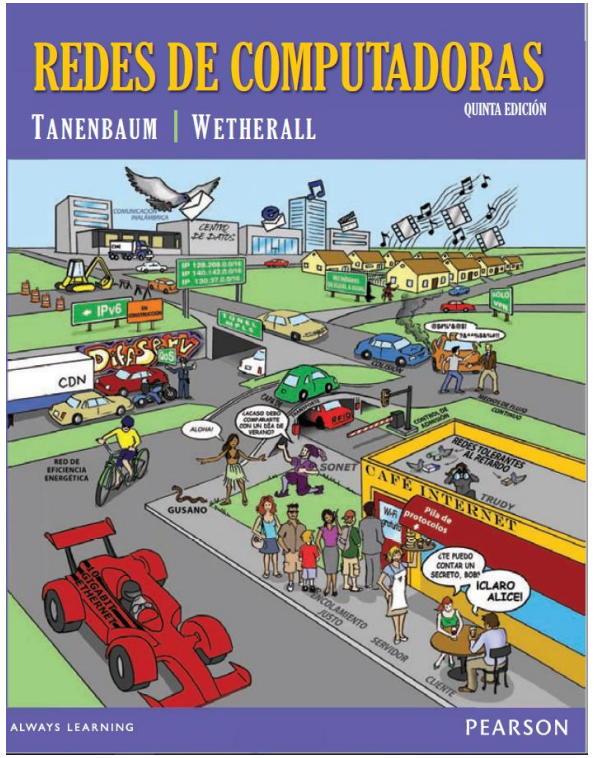
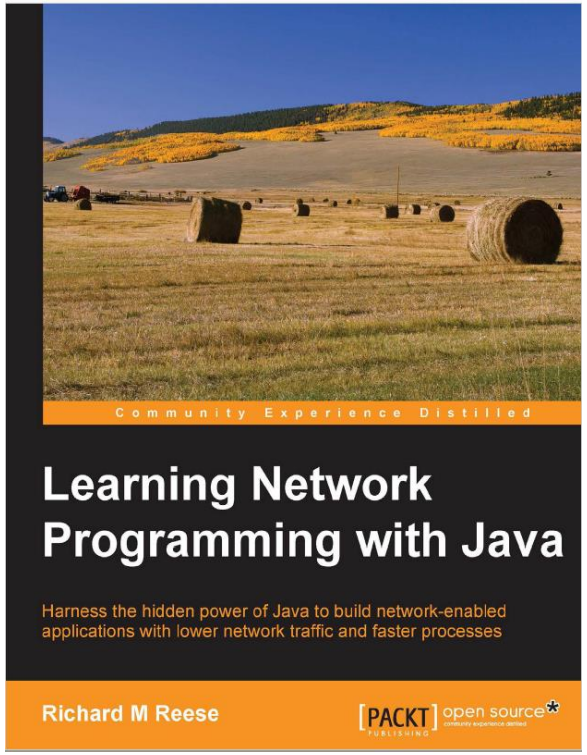
REDES DE COMPUTADORES Y LABORATORIO

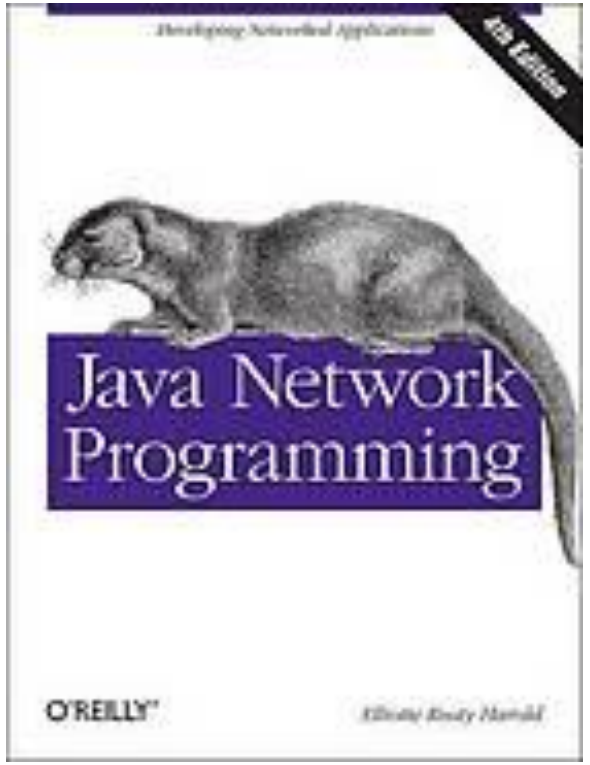
Christian Camilo Urcuqui López, MSc

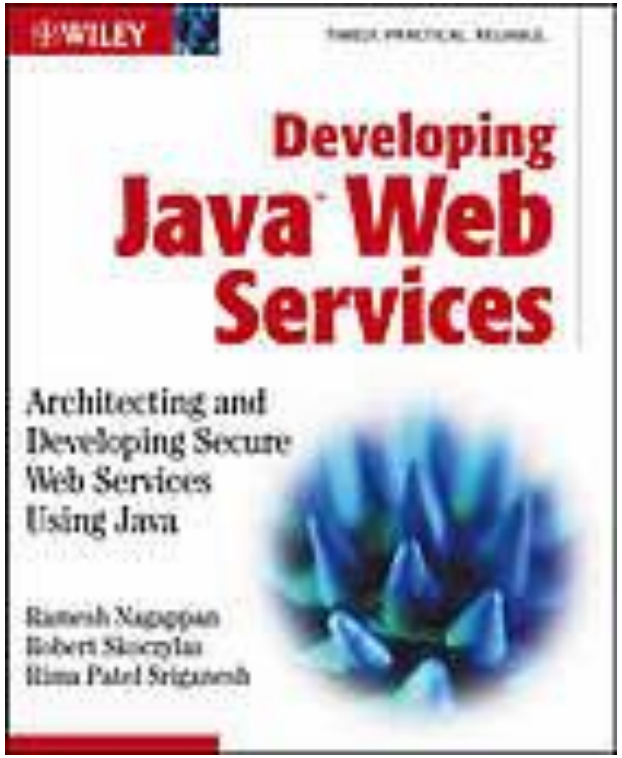
UNIVERSIDAD ICESI | A OTRO NIVEL

BIBLIOGRAFÍA









COMPETENCIAS

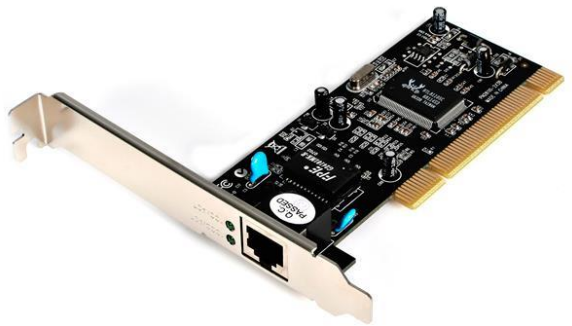
- Describa la capa de red
- Describa el protocolo IPv 4 e IP
- Aplique la Api de Java para el uso de las interfaces de red y la aplicación de multicastsocket

PROGRAMACIÓN EN RED

- Existen dos clases que heredan de InetAddress que permiten trabajar con los dos tipos de IP
  - Inet4Address para IPv4
  - Inet6Address para Ipv6
- Como hemos mencionado, Java reside en la capa de aplicación y nosotros no necesitamos preocuparnos por conocer el tipo de IP ya que hay mecanismos en las capas inferiores encargadas de la comunicación interred (recuerde tunelización).
- Recuerde que IPv4 cuenta con 32 bits y IPv6 con 128, es por ello que el método *isIPV4CompatibleAddress()* retorna una verdadero si y solo si existe una IPv4 interna es una IPv6, es decir, la dirección tiene la siguiente forma *0:0:0:0:0:0:0:xxxx*, donde los últimos 4 bytes no son cero.

