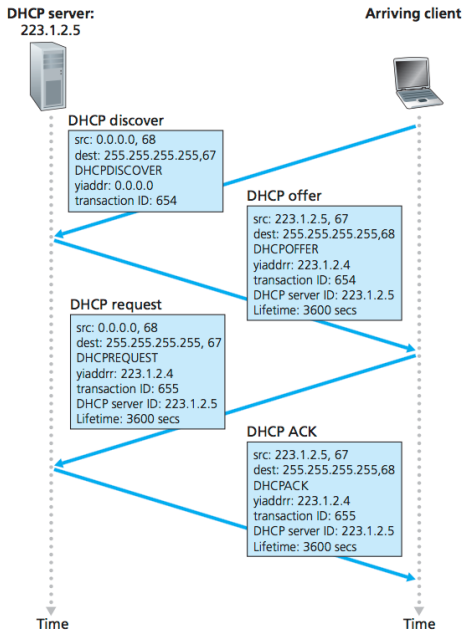
- Configuración manual de host: administrador asigna dirección IP y otros parámetros necesarios

- *Dinamic Host Configuration Protocol* [RFC 2131]
- Recibe la misma dirección IP cada vez que se conecta, o una distinta
- Además recibe máscara de red, dir. IP del router por defecto (primer salto) y servidor DNS local
- *Plug-and-play*, común en redes inalámbricas
- DHCP cliente-servidor (puerto 67 UDP)
    1. Descubrimiento servidor DHCP (agente *relay*)
    2. Ofrecimiento de servidor/es
    3. Solicitud
    4. Reconocimiento

Escenario DHCP Cliente-Servidor



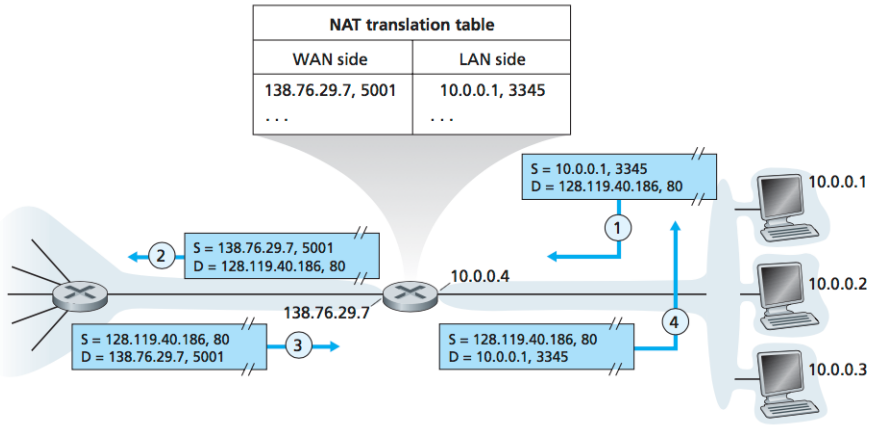
Interacción Cliente-Servidor DHCP



Traducción de Direcciones de Red, NAT

- *Network Address Translation*, NAT [RFC 2663]
- Problema direcciones subredes SOHO (*Small Office, Home Office*) donde ISP sólo asigna una dirección IP por conexión (mediante DHCP)
- Direcciones IP privadas
  - `10.0.0.0/8` (`10.0.0.0 - 10.255.255.255`)
  - `172.16.0.0/12` (`172.16.0.0 - 172.31.255.255`)
  - `192.168.0.0/16` (`192.168.0.0 - 192.168.255.255`)
- Apropiado para cliente (interno) - servidor (externo)
- *Universal Plug and Play*, UPnP
- P2P?

Traducción de Direcciones de Red



4.4.3. Internet Control Message Protocol, ICMP

Internet Control Message Protocol, ICMP

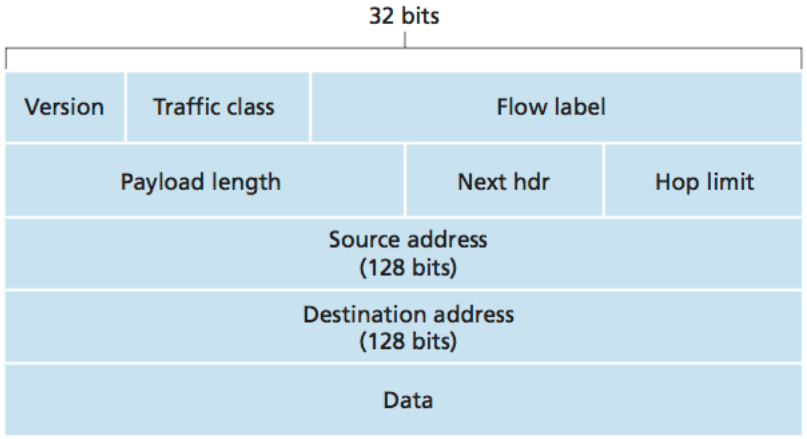
- *Internet Control Message Protocol*, ICMP [RFC 792]
- Intercambiar información de la capa de red entre hosts, routers
  - Normalmente situaciones de error
- Arquitectónicamente justo sobre IP
  - Paquetes ICMP encapsulados en datagramas IP
- Mensajes ICMP con *tipo* y *código*, además de cabecera y primeros 8 bytes del datagrama que causó envío del mensaje

Internet Control Message Protocol, ICMP

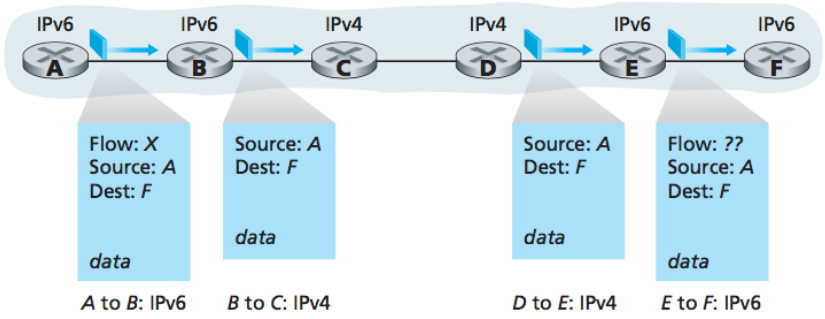
Tipo	Código	Descripción
0	0	Respuesta de eco (`ping`)
3	0	Red destino inalcanzable
3	1	Host destino inalcanzable
3	2	Protocolo destino inalcanzable
3	3	Puerto destino inalcanzable
3	6	Red destino desconocida
3	7	Host destino desconocido
4	0	Ralentizar fuente
8	0	Solicitud de eco (`ping`)
9	0	Anuncio de ruta
10	0	Descubrimiento de ruta
11	0	TTL expirado (`traceroute`)
12	0	Cabecera IP mal

4.4.4. IPv6

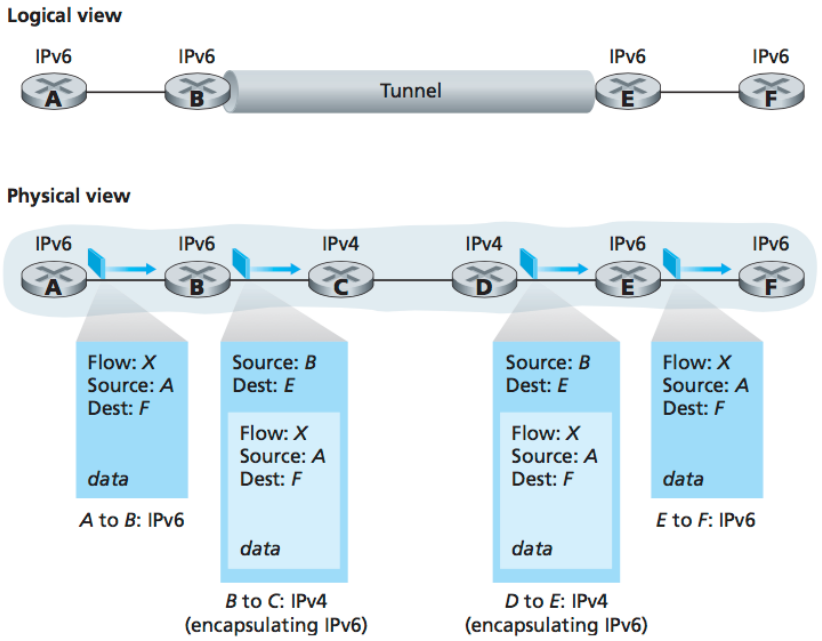
Formato Datagrama IPv6



Enfoque Pila Dual



Tunelado



4.4.5. Breve Incursión en la Seguridad IP

Seguridad IP

- IP diseñado sin consideraciones sobre seguridad
- *IPsec*
  - Origen: encripta segmento, añade campos al mismo y lo encapsula en un datagrama IP ordinario
  - Destino: desencapsula segmento encriptado, lo desencripta y entrega a la capa de transporte
- *Virtual Private Network*, VPN, proporciona un túnel cifrado con autenticación entre extremos
- Comunicación segura en redes públicas (Internet)

4.5. Algoritmos de Encaminamiento

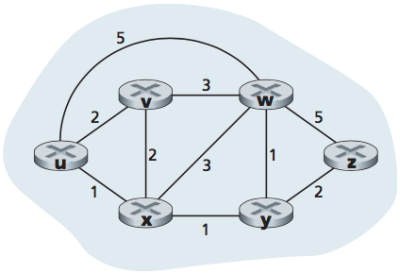
Algoritmos de Encaminamiento

- Cada host conectado a su router *por defecto* (primer salto)
  - Host transfiere paquetes salientes a su router por defecto para que se encargue (router origen)

    - Router del host destino, reenviará paquetes al host destino

- Problema de *encaminamiento*: encontrar una ruta entre el router origen y el destino

Modelo de Grafo para una Red de Computadoras



- *Grafo* $G = (N, E)$, conjunto N de nodos y colección E de arcos, donde un arco es un par de nodos.
- Los arcos pueden tener un valor representando el *coste* de recorrerlo, $c(x, y)$
- *Camino* $x_1, x_2, \ldots, x_p$
- *Coste del camino* es la suma del coste de los arcos del camino
    - *Camino de menor coste*
    - *Camino más corto*

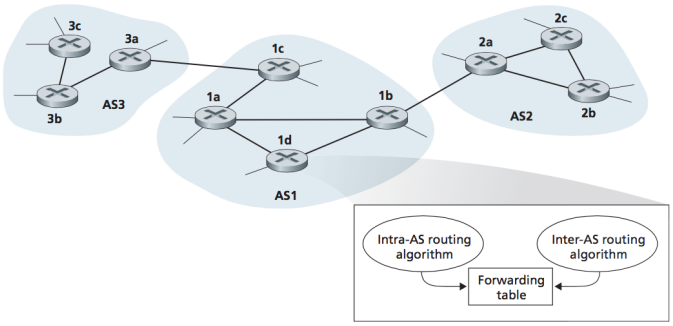
Algoritmos de Encaminamiento

- *Algoritmo encaminamiento global*, conocimiento completo de la red
    - Algoritmos de *estado de enlace*
- *Algoritmo encaminamiento descentralizado*, cada router sólo conoce el coste de sus enlaces
    - Algoritmos de *vector de distancias*
- Algoritmos *estáticos*, *dinámicos*
- Algoritmos *sensibles* o *insensibles* a la carga

4.5.1. Encaminamiento Jerárquico

Sistemas Autónomos

- Gran cantidad de routers intercambiando información
  - Escala
  - Autonomía administrativa
- *Sistema autónomo*, *Autonomous System*, AS
- Routers bajo una misma autoridad administrativa
- Protocolo encaminamiento dinámico intra-AS
- Routers *gateway*, conectan con otros ASs, inter-AS



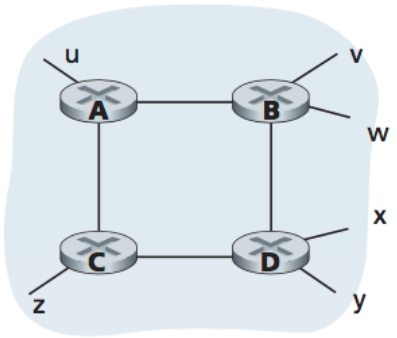
4.6. Encaminando en Internet

4.6.1. Routing Internet Protocol, RIP

Routing Internet Protocol, RIP

- *Routing Information Protocol*, RIP [RFC 2453]
- UDP, puerto 520
- Vector de distancias, intra-AS
- Coste enlace = 1
- Coste camino = *saltos* desde router origen a la subred destino
- Coste máximo camino 15
  - Diámetro máximo AS
- Intercambio entre vecinos de hasta 25 destinos (*anuncios*) cada 30 s
- Si no recibe anuncios en 180 s, router o enlace caídos, cambia tabla encaminamiento y propaga

Saltos desde Router A hacia subredes



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

Porción de un Sistema Autónomo

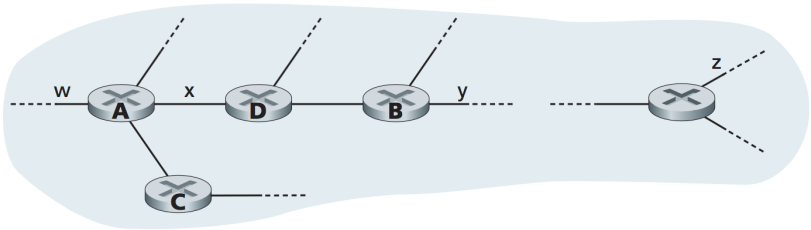
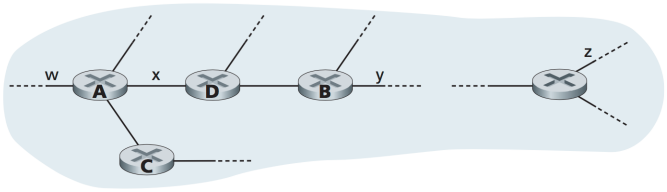


Tabla Encaminamiento en D antes de Recibir Anuncio de A

Subred Destino	Siguiente Router	Num. Saltos
w	A	2
y	B	2
z	B	7
x	–	1
...	...	...



Anuncio del Router A

Subred Destino	Siguiente Router	Num. Saltos
z	C	4
w	–	1
x	–	1
...	...	...

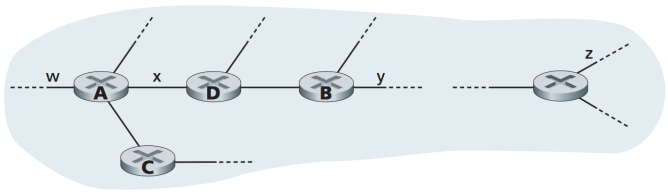
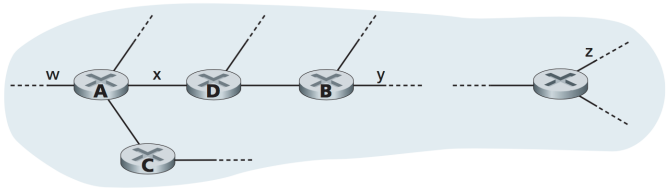


Tabla Encaminamiento de A tras Recibir Anuncio de A

Subred Destino	Siguiente Router	Num. Saltos
w	A	2
y	B	2
z	A	5
x	–	1
...	...	...