THE NETWORKINTERFACE CLASS

- Un equipo puede tener varias interfaces de red, es decir, tarjetas de red que pueden ser físicas o virtuales, también son conocidas como Network Interface Controller (NIC).
- Cada NIC física tiene un número de identificación único de 48 bits en hexadecimal asignado por los fabricantes, conocido como **dirección MAC** (Media Access Control) también conocido como dirección física.
- Las direcciones MAC son administradas por el IEEE.
- Los tipos de tarjetas difieren en su tipo de cableado o arquitectura:
  - Token Ring
  - ARCNET
  - Ethernet
  - WI-FI



THE NETWORKINTERFACE CLASS

THE NETWORKINTERFACE CLASS

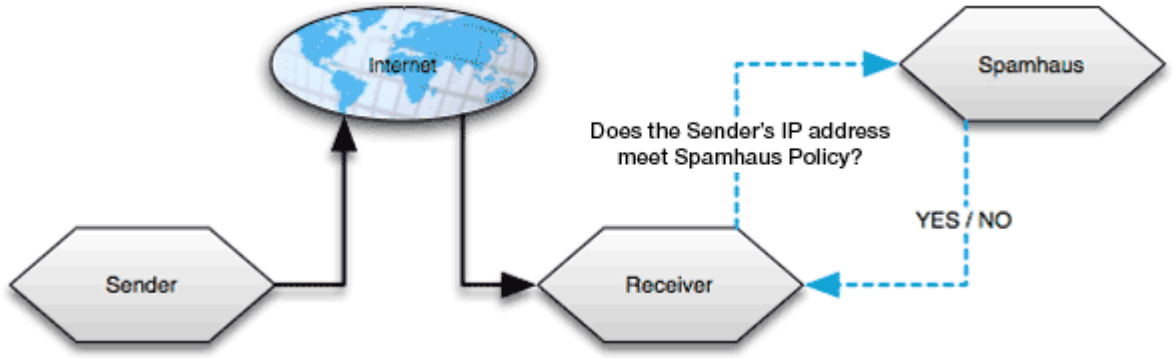
- La clase NetworkInterface representa una dirección IP local.
- La clase NetworkInterface provee los métodos para enumerar todos las direcciones locales independiente de la interfaz objetos InetAddress.
- Factory Methods
  - **public static NetworkInterface getByName(String name) throws SocketException**
    - Retorna el objeto NetworkInterface asociado a un nombre particular. Usualmente, en los sistemas UNIX los nombres de las interfaces Ethernet son de la forma eth0, eth1 y así sucesivamente.
  - **Public static NetworkInterface getByInetAddress(InetAddress address) throws SocketException**
    - Retorna un NetworkInterface asociado a la interfaz de red especificada a la dirección de red.
  - **public static Enumeration getNetworkInterfaces() throws SocketException**
    - Tenemos un objeto Enumeration que lista todas las interfaces de red del equipo local.

THE NETWORKINTERFACE CLASS

- Getter Methods
  - **public Enumeration getInetAddress()**
    - Una sola interfaz puede tener más de una dirección IP asociada. Esta situación no es muy común estos días, pero sucede en algunas ocasiones.
  - **public String getName()**
    - El método retorna el nombre asociado al objeto NetworkInterface, por ejemplo eth0 o lo
  - **public String getDisplayName()**
    - El método retorna un nombre más "amigable" para al objeto NetworInterface (por ejemplo, "Ethernet Card 0"). En algunos sistemas operativos retornara el mismo valor del método getName()

EJEMPLOS

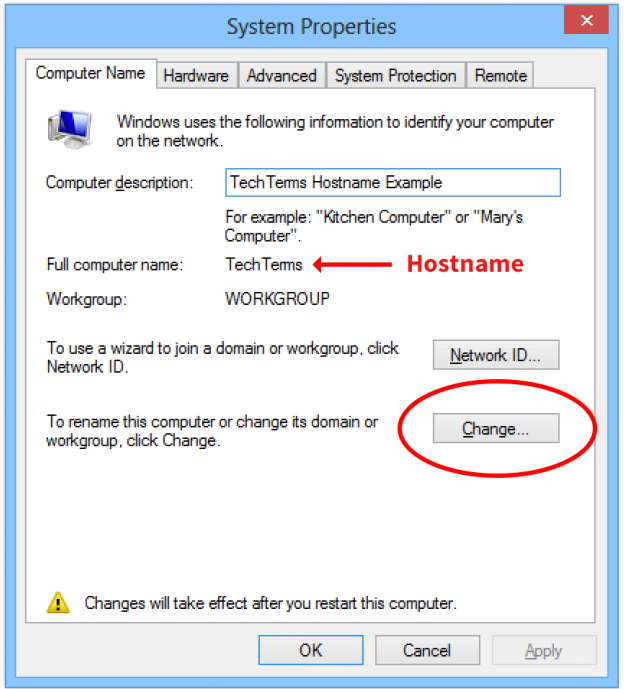
- **SpamCheck.**
  - La idea es identificar las IP que hacen referencias a spammer
  - https://www.spamhaus.org/whitepapers/dnsbl_function/

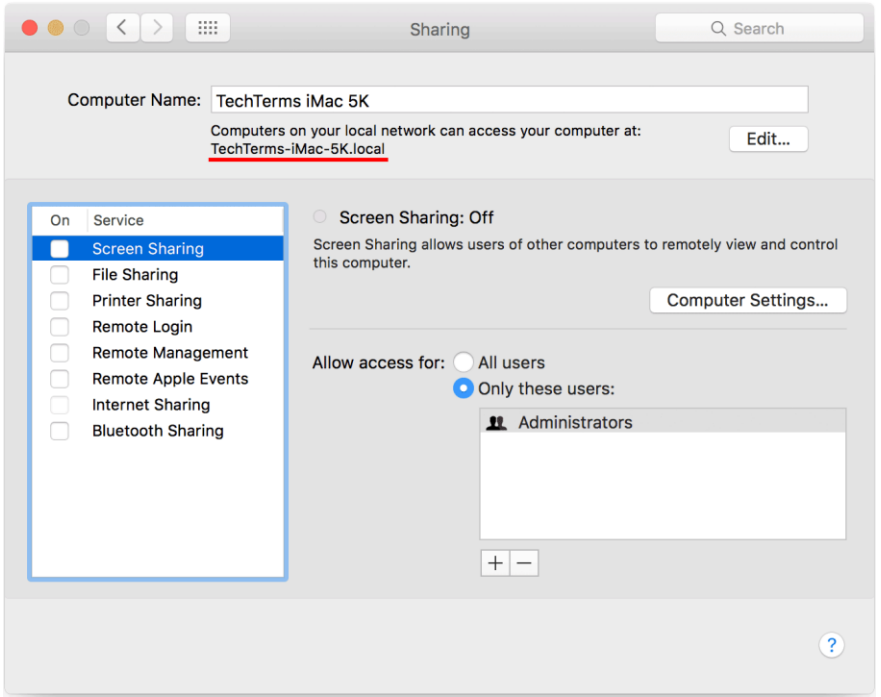


  - https://www.spamhaus.org/lookup/

EJEMPLOS

- **Procesamiento de archivos de log de un servidor web.**
  - Es el encargado de monitorear y registrar los accesos a la aplicación web, por defecto, se almacenan las direcciones IP de quien pide servicios al servidor.
  - El *hostname* es el nombre de un equipo, este parámetro es único y relativamente informal que se le da a un dispositivo conectado a una red de comunicaciones.