4.6.2. Open Shortest Path First, OSPF

Open Shortest Path First, OSPF

- *Open Shortest Path First*, OSFP [RFC 2328]
- Estado de enlace, intra-AS,
- Algoritmo Dijkstra camino menor coste a todas las subredes
- Coste enlaces configurable
- Difusión en cambios en enlace o periódico (30 min)
  - Directamente sobre IP
- Mensajes `HELLO`

Open Shortest Path First, OSPF

- *Seguridad*, autenticación contraseña compartida
- *Múltiple caminos con mismo coste*
- Anuncios enviados sólo entre routers, *multidifusión*, *multicast*
- *Jerarquía* routers en sistema autónomo, *áreas*

4.6.3. Border Gateway Protocol, BGP

Border Gateway Protocol, BGP

- *Border Gateway Protocol*, BGP [RFC 4271]
- Protocolo inter-AS *de facto* en Internet
- Vector de caminos (símil DV)
- Permite que subred anuncie su existencia y a toda Internet, cómo alcanzarla

4.7. Resumen

Resumen

- Capa de red presente en cada host y router de la red
- Routers procesan y reenvían todos los paquetes que cruzan la red
- Cada paquete es tratado independientemente
- Estructura de datagramas con información necesaria para ser encaminados hacia su destino
- IPv4 e IPv6
- Algoritmos de encaminamiento
  - Vector de Distancias (RIP)
  - Estado de Enlace (OSPF)
- Sistemas autónomos
  - Intra-AS: RIP, OSPF
  - Inter-AS: BGP

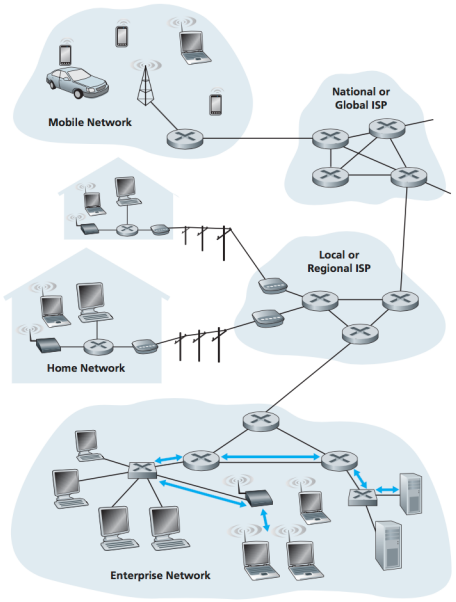
5. Capa de Enlace

5.1. Introducción

Introducción

- *Nodo* transmite *marcos* sobre *enlaces* a nodo adyacente
- Datagrama encapsulado en marco atraviesa enlace, salto a salto, posiblemente de distinto tipo

Introducción a la Capa de Enlace



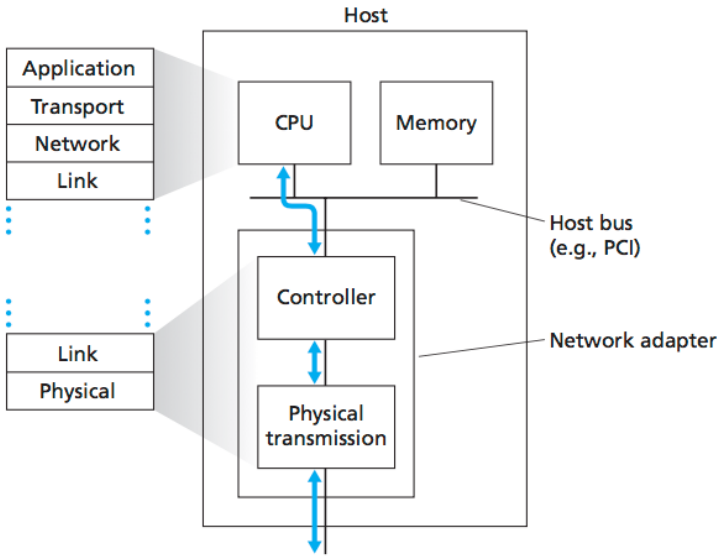
5.1.1. Servicios Proporcionados por la Capa de Enlace

Servicios Capa Enlace

- Enmarcado
- Acceso al enlace
- Entrega fiable
- Detección y corrección de errores

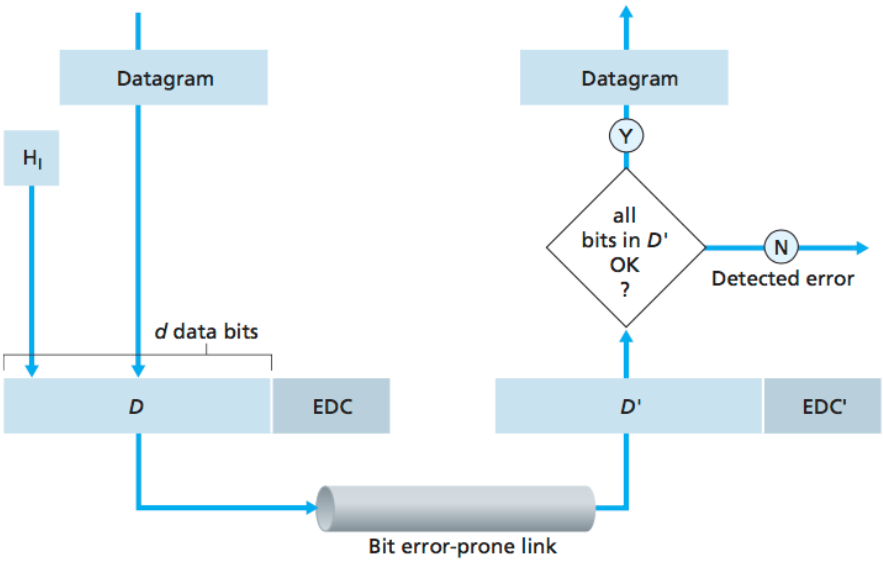
5.1.2. Dónde está Implementada la Capa de Enlace

Interfaz de Red y Pila de Protocolos



5.2. Técnicas de Detección y Corrección de Errores

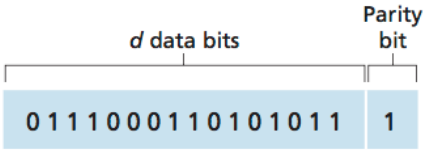
Escenario de Detección y Corrección de Errores



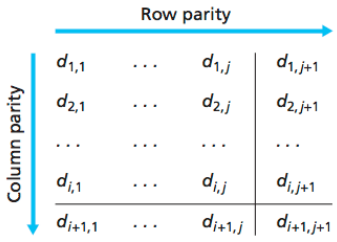
5.2.1. Comprobación de Paridad

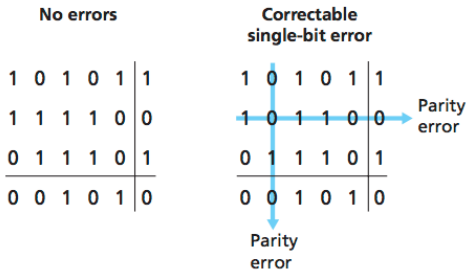
Comprobación de Paridad

- *Paridad par* o impar, añadir un bit para conseguir número de 1s sea par o impar
- *Corrección adelantada de errores*, *Forward Error Correction*, FEC



Paridad Par Bidimensional





5.2.2. Códigos de Redundancia Cíclica

Códigos de Redundancia Cíclica

- Datos a enviar, D, d bits
- CRC, R, r bits
- G, generador, $r + 1$ bits (bit más significativo a 1)
- $D\,concat\,R$, tal que número $d + r$ bits es divisible exacto por G (aritmética módulo 2)

- $D \cdot 2^r XOR\, R = nG$
- Puede detectar errores en ráfaga de hasta r bits
- Y cualquier número impar de errores de bit

5.2.3. Métodos de Suma de Comprobación

Suma de Comprobación

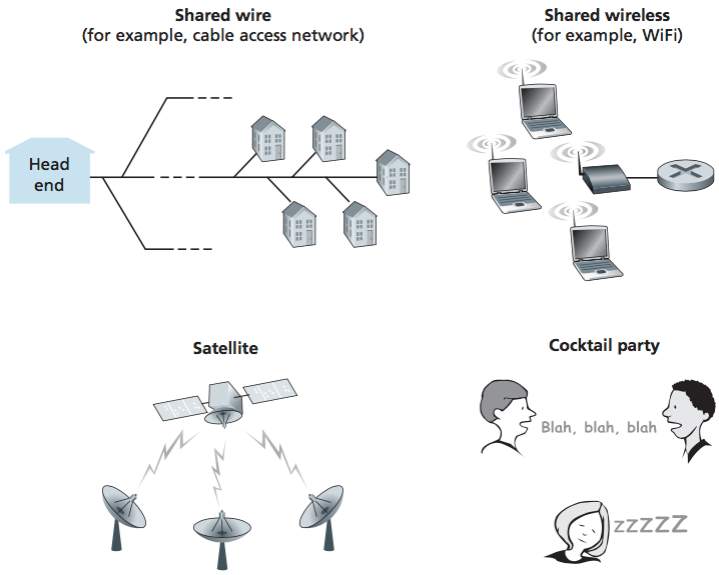
- *d* datos, tratados como una suma de *k* bits
- IP, checksum sobre cabecera
- TCP y UDP, checksum sobre paquete completo
- Por qué checksum en transporte y CRC en enlace?
  - Checksum software
  - CRC hardware

5.3. Protocolos y Enlaces de Acceso Múltiple

Enlaces de Acceso Múltiple

- Enlaces *punto a punto*
- Enlaces de *difusión* (*broadcast*)

Canales de Acceso Múltiple



Enlaces de Acceso Múltiple

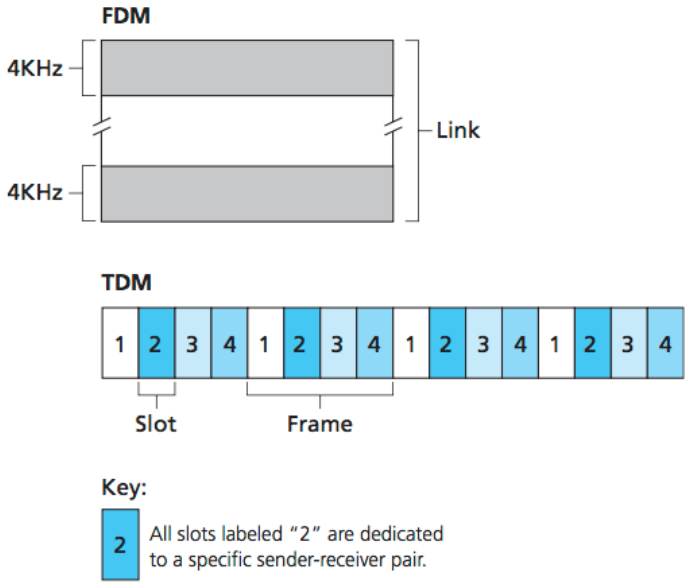
- Cómo coordinar el acceso de varios nodos al canal de difusión?
- Si *colisión*, pérdida de paquetes, desperdicio del canal
- *Protocolos de acceso múltiple*
  - Partición del canal
  - Acceso aleatorio (*arbitrario*)
  - Asignación de turnos

Características Ideales Acceso Múltiple

- Si canal de R bps
  1. Si un sólo nodo, emite a R bps
  2. Si M nodos, cada uno emite R/M bps por término medio en t segundos
  3. Descentralizado
  4. Simple

5.3.1. Protocolos de Partición del Canal

Ejemplo TDM y FDM de 4 Canales

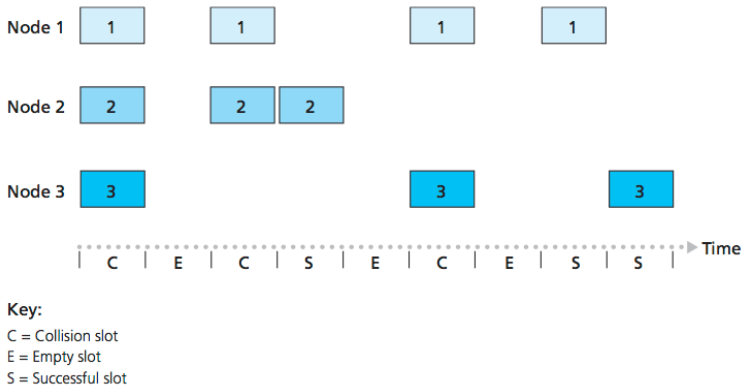


5.3.2. Protocolos de Acceso Aleatorio

Acceso Aleatorio

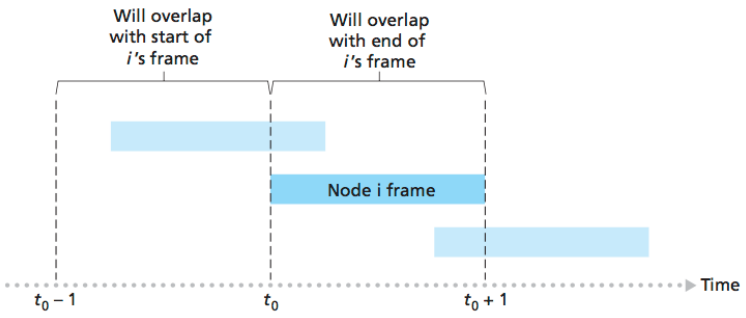
- Si un nodo tiene que transmitir, lo hace a R bps
- Si colisión, lo vuelve a intentar tras un retardo aleatorio

Aloha Ranurado



- Eficiencia 37 %
- Qué podemos mejorar?

Interferencia entre transmisiones en Aloha Puro

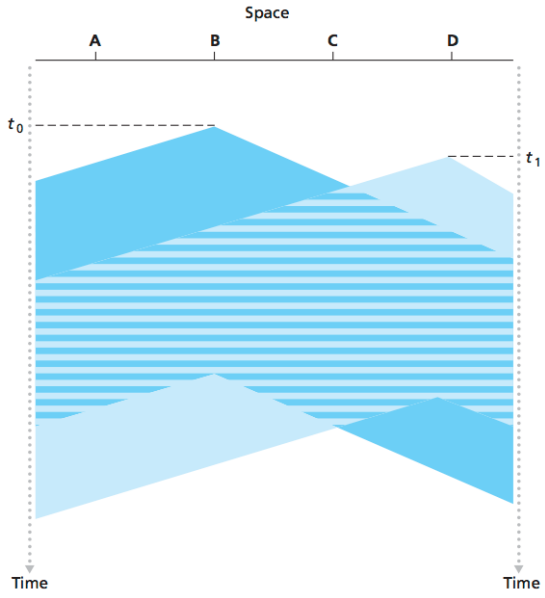


- Eficiencia 18 %
- Qué podemos mejorar?

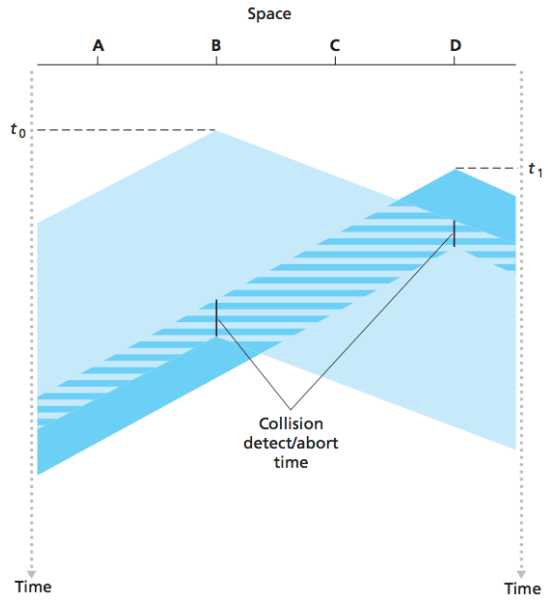
Acceso Múltiple con Detección de Portadora, CSMA

- En ALOHAs, los nodos no prestan atención al resto
  - Transmiten incluso cuando otros lo hacen
  - No interrumpen la transmisión en caso de colisión
- Reglas conversación humana
  - Escuchar antes de hablar $\rightarrow$ *Detección de portadora*
  - Si alguien más habla al mismo tiempo, callar $\rightarrow$ *Detección de Colisión*
- *Carrier Sense Multiple Access*, CSMA
- *Carrier Sense Multiple Access with Collision Detection*, CSMA/CD

Colisión en CSMA



Colisión en CSMA/CD



Cuanto tiempo esperar tras colisión?

- Dependerá de cuantas colisiones se produzcan
- Tiempo espera exponencial (*binary exponential backoff time*)
- Si n colisiones, elegir un valor k aleatoriamente de $0, 1, 2, \ldots, 2^n - 1$
- Ethernet, $K \cdot tiempo_{transmitir\ 512\ bits}$, con $n \leq 10$
- $Eficiencia = \frac{1}{1+5d_{prop}/d_{trans}}$
  - si $d_{prop} \to 0 \Rightarrow eficiencia \to 1$
  - si $d_{trans} \to \infty \Rightarrow eficiencia \to 1$

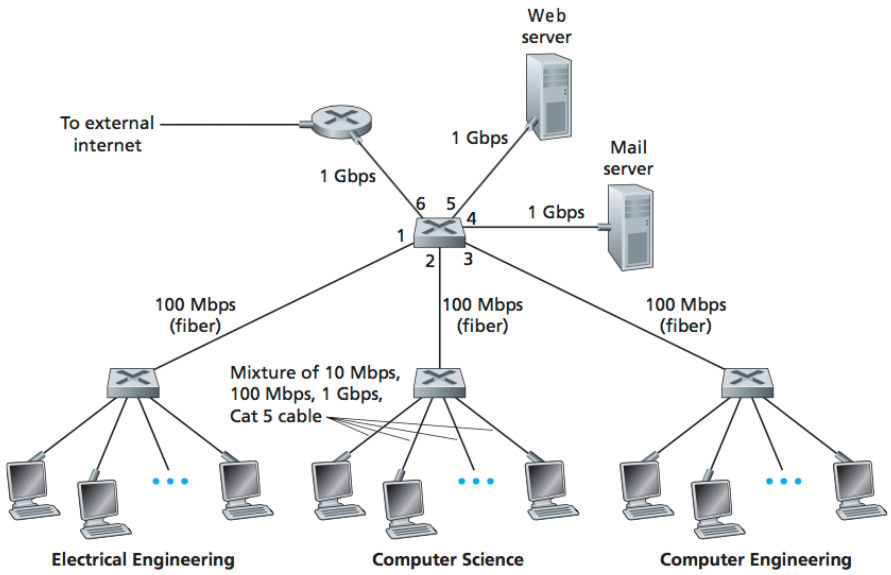
5.3.3. Protocolos de Turnos

Protocolos de Turnos

- *Encuesta*
  - Nodo maestro interroga cíclicamente resto nodos sobre si desean transmitir
  - Bluetooth
- *Paso de testigo*
  - Un paquete especial, *testigo*, circula por la red y habilita a enviar a los nodos que lo reciben
  - FDDI, Token Ring

5.4. Redes de Área Local Conmutadas

Red Institucional con Switches



5.4.1. Direccionamiento de la Capa de Enlace y ARP

Direccionamiento de la Capa de Enlace y ARP

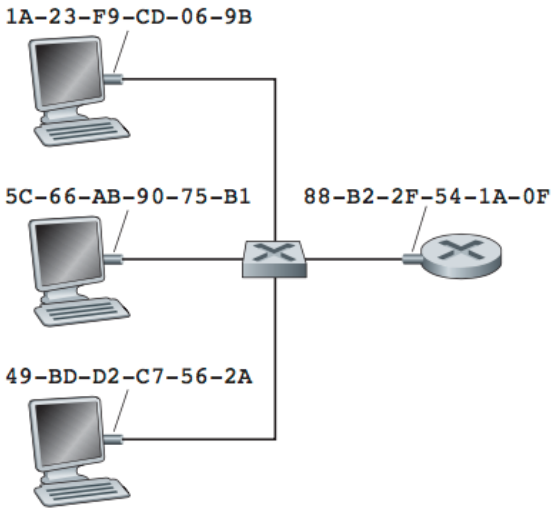
- Host y routers, tienen direcciones del nivel de enlace (vía interfaces)
- Por qué direcciones de red y de enlace?
- *Address Resolution Protocol*, ARP traduce direcciones IP a direcciones de enlace

Direcciones MAC

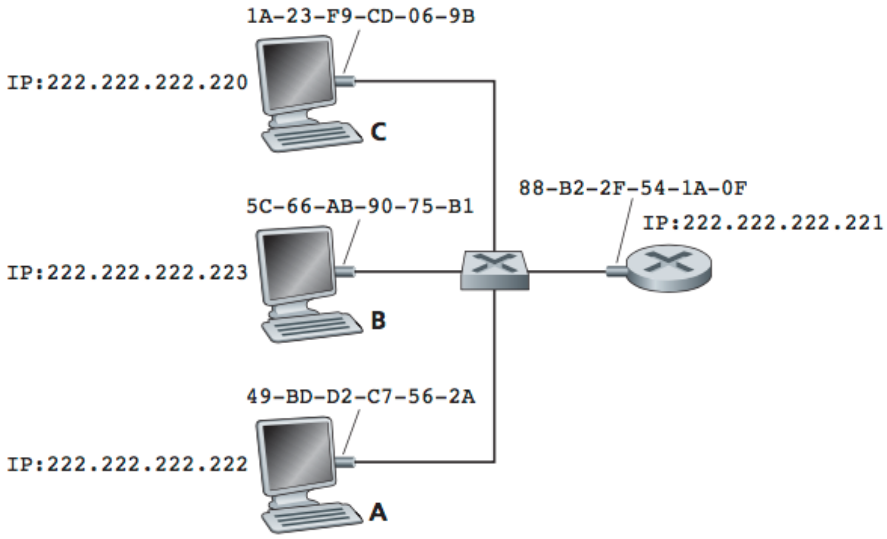
- Dirección de enlace asociada a la interfaz de red
  - Dirección *Medium Access Control*, MAC
  - Dirección física
  - Dirección LAN
- 6 bytes, 48 bits, hexadecimal (`ac:bc:32:a9:11:a1`)
- Estructura plana
- $\nexists$ dos iguales
  - IEEE fija bloques 24 bits al fabricante

- Aunque asignadas por hardware, pueden cambiarse por software
- Invariables respecto a la localización de la interfaz en la red
- *Broadcast* `ff:ff:ff:ff:ff:ff`

Interfaces y Direcciones MAC



Interfaces y Direcciones MAC e IP



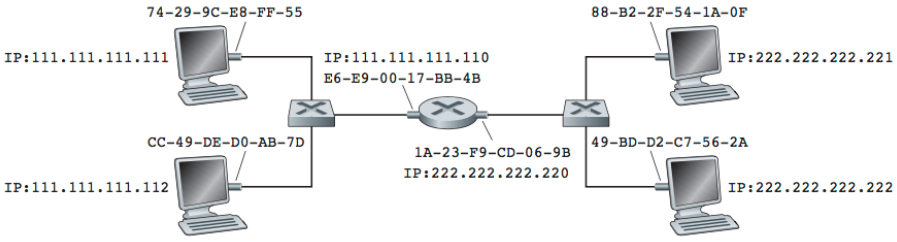
Address Resolution Protocol, ARP

- *Address Resolution Protocol*, ARP [RFC 826]
- Resuelve direcciones IP de interfaces conectadas en la misma subred
- *Tabla ARP* de interfaz 222.222.222.220

Dir. IP	Dir. MAC	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

- MAC(222.222.222.222) ?
- Consulta ARP dirigida a FF-FF-FF-FF-FF-FF
- Respuesta ARP normal
- Plug and Play

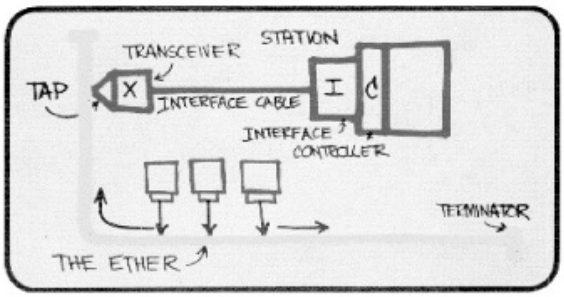
Subredes Interconectadas por un Router



5.4.2. Ethernet

Ethernet

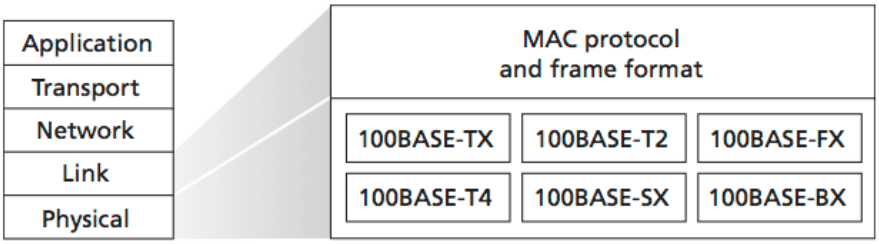
- IEEE 802.3 CSMA/CD [IEEE 802.3]
- Tecnología dominante en LAN cableadas
  - 1970s, Bob Metcalfe y David Boggs
- Originalmente bus (*broadcast*)
- 1990s, estrella basada en *hub*, (*broadcast*)
- 2000s, estrella basada en *switch*, separación *dominios colisión*
  - Conmutador de almacenamiento y reenvío de nivel 2



Marco Ethernet

Estándares Ethernet

- Ethernet especifica capa enlace y capa física
- 100BaseT
  - 100 Mbps
  - Banda base
  - Pares de cobre trenzado



- Evolución 10, 100, 1G, 10G
- Mismo formato de marco

5.4.3. Conmutadores de la Capa de Enlace

Switches

- Recibir paquete entrante y reenviarlo por puertos salientes

    - Transparente a los hosts
    - *Buffers* para adaptar tasas entrada y del enlace
- *Filtrado*
- *Reenvío*

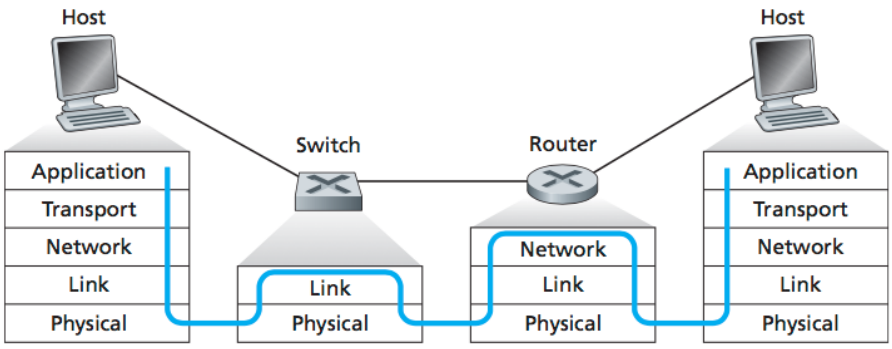
Dirección	Interfaz	Tiempo
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	2	9:36

- Autoaprendizaje, *plug-and-play*

Propiedades de los Switches

- Eliminación de las colisiones (*dominios de colisión*)
- Enlaces heterogéneos, convivencia varios tipos
- Facilita la gestión
    - Fallos interfaces (*jabbering*)
    - Corte cable
    - Estadísticas uso

Procesamiento de Paquetes en Hosts, Switches y Routers



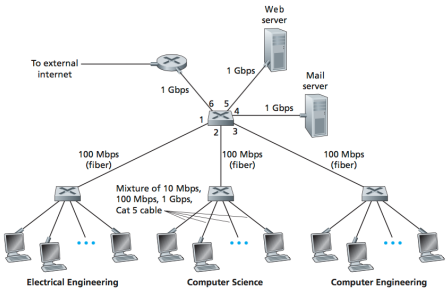
Comparación Hubs, Switches y Routers

	Hubs	Switches	Routers
Aislamiento del tráfico	No	Sí	Sí
Dominios colisión separados	No	Sí	Sí
Dominios difusión separados	No	No	Sí
Plug-and-play	Sí	Sí	No
Encaminamiento óptimo	No	No	Sí

- Utilizar switch o router?
  - Número hosts (tráfico difusión)
  - *Conmutar dentro, encaminar fuera*

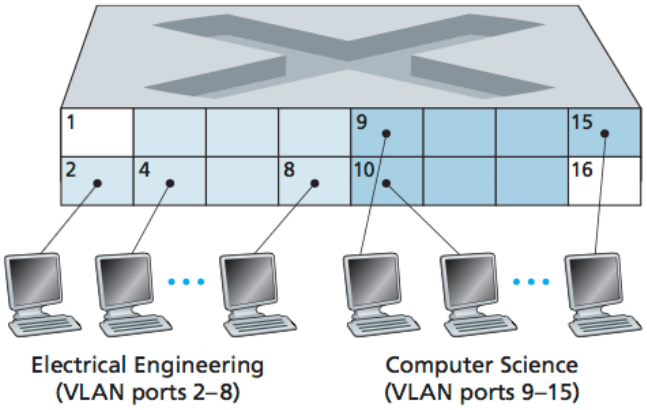
5.4.4. Redes de Área Local Virtuales, VLANs

Redes de Área Local Virtuales, VLANs



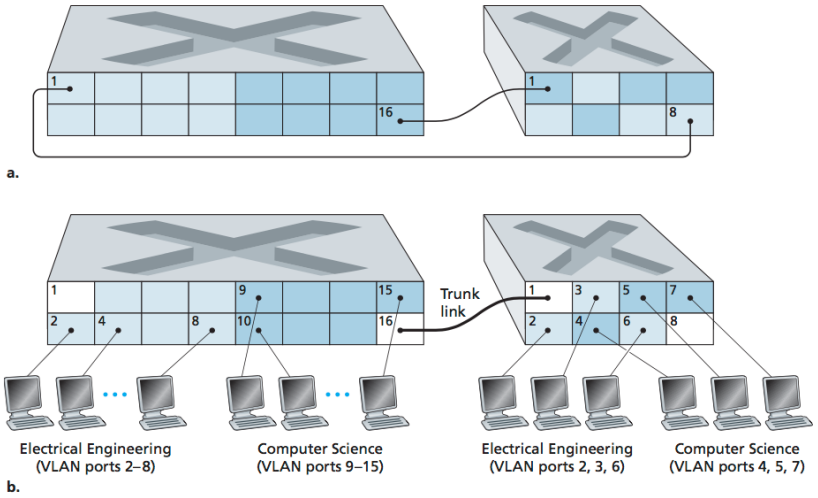
- Problemas
  - Dominios difusión compartidos
  - Asociación switch-host lógica y física
- *VLANs*, múltiples LANs virtuales definidas sobre una única LAN física

Dos VLANs en un Switch



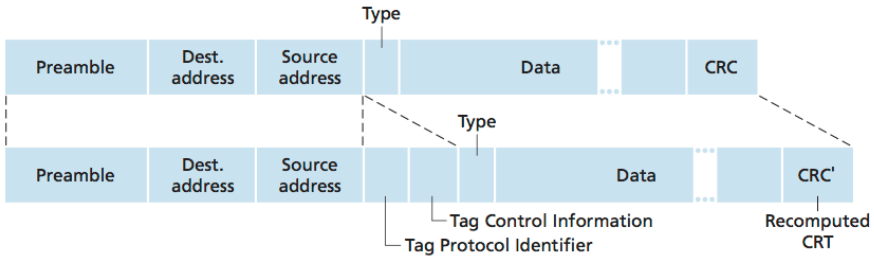
- Cómo conectar ambas VLANs (subredes)?
- Extender VLAN en varios switches?