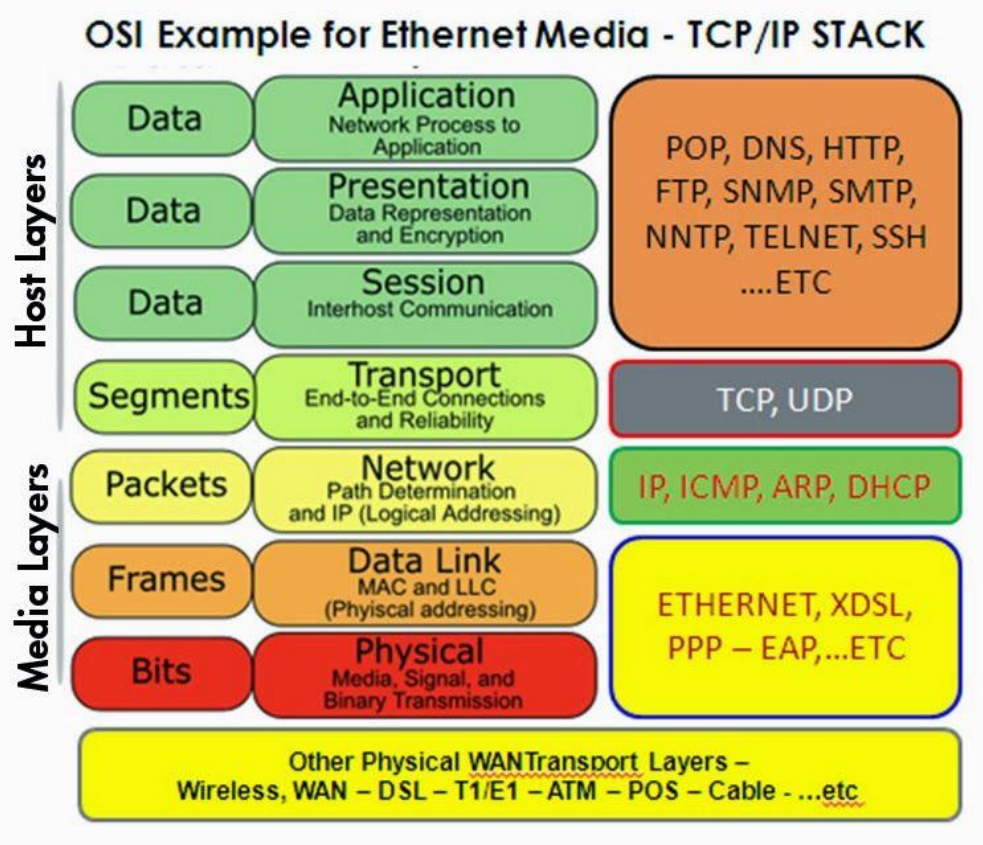
EJEMPLOS

- **Procesamiento de archivos de log de un servidor web.**
  - Usualmente los servidores web tienen un archivo log.

```
205.160.186.76 unknown - [17/Jun/2013:22:53:58 -0500]
                                    "GET /bgs/greenbg.gif HTTP 1.0" 200 50
```

- El libro propone dos formas de abordar la lectura de direcciones IP de un archivo log para traducirlos a hostname a través del DNS. Una con un conjunto de hilos que por cada entrada con la finalidad de ser más rápido las consultas de los nombres a los servidores DNS.

LA CAPA DE RED



- Encargada del enrutamiento y la interconexión entre redes.
- ¿Cómo la capa de red permite un servicio orientado a conexión y otro sin conexión?

SERVICIO SIN CONEXIÓN

- Los paquetes se trasmiten por separado en la red y se enrutan de manera independiente.
- Los paquetes son conocidos como **datagramas** y la red se conoce como **red de datagramas**.

SERVICIO SIN CONEXIÓN

- Cada enrutador tiene una tabla interna que le indica a dónde enviar paquetes para cada uno de los posibles destinos.

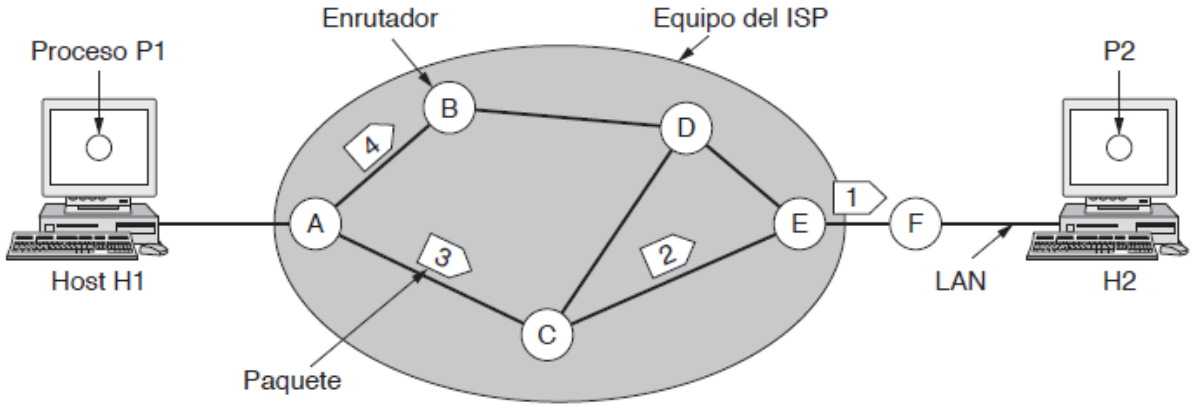


Tabla de A (al principio)

Destino	Línea
A	–
B	B
C	C
D	B
E	C
F	C

Tabla de A (más tarde)

A	–
B	B
C	C
D	B
E	B
F	B

Tabla de C

A	A
B	A
C	–
D	E
E	E
F	E

Tabla de E

A	C
B	D
C	C
D	D
E	–
F	F

El algoritmo de enrutamiento es el encargado de definir la ruta

SERVICIO SIN CONEXIÓN

- IP (Protocol Internet), que constituye la base de Internet, es el ejemplo dominante de un servicio de red sin conexión. Cada paquete transporta una dirección IP de destino que los enrutadores usan para reenviar cada paquete por separado.
- Las direcciones son de 32 bits en los paquetes IPv4.
- Las direcciones son de 64 bits en los paquetes IPv6.

8 bits = 1 byte

SERVICIO SIN CONEXIÓN

- IP (Protocol Internet), que constituye la base de Internet, es el ejemplo dominante de un servicio de red sin conexión. Cada paquete transporte una dirección IP de destino que los enrutadores usan para reenviar cada paquete por separado.
- Las direcciones son de 32 bits en los paquetes IPv4.
- Las direcciones son de 64 bits en los paquetes IPv6.

1 0 0 0 0 0 1 1 | 0 1 1 0 1 1 0 0 | 0 1 1 1 1 0 1 0 | 1 1 0 0 1 1 0 0

$2^7 + 2^1 + 1 = 131.$ 108. $2^6 + 2^5 + 2^4 + 2^3 + 2 = 122.$ 204

131. 108. 122. 204
Notación decimal punteada

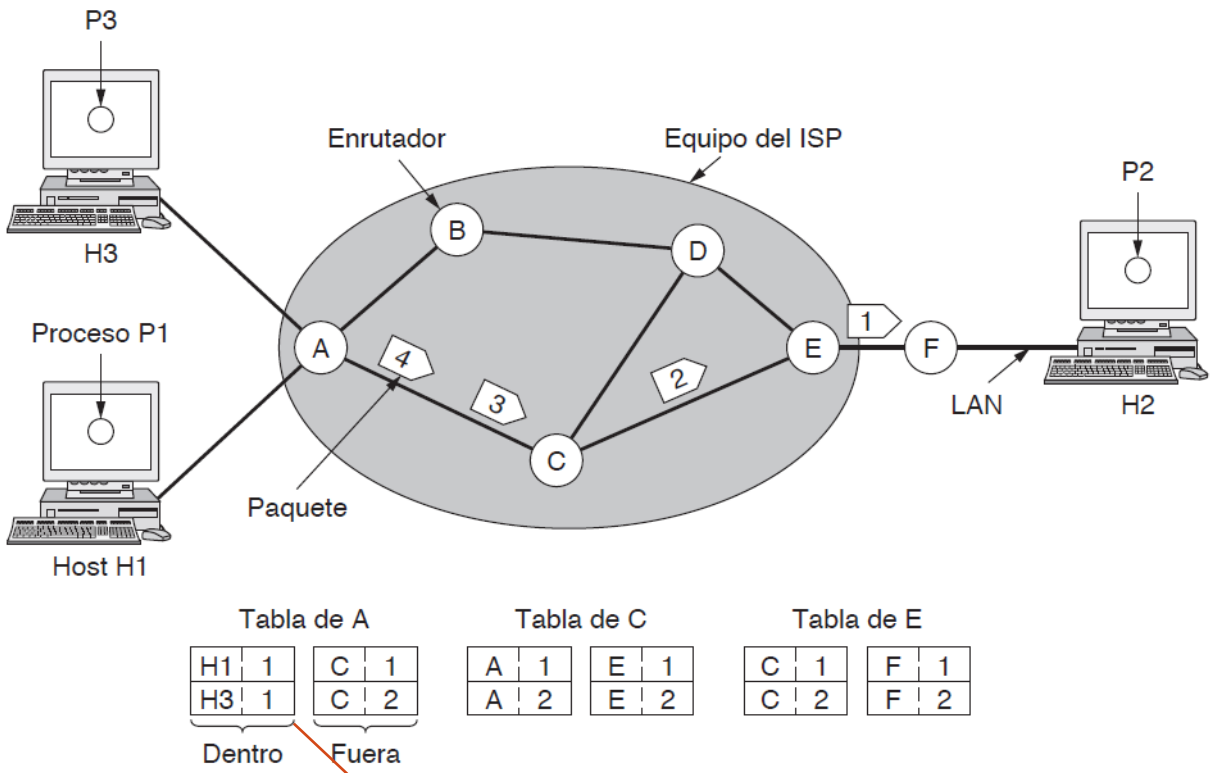
SERVICIO ORIENTADO A LA CONEXIÓN

- Hay que establecer una ruta del enrutador de origen al enrutador de destino antes de poder enviar cualquier paquete de datos. Esta conexión se conoce **VC (circuito virtual)**. El objetivo de un VC es evitar la necesidad de elegir una nueva ruta para cada paquete enviado.
- La red se denomina **red de circuitos virtuales**.

SERVICIO ORIENTADO A LA CONEXIÓN

- Cuando se establece una conexión, **se elige una ruta** de la máquina de origen a la máquina de destino como parte de la configuración de conexión y se almacena en tablas dentro de los enrutadores.
- En la ruta se dirige todo el tráfico de red a través de la conexión, cuando se termina la conexión se termina el circuito virtual.
- Cada paquete lleva un identificador que indica a cuál circuito virtual pertenece.

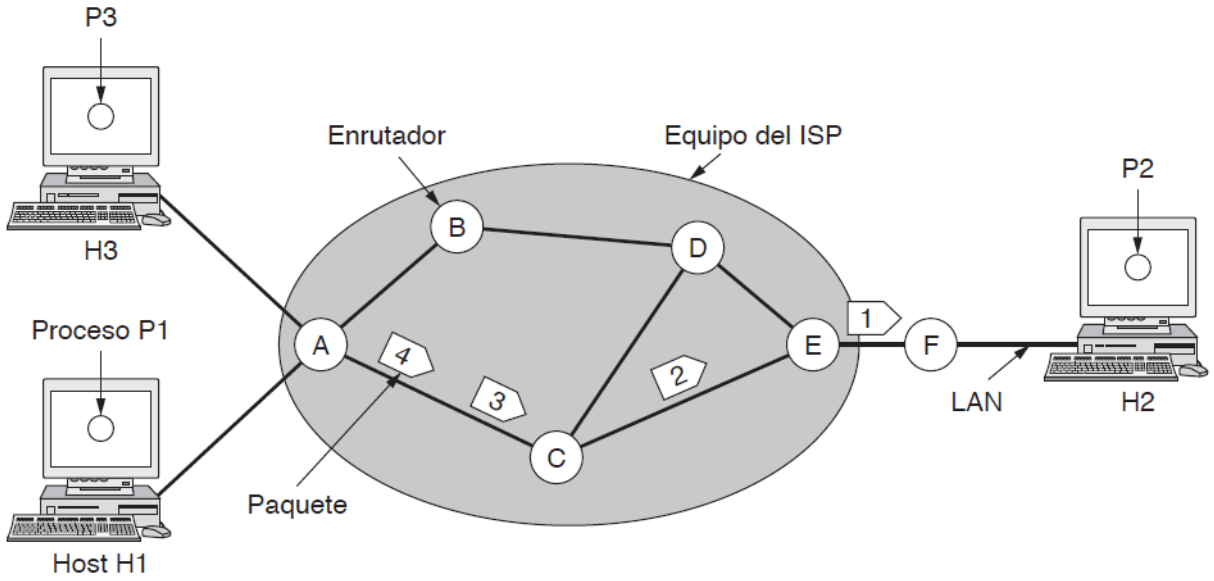
SERVICIO ORIENTADO A LA CONEXIÓN



- En este ejemplo hay dos conexiones realizadas por H1 y H3.
- Existen dos identificadores para las conexiones.
- Los enrutadores tienen que tener la capacidad de reemplazar los identificadores de conexión en los paquetes de salida.
- Para este ejemplo, el enrutador A genera un nuevo identificador para el tráfico de salida en la segunda conexión.
- Conmutación mediante etiquetas, **MLPS** (MultiProtocol Label Swithching)

Ambos tienen 1 ya que es la primera conexión e indica la creación del circuito virtual

SERVICIO ORIENTADO A LA CONEXIÓN



- **MLPS** es un servicio de red orientado a conexión, los paquetes IP se envuelven en un encabezado MPLS que tiene un identificador de conexión.

Ambos tienen 1 ya que es la primera conexión e indica la creación del circuito virtual

CIRCUITOS VIRTUALES Y REDES DE DATAGRAMAS

Asunto	Red de datagramas	Red de circuitos virtuales
Configuración del circuito.	No necesaria.	Requerida.
Direccionamiento.	Cada paquete contiene la dirección de origen y de destino completas.	Cada paquete contiene un número de CV corto.
Información de estado.	Los enrutadores no contienen información de estado sobre las conexiones.	Cada CV requiere espacio de tabla del enrutador por cada conexión.
Enrutamiento.	Cada paquete se enruta de manera independiente.	La ruta se elije cuando se establece el CV; todos los paquetes siguen esa ruta.
Efecto de fallas del enrutador.	Ninguno, excepto para paquetes perdidos durante una caída.	Terminan todos los CVs que pasaron por el enrutador defectuoso.
Calidad del servicio.	Difícil.	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV.
Control de congestión.	Difícil.	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV.

CIRCUITOS VIRTUALES Y REDES DE DATAGRAMAS

- En una primera fase el tiempo de configuración y el tiempo de análisis de la dirección es más alto en un **circuito virtual**, pero, una vez mapeada la tabla del enrutador ya es más fácil hacer el enrutamiento del paquete. En una **red de datagramas** no requiere de configuración pero sí utiliza un proceso más complicado para la localización del destino.
- Una red de datagramas necesita tener una entrada por cada destino posible, mientras que una red de circuitos virtuales sólo necesita una entrada para cada CV.
- Las CV tienen ventajas en cuanto a garantizar la calidad del servicio y evitar congestiones en la red.
- Los CV tienen problemas de vulnerabilidad, es decir, si falla un enrutador y pierde su memoria, todos los circuitos virtuales que pasan por él tendrán que abortarse.