Dos VLANs en Dos Switches



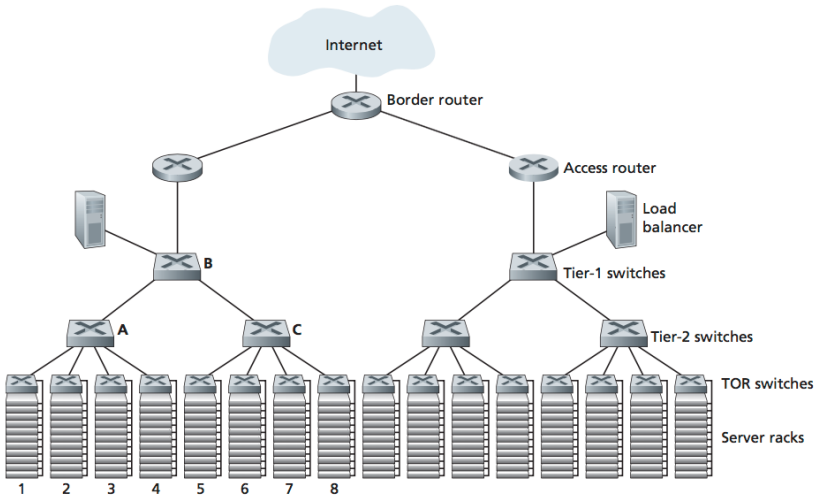
Marco Ethernet y 802.1Q

- *VLAN trunking*
- IEEE 802.1Q, sólo marcos cruzando un enlace tipo *trunk* (troncal)
- Incluye identificador VLAN 12 bits

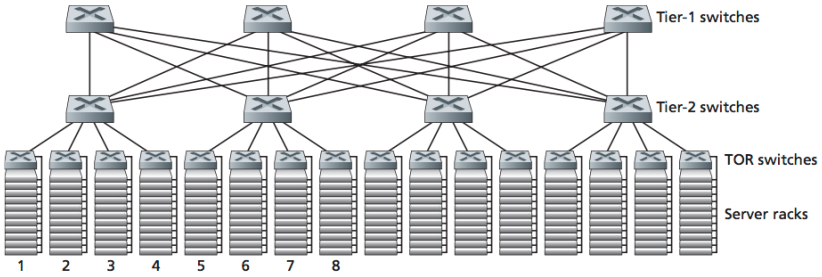


5.5. Red de Centro de Datos

Topología Red Jerárquica con Reparto de Carga

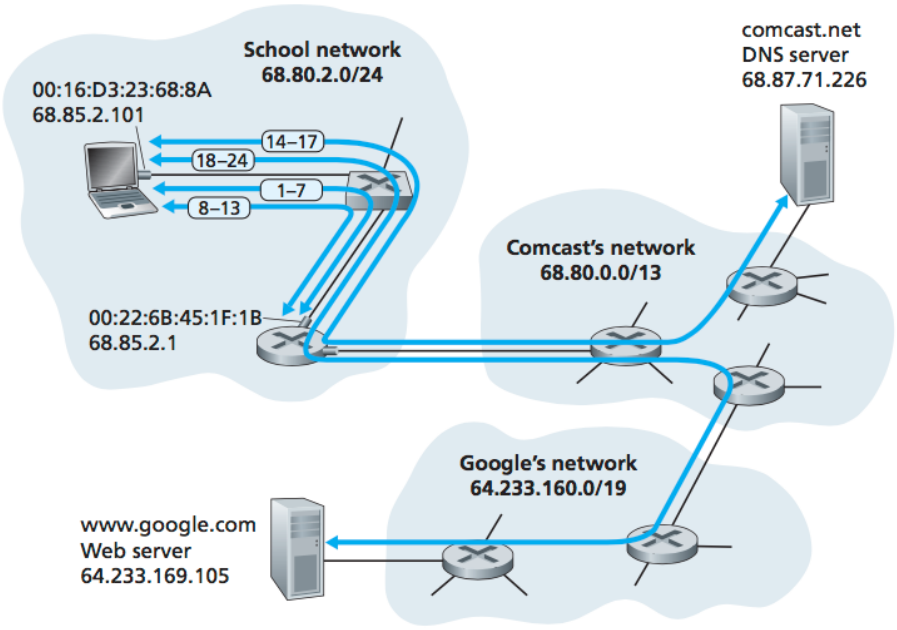


Topología Red Altamente Conectada



5.6. Retrospectiva: Un Día en la Vida de una Solicitud de una Página Web

Un Día en la Vida de una Solicitud Web



Un Día en la Vida de una Solicitud Web

- (1 - 7) Obtener configuración red, DHCP
- (8 - 13) Enviar solicitud DNS, ARP para alcanzar router por defecto
- (14 - 17) Resolver nombre dominio, DNS
- (18 - 24) Establecer conexión TCP para soportar solicitud y respuesta HTTP

5.7. Resumen

Resumen

- Enlace mueve datagramas en marcos, de un nodo al siguiente adyacente
- Encapsulación datagrama en marco
- Diferentes protocolos, diferentes servicios
- Punto a punto, acceso múltiple
- Control de acceso múltiple, CSMA/CD
- Detección y corrección de errores de bit
- Ethernet

- Direcciones MAC y direcciones IP, ARP
- Redes conmutadas, switches
- Redes virtuales, VLANS y protocolo de trunking
- Centros de datos
- Acompañamos a una petición HTTP a través de la red

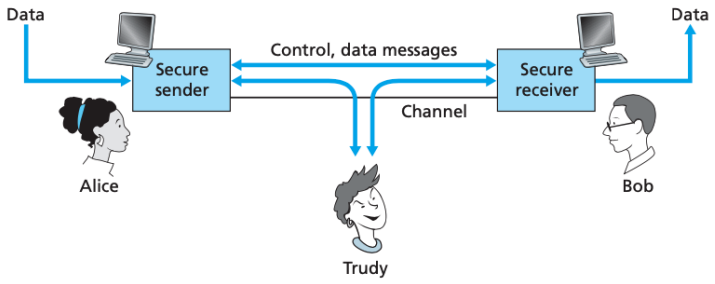
6. Seguridad en Red

6.1. Qué es Seguridad en Red?

Qué es Seguridad en Red?

- Comunicación *segura*
  - Confidencialidad
  - Integridad
  - Autenticación entre extremos
  - Disponibilidad
- Seguridad operativa en organizaciones

Emisor, Receptor e Intruso

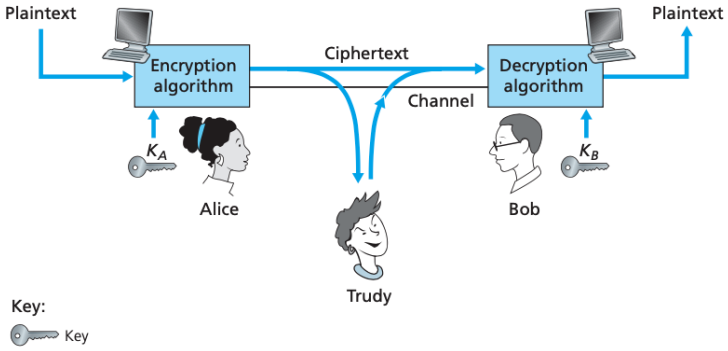


Qué Puede Hacer un Atacante?

- Fisgar, modificar, insertar o borrar mensajes
- Ataques de seguridad:
  - Husmear comunicaciones
  - Suplantar identidades
  - Denegar servicio
- *Criptografía* como medio de asegurar confidencialidad, integridad y autenticidad

6.2. Principios de Criptografía

Principios de Criptografía



6.2.1. Criptografía de Clave Simétrica

Criptografía de Clave Simétrica

- $K_A = K_B$, secreto compartido
- Ejemplos
  - Cifrado César desplaza alfabeto i caracteres, $K = i$
  - Cifrado por sustitución monoalfabético o polialfabético

Plaintext letter:	a b c d e f g h i j k l m n o p q r s t u v w x y z
Ciphertext letter:	m n b v c x z a s d f g h j k l p o i u y t r e w q

Plaintext letter:	a b c d e f g h i j k l m n o p q r s t u v w x y z
$C_1(k = 5)$:	f g h i j k l m n o p q r s t u v w x y z a b c d e
$C_2(k = 19)$:	t u v w x y z a b c d e f g h i j k l m n o p q r s

- Actualmente, cifrado de flujos (WiFi) y cifrado de bloques (SSL)

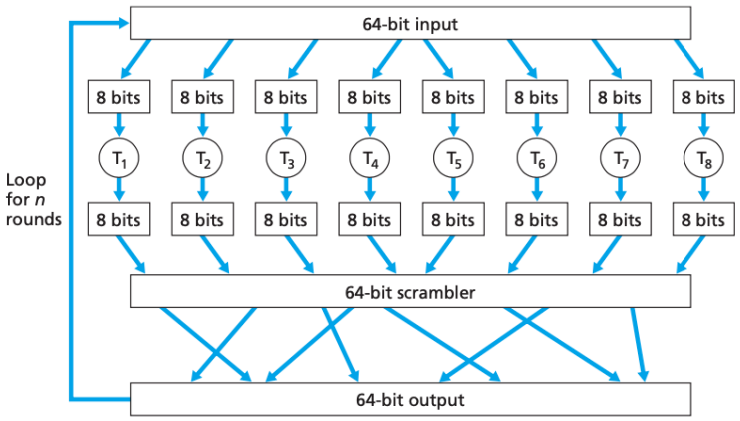
Romper Cifrado

- Dependiendo de la información que tenga el intruso
  - Ataque de texto cifrado
    - Ataque de fuerza bruta
    - Análisis estadístico
  - Ataque de texto plano conocido
    - Conoce alguno de los pares (texto plano, texto cifrado)
  - Ataque de texto plano elegido
    - Produce texto que será cifrado
    - *Pangramas*, [*Jovencillo emponzoñado de whisky: ¡qué figurota exhibe!*]

Cifrado de Bloques

- Usado en protocolos Internet
- Procesa el mensaje en bloques de k bits, cifrados independientemente
- $2^k!$ correspondencias
- Ataque fuerza bruta, aumentar k, difícil implementar tabla correspondencias
- Utilización funciones que simulan las correspondencias
  - Función conocida
  - Clave secreta
- *DES Data Encryption Standard*, bloques 64 bits, clave de 56 bits
- *AES Advanced Encryption Standard*, bloques 128 bits y claves de 128, 192 y 256 bits
- *3AES*

Cifrado de Bloques



Cifrado de Bloques

- Problema con el cifrado de grandes flujos de datos
- Añadir algo de aleatoriedad, dos bloques iguales no presenten el mismo cifrado
  - m_i, bloque i-ésimo de k bits
  - c_i, bloque i-ésimo cifrado
  - K_S, algoritmo de cifrado con clave S

- r_i, número aleatorio k bits
- $c_i = K_S(m_i \oplus r_i)$
- Emisor envía c_1, r_1, c_2, r_2, ...
- $m_i = K_S(c_i \oplus r_i)$
- Si $m_i = m_j \Rightarrow c_i \neq c_j$ con mucha probabilidad

Cifrado de Bloques Encadenado

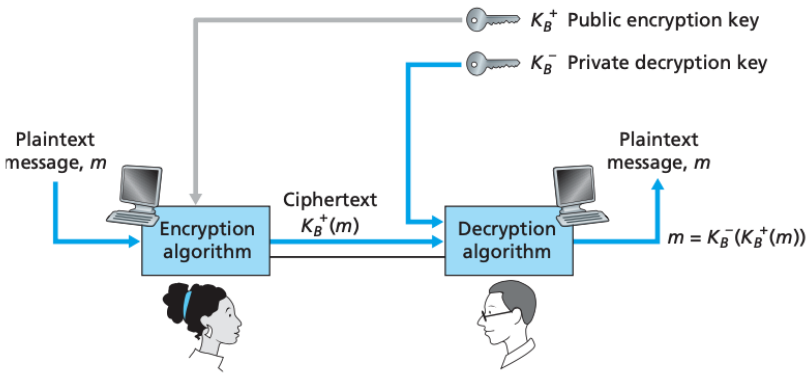
- *CBC*, *Cipher Block Chaining*
- Enviar únicamente un valor aleatorio con el primer bloque
- Calcular el resto a partir del bloque anterior
  - Emisor genera c_0 y envía al receptor como texto plano (*vector de inicialización*)
  - $c_1 = K_S(m_1 \oplus c_0)$
  - $c_i = K_S(m_i \oplus c_{i-1})\ \forall i > 1$

6.2.2. Criptografía de Clave Pública

Criptografía de Clave Pública

- Problema distribución de claves sobre canal inseguro
- 1970s Criptografía de clave pública
  - Diffie y Hellman, aritmética exponencial y modular
  - Rivest, Shamir y Adleman (RSA), factorización de grandes números
- *Encriptación* y *autenticación* (firma electrónica)
- Dos claves, una pública K^+ y otra privada K^- secreta
- Alice envía $K_B^+(m)$ a Bob
- Bob obtiene mensaje $m = K_B^-(K_B^+(m))$
- $K_B^-(K_B^+(m)) = K_B^+(K_B^-(m)) = m$

Criptografía de Clave Pública



Clave de Sesión

- RSA requiere muchos recursos
- DES 10^2 veces más rápido que RSA en software, 10^5 en hardware
- Combinar clave pública con simétrica
- Alice genera K_S clave simétrica, *clave de sesión*
- Alice envía $K_B^+(K_S)$
- Bob recupera $K_S = K_B^-(K_B^+(K_S))$
- $K_S(m)$

6.3. Integridad del Mensaje y Firmas Digitales

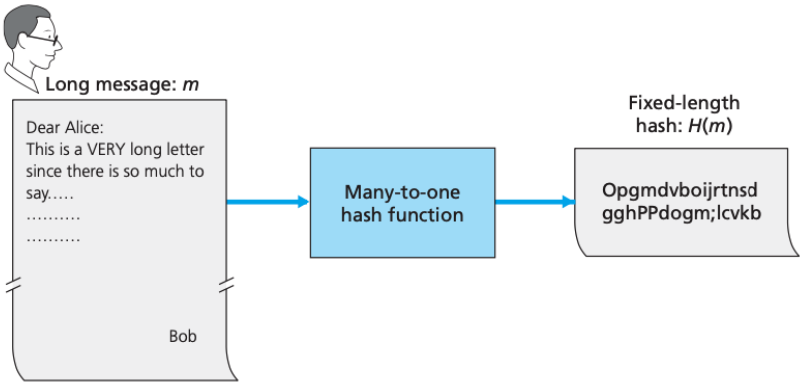
Integridad del Mensaje y Firmas Digitales

- *Autenticación del mensaje*
  - El mensaje ha sido generado por Alice
  - El mensaje no ha sido alterado en el camino hacia Bob
- Funciones de *dispersión criptográficas* (*hash*)

6.3.1. Funciones de Dispersión Criptográficas

Funciones de Dispersión Criptográficas

- Es computacionalmente imposible encontrar dos mensajes distintos x e y tales que $H(x) = H(y)$
- *MD5*, resumen de 128 bits
- *SHA-1*, resumen de 160 bits