EL PROTOCOLO IP VERSIÓN 4

- Un datagrama IPv4 consiste de dos partes: el encabezado y el cuerpo o carga útil.
- El ***encabezado* tiene una parte fija** de 20 bytes y una parte ***opcional* de longitud variable.**
- Los bits se transmiten en orden de izquierda a derecha y de arriba hacia abajo, comenzado por el bit de mayor orden del campo ***Versión***.

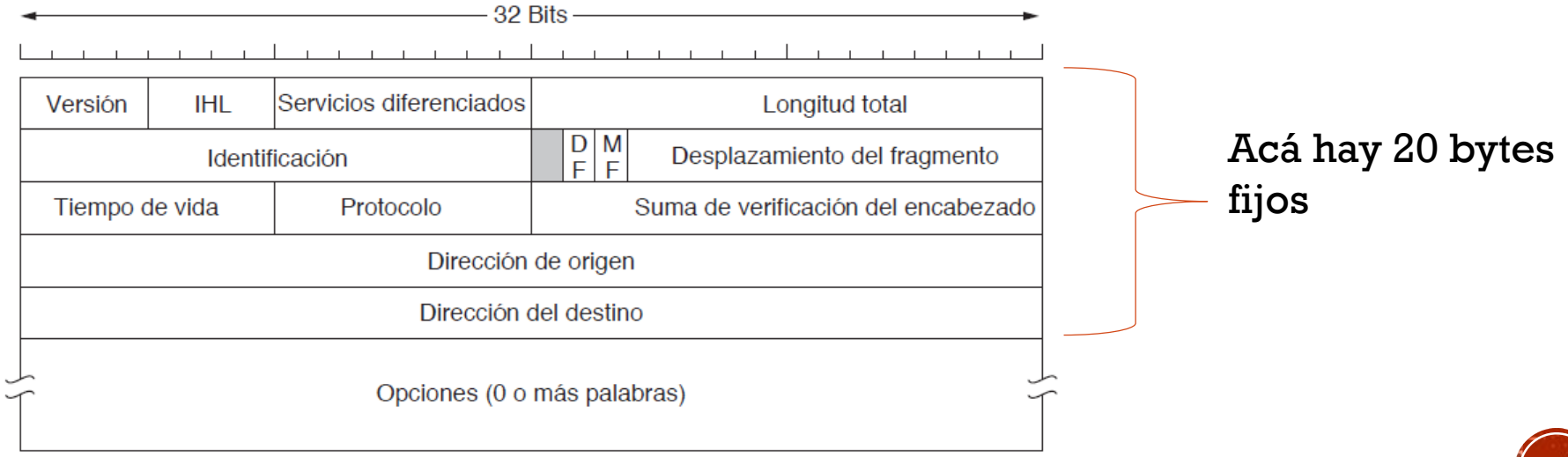


Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).

EL PROTOCOLO IP VERSIÓN 4

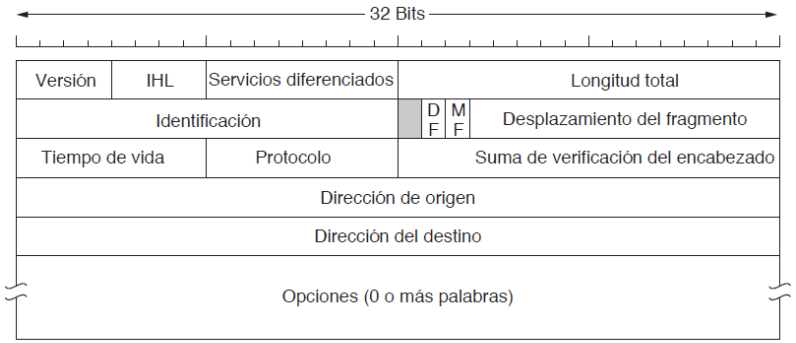


Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).

- Definido en el RFC 791.
- 20 bytes hasta 60 bytes.
- TCP, UDP, ICMP, IGMP.
- 12 variables.
- ***Versión***, lleva la versión del protocolo al que pertenece el datagrama (por ejemplo, 4 para IPv4). **(4 bits).**
  - 0100 -> IPv4
  - 0110 -> IPv6
- Dado que el encabezado no es constante, el tamaño de cabecera (***IHL, Internet Header Length)*** indica la longitud de la cabecera IP. El valor mínimo es 5, cuando no hay opciones. El valor máximo de **4 bits** es **15**, lo que limita el encabezado a **60 bytes** y por lo tanto, el campo ***Opciones a 40 bytes***.

EL PROTOCOLO IP VERSIÓN 4

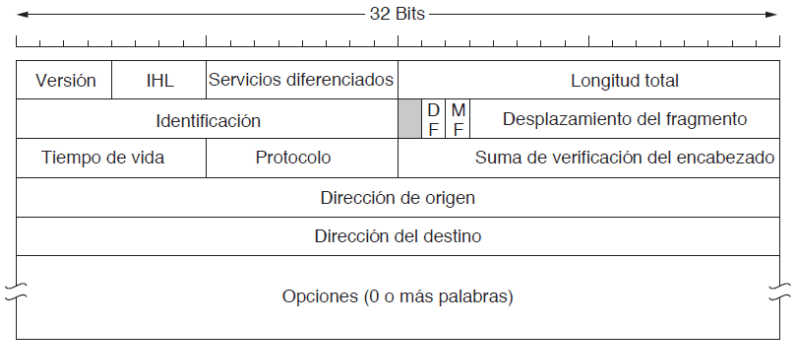


Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).

- Definido en RFC 2474
- El ***Servicio diferenciado***, distingue entre las diferentes clases de servicios (confiabilidad y velocidad). Tiene 8 bits, los 2 bits inferiores se utilizan para transportar información sobre la notificación de congestión (QoS).
- El campo **Longitud total** incluye todo el datagrama: tanto el encabezado como los datos. La longitud máxima es de 65535 bytes.
- El campo ***Identificación*** permite al host destino identificar a qué paquete pertenece un fragmento recién llegado. Todos los fragmentos de un paquete deben tener el mismo valor.

EL PROTOCOLO IP VERSIÓN 4

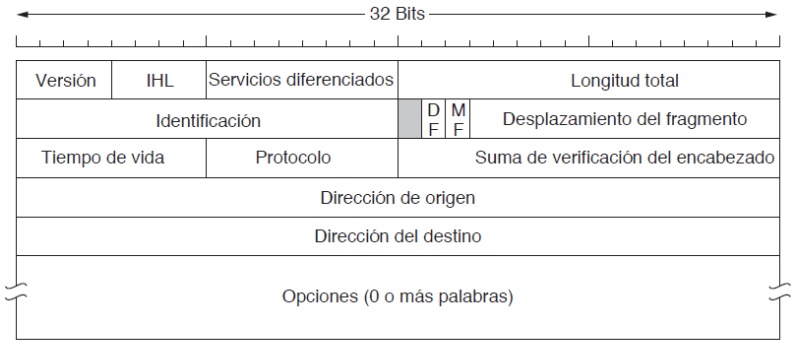


Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).

- Luego viene un bit sin uso, se supone que es para identificar tráfico malicioso.
- ***DF*** es de un bit y significa no fragmentar, en un principio se utilizaba para decirle al enrutador que no fragmenten el paquete. Actualmente, se utiliza para la definición de la ruta, es decir el emisor conoce si llegará una pieza o recibirá de vuelta un mensaje de error.
- ***MF*** significa más fragmentos. Todos los fragmentos excepto el último tienen establecido este bit.
- El ***Desplazamiento del fragmento*** indica a qué parte del paquete actual pertenece este fragmento. Dado a que proporcionan 13 bits, puede haber un máximo de 8192 fragmentos.

EL PROTOCOLO IP VERSIÓN 4

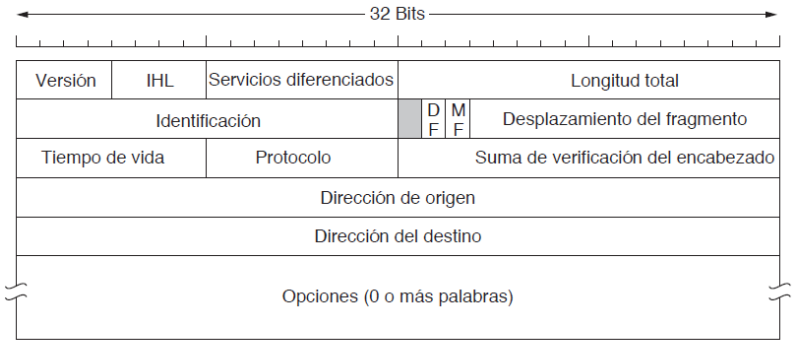


Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).

- ***Tiempo de vida (TtL)*** es un contador que se utiliza para limitar el tiempo de vida del paquete. Actualmente, cuenta los saltos de enrutador a enrutador, cuando llega a cero, el paquete se descarta y se envía de regreso un paquete de aviso al host de origen. Esta característica sirve para que los paquetes anden vagando eternamente.
- ***Protocolo***, indica a cual proceso de transporte debe entregar el paquete, por ejemplo, TCP, UDP y otros más (Protocolos).
- ***Suma de verificación*** es útil para detectar errores mientras el paquete viaja por la red. Se recalcula en cada salto.
- **Dirección de origen y dirección destino** indican la dirección IP de las interfaces de red la fuente y del destino.

EL PROTOCOLO IP VERSIÓN 4

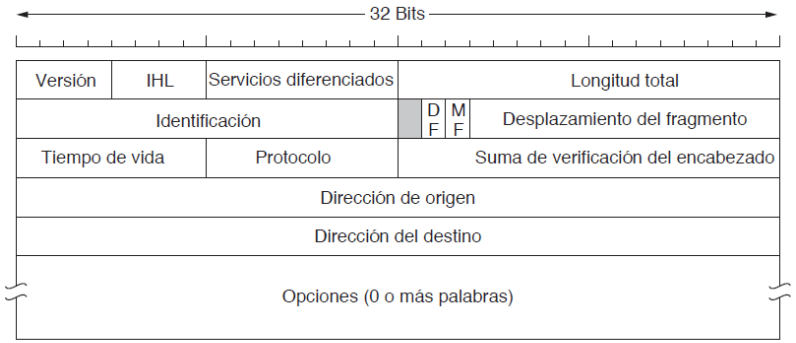


Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).

- ***Opciones*** se diseño para proporcionar un recurso que permitiera que las versiones subsiguientes del protocolo incluyeran información que no estuviera presente en el diseño original. Las opciones son de longitud variable. Cada una empieza con un código de 1 byte y se rellena con múltiplos de 4 bytes.

Opción	Descripción
Seguridad.	Especifica qué tan secreto es el datagrama.
Enrutamiento estricto desde el origen.	Proporciona la ruta completa a seguir.
Enrutamiento libre desde el origen.	Proporciona una lista de enrutadores que no se deben omitir.
Registrar ruta.	Hace que cada enrutador adjunte su dirección IP.
Estampa de tiempo.	Hace que cada enrutador adjunte su dirección y su etiqueta de tiempo.

Figura 5-47. Algunas de las opciones del protocolo IP.

EL PROTOCOLO IP VERSIÓN 4

- La opción de seguridad, indica qué tan secreta es la información.
- La opción de *Enrutamiento estricto desde el origen* proporciona la ruta completa desde el origen hasta el destino como una secuencia de direcciones IP.
- La opción de *Enrutamiento libre desde el origen* requiere que el paquete pase por los enrutadores indicas en la lista, y en el orden especificado.
- La opción de *Registrar ruta* indica a cada enrutador a lo largo de la ruta que adjunte su dirección IP al campo de *Opciones.*
- La opción *Estampa de tiempo*, registra la dirección IP de 32 bits y una estampa de tiempo de 32 bits. Útil para la medición en las redes.

```
Internet Protocol Version 4, Src: 192.168.0.100, Dst: 200.3.192.46
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
   Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
   Total Length: 40
   Identification: 0x71e0 (29152)
   Flags: 0x4000, Don't fragment
      0... .... .... .... = Reserved bit: Not set
      .1.. .... .... .... = Don't fragment: Set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
   Time to live: 128
   Protocol: TCP (6)
   Header checksum: 0x0000 [validation disabled]
   [Header checksum status: Unverified]
   Source: 192.168.0.100
   Destination: 200.3.192.46
```

DIRECCIONES IP

- Una dirección IP en realidad no se refiere a un host, sino a una interfaz de red, por si un host está en dos redes, debe tener dos direcciones IP. En contraste, los enrutadores tienen varias interfaces y, por lo tanto, múltiples direcciones IP.
- Hay una porción de red de longitud variable.
- Cada uno de los 4 bytes se escribe en decimal, de 0 a 255.
- El prefijo se escribe después de la dirección IP como una barra diagonal. Si el prefijo, contiene 2^8 direcciones y, por lo tanto, deja 24 bits para la porción de red. Se escribe como 128.208.0.0**/24**