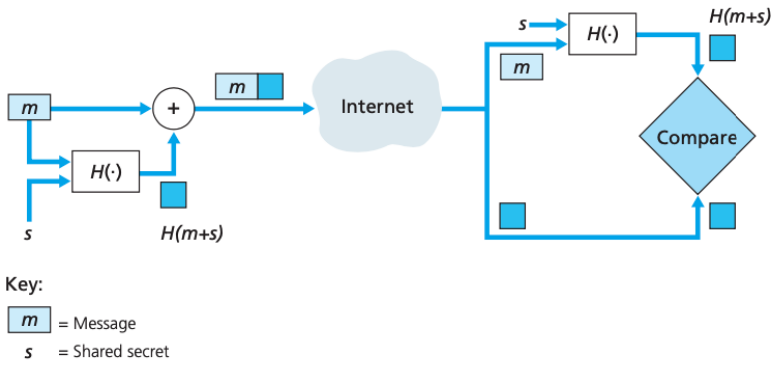


6.3.2. Código de Autenticación de Mensaje

Código de Autenticación de Mensaje

- Trudy puede hacer pasar un mensaje como de Alice
  - Alice: m, $H(m) = h$, envía "Soy Alice, (m, h)"
  - Bob: si $h = H(m)$, m OK
  - Trudy: m', $H(m') = h'$, envía "Soy Alice, (m', h')"
  - Bob: si $h' = H(m')$, m' OK (!!)
- *Código de autenticación*, s secreto compartido (MAC, *Message Authentication Code*)
  - Alice: m, s, $H(m + s) = h$, envía "Soy Alice, (m, h)"
  - Bob: si $h = H(m + s)$, m OK
- *HMAC*, puede funcionar con MD5 o SHA-1

Código de Autenticación de Mensaje



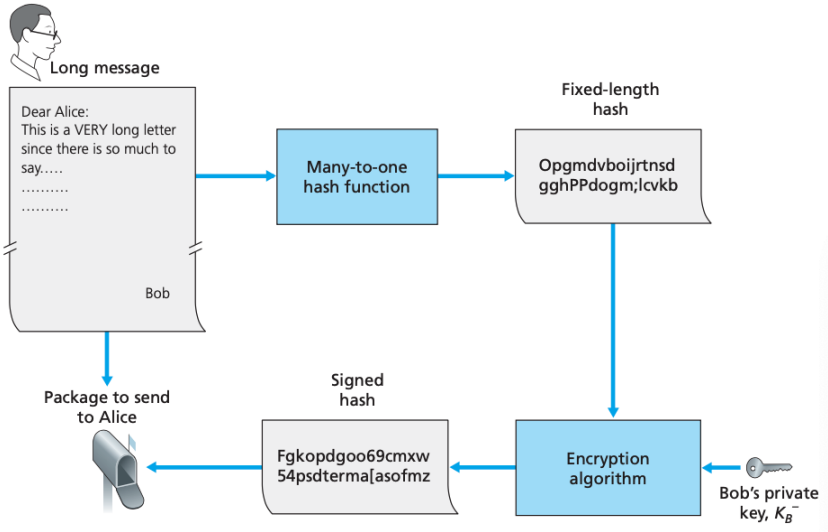
- Cómo se distribuye s?
- $K_B^+(s)$

6.3.3. Firma Digital

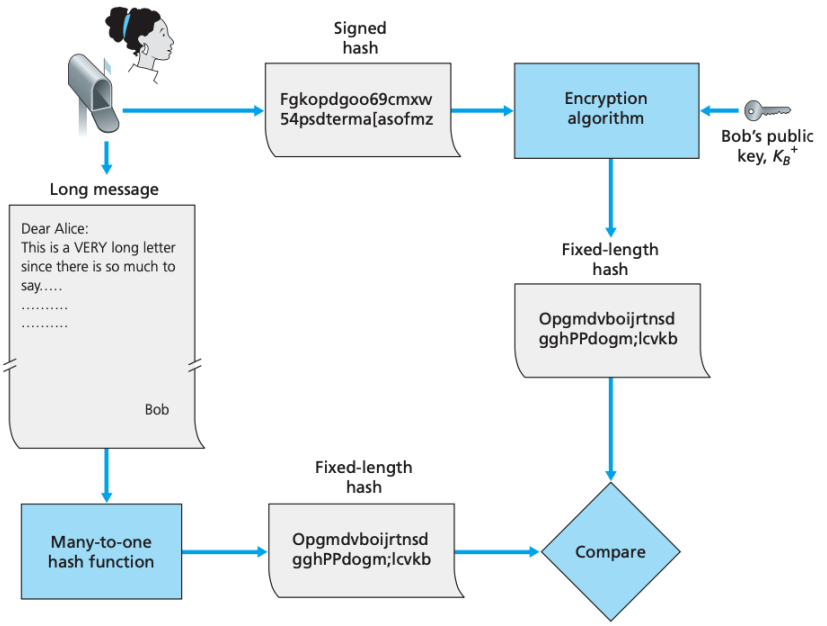
Firma Digital

- Creador/poseedor de un documento
- Conformidad con lo que se dice en el documento
- *Firma digital* técnica criptográfica
  - Verificable
  - Infalsificable
- Bob firma m como $K_B^-(m)$
  - Verificable: $\nexists\ K_X^+\ con\ X \neq B / K_X^+(K_B^-(m)) = m$
  - Infalsificable: Sólo Bob conoce K_B^-
- Integridad del mensaje
  - m', $K_b^+(K_b^-(m)) \neq m'$
- Computacionalmente costoso
- Firmar m como $K_B^-(H(m))$

Firma Digital

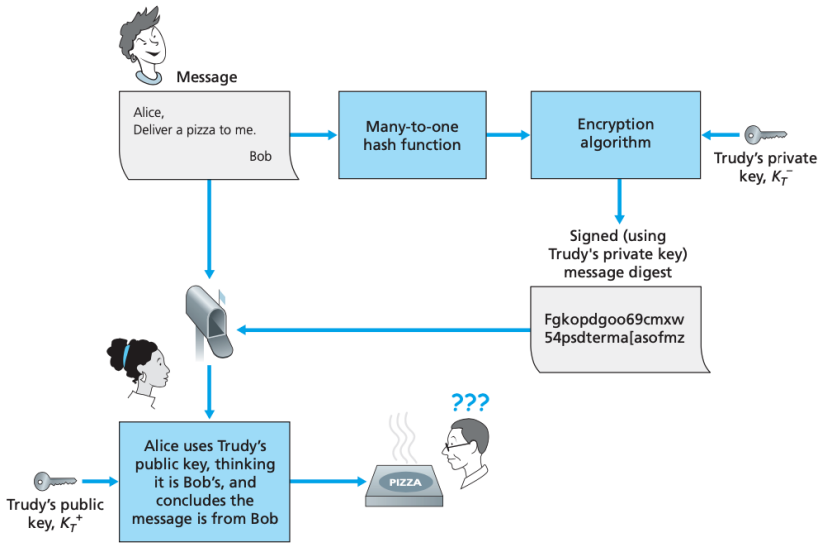


Firma Digital



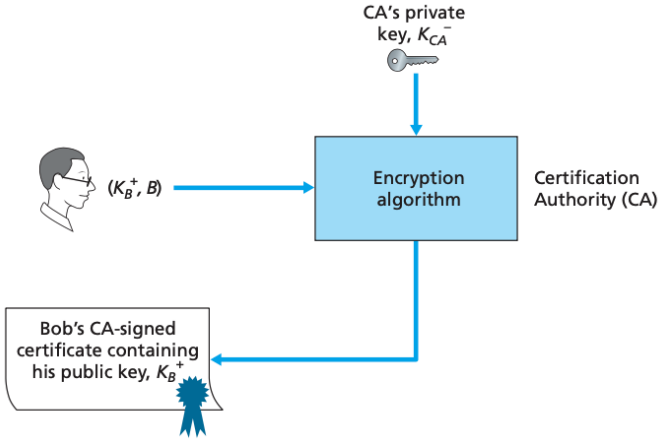
Certificación de Clave Pública

- Certificar que una clave pública pertenece a una entidad



Certificación de Clave Pública

- *Autoridad de Certificación*
  - Valida identidades
  - Emite certificados que ligan identidad a clave pública
  - Firma el certificado con su clave privada



Certificación de Clave Pública

- *[X.509]*, servicio autenticación y sintaxis para certificados

- *[RFC 1422]*, compatible X.509 añade arquitectura gestión claves

Campo	Descripción
Versión	Número de versión de la especificación X.509
Número de serie	Identificador único para este certificado
Firma	Algoritmo utilizado por CA para firmar este certificado
Nombre del emisor	Identifica la CA que emite el certificado
Periodo validez	Inicio y final del periodo de validez del certificado
Nombre del sujeto	Identidad de la entidad asociada a este certificado
Clave pública del sujeto	Clave pública de la entidad así como el algoritmo a utilizar con ella (parámetros)

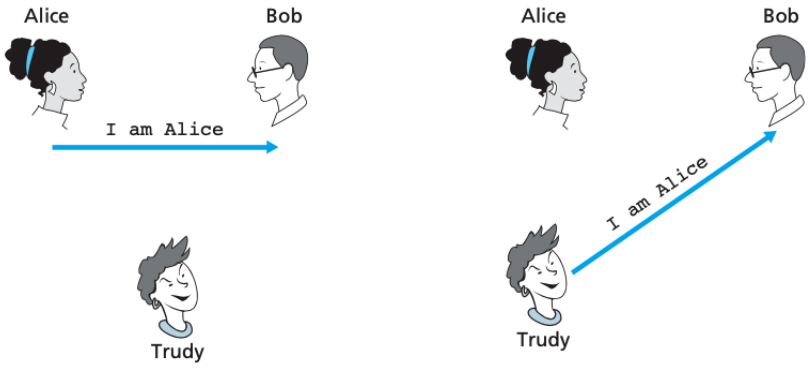
6.4. Autenticación entre Extremos

Autenticación entre Extremos

- Cómo pueden autenticarse dos entidades sólo con los datos y mensajes intercambiados en la red?
- *Protocolo autenticación* previo a otra comunicación
- Escenario: Alice necesita autenticarse frente a Bob

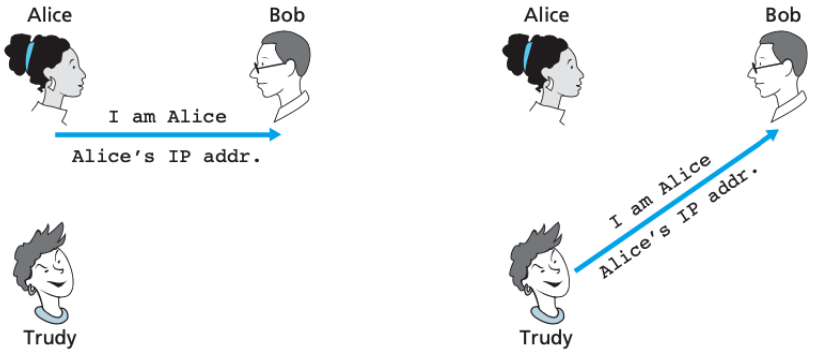
6.4.1. Protocolo Autenticación `ap1.0`

Protocolo Autenticación `ap1.0`



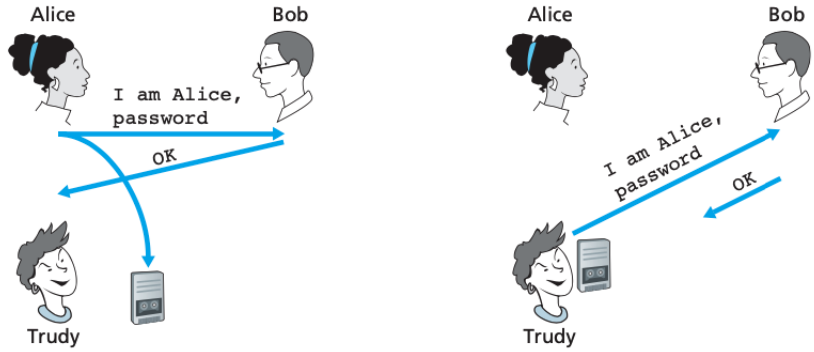
6.4.2. Protocolo Autenticación `ap2.0`

Protocolo Autenticación `ap2.0`



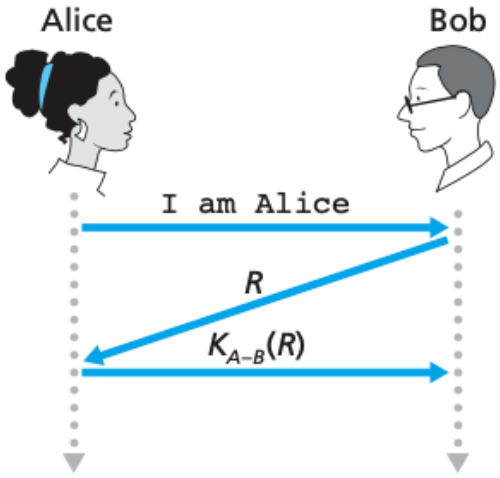
6.4.3. Protocolo Autenticación `ap3.0`

Protocolo Autenticación `ap3.0`



6.4.4. Protocolo Autenticación `ap4.0`

Protocolo Autenticación `ap4.0`



6.5. Asegurando el Correo Electrónico

Asegurando el Correo Electrónico

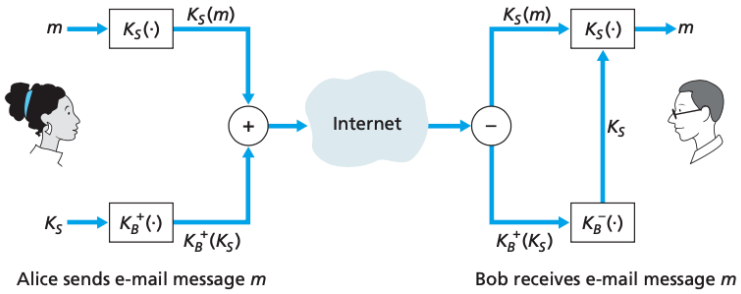
- Cómo utilizar herramientas seguridad en Internet
- Se puede aportar seguridad en las 4 capas superiores
  - Aplicación
  - SSL
  - IPsec
  - Seguridad en IEEE 802.11
- Por qué no basta con seguridad en capa de red?
  - IPSec, encripta datos en datagramas y autentica dir. IP origen
  - Más sencillo implementar nuevos servicios en capas altas

6.5.1. Correo Electrónico Seguro

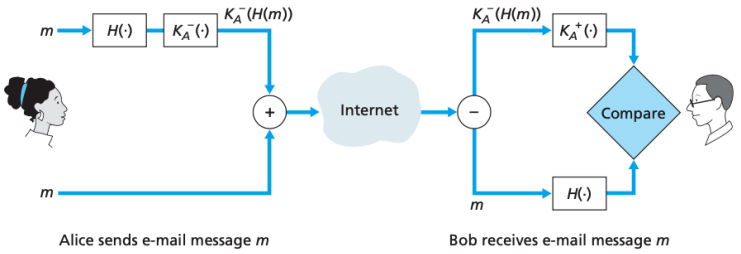
Correo Electrónico Seguro

- Confidencialidad
- Autenticación emisor
- Integridad del mensaje
- Autenticación del receptor

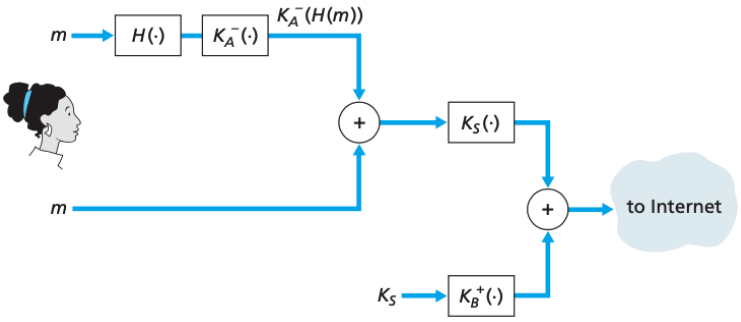
Correo Electrónico Seguro: Confidencialidad



Correo Electrónico Seguro: Integridad y Autenticación Emisor



Correo Electrónico Seguro: Confidencialidad, Integridad y Autenticación Emisor



6.5.2. PGP

PGP

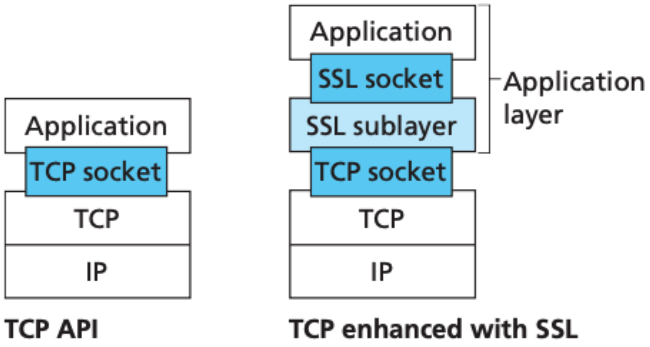
- *PGP*: Pretty Good Privacy (Zimmermann, 1991)
- Esquema de encriptación de correo electrónico (estándar *de facto*)
- MD5 o SHA para resumen mensaje

- CAST, 3DES o IDEA para encriptación clave simétrica
- RSA, para encriptación clave pública
  - Clave pública en página web o servidor público
  - Clave privada protegida por contraseña
- *Red de Confianza*

6.6. Asegurando Conexiones TCP: SSL

Asegurando Conexiones TCP: SSL

- *Secure Sockets Layer*, SSL
- Soportado por Web (`https`)
- Aporta a TCP confidencialidad, integridad de datos, autenticación de extremos (cliente y servidor)
- Librería SSL similar a sockets

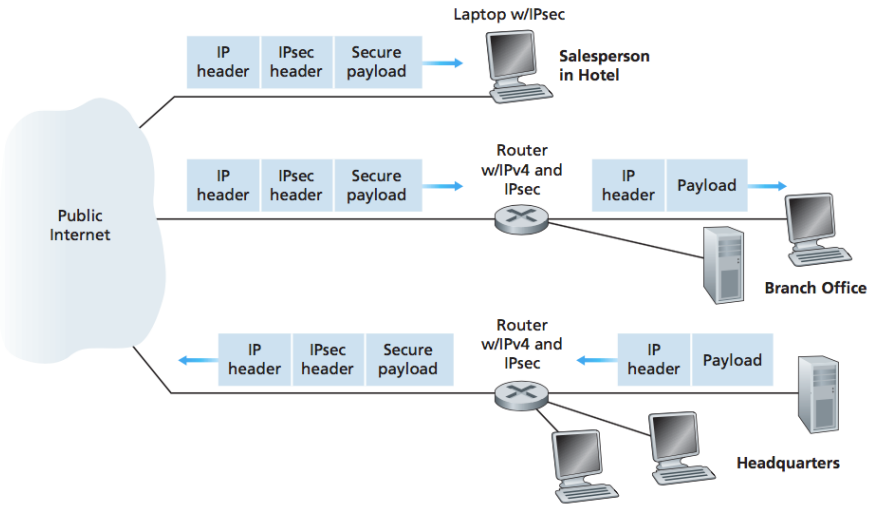


6.6.1. Seguridad en la Capa de Red: IPSec y Redes Privadas Virtuales

Seguridad en la Capa de Red: IPSec y Redes Privadas Virtuales

- *IPsec* asegura datagramas IP entre cualquiera dos entidades (hosts, routers)
- Creación de Redes Privadas Virtuales, *VPNs*, sobre la red pública Internet
- Confidencialidad
- Autenticación de origen
- Integridad de datos
- Prevención ataque de reproducción

VPN

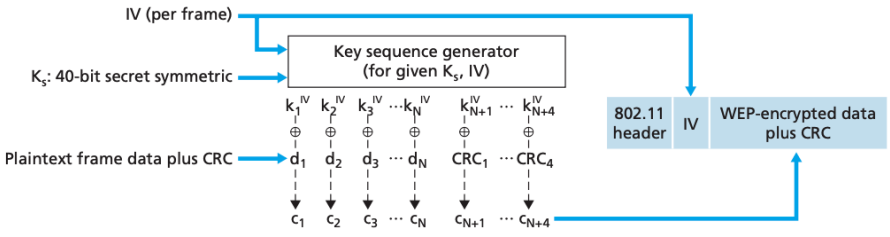


6.7. Asegurando Redes Inalámbricas

Asegurando Redes Inalámbricas

- 802.11 con graves carencias de seguridad
- *WEP*, *Wired Equivalent Privacy*, con deficiencias conocidas
  - Clave 64 bits: *IV* de 24 bits + K_S de 40 bits
  - IV transportado en plano
  - Problemas con repetición de claves
- *802.11i* más robusto

WEP



6.8. Seguridad Operativa: Cortafuegos y Detectores de Intrusión

Seguridad Operativa: Cortafuegos y Detectores de Intrusión

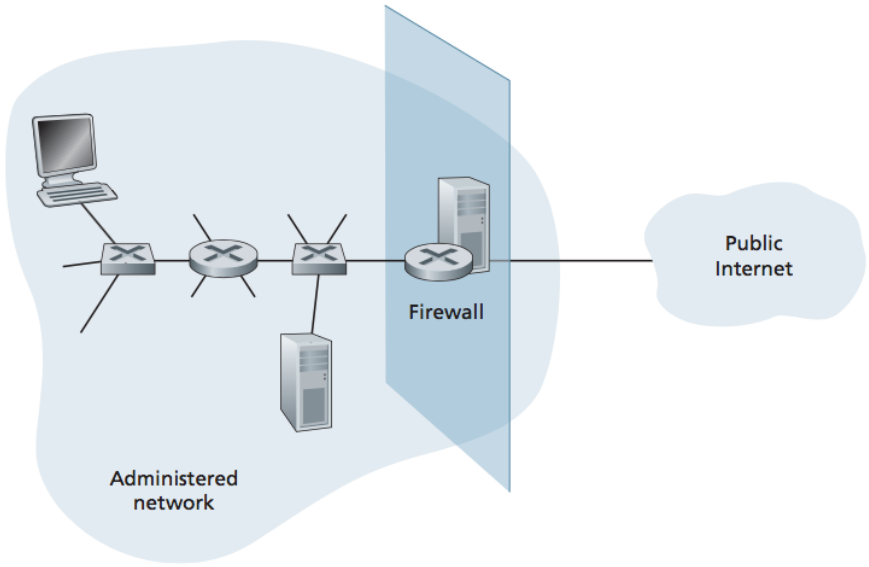
- Internet no es un sitio seguro
- Desde perspectiva administrador de la red
  - chicos buenos, internos a la organización, sin restricción
  - chicos malos, externos a la organización, accesos bajo escrutinio
- Castillos y edificios oficinas, acceso único donde aplicar controles
- Red computadoras con *cortafuegos*, *detectores de intrusión*

6.8.1. Cortafuegos

Cortafuegos

- *Cortafuegos*, (*firewall*), combinación hardware y software que aísla red/Internet
  - Tráfico entrante y saliente cruza el cortafuegos
  - Sólo tráfico autorizado pasará (*políticas de seguridad*)
  - Inmune a los ataques
- Tipos
  - Filtrado de paquetes
    - Sin estado
    - Con estado
  - Pasarela de aplicación

Cortafuegos



Filtrado de Paquetes

- Examinar cada datagrama aislado y determinar si debe pasar o es desechado
- Política de filtrado
  - Dir. IP origen y destino
  - Tipo de protocolo: TCP, UDP, ICMP, ...
  - Puertos TCP/UDP origen y destino
  - Flags
  - Tipo mensaje ICMP
  - Reglas diferentes por interfaz
  - Reglas diferentes entrantes y salientes
- *Lista de control de acceso*, *Access Control List* (ACL)
- `iptables`

Lista Control de Acceso

Acción	Dirección Origen	Dirección Destino	Protocolo	Puerto Origen	Puerto Destino	Flags
Permitir	222.22/16	fuera de 222.22/16	TCP	>1023	80	*
Permitir	fuera de 222.22/16	222.22/16	TCP	80	>1023	ACK
Permitir	222.22/16	fuera de 222.22/16	UDP	>1023	53	-
Permitir	fuera de 222.22/16	222.22/16	UDP	53	>1023	-
Denegar	*	*	*	*	*	*

Filtrado de Paquetes con Estado

- Considerar contexto de conexión para filtrar paquetes
  - Observa SYN
  - Observa FIN
  - Observa inacción
- *Tabla de conexiones*

Dir. Origen	Dir. Destino	Puerto Origen	Puerto Destino
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

Lista Control de Acceso

Acción	Dirección Origen	Dirección Destino	Prot.	Puerto Orig.	Puerto Dest.	Flags	Conexión
Permitir	222.22/16	fuera de 222.22/16	TCP	>1023	80	*	-
Permitir	fuera de 222.22/16	222.22/16	TCP	80	>1023	ACK	X
Permitir	222.22/16	fuera de 222.22/16	UDP	>1023	53	-	-
Permitir	fuera de 222.22/16	222.22/16	UDP	53	>1023	-	X
Denegar	*	*	*	*	*	*	

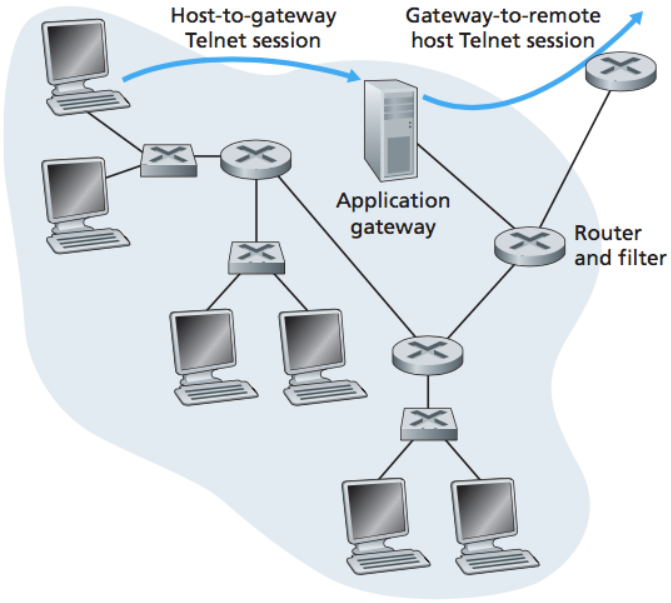
Pasarela de Aplicación

- Cómo controlar conexiones si dependen de la capa de aplicación?
- *Pasarela de aplicación* (proxy), servidor específico por el que cruzan todos los mensajes entrantes y salientes
- Dependiente de aplicación

- Autenticación usuarios
- Servidor para clientes internos
- Cliente para servidor externo

- Combinado con filtrado

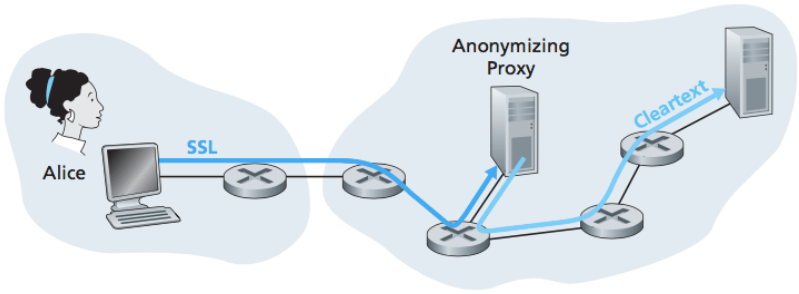
Pasarela de Aplicación



Pasarela de Aplicación

- Inconvenientes
  - Una para cada aplicación
  - Degradación de prestaciones
  - Configuración clientes

Anonimato y Privacidad

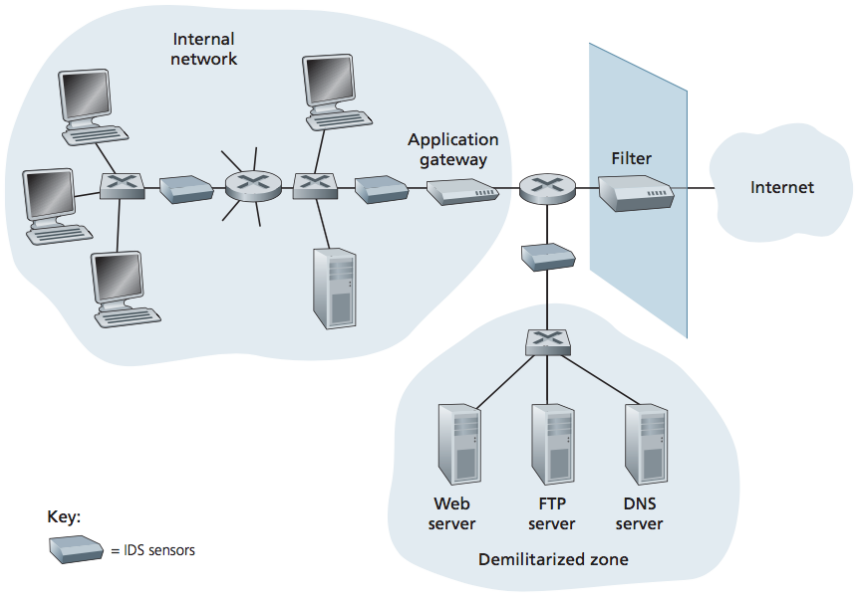


6.8.2. Detectores de Intrusión

Detectores de Intrusión

- Inspección más profunda del tráfico
- Paquetes o series de paquetes sospechosos
- *Detección de Intrusión* (IDS), avisar
- *Prevención de Intrusión* (IPS), bloquear
- Basados en firmas o anomalías
- Jerarquía de detectores
- `snort`

Arquitectura de Red Segura



6.9. Resumen

Resumen

- Mecanismos para comunicación segura: confidencialidad, autenticación de extremos e integridad del mensaje
- Criptografía como herramienta, de clave simétrica y pública
- Integridad basada en códigos de autenticación y firma digital (resumen, hash criptográfico)
- Nonces como herramienta frente a reproducción de mensajes
- Seguridad en las capas TCP/IP: correo electrónico seguro, SSL, IPsec, WEP
- Arquitectura de red segura: cortafuegos (filtrado de paquetes y pasarelas de aplicación), detectores de intrusión y DMZ