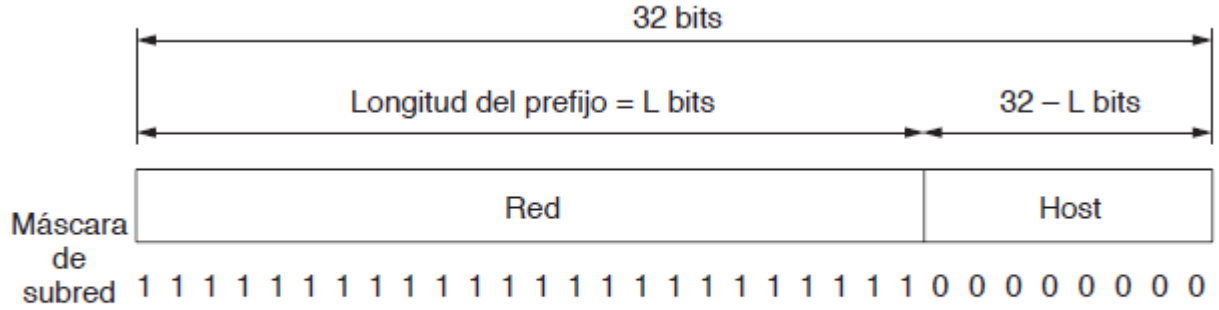
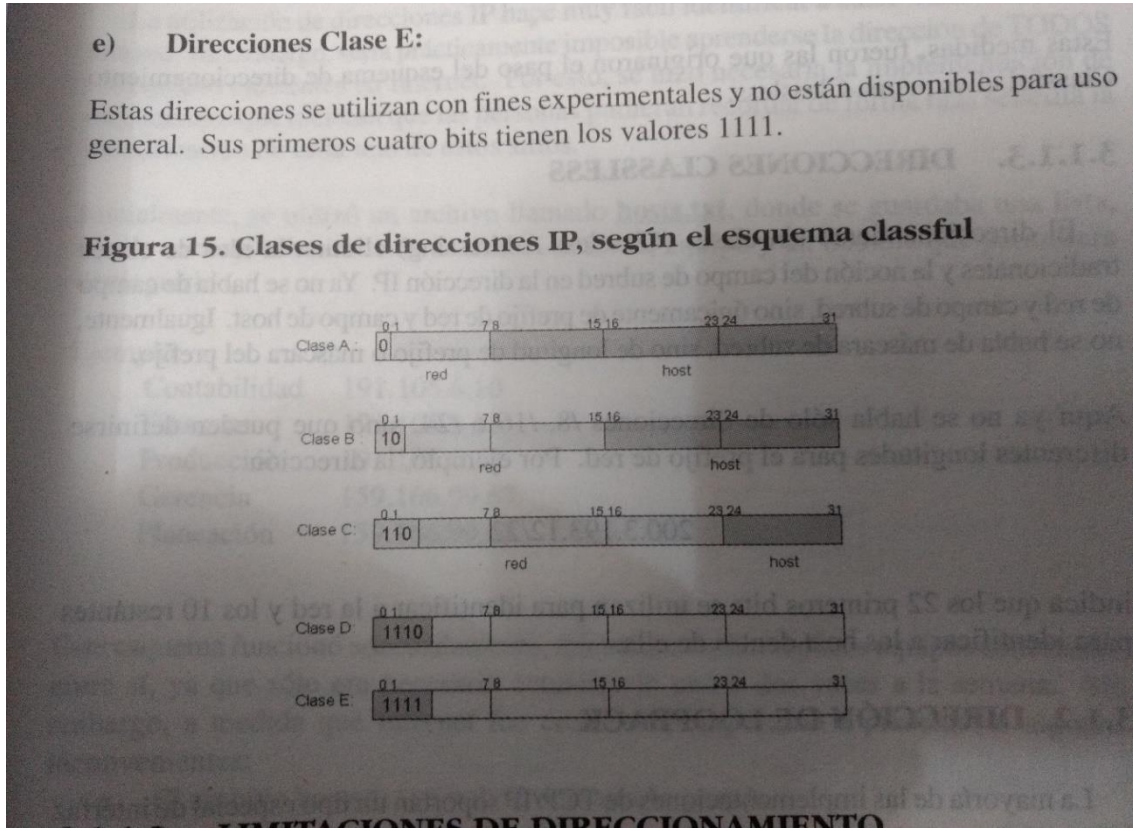


Figura 5-48. Un prefijo y una máscara de subred del protocolo IP.

DIRECCIONES IP

- La cantidad de bits de red y host depende de la clase a la que pertenece la dirección.



e) **Direcciones Clase E:**

Estas direcciones se utilizan con fines experimentales y no están disponibles para uso general. Sus primeros cuatro bits tienen los valores 1111.

Figura 15. Clases de direcciones IP, según el esquema classful

DIRECCIONES IP

- CLASE A - Soporta redes en Internet grandes.
- CLASE B – Soporta redes en Internet moderadas.
- CLASE C – Soporta redes en Internet pequeñas.
- CLASE D – Soporta Redes Multicast.
- CLASE E - Sin uso. Redes experimentales (uso futuro).

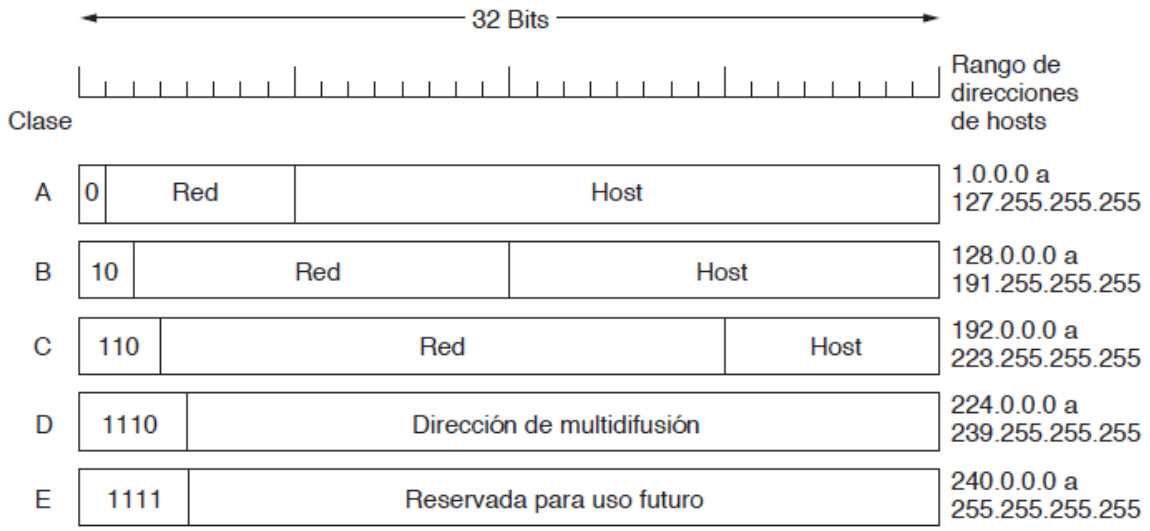
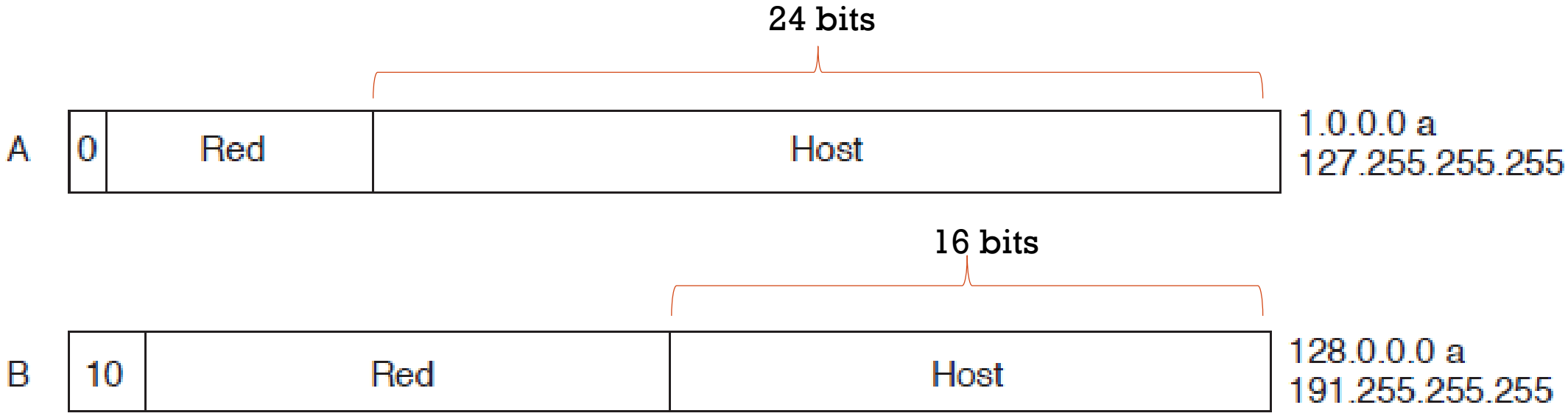


Figura 5-53. Formatos de direcciones IP.