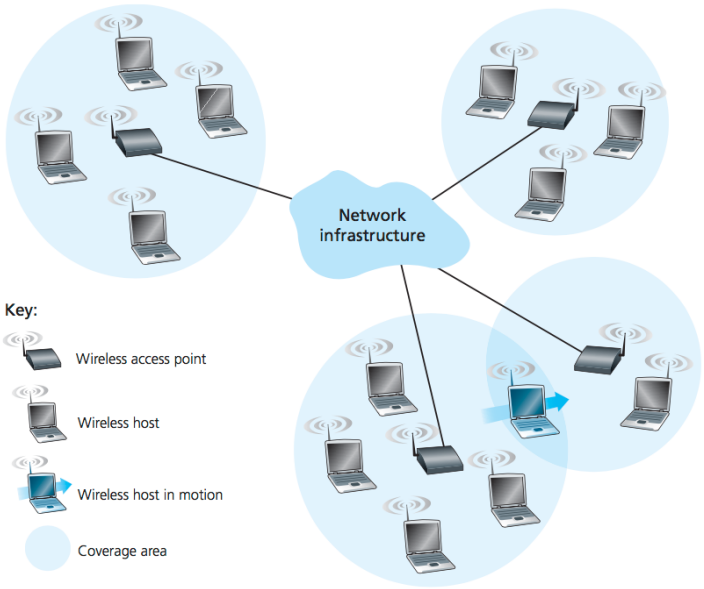
7. Redes Inalámbricas

7.1. Introducción

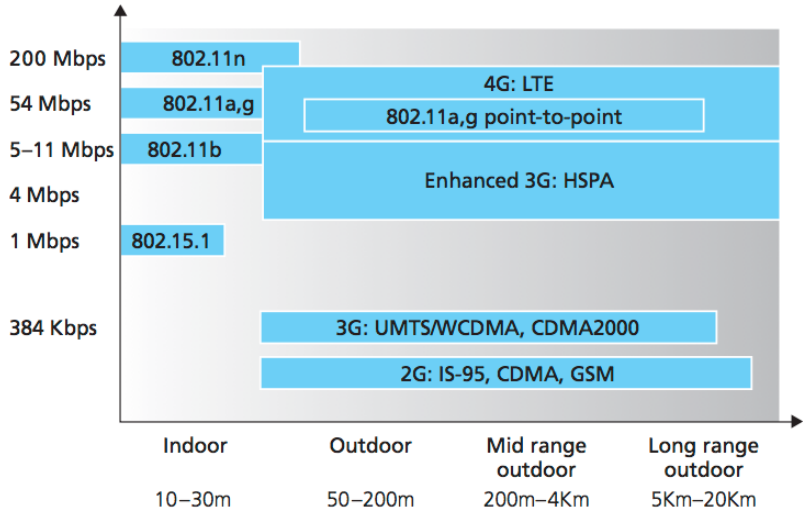
Elementos Red Inalámbrica

- *Host inalámbricos*
  - Móviles o no
- *Enlaces inalámbricos*
  - En extremo de la red
- *Estación Base*
  - Intermediario red cableada e inalámbrica
  - Sin equivalente en redes cableadas
  - Enviar y recibir paquetes a nodos inalámbricos
  - Coordinar transmisión entre múltiples nodos *asociados*
  - Redes *Infraestructura* o *ad-hoc*
  - Transferencia de nodos cambiando de estación base, *handoff*
- *Red infraestructura*

Elementos de una Red Inalámbrica



Características del Enlace en Redes Inalámbricas



Clasificación Redes Inalámbricas

- Criterios:
  - Si un paquete cruza una o más redes inalámbricas
  - Si existe o no una infraestructura de red
- Salto único, basada en infraestructura: WiFi eduroam (*)
- Salto único, sin infraestructura: WiFi ad-hoc
- Múltiples saltos, basada en infraestructura: redes de malla
- Múltiples saltos, sin infraestructura: MANETs y VANETs

7.2. Características de Enlaces y Redes Inalámbricas

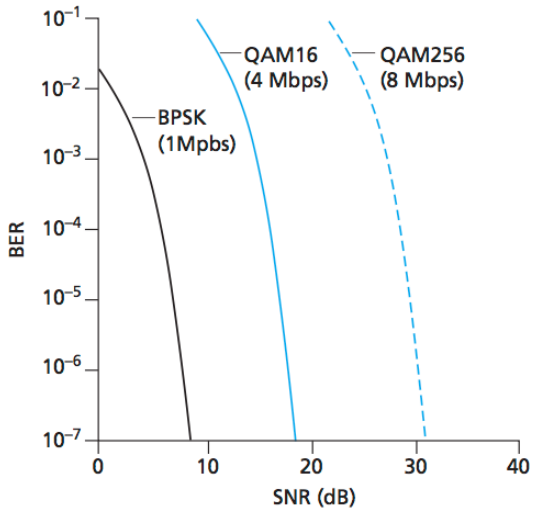
Diferencias Enlaces Inalámbricos frente a Cableados

- Disminución de la intensidad de la señal
- Interferencias con otras emisoras, ruido
- Propagación multicamino
- Errores de bit más comunes
  - *Relación Señal-Ruido*, SNR
  - *Tasa de Error*, BER

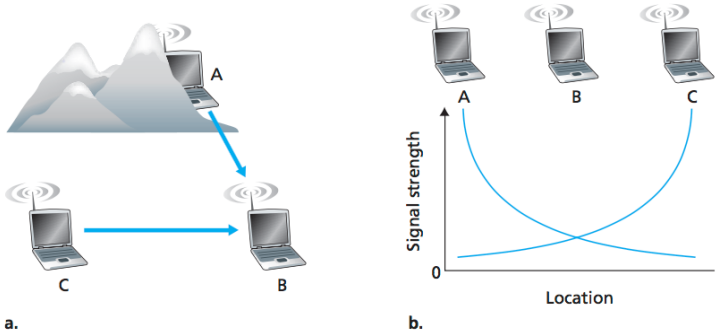
- A mayor SNR, menor BER
- A mayor tasa de transmisión, mayor BER
- Selección dinámica técnica modulación

- Medio compartido, problema nodo oculto
  - Obstáculos
  - Distancia

Tasas de Error y Transmisión y BER



Problema del Terminal Oculto

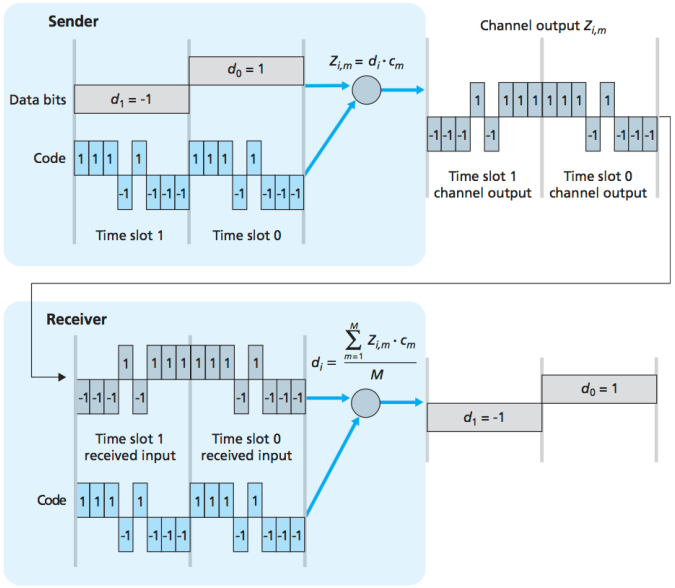


7.2.1. CDMA

Acceso Múltiple por División de Código

- *Code Division multiple access*, *CDMA*
- Protocolo de reparto de canal
- Cada bit se transmite multiplicado por una señal (código) que cambia más frecuentemente que la secuencia de bits
- Señales de bit interfiriendo son aditivas
- Asignar códigos distintos a cada transmisión

Ejemplo simple de CDMA



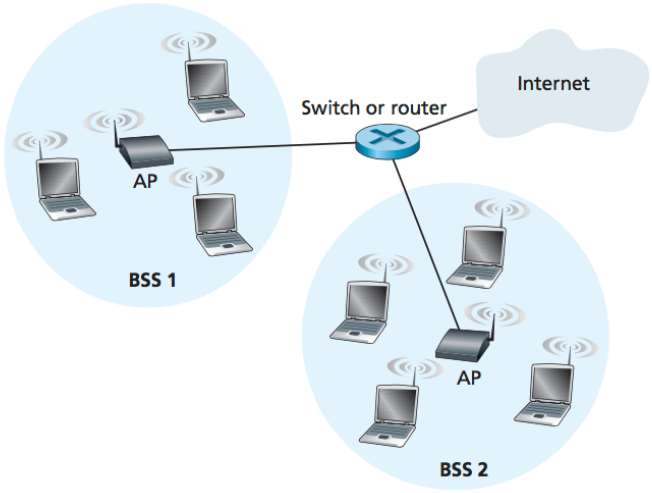
7.3. WiFi: Redes Inalámbricas 802.11

Estándares 802.11

Estándar	Tasa Datos	Banda Frecuencia
802.11a	54 Mbps	5 GHz
802.11b	11 Mbps	2,4 GHz
802.11g	54 Mbps	2.4 GHz
802.11n	600 Mbps	2,4 o 5 GHz
802.11ac	6 Gbps	5 GHz

7.3.1. Arquitectura 802.11

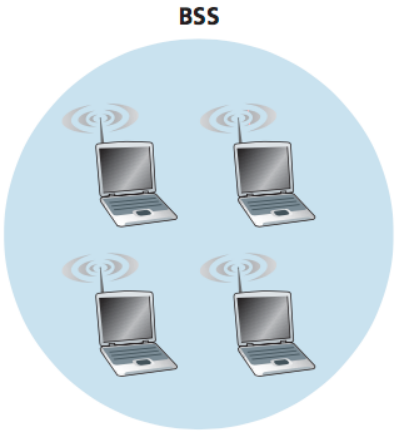
Arquitectura 802.11



Infraestructura 802.11

- *BSS*, conjunto de servicios básico
  - 1 estación base, *AP* (*Access Point*)
  - n estaciones inalámbricas
- Interfaces inalámbricas con dirección MAC (también AP)
- Redes *infraestructura* y *ad-hoc*

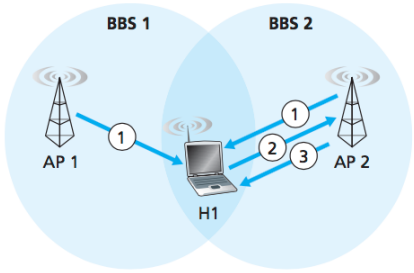
Red 802.11 tipo ad-hoc



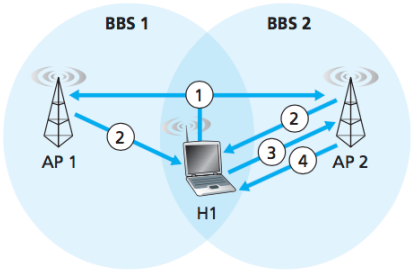
Canales y Asociación

- Asignar al AP
  - *SSID*, identificador del BSS
  - Canal
    - Hasta 11 canales solapados en 2.4 GHz
    - No solapados si separados por 4 o más canales (1, 6, 11)
- *Asociación*, crear enlace virtual inalámbrico entre estación y AP
- *Beacon*, marco periódico anunciando SSID y MAC del AP
  - Escaneo pasivo y activo
- Autenticación
  - WEP
  - WPA, WPA2
  - Servidor autenticación (RADIUS o DIAMETER)
  - 802.11i
- Una vez asociado, listo para configurar subred IP (DHCP)

Escaneo Activo y Pasivo de Puntos de Acceso



a. Passive scanning
1. Beacon frames sent from APs
2. Association Request frame sent: H1 to selected AP
3. Association Response frame sent: Selected AP to H1



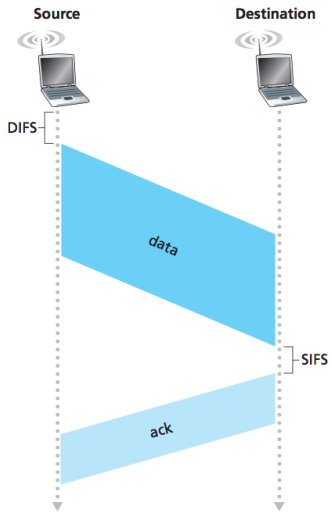
a. Active scanning
1. Probe Request frame broadcast from H1
2. Probes Response frame sent from APs
3. Association Request frame sent: H1 to selected AP
4. Association Response frame sent: Selected AP to H1

7.3.2. Protocolo MAC 802.11

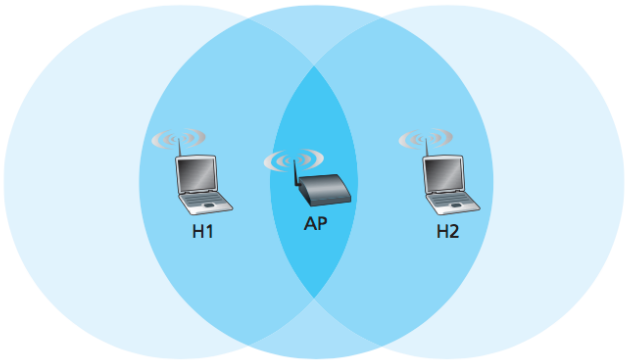
Protocolo MAC 802.11

- Acceso aleatorio, múltiples estaciones (hosts y APs)
- *CSMA/CA*, *Collision Avoidance*, imposibilidad detectar colisiones
  - No emitir y recibir al tiempo, y nodos ocultos
  - *Retardo aleatorio* si canal ocupado (*backoff*)
- Reconocimientos y retransmisiones a nivel enlace
  - Errores bit
  - Reenvío si no ACK
- Una vez que una estación comienza a transmitir un paquete, éste es transmitido completamente

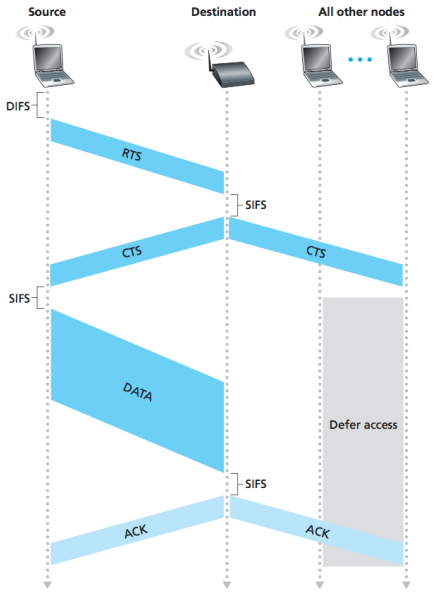
Reconocimientos en Capa de Enlace



Ejemplo de Terminal Oculto

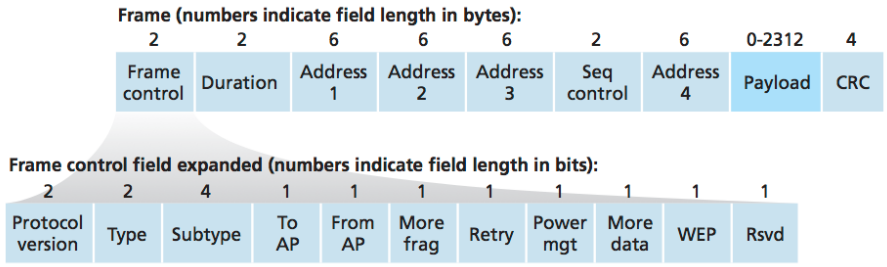


Evitación de Colisiones con Marcos RTS y CTS



7.3.3. Marco 802.11

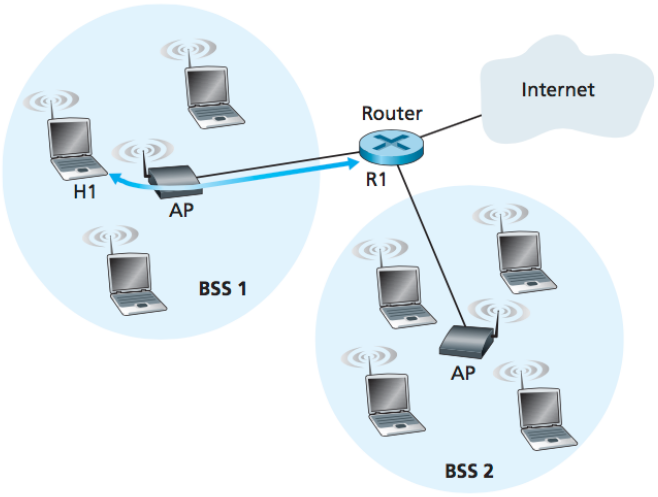
El Marco 802.11



Campos de Direcciones en 802.11

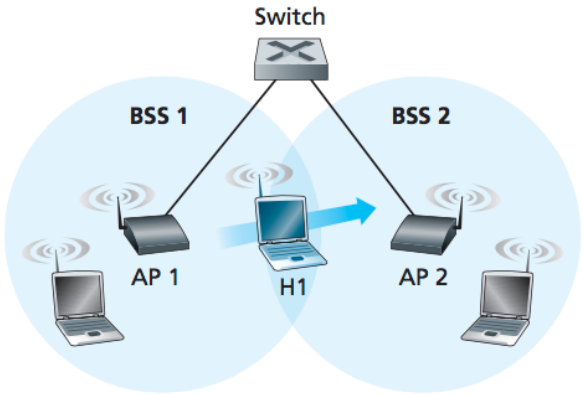
- Dirección 1: MAC estación que recibe el marco (AP o host)
- Dirección 2: MAC estación transmite marco (host o AP)
- Dirección 3: MAC interfaz del router
- Dirección 4: Modo ad-hoc

Campos de Direcciones en 802.11 en Modo Infraestructura



7.3.4. Movilidad Dentro de una Misma Subred IP

Movilidad en Subred



7.3.5. Características Avanzadas en 802.11

Características Avanzadas en 802.11

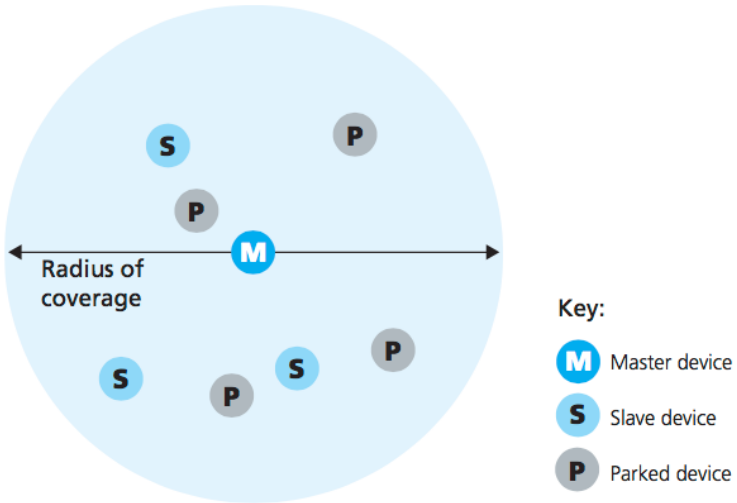
- No especificadas completamente en el estándar
- Adaptación de técnica de modulación según características canal
- Gestión energía alternando entre estado activo o dormido

7.3.6. Redes de Área Personal: Bluetooth y Zigbee

Redes de Área Personal: Bluetooth y Zigbee

- *Bluetooth* IEEE 802.15.1
  - Reemplazo de cables, red inalámbrica ad-hoc de corto alcance y bajo consumo
  - *Wireless Personal Area Networks*, *WPANs*
  - 2,4 GHz, TDM con slots de 625 μs y 79 canales
  - 4 Mbps, *Frequency-hopping spread spectrum*, (FHSS), Hedy Lamarr
  - $\leq$8 nodos: 1 maestro, resto esclavos (y <255 aparcados)
- *Zigbee* IEEE 802.15.4
  - Red baja energía, tasa de datos y carga

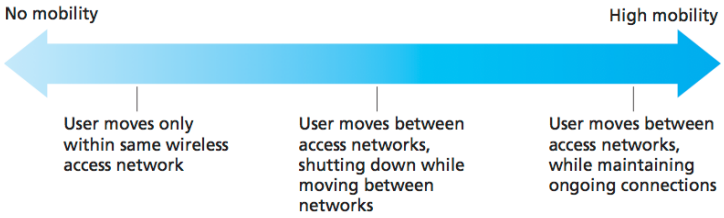
Piconet Bluetooth



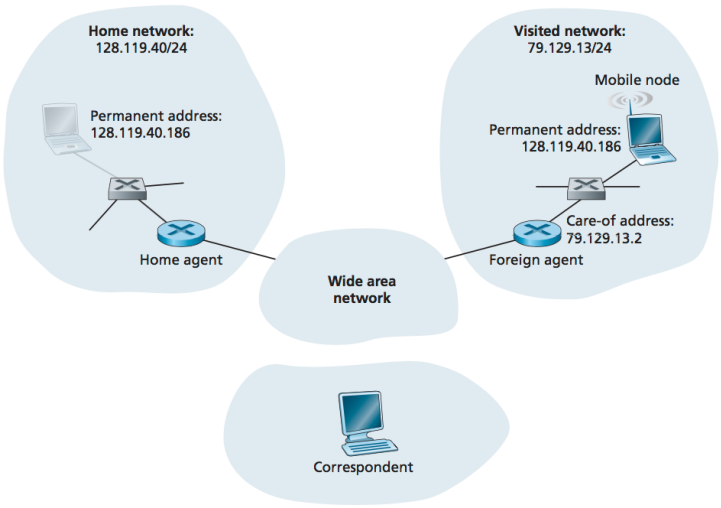
7.4. Gestión de la Movilidad: Principios

Movilidad

- Desde el punto de vista de la red, cómo de móvil es un nodo?
- Cómo de importante es para el nodo mantener la misma dirección?



Arquitectura de Red para Movilidad



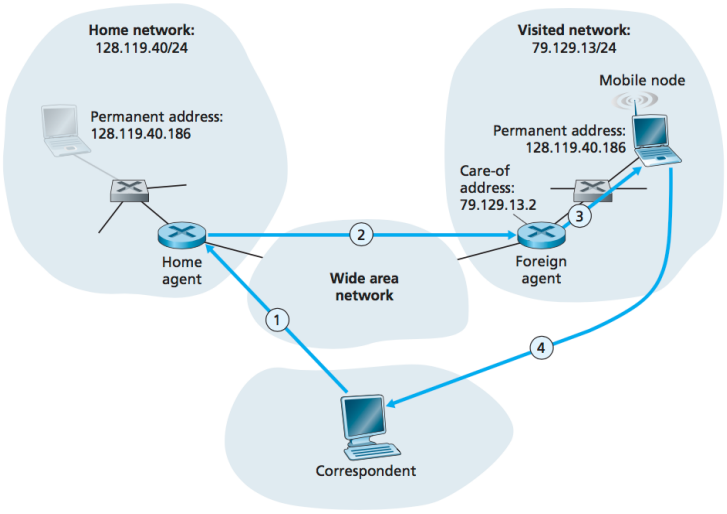
7.4.1. Direccionamiento

Direccionamiento

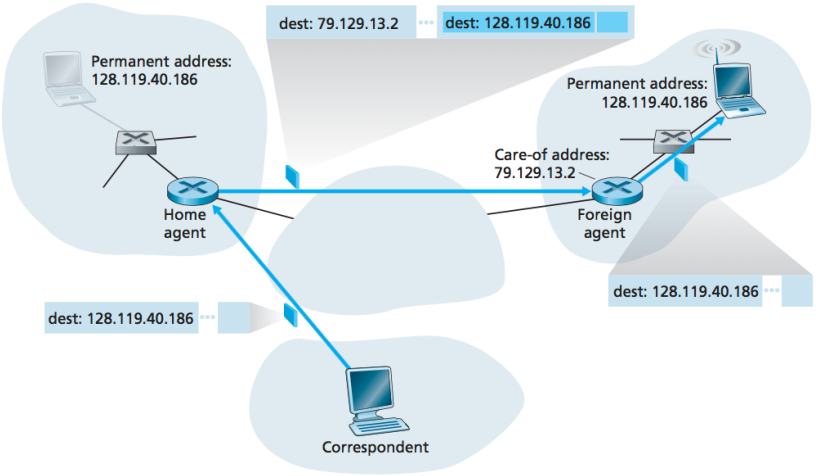
- Mantener dirección IP mientras se mueve entre redes
- Reenvío del tráfico a la red foránea
- Publicar nueva ruta a ese destino
  - Escalabilidad
  - Routers manteniendo tablas con millones de nodos en movimiento
- Protocolo entre red doméstica y foránea
  - Traspasar problema al extremo de la red
  - Dirección foránea, *care-of address* (COA), utilizada para reenviar tráfico entre agente doméstico y foráneo

7.4.2. Encaminamiento hacia Nodo Móvil

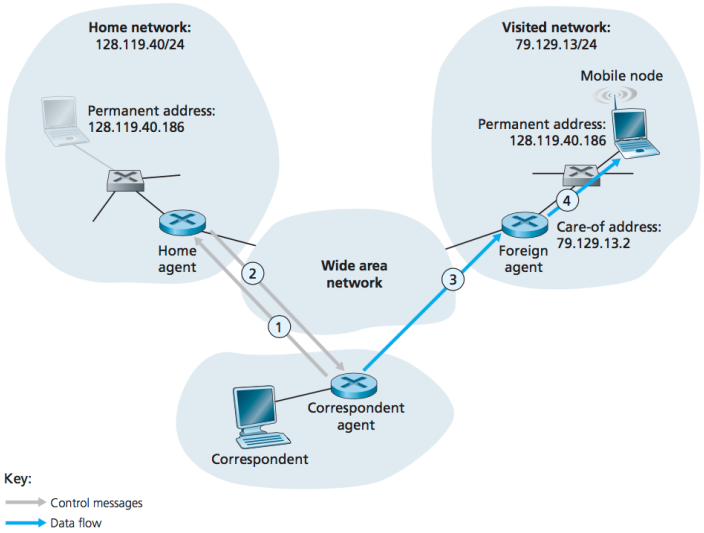
Encaminamiento Indirecto



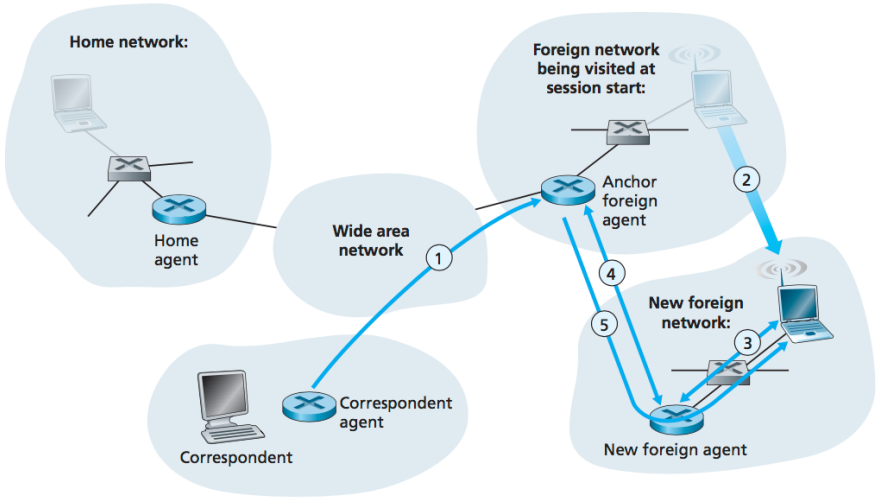
Encapsulación y Desencapsulación



Encaminamiento Directo



Transferencia entre Redes con Encaminamiento Directo

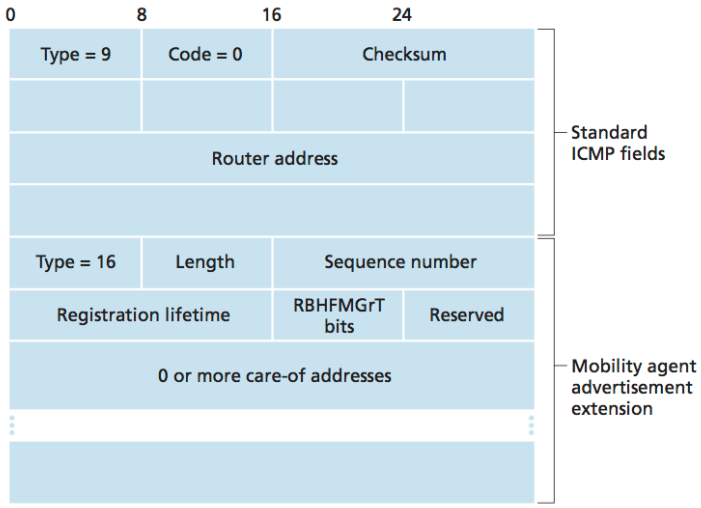


7.5. IP móvil

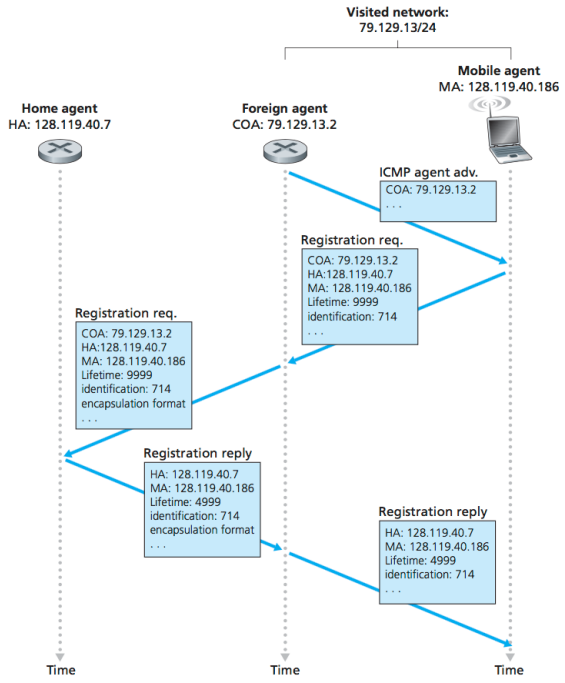
IP móvil

- *Mobile IP* [RFC 5944] para IPv4
  - Descubrimiento de agente
  - Registro con el agente doméstico
  - Encaminamiento indirecto

Mensaje ICMP de Descubrimiento de Ruta con Extensión de Anuncio de Agente de Movilidad



Anuncio de Agente y Registro IP Móvil



7.6. Redes Inalámbricas y Movilidad: Impacto en Protocolos de Capas Superiores

Redes Inalámbricas y Movilidad: Impacto en Protocolos de Capas Superiores

- Inalámbricas difieren de cableadas
  - Enlace
  - Red
  - Transporte?
- *Best-effort* sigue vigente, TCP/UDP funcionan sobre inalámbricas
- Diferencia de prestaciones
  - Perdidas por congestión red o transferencia entre APs o corrupción
  - Reacción TCP: reducción de tasa de envío
- Aplicación afectadas por el relativo bajo ancho de banda, *aplicaciones sensibles al contexto y localización*

7.7. Resumen

Resumen

- Redes inalámbricas han cambiado profundamente las redes de computadoras: conexión ubicua y continua
- Diferencias en el tipo de enlace y movilidad
- Redes inalámbricas, (WLANs) IEEE 802.11 Wifi
- Redes de área personal (PANs) IEEE 802.15 Bluetooth y Zigbee
- Movilidad, manteniendo la subred o no
- Mobile IP, conjunto de protocolos que permiten la movilidad manteniendo la IP
- TCP confundido por las pérdidas de paquetes no debidas a congestión

8. Referencias

Referencias

[KuroseRoss, 2012] James F. Kurose, Keith W. Ross *Computer Networking: a Top-Down Approach*. Pearson, 6/ed, 2012.

Hecho con...
LaTeX y BEAMER