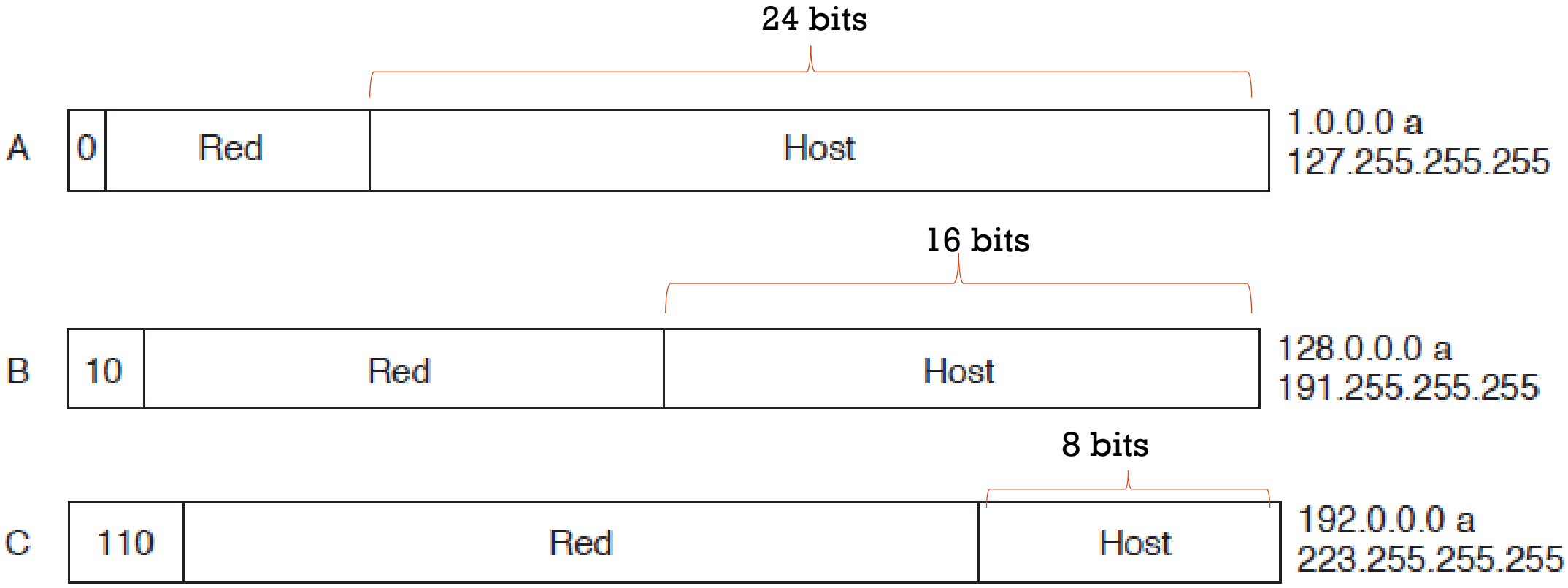
DIRECCIONES IP

- Para clase A tenemos 24 bits para el host y por lo tanto tenemos 8 bits para la red. Recuerde que tenemos un bit al inicio que identifica la clase.



DIRECCIONES IP

- Para clase A tenemos 24 bits para el host y por lo tanto tenemos 8 bits para la red. Recuerde que tenemos un bit al inicio que identifica la clase.



¿CÓMO SE DETERMINA LA CLASE?

- Suponga la dirección IP 192.168.100.50
- Tenemos que referirnos al primer octeto y ver en que rango esta de la clase

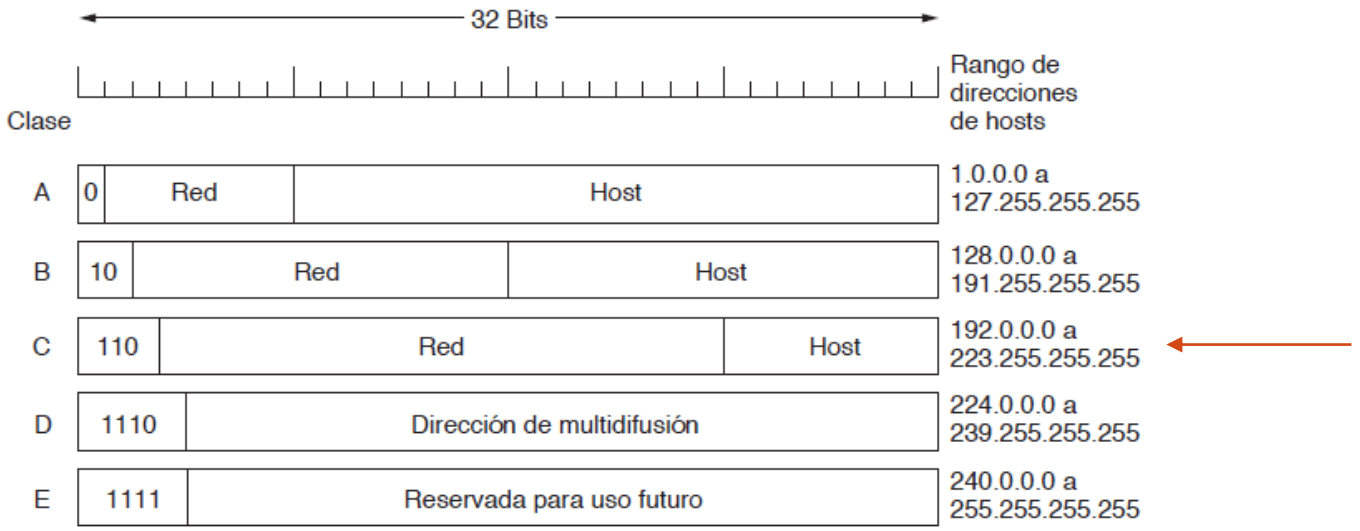
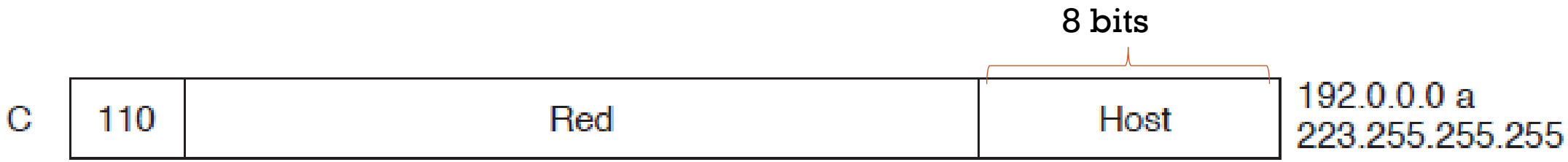


Figura 5-53. Formatos de direcciones IP.

- Ahora suponga la dirección IP 200.100.210.200
- Es de clase C y tiene 24 bits para red y 8 bits para host



200.100.210.200 → 200.100.210.200 / 24

Prefijo de red

Red
3 octetos
24 bits

Host
1 octeto
8 bits

MÁSCARAS Y PREFIJOS

- Las máscaras y los prefijos representan lo mismo, es decir, la cantidad de bits de la dirección IP que representan a la red.

192.168.168.100 / 24 → 11111111111111111111111100000000

172.16.16.10 / 16

255.255.255.0

11111111111111110000000000000000

255.255.0.0 → Máscara de red

LECTURAS

Material utilizado	1. Arboleda, L. (2012). Programación en Red con Java. 2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.". 3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación. 4. Reese, R. M. (2015). Learning Network Programming with Java. Packt Publishing Ltd.
Actividades DESPUÉS	A1. Leer del libro 3, la sección 5.6.1 hasta la sección 5.6.3, y 5.6.4 A2. Leer la sección 9 del libro 1

REFERENCIAS

1. https://3.bp.blogspot.com/-RPoJvwyN3Ic/WtQVBltajaI/AAAAAAAAALWY/tzutcLAvXlwvHM0TSOyho2GxduDG14HIwCLcBGAs/s640/Site-to-site-ipsec-example.png
2. http://www.rfwireless-world.com/images/IPSec-Transport-mode-vs-IPSec-Tunnel-mode.jpg
3. http://richardgoyette.com/Infosec/Alice/images/encrypt3.jpg
4. https://i.pinimg.com/originals/3e/53/42/3e534245a610e82dd09bf17e5c828c84.jpg